A Simulation-based Methodology for the Assessment of Server-based Security Architectures for Mobile Ad Hoc Networks (MANETs)

A thesis submitted for the degree of Doctor of Philosophy

By

Salaheddin Darwish

Department of Computer Science

Brunel University

December 2014

ABSTRACT

A Mobile Ad hoc Network (MANET) is typically a set of wireless mobile nodes enabled to communicate dynamically in a multi-hop manner without any pre-existing network infrastructure. MANETs have several unique characteristics in contrast to other typical networks, such as dynamic topology, intermittent connectivity, limited resources, and lack of physical security. Securing MANETs is a critical issue as these are vulnerable to many different attacks and failures and have no clear line of defence. To develop effective security services in MANETs, it is important to consider an appropriate trust infrastructure which is tailored to a given MANET and associated application. However, most of the proposed trust infrastructures do not to take the MANET application context into account. This may result in overly secure MANETs that incur an increase in performance and communication overheads due to possible unnecessary security measures.

Designing and evaluating trust infrastructures for MANETs is very challenging. This stems from several pivotal overlapping aspects such as MANET constraints, application settings and performance. Also, there is a lack of practical approaches for assessing security in MANETs that take into account most of these aspects. Based on this, this thesis provides a methodological approach which consists of well-structured stages that allows the exploration of possible security alternatives and evaluates these alternatives against dimensions to selecting the best option. These dimensions include the operational level, security strength, performance, MANET contexts along with main security components in a form of a multi-dimensional security conceptual framework. The methodology describes interdependencies among these dimensions, focusing specifically on the service operational level in the network. To explore these different possibilities, the Server-based Security Architectures for MANETs (SSAM) simulation model has been created in the OMNeT++ simulation language. The thesis describes the conceptualisation, implementation, verification and validation of SSAM, as well as experimentation approaches that use SSAM to support the methodology of this thesis. In addition, three different real cases scenarios (academic, emergency and military domains) are incorporated in this study to substantiate the feasibility of

the proposed methodology. The outcome of this approach provides MANET developers with a strategy along with guidelines of how to consider the appropriate security infrastructure that satisfies the settings and requirements of given MANET context.

ACKNOWLEDGEMENTS

First and Foremost, my sincere gratefulness goes towards to the most gracious and the most merciful God (**ALLAH**) for all his blessings and bounties he have bestowed me since I was born, without his blessings I would not be in this place at all. Most importantly, I would like to express my sincere gratitude to my supervisor, **Dr Simon JE Taylor**, without his support, patience and guidance this thesis would have been impossible to achieve. I was fortunate enough to have him by my side, meetings with him was enjoyable and discussions were insightful. Also, I wish to thank my second supervisor **Dr George Ghinea** for his professional guidance and critical comments which helped very much to improve my work.

I am deeply in debt to my darling parents, **Mr Muhammad** and **Mrs Amina**, for their care, patience, support, prayers to finish this work. Also, I love to thank my lovely sister, **Mrs Suzan** and my supportive brothers, **Dr Bilal**, **Dr Baraa** and **Mr Mouaaz** for their love, concern and encouragement along the way. I owe my sincerest gratitude to my sweetheart wife, **Mrs Altaf** and our gorgeous sons, **Muhammad** and **Zain**, for their understanding and endless love at all times. Eventually, I would like to express thanks to my father-in-law **Mr Abdulalhadi** and my mother-in-low **Mrs Fatema** for their prayers and support during my study.

Last but not least, I would also like to take the opportunity of thanking all my friends, fellow colleagues, the academic and support staff in the computer science department at Brunel University, specifically **Dr Anastasia Anagnostou, Dr Mahir Arzoky, Dr Bachar Alrouh, Dr Mohammad Hassona** and **Dr Muhammad Mersad Ghorbani**, I am grateful to all of those with whom I have had the pleasure to work during the course of my Ph.D.

DECLARATION

The following papers have been published (or submitted for publication) as a direct result of the research discussed in this thesis:

Darwish, S.; Taylor, S. J.E.; Ghinea, G., (2012). Security Server-Based Architecture for Mobile Ad Hoc Networks *In: Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE 11th International Conference on* , vol., no., pp.926,930, 25-27 June 2012.

ABBREVIATIONS

3G Networks	Third Generation Cellur Networks
4G Networks	Fourth Generation Cellur Networks
AA	Attribute Authority
AAA	Authentication Authorisation Accounting
AcSc	Achievement Score
AES	Advance Encyption Standard algorithm
AODV	Ad hoc On-Demand Distance Vector
AVP	Attribute Value Pair
CA	Certification Authority
CAS	Central Authority Server
CBR	Constant Bit Rate
CCA	Central Certification Authority
CH	Cluster Head
CKM	Composite Key Management
CNA	Current Number of Attempts
CRL	Certificate Revocation List
CSMA	Carrier Sense Multiple Access
DARPA	Defence Advanced Research Project Agency
DAS	Delegated Authority Server
DCA	Delegated Certificate Authority
DDMZ	Dynamic Demilitarised Zone
DDoS	Distributed Denial of Service
DICTATE	Distributed CerTification Authority with probabilisTic frEshness
DoS	Denial of Service
DSA	Digital signature Algorithm
DSDV	Destination-Sequenced Distance-Vector
DSR	Dynamic Source Routing
DSS	Digital Signature Standard
EAP	Extensible Authentication Protocol
ECC	Elliptic curve cryptography
ETSI	European Telecommunications Standards Institute
GloMo	Global Mobile Information Systems
GNED	Graphical Network Editor
GUI	Graphical User Interface
HIPERLAN	High Performance Radio Local Area Network
HMAC	Hash Message Authentication Code
IETF	Internet Engineering Task Force
IrDA	Infrared Data Association
ISO	International Organization for Standardization
ITU	International Telecommunications Union
KDC	Key Distribution Centre
LLSC	Location Limited Side Channels
MAC	Medium Access Control
MAC	Message Authentication Code
MANET	Mobile Ad hoc Network
MAS	Master AAA Server
MCA	Mobile Certification Authority
MD	Message Digest
MIT	Massachusetts Institute of Technology
MNA	Max Number of Attempts for the authentication re-try

MOCA	MObile Certification Authority
MRA	Master Root Authority
NED	Network Description
NetTRUST	NETworks Trust infrastRUcture baSed on Threshold cryptography
NPC	Number of received Partial Certificates
NTDR	Near-Term Digital Radio
OLSR	Optimized Link State Routing
OMNeT++	Objective Modular Network Testbed in C++
OSI	Open Systems Interconnection
P2P	Peer-To-Peer
PAN	Personal Local Networks
PCA	Policy Certificate Authority
PDA	Personal Digital Assistant
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
PRNET	Packet Radio Network
QoS	Quality of Service
RA	Registration Authority
RCA	Root Certificate Authority
RERR	Route ERRor control messages
RF	Radio Frequency
SPP	Stationary Poisson Process
RREP	Route REPly control message
RREQ	Route REQuest control message
RRW	Rank Reciprocal Weighting
RSA	Rivest-Shamir-Adleman algorithm
RTT	Round Trip Time
RTT-STDev	The Standard Deviation of Round Trip Time
RWM	Round Waypoint Mobility
SASs	Slave AAA Server
SEKM	Secure and Efficient Key Management
SHA	Secure Hash Algorithm
SPC	Single Point of Compromise
SPF	Single Point of Failure
SRWM	Steady-state Random Waypoint Mobility
SSAM	Server-based security Architecture for MANETs
SURAN	Survivable Radio Networks
TAS	Threshold Authority Server
TC	Threshold Cryptography
TCA	Temporary Certificate Authority
TTP	Third Trust Party
VV&T	Validation and Verification and Testing
WLAN	Wireless Local Area Network

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction

1.1 Background and Motivation

Nowadays, as a result of remarkable advances in mobile computing and wireless communication technologies, mobile devices (e.g. smartphones, PDAs, tablets, laptops, etc.) are having a greater impact on many diverse applications in military, civil, heath and space domains. Consequently, a mobile ad hoc network (MANET), which offers a means of interconnection between wireless devices without a predefined infrastructure, has received a substantial attention by different communities. A new evolutional market is emerging which aims to deploy and run new network services through exploiting current mobile devices in order to access resources anywhere and anytime (Guarnera *et al.*, 2002; Chlamtac *et al.*, 2003), for example, in mobile gaming, live or on-demand multimedia streaming and video conferencing, etc.

Although a MANET exhibits great potential, it has a number of challenging properties network- and application-wise. Network-wise, each node is able to join and leave the network freely, consequently this changes its topology frequently (Toh, 2001; Murthy and Manoj, 2004). In addition, the nature of its resources is constrained, as the majority of its nodes are low-end devices normally powered by a battery. Also most types of MANETs exploit the current wireless technologies (e.g. Wi-Fi, Bluetooth and IrDA) that already operate with limited bandwidth and intermittent communications (Chlamtac *et al.*, 2003). Application-wise, a MANET may be utilised for running particular applications which have different settings and requirements. For example, an application may use MANETs for spontaneous communication, i.e. no prior relationship between nodes, or for planned communication. Furthermore, a MANET may operate for short time or long time period according to application time requirements (Hoeper and Gong,

2004; Dawoud *et al.*, 2011). Hence, the demand of an application will shape the complexity and cost of a MANET initialisation, operation, and management.

Furthermore, securing a MANET is very important but at the same time very problematic, due to the fact that this network, unlike other typical networks, is vulnerable to different attacks and also it is characterised with no clear line of defence (Yang *et al.*, 2004; Djenouri *et al.*, 2005; Carvalho, 2008). This is as a result of the open wireless medium used and a number of specific constraints in its properties, for example the limitation in resources capability, the lack of physical protection, and the variances in security requirements of MANET applications. Trust is a very important term which most security solutions (e.g. authentication, authorisation, confidentiality, etc.) rely on in their deployment. In fact, in the literature (Zhou and Has (1999), Yi *et al.* (2003), Bechler *et al.* (2004), Luo *et al.* (2004), Ngai and Lyu (2004), Hadjichristofi *et al.* (2005), Luo *et al.* (2005) Rachedi *et al.* (2006, 2007), Raghani *et al.* (2006), Dong *et al.* (2007), Omar *et al.* (2007), Wu *et al.* (2007), Saremi *et al.* (2009), Al-Bayatti *et al.* (2009) and Larafa and Lauren (2011)) particularly there are various credential-based authoritarian trust models whose mission is to describe how to manage trust relationship between entities by an authority using a particular credential (e.g. certificates). This stems from the fact that relying on an authority can guarantee securing high-value communications with high confidence, particularly in large-scale networks as every node in the network should have a strong trust on the authority for handling their security matters. For this reason, to develop effective security services in MANETs, initially it is imperative to select the proper trust infrastructure which is tailored to the context of MANETs in question. This context consists of MANET characteristics and application requirements.

Conversely, it may be argued that any security solution for MANETs is not the overall story. Improving security strength in MANETs would definitely incur an increase in computation, communication, and management overhead due to additional functionalities that must be implemented for supporting security. Therefore, network performance, in the sense of scalability, service availability,

robustness of the security solutions, becomes an important factor in resource-constrained MANETs. While many existing proposals focus on the cryptographic perspective of their security solutions, they ignore unintentionally the consideration of the network performance factor (Yang *et al.*, 2004). Hence, it is important that both dimensions of security strength and network performance should be equally taken into account for better security design of MANETs and this can be achieved by making an appropriate trade-off between them.

In addition, the design paradigm of MANETs should arguably follow the standard layer-based stack – the OSI (ISO/IEC 7498-1) model which leverages the strict protocol-layer separation and horizontal interaction so as to facilitate the MANET development and deployment by supporting modularity, flexibility, simplicity and interoperation in networking system design. Some design proposals show a compliance with the OSI model for tackling MANET security, such as Yu *et al.* (2003), Yang *et al.* (2004) and Sehgal *et al.* (2011). However, other design approaches have been proposed, sacrificing flexibility and interoperation aspects, for example, a vertical layer integration approach in Corson *et al.* (1999) and a cross-layer design approach in Conti et al. (2004) and Messerges et al. (2003). Hence, the aspect of having a layered design (i.e. operational levels called in this study) is vital for a MANET design as each layer has its own different activities and issues (e.g. security, performance, routing, etc.) which can be tackled independently and effectively during the MANET development.

Therefore, developing an effective and efficient security solution (especially in the domain of security/trust infrastructure) for MANETs is not a trivial issue. This is because there are several dimensions (i.e. levels, strength, performance and constraints) which have a significant impact on the security design of MANETs. Few works have addressed some of these issues. For example, in Balakrishnan and Varadharajan (2005), three dimensions were identified and must be taken into consideration when securing MANETs: cryptography, resources and behavioural trust. On the other hand, most published studies in MANET security have only focused on routing and networking security (Cho *et al.*, 2011) while overlooking

security design in transport and application layers (i.e. representing the service level where a client-server model is predominant). Only Martucci *et al.* (2004) and Aljnidi and Leneutre (2007) have focused on the application layer for their security solution. Overall, it appears that few publications, if any, attempt to adopt a balanced view of MANET that reflects an operational level, security, performance and aspects of a given application along with MANET capabilities.

Therefore, this thesis argues that the security alternatives need more scrutiny along with those pivotal overlapping factors discussed before (e.g. security strength, performance and the contexts of MANET constraints and application settings), especially in the service level of MANETs. Accordingly, this study is dedicated to presenting an exhaustive evaluation for different authority-server-based security architectures for large-scale MANETs which are primarily composed of a specific server architecture and authentication protocol, cryptosystem, and security credential. The outcome of this multi-dimensional evaluation, which mainly consists of a set of indicators and recommendations, has the potential to enable a systematic approach to sensibly choose a security architecture under a certain context of constraints and settings. This approach can assist MANET developers to come up with pragmatic solutions for security problems in MANETs.

1.2 Research Aim and Objectives

The previous section has underlined that there is a number of different aspects which have influence on implementing a security/trust infrastructure in the service level of large-scale MANETs. Selecting a security solution for a given MANET often requires an insight into the trade-offs between different aspects which are security strength, performance and context. Therefore, the main aim of this research is:

"To develop a methodological approach that enables MANET developers to select the best appropriate server-based security architectures that satisfy the security, performance and context requirements of a given application"

To fulfil the aim of this study, the following five objectives will be met.

Objective 1: Conduct a literature review to evaluate the current and most common security fundamentals and authoritarian trust models in the MANET domain to highlight their capabilities, requirements and limitations which will be a cornerstone for developing the proposed approach in the aim of this study.

Objective 2: Design a conceptual security framework for identifying the building blocks of the proposed approach of this study: the security operational level design; the security/trust infrastructure; security strength; performance; context and addressing the relationship between them as well as establishing this approach.

Objective 3: Design a new security model of server-based security architectures for MANETs (SSAM), based on the components of the security/trust infrastructure for the service level of MANETs and also for checking the applicability of the proposed approach designed in pervious objectives.

Objective 4: Develop a simulation model for the proposed server-based security architectures for MANETs (SSAM) and experimentally test them under different scenarios using a network simulator in order to understand the cost of applying them on the performance and communication of MANETs, i.e. to create a performance evaluation model.

Objective 5: Develop the proposed approach based on the results collected from the simulation experiments from the previous objective and

the consideration of the other security strength and context requirements.

Objective 6: Evaluate the proposed approach through the use of real case scenarios so as to demonstrate and validate the benefits and value of the approach.

1.3 Research Approach

The general theme of this research clearly abides by the experimental simulation approach (Balci, 1990; Law and McComas, 1991; Balci, 1994; Nance, 1994) which required the SSAM model to be evaluated in terms of the performance and communication using the OMNeT++ simulation tool. The results generated from this experimentation would be used in validating the proposed approach of selecting appropriate security alternatives in this study, based on different criteria (security strength, performance, etc.). Therefore, the simulation appeared to be a suitable scientific methodology for conducting this research.

For successful and credible research outcomes, the simulation methodology is divided into three main stages, (1) the problem definition, (2) the model development and (2) decision support stages. In each stage, a simulation study typically has distinct processes that are required to be performed in order to reach the various phases within the stage, as shown in Figure 3-2. The identified *problem*, *system* boundaries and *objectives* phases are formulated in the problem definition stage. The model development stage presents how the simulation model will be evolved through different phases, *conceptual*, *communicative*, *programmed* and *experimental* models and *model results* (generated from simulation experimentation). In decision support stages, the results from the previous stage will be incorporated to support decision. Chapter 3 explains this approach in more detail.

1.4 Terminology

For avoiding ambiguity, three terms which are used in this thesis extensively are important to be defined and described as follows:

1- **The multi-dimensional security framework:** This framework is defined as a conceptual structure of a security/trust infrastructure (i.e. security building blocks) for MANETs and its vital dimensions (i.e. operational levels, security strength, performance, MANET contexts). The framework is introduced to serve as a guide for the designing of better security/trust infrastructures for a security service over MANETs taking into consideration these proposed dimensions.

2- **The methodological, systematic or pragmatic approach ("Methodology"):** this methodology is defined in this thesis as a set of well-structured steps which are necessary to follow in order to effectively evaluate security alternatives in MANETs and then find the most appropriate ones which fulfils particular requirements. Simulation experimentation along with a decision making technique is used in the evaluation and selection of those alternatives. However, the security alternatives in this context indicate the security/trust infrastructures for MANETs (i.e. the SSAM model) which are already represented in the framework defined before. Also, the dimensions in this framework are incorporated in this methodology to establish the requirements for sake of evaluation.

3- **The security architecture:** this architecture presented in the SSAM model is defined as an initial and incomplete design of the security infrastructure for deploying a security service in MANETs. The architecture is intended to describe the types of security servers and authentication protocols and the strategies of calling and re-authentication which are in use.

1.5 Thesis Structure

This thesis is structured into seven chapters. This chapter (Chapter 1) showed the research motivation, clarified the aim and objectives of this research, the research approach being adopted and finally the thesis structure.

Chapter 2: This chapter provides a literature review of the related topics that are the knowledge body of this research. It begins to present the MANET evolvement, their unique characteristics and potential applications. Also it outlines most common MANET challenges that need to be investigated. Thereafter, most MANET-related security fundamentals, which may be incorporated in the proposed security approach of this study, are identified, such as security requirements, MANET threats and attacks, and relevant security mechanisms. Finally, the notion of trust management and the current related trust models in MANETs are discussed. This chapter is to establish grounding to the proposed approach through characterising a number of elements, such as distinct MANET attributes, MANET security and trust infrastructures.

Chapter 3: This chapter discusses the research methodology followed in this thesis. It starts with an outline of the overall research approach for this research. Thereafter, the chapter sheds light on the simulation technique and its experimental approach which are used to conduct performance and communication testing for the SSAM model. Three different stages of the simulation approach are illustrated along with how these stages are applicable to this study: the problem definition, the model development and the decision support. In addition, the verification validation and testing techniques (VV&T) are identified and used in the simulation approach of this research to establish its credibility.

Chapter 4: This chapter consists of two parts (Part A and B). Part A mainly presents the conceptual security framework for MANETs which leverages various and crucial security-design-related facets as to support designing and developing a trust/security infrastructure in MANET (i.e. an operational level, security

components, security strength, performance, the context of MANET constraints and application settings). The aim is to introduce a methodological approach which would assist a MANET security developer to come up with an effective security solution for a certain MANET context. Part B describes the proposed security model (SSAM) with its components (i.e. a server architecture, an authentication protocol, a cryptosystem, a security credential, a strategy of calling, and MANET settings). Also, it presents in detail the model design of SSAM activities, communications and processes. The SSAM model is intended to provide a new security design for MANETs in the service operational level and to prove the feasibility of the suggested multi-dimensional approach generated from the framework proposed in Part A.

Chapter 5: Along with the proposed model assumptions, this chapter presents the implementation and experimentation of the SSAM model being designed in Chapter 4. The OMNeT++ simulation tool is chosen for creating the SSAM prototype. This prototype relies on defining the necessary network and node structures and creating the related C++ classes. Furthermore, all important configurations and initialisations for this prototype are systematically addressed in order to appropriately conduct performance testing. There are two particular types of configurations that are elaborated in this chapter, the security and network configurations for SSAM. The security configurations are related to the SSAM server architecture, massage sizes in different authentication protocols, processing time in authentication process and the re-authentication scheme being used. The network configurations refer to various related MANET components, such as mobility, traffic, churn models, and transport, routing and MAC protocols. Eventually, the experimental design for running SSAM simulation is described through determining the necessary metrics, the test cases and the number of replications for simulation.

Chapter 6: This chapter initially presents the analysis of quantitative results produced from testing the performance and communication of the proposed security architectures in SSAM under different network scenarios (i.e. the *Churn*

and *No-Churn* scenarios). The strength of each security architecture in SSAM is analysed against particular server-related security problems (i.e. Single-Point-of-Failure (SPF) and Single-Point-of-Compromise (SPC)) and protocol robustness. The outcome of the analysis represents three different dimensions, such as performance, communication (i.e. MANET constraints) and security strength. This is incorporated in the proposed approach (i.e. methodology) of this study. As a result, this approach is applied in the context of the SSAM mode for validation purposes. As part of the evaluation stage in this approach, a simple ranking system, which is based on achievement scoring, reciprocal ranking weighting (RRW) and weighted averaging, is proposed to facilitate ranking the security alternatives. The value of this proposed evaluation approach is justified by employing this approach to three different real case scenarios (academic, emergency and military contexts).

Chapter 7: This chapter concludes by recapping the whole work done in this research and describing the thesis contributions and the main limitations of this work. It also suggests the future directions of research to complement this work.

Chapter 2: MANETs and Security Background

2.1 Overview

As explained in the previous chapter about the road map of this thesis, the main purpose of this chapter is essentially to introduce the common reviewed topics associated with MANETs and security in MANETs. On one hand, it covers the MANET definition and history, unique MANET characteristics, MANET application domains and MANET current challenges. On the other hand, it also addresses the security background relevant to this MANET research and mainly includes security requirements, cryptographic mechanisms, trust management and the related credential-based trust models. Eventually, this chapter attempts to clear up the vital fundamentals of MANETs and security issues. These can be exploited to create a better understanding of some keystones that are required to be considered for developing this study proposed approach of planning and designing security service in MANETs.

2.2 Mobile Ad hoc Networks (MANETs)

This section introduces MANETs: definition; history; characteristics; applications, challenges. A definition and history of MANETs are briefly presented. The MANET characteristics with their relevant sub-attributes are described in detail. In addition, in this section, the various application domains that can utilise MANET capabilities are discussed and then MANET challenges are highlighted.

2.2.1 Definition and History

In fact, the ad hoc networking paradigm is considered one of the most attractive as well as challenging paradigms in the networks development. As shown in Figure 2-1, a Mobile Ad hoc Network (MANET) is defined as a self-organised network which consists of a set of wireless mobile nodes enabled to communicate dynamically on the fly in multi-hop manner without any pre-existing network infrastructure (infrastructure-less) (Chlamtac *et al.*, 2003). Unlike cellular networks and WLANs, each node has two roles simultaneously; it becomes a router to handle packet to other nodes and an application node (i.e. a user or service provider) to handle its own communication (Toh, 2001; Chlamtac *et al.*, 2003; Murthy and Manoj, 2004). Therefore the nodes communicate directly with each other without any intermediate body. Each node can also join and leave the network at any time.

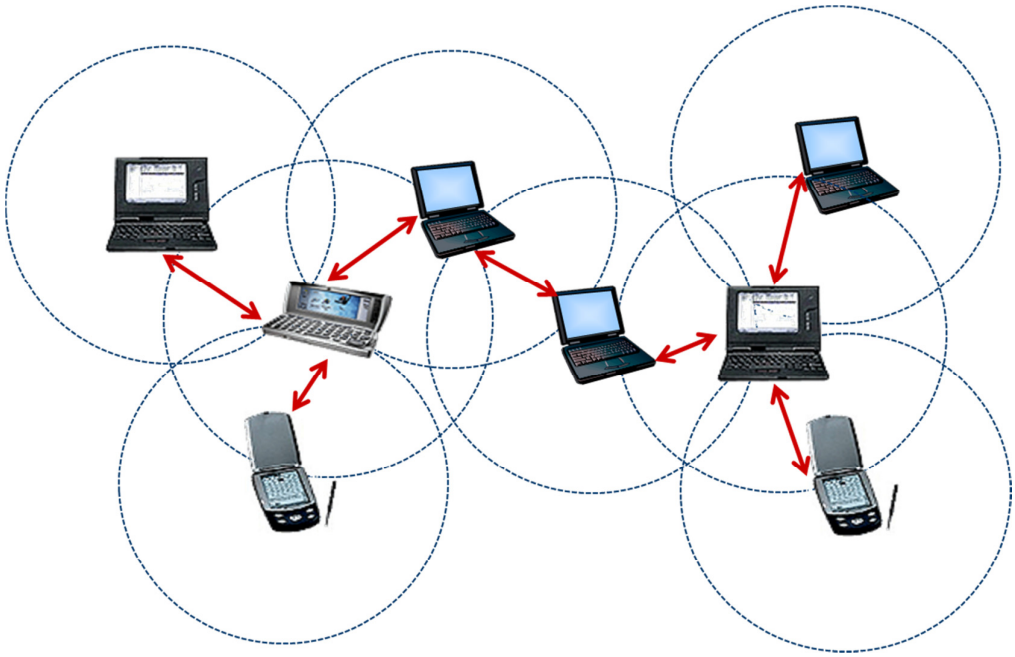


Figure 2-1: The Mobile Ad hoc Network - MANET

Historically the first ad hoc network was coined in tactical network applications in order to improve battlefield communications survivability. In 1972

PRNET (Packet Radio Networks) under the DARPA project was developed by US Department of Defence aiming to provide effective packet-switched multi-hop networking (i.e. bandwidth sharing and store-and-forward routing) to mobile battlefield elements like soldiers, tanks, aircrafts in a hostile environment without relying on any fixed infrastructure and central control. PRNET basically relied on the combination of ALOHA and CSMA channel access protocols so as to support the dynamic sharing of the broadcast radio channel. In addition, to avoid limited radio coverage, it used distance-vector routing protocols to facilitate multi-user interconnection.

A few years later in 1983, Survivable Radio Networks (SURAN) came out as a new version of PRNET, which had significant improvements to the issues of network scalability, security, capacities and power management. Its goals were to devise network algorithms to boost a network that could use small, low-cost and light-energy radios for interconnection, scale well to thousands of nodes (using hierarchical link-state routing protocols) and have resilience to electronic attacks (Freebersyser and Leiner, 2001).

Almost a decade later, the notion of commercial ad-hoc networks began to gain more attention with the advent and widespread popularity of notebook computers and other new viable wireless communication technologies (e.g. IrDA, RF etc.). The IEEE 802.11 Wireless Local Area Network (WLAN) working group adopted the term ‘Ad Hoc’ for such networks and have them in its standardisation initiatives. The European Telecommunications Standards Institute (ETSI) started standardisation of the High Performance Radio Local Area Network/1 (HIPERLAN/1) which utilises ad hoc networking in its architecture. Meanwhile the US Department of Defence funded other projects related to Ad Hoc networks, including the Global Mobile Information Systems (GloMo) and Near-Term Digital Radio (NTDR). GloMo was developed to use an Ethernet-based technology to offer a multimedia connectivity any anywhere and anytime in handheld devices while NTDR was based on a self-organised two-tier ad-hoc

network which made use of clustering and link-state routing (Ramanathan and Redi, 2002).

From the mid 1990's, ETSI released the first edition of the HIPERLAN//1 standard (Halls, 1994) and then the HIPERLAN//2 standard in following couple of years (Khun-Jush *et al.*, 2000). The IEEE 802.11 subcommittee brought to light its first standard, IEEE 802.11 (IEEE), a medium access protocol that was based on collision avoidance and tolerated hidden terminals, enabling to create a mobile ad hoc network prototypes from notebooks fitted with 802.11 PCMCIA cards. The following standards, IEEE 802.11a and IEEE 802.11b IEEE 802.11g were the successors of this standard. The Internet Engineering Task Force (IETF) founded the MANET charter (MANET, 2012) aiming to investigate and standardise routing protocols for ad hoc networks. As results of these efforts, a number of reactive and proactive routing protocols (e.g. AODV, DSR, and OLSR etc.) have been developed. The Bluetooth technology became also recognisable as an example of exploiting ad hoc networking in building Personal Local Networks (PAN). The Bluetooth special interest group released standard v. 1.0 in 1999. Furthermore, Bluetooth standardisation started to be carried out in close collaboration with the IEEE 802.15 PAN working group, leading to the release of the IEEE 802.15.1 standard based on the Bluetooth specification v. 1.1 in 2002.

It appears that on-going research in MANET technology is moving towards the standardisation of different existing systems with different network controls in a unified application framework. It is also noticeable that wireless devices are becoming smaller, smarter and more affordable. General-purpose MANETs therefore offer organisations a low-cost way to keep these devices connected thereby increasing MANET demand.

2.2.2 MANET Characteristics

MANETs as perceived from the MANET definition and their history stated in pervious section, have several exceptional characteristics MANETs in contrast to other typical networks. In this thesis for the sake of simplicity, the MANET characteristics are sorted into two main classes, the network- and node-related attributes. Each main class has its key attributes and their following sub-attributes which contribute to their key ones as shown in Figure 2.1. The network-related ones features that are pertinent to the system structure of MANETs whereas the node-related ones indicate to physical capabilities of their nodes. Each is now described below.

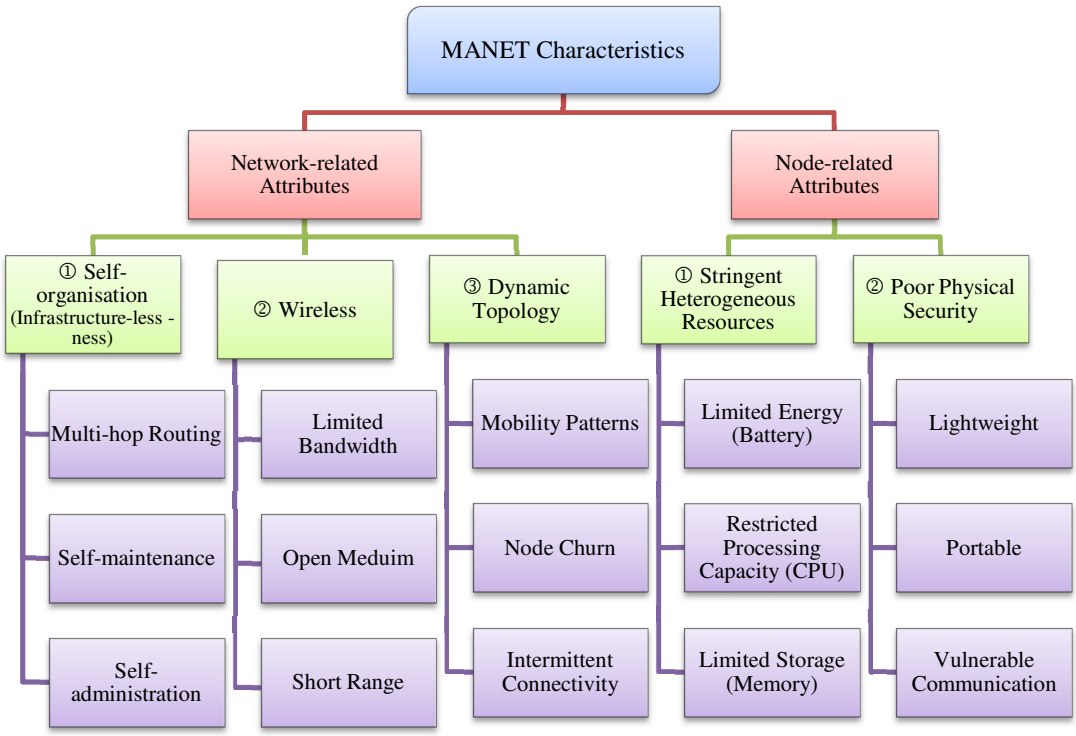


Figure 2-2: Mobile Ad hoc Network Characteristics

2.2.2.1 Network-related Attributes

In the context of MANETs, this section discusses the main attributes of a networking system along with the relevant sub-attributes which contribute to the key ones. These attributes primarily involve most aspects of the medium technology, interconnection infrastructure, and network topology.

1. Self-organisation (Infrastructure-less-ness)

MANETs have been developed originally to suit given applications in particular circumstances where installing a long-term infrastructure is not feasible, for example, in battlefields or search-and-rescue missions. Therefore, the *self-organisation* attribute fundamentally stems from the fact that MANETs presumably do not depend on any pre-defined infrastructure or centralised administration (e.g. central servers, base stations, fixed routers etc.) for operation and also in most cases they could spontaneously be deployed with no a priori knowledge of the physical location and networking environment. On the other hand, for self-maintenance and alleviating network disconnection, MANETs exploit a unique pattern of cooperative interconnection using *multi-hop routing* between network nodes. In other words, each node normally acts as an independent router to relay other nodes messages and at the same time act as a host to make use of available network services (Chlamtac *et al.*, 2003; Djenouri *et al.*, 2005; Savola and Uusitalo, 2006).

2. Wireless

MANETs primarily rely on the *open shared wireless medium* to link their nodes. Wireless links usually have a lower bandwidth compared to wire-line counterparts thanks to a number of wireless phenomena for example noise, fading, interference and congestion. As MANETs make use of current wireless technologies (e.g. Wi-Fi, Bluetooth etc.) which already have problems (e.g. unreliability, low capacity and a short range), they will apparently inherit those problems in addition to their own ones. On the other side, MANETs are typically characterised by very much more *restricted bandwidth* than other similar wireless technologies like cellular

networks or WLANs. That stems from the dual role for each node to perform (i.e. an access point and host) which may reduce the link capacities for interconnection in MANETs (Corson and Macker, 1999; Yang *et al.*, 2004; Scott *et al.*, 2006).

Furthermore, another reason of constraining the bandwidth of MANETs is that MANETs may occasionally operate in heterogeneous wireless environments where each node may have one or more radio interfaces that consist of different transmission/receiving capabilities and work across different frequency bands. This heterogeneity in node radio capacities can bring on asymmetric links which in turn lead to varying bandwidth-delay characteristics (Corson *et al.*, 1999; Chlamtac *et al.*, 2003).

3. Dynamic Topology

As the MANET nodes have wireless connectivity and mostly mobilise according to a certain *mobility patterns* (e.g. random human walk, vehicles mobility and etc.), these features allow nodes to freely roam and easily join, and leave the network, i.e. have specific *joining* and *churn* rates. As a result, the unreliable wireless links between nodes render unintentionally intermittent. Eventually, the network topology, changes arbitrarily and frequently. This makes the network unpredictable and amorphous i.e. changeable structure about both its size and shape (Corson *et al.*, 1999; Corson and Macker, 1999; Djenouri *et al.*, 2005).

2.2.2.2 Node-related Attributes

This section presents details of the attributes and their sub-attributes that are associated with capabilities of nodes participating in MANETs. It is noted that there are two important and critical attributes which need to be addressed in MANETs regard: stringent heterogeneous resources; poor physical security.

1. Stringent heterogeneous resources

Almost all MANET nodes are small, mobile and handheld devices for example smartphones, PDAs, etc. They often are powered by batteries and this makes them have a constraint in their power resources (i.e. *limited energy*). Besides, MANET

nodes in particular experience an exceptional energy challenge when compared to conventional wireless networks such as WLAN, etc. This is due to that fact that, as aforementioned, each node is assumed to act as both an end system and a router concurrently so additional energy is required to handle packets routing to other nodes (Chlamtac *et al.*, 2003; Djenouri *et al.*, 2005; Chadha and Kant, 2008).

On the other hand, MANETs predominantly make use of various embedded and off-the-shelf wireless devices and such flexible usage entail *heterogeneity* in processing and storage capabilities as each of those devices have their own different software/hardware specifications. Also, their processing and storage capacities are normally considered restricted for sake of saving power (e.g. using small CPU and memory resources) since most of those devices rely on limited batteries (Chlamtac *et al.*, 2003; Djenouri *et al.*, 2005; Savola and Uusitalo, 2006; Merwe *et al.*, 2007; Chadha and Kant, 2008).

2. Limited physical security

Contrary to conventional wired networks, the use of wireless communication and lightweight and *portable* devices in MANETs implies poor physical security and serious security vulnerabilities. Those devices and the information stored in them can be easily physically compromised by either loss or theft. On the other side, *wireless medium* and cooperative nature in this type of networks also enable adversaries to take advantages to intercept the flow of information over the air and perform any number of tests and analysis in order to lunch attacks for example spoofing, eavesdropping and denial-of-service (Corson *et al.*, 1999; Scott *et al.*, 2006; Merwe *et al.*, 2007).

To conclude, MANETs reveal how unique, diverse and challenging characteristics they possess. It is worth pointing out that in order to take advantages of most MANET potentials, those characteristics are therefore required to be taken into consideration when tackling any issues related to MANETs for example, security, performance, Quality of Services (QoS) and application deployment and implementation.

2.2.3 MANET Applications

While the first MANET applications and deployments have been in the tactical domain, non-military applications have also started to spread considerably since then. Particularly in the past few years, as a result of the flourish of new mobile devices as well as rapid advances in wireless communication, MANETs have gained a substantial attention and interests from commercial industry, as well as the standards community (Sarkar *et al.*, 2007). Correspondingly having new technologies such as Bluetooth, IEEE 802.11 and HYPERLAN greatly drives the deployment of MANET technology away from the military domain, and new ad hoc networking applications are beginning to emerge in specialised fields for example space, health-care, emergency services, disaster recovery and environment monitoring. This is owing to the fact that mobile ad hoc networking can be deployed anywhere where there is no support of a fixed infrastructure for communication or the existing infrastructure is not cost-effective to utilise.

Furthermore, the MANET flexibility and its capabilities enable this technology to suit diverse applications ranging from large-scale, mobile, highly dynamic networks, to small, static networks that are restricted by power sources for example, in personal area networking, home networking, law enforcement operation, search-and-rescue operations, commercial and educational applications (Jun-Zhao, 2001; Guarnera *et al.*, 2002; Chlamtac *et al.*, 2003; Hoeper and Gong, 2004). Table 2-1 presents some categories of current and possible future real scenarios for MANETs, as well as the services they may provide in each domain.

DOMAIN	APPLICATION	DESCRIPTIONS/SERVICES
Military	Tactical Networks	▪ Communication in army operations. ▪ Automated battlefields.
	Emergency & Health Care Services	▪ Search and rescue operations, as well as disaster recovery; e.g. - early retrieval and transmission of patients.

Civilian domain (i.e. non-military)		▪ An alternative to replace a fixed infrastructure in case of earthquakes, hurricanes, fire etc. ▪ Monitoring patients.
	Transportations & Vehicles	▪ Broadcast of news, road condition, weather, and music. ▪ A local ad hoc network with adjacent vehicles for road/accident guidance. ▪ Advertising location specific service, like petrol stations. ▪ Particular services for providing a travel guide for vehicles on the street.
	Commercial & Business Services	▪ E-Commerce: e.g. - Electronic payments from anywhere (i.e. taxi). ▪ To provide dynamic access to customer files stored in a central server on the fly. ▪ To offer consistent databases for all agents. ▪ Shared Email and internet Gateways.
	Education & Academia	▪ To facilitate setting up e-class or e-conference (e.g. video conferencing). ▪ To establish ad hoc communication during conferences, meetings, or lectures taking place in the same area (e.g. spreading hand-outs). ▪ To offer an extra way for student to get access to certain services offered by university (e.g. internet and email gateway) or interact with other available e-learning applications like student study profiles, enrolment, etc.

	Entertainment	▪ Multi-user games ▪ Robotic pets ▪ Outdoor Internet access
	Home and Workplace	▪ Smart homes for controlling different appliances. ▪ Home/Office Wireless Networking (WLAN), e.g. - shared whiteboard application; use printing facilities anywhere. ▪ Personal Area Networks (PANs).
Space	Space missions	▪ Control of unmanned robots. ▪ Spaceship, shuttle and satellite communications.
Nature	Sensor Networks (Akyildiz *et al.*, 2002)	▪ Environmental applications include tracing the activities of animals (e.g. birds and insects), chemical/ biological detection, vegetation etc. ▪ Tracking data highly correlated in time and space, e.g. - remote sensors for weather, earth activities and disasters.

Table 2-1 : MANETs Application Domains

2.2.4 MANET Challenges (Research Areas):

The unique properties and constraints of MANETs as well as the variety of its applications as discussed above, presents several crucial challenges to MANET design and deployment that must be addressed to fully harvest MANET benefits. These challenges however instigate a substantial body of research in the MANET domain in order to resolve its critical issues and facilitate the design and operation of the networks. In this regards; prevalent challenges and related research areas are briefly reviewed within the MANET domain as stated in Jun-Zhao (2001),

Chlamtac *et al.* (2003), Conti and Giordano (2007), Cordeiro and Agrawal (2011) and Goyal *et al.* (2011).

● **Wireless Channels & Medium Access Control Protocols:** MANETs normally rely on current standard wireless technologies for interconnection (MAC single-hop communication), for example, the IEEE 802.15.4 (also known as Zig-bee) for short-range low data rate (< 250 kb/s) networks, Bluetooth (IEEE 802.15.1) for personal area networks (PANs), the 802.11 standards for high-speed medium-range MANET, and the 802.16 standards suite for high-speed wide-range. On the other hand, these standards are in fact not originally developed for multi-hop ad hoc networking. There are several constraints when operating in an ad hoc mode (e.g. hidden terminals). Accordingly the performance can be degraded especially in large-scale scenarios, and users in this situation may be deterred using MANETs. Improvements in antennas, signal processing schemes and software defined radio are expected to help to enhance the performance and reliability of current wireless technologies (Chlamtac *et al.*, 2003; Conti and Giordano, 2007).

● **Routing:** As the topology of a MANET may change frequently due to nodes mobility and churn, establishing a routing infrastructure to enable any pair of nodes to communicate becomes a troublesome issue. There are different types of routing protocols that have been developed for providing effective and efficient routing in MANETs not only for single hop communication but also for multi-hop communication (Eriksson *et al.*, 2005; Sarkar *et al.*, 2007). They are grouped into six main categories: reactive, proactive, hybrid, location-aware, energy-aware and multicast routing protocols. For more example, see Conti and Giordano (2007).

● **Power Management:** Power management is a very critical issue in MANETs since most of the network nodes are light-weight mobile terminals powered by limited sources like batteries. Additionally, they have the dual role that each node takes (i.e. router and host roles). Therefore, it is imperative that communication-related utilities should be enhanced for reducing power consumption. Energy saving and power-aware routing should be taken seriously into account (Sarkar *et al.*, 2007).

- **Quality of Service (QoS):** Offering different quality of service levels in a dynamic environment like MANETs is not an easy task. Also, the inherent stochastic and unreliable nature of communications quality in MANET has a negative impact on providing fixed guarantees on the services offered to a device. Hence, aware or adaptive QoS must be developed over the traditional resource reservation to support the multimedia services (e.g. voice and video). On the other side, QoS in MANET is still an immature research area. Issues of QoS robustness, QoS routing policies, algorithms and protocols with multiple priorities are necessary to be investigated (Cordeiro and Agrawal, 2011).

- **Security:** Security concerns are considered to be the main obstacle that makes companies hesitant to fully adopt any particular technologies. MANET security issues need to be addressed and handled thoroughly. In addition to the common vulnerabilities of wireless networks, MANETs are vulnerable to other different attacks like relay, black hole , Sybil attacks and also have no clear line of defence (Yang *et al.*, 2004; Djenouri *et al.*, 2005; Carvalho, 2008). This is due to the open wireless medium used and a number of specific constraints in MANET properties for example limitation in resources capability, lack of physical protection, and the variances in security requirements of tackled MANET's applications. However, because of MANET mobility and infrastructure-less-ness, MANETs require presumably distributed operation for different schemes of authentication and key management. As the core of this research primarily addresses security and relevant topics in MANETs, this issue will be elaborated later in this chapter.

- **Inter-networking and Interoperation:** In some given cases, a number of MANETs is anticipated to communicate with other conventional networks (e.g. IP- based networks). However, defining the interface that facilitates interoperating between the two different networks is problematic. Also, adapting routing protocols in such a mobile device to work with other types of networks introduces another challenge as MANETs, unlike traditional networks, have a dynamic topology caused by mobility (Goyal *et al.*, 2011). On the other hand, much research on current wireless networks is mainly devoted to offer seamless

integration of all types of networks. Therefore, MANET design should cater for compatibility with other types of network, for example, wireless LANs, 3rd Generation (3G) and (4G) cellular networks (Cordeiro and Agrawal, 2011).

Eventually, although MANETs have great promise in terms of variety of applications and attractiveness of their characteristics, they nevertheless are confronted by several critical challenges, as shown clearly above. Each of these challenges can be deemed as a separate research topic that requires in-depth investigation. On the other hand, these six subjects discussed above are still considered as very prominent and broad topics in the field of MANET research.

2.3 MANET-related Security Fundamentals

Compared to the conventional networks, the distinctive facets of MANETs like open collaborated network architecture, shared wireless medium, stringent resource constraints, and highly dynamic network topology instigate many more nontrivial security concerns and threats. These concerns obviously make a demand for developing multi-fence security solutions that satisfy both wide protection and desired network performance. Accordingly, in order to perceive security challenges and dimensions in MANETs, it is important to shed light on the most common security fundamentals relevant to the security MANET domain. This section mainly not only covers most general principle of security in any networking systems but also indicates to the other important security issues related to MANET. It starts with addressing the key security requirements known as security services and the potential attacks in MANETs. In addition, in this section a number of important related security techniques required to fulfil particular security requirements and prevent specific attacks are discussed.

2.3.1 Security Requirements (Security Services)

When tackling security of a given networking system like MANETs, security requirements must be addressed as a vital element in order to achieve a standard level of protection from any particular attacks or threats that a networking system encounters. However, satisfying all those requirements in a networking system is not an easy task. Security experts therefore have identified a set of key requirements that cover the major security needs of any systems. These are also identified by well-known organisations such as, the International Telecommunications Union (represented by their ITU-T Recommendation X.805 and X.800 (ITU, 1991, 2003)) and the International Organization for Standardization (by their standard ISO 7498-2 (ISO, 1989)). The key requirements includes confidentiality, integrity, non-repudiation, authentication, authorisation, availability (Djenouri *et al.*, 2005; Stallings, 2010).

- **Confidentiality** means that messages or data should be protected from any unauthorised disclosure during transmission. In other words, confidentiality ensures that the content cannot be interpreted by unauthorised entities. Confidentiality can be implemented by using any of the well-known encryption methods (e.g. symmetric and asymmetric ciphers).

- **Integrity** ensures the correctness or accuracy of data or message content during transmission. These must be protected against unauthorised alteration such as, deletion, injection and replication and it must be indicated. Technologies of digital signature and hash functions normally are being used to achieve integrity.

- **Non-repudiation** guarantees the identity of the sender of a service request (e.g. proof of obligation, intent, or commitment; proof of data origin; delivery and submission of a request). This can be delivered by using digital signature on service invocation to refer to the authentic signer.

- **Authentication** validates the claimed identities of entities that are communicating (e.g., device, service or application) and to make sure that no

entity is attempting a masquerade or an unauthorised replay of a preceding communication. This particular requirement can be considered as the first line of defence against intruders as most other security requirements (e.g. confidentiality, integrity etc.) rely on authenticated entities for their deployment. There are two types, peer entity authentication and data origin authentication. Peer entity authentication caters for validating the identity of a peer entity in an association while data origin authentication is to assure the source of a data unit being received.

- **Authorisation (Access control)** is a means to prevent unauthorised use of network resources. In other words, access control only allows legitimate personnel or devices to use stored information, services and applications by managing the level of access to those resources. Clearly, access control is tied to the authentication requirement. There are a number of approaches to access control most common for example, Role Based Access Control (RBAC), Discretionary Access Control (DAC), Mandatory Access Control (MAC) and Attribute Based Access Control (ABAC).

- **Availability** ensures that the provision of access to network services is available whenever they are required, even the services are under attacks (e.g. denial of service DoS). Therefore this requirement can be implemented by a proper management and control of system resources (i.e. access control services and other security services).

2.3.2 Security Threats and Attacks in MANETs

Due to the open and shared operation medium of MANETs, and also the lack of any central administration or clear line of defence in this particular type of networks, MANETs tend to be more vulnerable to security threats and attacks than any other conventional networks like wired ones. While security threats and attacks in MANETs are not the core of this work, the following is intended to

show the different forms of attacks and suggests that there is no uniform blanket approach to MANET security.

The potential attacks of MANETs can be split into two main streams according to their nature: passive attacks and active attacks. A ***passive attack*** can be identified when an adversary gains access to a protected asset of the network but without modifying any content of that asset or can track and learn about activities within the network but without disrupting the operation (e.g. eavesdropping and traffic analysis). Avoiding such passive attacks is very challenging because the network operations and resources are not touched to indicate that there is an attack (i.e. attacks may be undetectable). The appropriate way to overcome these attacks is to make use of encryption methods which normally protect the message content being transmitted, so that it becomes hard for eavesdroppers to gain any information about what is being transmitted. However an ***active attack*** can be recognised when an adversary attempts to temper message or data contents (e.g. modification, injection and deletion) being exchanged in the networks. Active attacks usually have several forms: masquerading (i.e., impersonation and man-in-the-middle attacks), replay (i.e., retransmitting messages), jamming, message spoofing, message modification, and denial-of-service (DoS) (i.e. causing excessive resource consumption in the network). These attacks can be alleviated by using security mechanisms, for example encryption techniques, firewalls and intrusion detection systems, etc. (Djenouri *et al.*, 2005; Abusalah *et al.*, 2008; Cho *et al.*, 2011).

On the other hand, since most MANET nodes usually have a dual role of network operation, both types of attacks discussed before can be performed by either *insider* or *outsider* nodes. However, attacks originating from insiders are considered much more damaging and difficult to prevent when compared with outside attacks. This is due to the fact that the insider typically owns valuable and secret information (e.g. access privileges) about the network system which can be exploited to disrupt the network easily (Wu *et al.*, 2007a; Cho *et al.*, 2011).

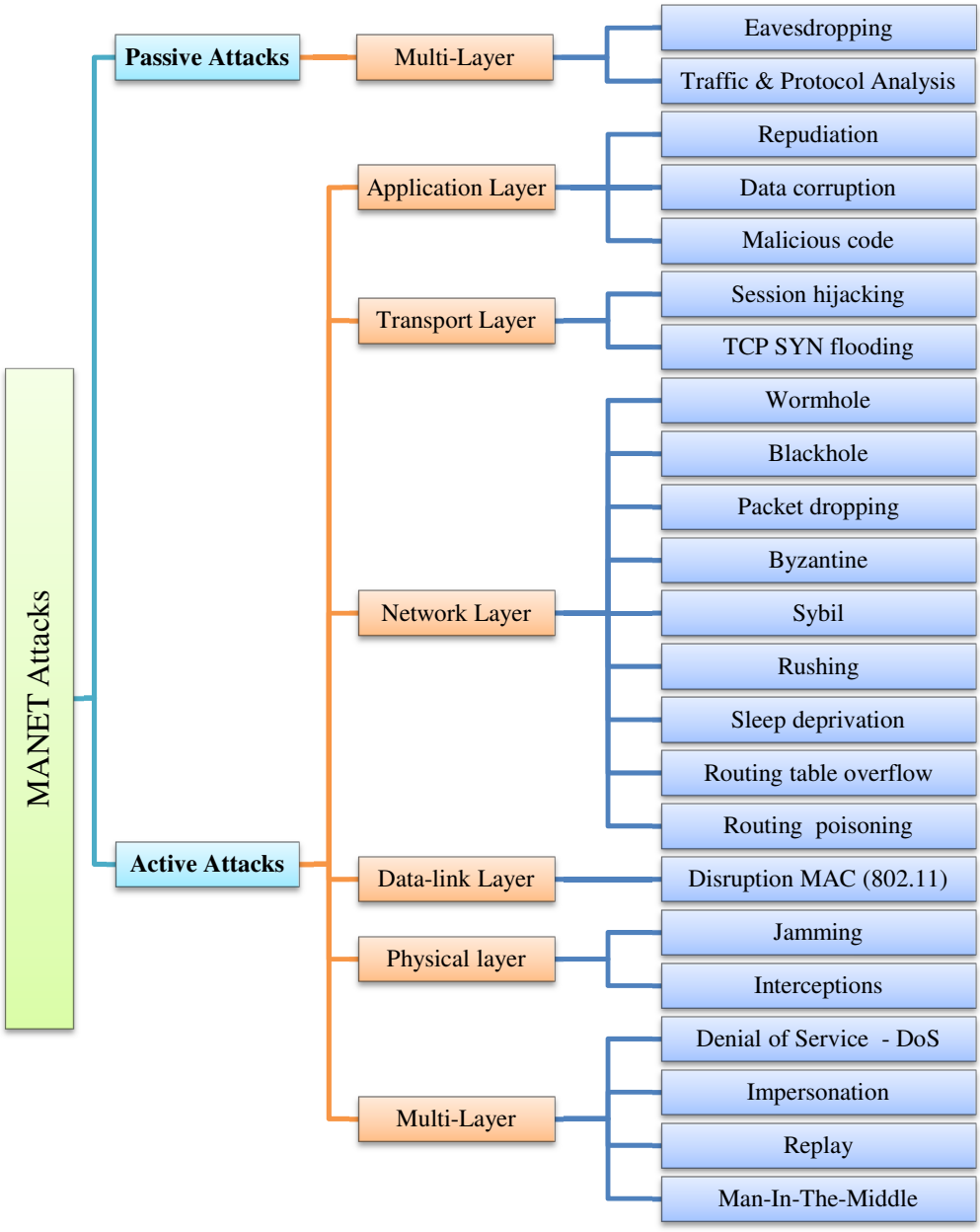


Figure 2-3: The MANET Attacks Categories (Wu *et al.*, 2007a; Cho *et al.*, 2011).

To realise the variety of threats that a MANET may encounter, Figure 2-3 outlines a representative, but not exhaustive list of attacks occurring in MANETs. Those attacks are arranged based on their types (active or passive) and also according to where they can be launched in the OSI model (either in one specific layer or in multilayers). For more detail about those attacks, see Cho *et al.* (2011)

and Wu *et al.* (2007a). Last but not least, MANET threats become an important factor that has an impact on how to design and deploy any security solution for MANET. Therefore it is required to take those threats into account seriously in the proposed security solution for more protected MANETs otherwise this will make this security solution ineffective in term of attack resistance.

2.3.3 Security Techniques (Cryptographic Fundamentals)

Cryptography is known as the art and science of secrecy (Scheneier, 1996). This science basically exists to facilitate how to manage, store, transmit sensitive data securely across insecure networks (e.g. internet) so that no one can intercept or interpret it except the intended recipient. On the other hand, the main mission of cryptographic techniques are presumably to meet the requirement of confidentiality, integrity, authentication, authorisation and non-repudiation and also to protect from any potential attacks (Scheneier, 1996; Stallings, 2010). Several security techniques can be recognised in the knowledge body of security. However, this study covers only those pertinent techniques which contribute to the main goals of this study. Briefly the next subsections exclusively describe those techniques in details: encipherment, hash functions, message authentication codes (MACs), digital signatures, authentication exchanges, digital certificates and threshold cryptography. Also, to understand the relationship between these prescribed security techniques and the key security requirements, Table 2-2 primarily presents how using a specific security technique leads to satisfy certain security requirements annotated by "*Y*".

Security Techniques \ Security Requirements	Confidentiality	Integrity	Non-repudiation	Authentication	Authorisation	Availability
Encipherment	*Y*					
Hash Functions		*Y*				
Message Authentication Code		*Y*		*Y*		
Digital Signature		*Y*	*Y*	*Y*		
Authentication Exchange				*Y*		*Y*
Digital Certification				*Y*	*Y*	
Threshold Cryptography	*Y*	*Y*	*Y*	*Y*		*Y*

Table 2-2: Security Techniques Vs Security Requirements, "*Y*"= Yes

2.3.3.1 Encipherment

In cryptography, to provide confidentiality features, plaintext is transformed into cipher-text by using an encryption cipher while an intended recipient is able to decrypt the cipher-text back into plaintext by a decryption cipher. The encryption and decryption are normally characterised and controlled by security keys which are considered as an active element of running any cryptographic algorithms, and must often be kept secret in particular cases. There are two primary types of cryptographic algorithms: symmetric and asymmetric key algorithms. Symmetric key algorithms (e.g. AES) make use of the same secret key for encryption and decryption, whereas asymmetric key algorithms (e.g. RSA, ECC) make use of two different keys, public and private keys for encryption and decryption. On the other side, in term of processing cost, symmetric key ciphering is less expensive than asymmetric. A symmetric key cipher is therefore suitable for use when there is a limitation in resources being used (i.e. limited power and processing capacity). Typically the asymmetric method is used only to exchange the shared secret (Smart, 2003; Stallings, 2010).

2.3.3.2 Hash Functions and Message Authentication Code

A cryptographic hash function is used to apply a one-way compression function on a block of data of any size to convert to an output of fixed length *n* (i.e. called a data digest). The common purpose of using hash functions is to provide a measure of data integrity but hash functions can be incorporated with other cryptographic techniques to achieve other security requirements (for example, a digital signature, a message authentication code (MAC) and hash chains for authentication). There are several families of standard hashing algorithms such as the message digest (MD) family (e.g. MD2, MD4 and MD5); the secure hash algorithm (SHA) family; (e.g. by SHA-0, SHA-1, SHA-256, SHA-384 and SHA-512) (Menezes *et al.*, 1996; Cayirci and Rong, 2008).

On the other hand, it is worth pointing out that a message authentication code (MAC) algorithm also belongs to hashing techniques. However, it uses a secret key as an input in the compression process of any standard hash functions without using any particular type of encryption (e.g. HMAC). Furthermore, it is a light-weight algorithm in terms of processing overhead and also it meets security requirements of integrity and authentication (Cayirci and Rong, 2008; Stallings, 2010).

2.3.3.3 Digital Signature

A digital signature is based on an asymmetric key algorithm (e.g. RSA or Elgamal) to be produced. In this technique, the sender initially utilises a hash function (e.g. SHA) to create the digest value of a message and then applies encryption using its private key on that digest in order to generate a valid digital signature attached with an original message. The recipient on the other end is able to verify the origin and integrity of a message by repeating hashing and decryption using the sender public key. This security technique assures that the message or data has not been tampered after running the digest calculation (integrity) and they

are authentic and come from the owner of the public key (authentication). Furthermore, digital signature techniques can be used to provide a measure of origin non-repudiation for data (i.e. the recipient of data with entity sender's signature on it can identify that the sender sent the data (which even sender cannot deny)). There are a number of standard algorithms for the digital signatures, for example, RSA signature, the digital signature algorithm (DSA) or Digital Signature Standard (DSS), etc. (Scheneier, 1996; Smart, 2003).

2.3.3.4 Authentication Exchange

Authentication exchange techniques, often referring to authentication communication protocols, are defined as a series of handshakes for cryptographically protected messages in order to enable two communicating entities to validate one another's identity (i.e. to accomplish entity authentication) mutually or unilaterally (known as mutual or one-way authentication). Furthermore, these exchanges may be exploited to establish a secure connection either by transporting secret keys from each entity or by deriving session keys between each other as an additional feature to the authentication process (Boyd and Mathuria, 2003). On the other hand, depending upon the type of authentication exchange, these techniques usually need different cryptographic methods like encipherment, digital signature or integrity mechanisms for protecting the messages being exchanged. Therefore, authentication exchange techniques are split into three main categories according to underlying cryptographic primitives used: symmetric and asymmetric and hybrid (symmetric and asymmetric) authentication protocols. Besides, in a given authentication technique, non-cryptographic mechanisms (typically time-stamps or random nonces) can be involved during messages exchange so as to confirm the freshness of messages in authentication for thwarting such particular attacks like a replay attack (Menezes *et al.*, 1996).

As implied, authentication protocols mainly aim to fulfil authentication requirements yet on the other side, especially when dealing with a limited network bandwidth, these protocols can contribute to availability requirements (i.e. service availability) through the number of protocol messages being exchanged in a given network and the size of these messages' contents being piggybacked. In other words, the lower the number of messages and the smaller the size of the authentication protocol messages, the less load on the network (i.e. more available) avoiding such a denial of service (DoS) attack.

There are several prominent examples of authentication exchange techniques that have been standardised for adoption and interoperability purposes like SSL/TLS (Dierks, 2008) or an one-way pass, two-way and three-way passes authentication protocols in (ISO/IEC 9798-2, 2008) — using symmetric encryption and (SO/IEC 9798-3, 1998/Cor.1:2009) and X.509 (ITU-T, 1989; Chokhani *et al.*, 2003; ITU-T, 2008) — using public-key encryption, etc..

Finally, the use of authentication exchanges is very demanding as it is a key part of the provision of security in any circumstances where the medium of communications in a particular network (e.g. MANETs) is unreliable and vulnerable to attacks. Therefore, it can be considered as a first line of defence against adversaries trying to fraudulently access services in any network; use of this technique in the network becomes inevitable (Menezes *et al.*, 1996; Stallings, 2010).

2.3.3.5 Digital Certificates

Relying upon public-key cryptography, a digital certificate is typically developed as a form of identification to support the authenticity of public keys as well as the identities being used during communication in order to prevent from some particular attacks like man-in-middle. The certificate is generally defined as an electronic document which makes use of a digital signature to bind together a public key with identity-information of the certificate owner, for example, a name

of a person or an organisation, their address, emails or attributes, etc. In addition, there are two types of digital certificate: a public-key certificate and an attributes certificate. The certificate according to its type is assumed to be issued by some trusted certification authority (CA) or attribute authority (AA) which enables any interested party to check the integrity of the certificate by verifying the signature of that authority. This is shown in Figure 2-4 and represents an example of a public-key certificate.

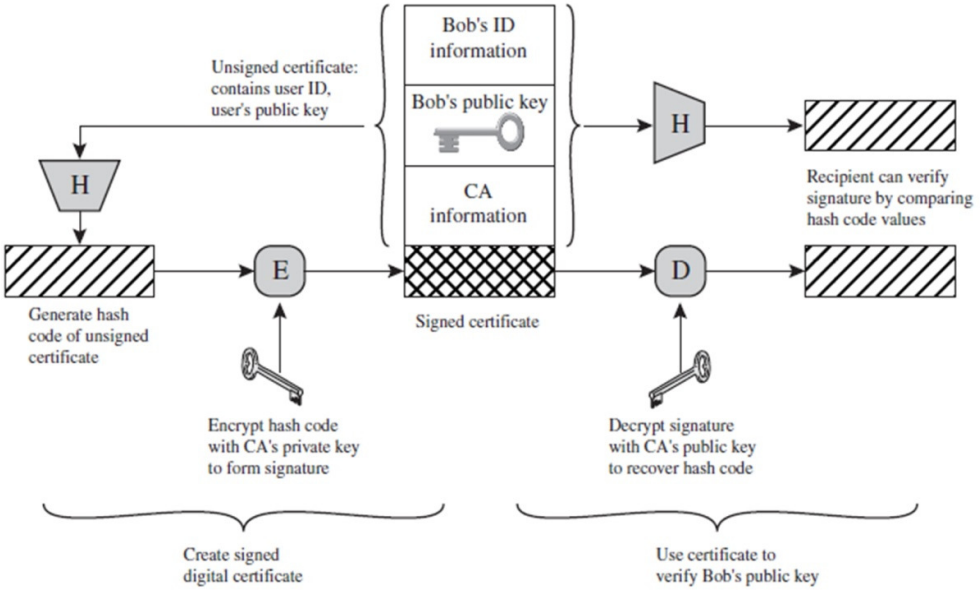


Figure 2-4: The Digital Certificate Generation and Verification (H: a hash function; E: encryption; D: decryption) (Stallings, 2010, p.430)

It is important to note that digital certificates can be incorporated with other security techniques like authentication exchanges, digital signature, hashing functions or asymmetric encryption to leverage the most requirements of integrity, confidentiality, authentication, and authorisation. On the other hand, digital certificates take several forms depending specifically upon a trust model being adopted, for example, X509 (Chokhani *et al.*, 2003; Cooper *et al.*, 2008; ITU-T, 2008) or Pretty Good Privacy (PGP) (Abdulrahman, 1997). The X509 framework is a commonly used standard based on a the Public Key Infrastructure -PKI

scheme for certifying and revoking digital certificates via a third trusted party (TTP) so-called CA whereas PGP is similar but relied on Web of Trust to generate certificates via trusted users themselves. Both models will be discussed in the subsequent sections of this chapter.

2.3.3.6 Threshold Cryptography (TC)

Initially, the notion of *threshold cryptography* (TC) (Desmedt and Frankel, 1990) is constructed as a result of emerging and applying secret sharing techniques first developed by Shamir (1979) for a confidentiality purpose among a group of participants. In Shamir's (1979) secret sharing proposal, the *k* out of *n* scheme *(k* $\leq n$*)* is essentially defined as a cryptographic technique which enables to break a secret *S* into n different shares S_i ($1 \leq i \leq n$) according to a random polynomial used so that the knowledge of at least *k* shares is necessary to recover the initial secret *S* by means of Lagrange interpolation.

With the properties of better fault tolerance without increased risk, threshold cryptography can offer an approach to facilitate trust distribution and control sharing for critical activities (e.g. signing docs) by enabling *k* of *n* parties to perform the critical action cooperatively. In other words, this cryptography can usually be exploited to distribute the duty among a number of trusted entities so as to provide a given cryptographic service in collaborative manner. As the public key cryptosystem becomes very prominent with many standard algorithms (e.g. RSA, DSA, ElGamal, ECC, etc.), many schemes are proposed to cater for incorporating this technique with public-key cryptography particularly in order to generate a digital signature by multiple signers, so-called the *threshold signature* such as Desmedt and Frankel (1992), Wang *et al.* (1998), Gennaro *et al.* (2000), Shoup (2000), Kong *et al.* (2001), Saxena *et al.* (2003, 2007), Kim *et al.* (2011) and Dossogne *et al.* (2013).

On the other side, most schemes based on *threshold cryptography* mainly consists of two main dimensions of implementation: a sharing management

dimension where shares of the secret *S* are distributed, updated and verified among participants, and a usage dimension (i.e. a computation protocol) where a distributed protocol makes use of these shares jointly so that the function of the secret *S* can be applied, for example signing partial certificates using private shares of a private key in public key cryptosystem (Dossogne *et al.*, 2013).

In addition to a confidentiality requirement being fulfilled by secret sharing, *threshold cryptography* can also adhere to the other security requirements such as integrity, authentication and availability. Furthermore, a system running any threshold cryptographic schemes *TC(k, n)* is characterised with good security robustness and high availability since it depends on multiple players (*n*) to function and adversaries are required to compromise *k* out of *n* active players of a system in order to break the whole system. To conclude, the area of threshold cryptography is considered be different from the area of secret sharing as its aim is to perform a cryptographic function of the secret *S* without disclosing the actual secret (i.e. never reconstructing the secret *S* from its shares $< s_1, .., s_n >$ but rather employing those shares as a shared input into that cryptographic function). However, the sharing techniques are usually the same in both areas.

2.4 The Trust Management

There are several interpretations of "Trust" differing from one domain to another. One view taken from the social sciences is the degree of subjective belief about particular entity's behaviour (Cook, 2001). It is arguably fundamental to the security design in networks like MANETs. Furthermore, trust becomes a vital element which the most common security services (e.g. authentication and authorisation protocols, confidentiality, etc.) require in their deployment. Throughout this study, only in-network trust elements which are managed and maintained within MANET nodes are presented.

Alternatively, trust which aims to help in establishing a relationship between entities usually can be recognised in different forms. As stated in Aivaloglou *et al*

(2006), Li and Singhal (2007), Yunfang (2007), these forms can be classified into two key categories as shown in Figure 2-5 with a number of examples: **(1)** the so-called *credentials-based trust* (e.g. certificates, and keys, etc.) and **(2)** the *monitored behavioural trust*,(e.g. reputation and recommendation (Louta *et al.*, 2010)). In the first case, the security frameworks that have adopted credentials-based trust, typically exploit credentials like certificates for pre-deployment knowledge of trust relationships within the network nodes. Also, these credentials must be disseminated, maintained and managed, either independently or collaboratively by the nodes. On the other hand, having a valid credential is an indispensable criterion that trust decisions rely on and also this is to confirm trustworthiness of the target node by a credential authority (e.g. CA) or by other nodes that the issuer trusts.

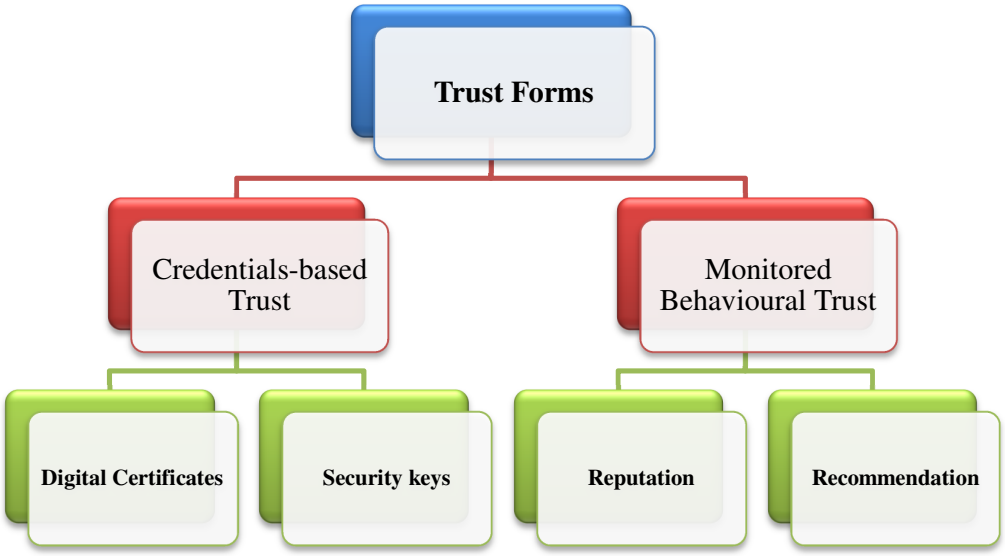


Figure 2-5: Trust Forms: Credentials-based Trust & Monitored Behavioural Trust with their examples

In contrast, in the second case of monitored behavioural trust frameworks, each node plays the role of constantly observing the behaviour of its neighbouring nodes so as to evaluate trust which is usually perceived as a reputation (i.e. computational trust) among nodes. A reputation of a particular node being

monitored evolves increasingly or decreasingly according to the experience and perception of other nodes about that node, which is appraised from voting, rating and recommendation (Cho *et al.*, 2011). On the other side, it is worth pointing out that the monitored behavioural framework is considered as a reactive approach, working under the assumption that nodes' identities in the network must be verified via pre-installed authentication techniques. Since MANETs primarily depend on the cooperation of the most nodes, their monitored behavioural trust framework proposed in so-called cooperation enforcement schemes (Marias *et al.*, 2006; Louta *et al.*, 2010) differs from the other same frameworks with the extra effort of alleviating problem of node selfishness. Therefore, the reputation of a MANET node can build up as long as it performs correctly the tasks of route discovery and data forwarding. Otherwise, in case the node misbehaves either through accessing network resources in an unauthorized way or not cooperating in routing, it will be regarded as a malicious or selfish node and be isolated as a result of misbehaviour detection. For more example about MANET cooperation enforcement schemes, they are well presented in Marias *et al.* (2006) and Louta *et al.* (2010).

Last but not least, in the communication and networking field, it is important to realise that trust management is considered as a generalised approach which is supposed to incorporate both credentials-based and monitored behavioural trust frameworks fully or partial in order to manage trust effectively. According to Cho *et al.* (2011), trust management typically must tackle the issue of the formulation of evaluation rules and policies, representation of trust evidence, and evaluation and management (i.e. issuance and revocation) of trust relationships among nodes. Additionally, trust management must be concerned with collecting the information which is essential to form a trust relationship and also dynamically monitoring and regulating the existing trust relationship (Li and Singhal, 2007). Trust management can be involved in diverse security issues such as intrusion detection, authentication, access control, key management, and isolating misbehaving nodes for effective routing.

The dynamic topologies and constrained resources of MANETs, mean that trust management comprising of trust establishment, trust update and trust revocation encounters more exceptional challenges than in traditional centralised settings. For example, acquiring trust evidence so as to evaluate trustworthiness is problematic owing to frequent changes in topology instigated from node mobility or node failure. Furthermore, resource and power limitations in MANETs also become an obstacle to process trust evaluation efficiently. In MANETs, there are a few attempts to propose a trust management scheme that normally integrates the two types of trust approaches (credentials-based and reputation-based) partially or fully such as Buchegger and Le Boudec (2002), Hadjichristofi *et al.* (2005a; 2005b), Yunfang (2007) and Toubiana and Labiod (2008). However, this study in MANETs is dedicated primarily to investigate and evaluate the current credential-based trust frameworks as shown in this thesis and the domain of reputation models are out of scope of this study. In next section, the state of art credential-based trust models in MANETs and their important relevant issues will be characterised and described with more details.

2.5 The Credential-based Trust Models In MANETs

As discussed in the previous section, the credential-based trust models are a main part of a trust management system in MANETs. These are deemed as one direction towards specifically facilitating the authentication and credential management processes. This is because these models essentially aim for establishing, maintaining and managing the trust relationship among nodes using security credentials and therefore they are able to interact with each other securely (i.e. authentication and authorisation). There is a variety of trust models proposed in the literature which MANETs may take advantages of.

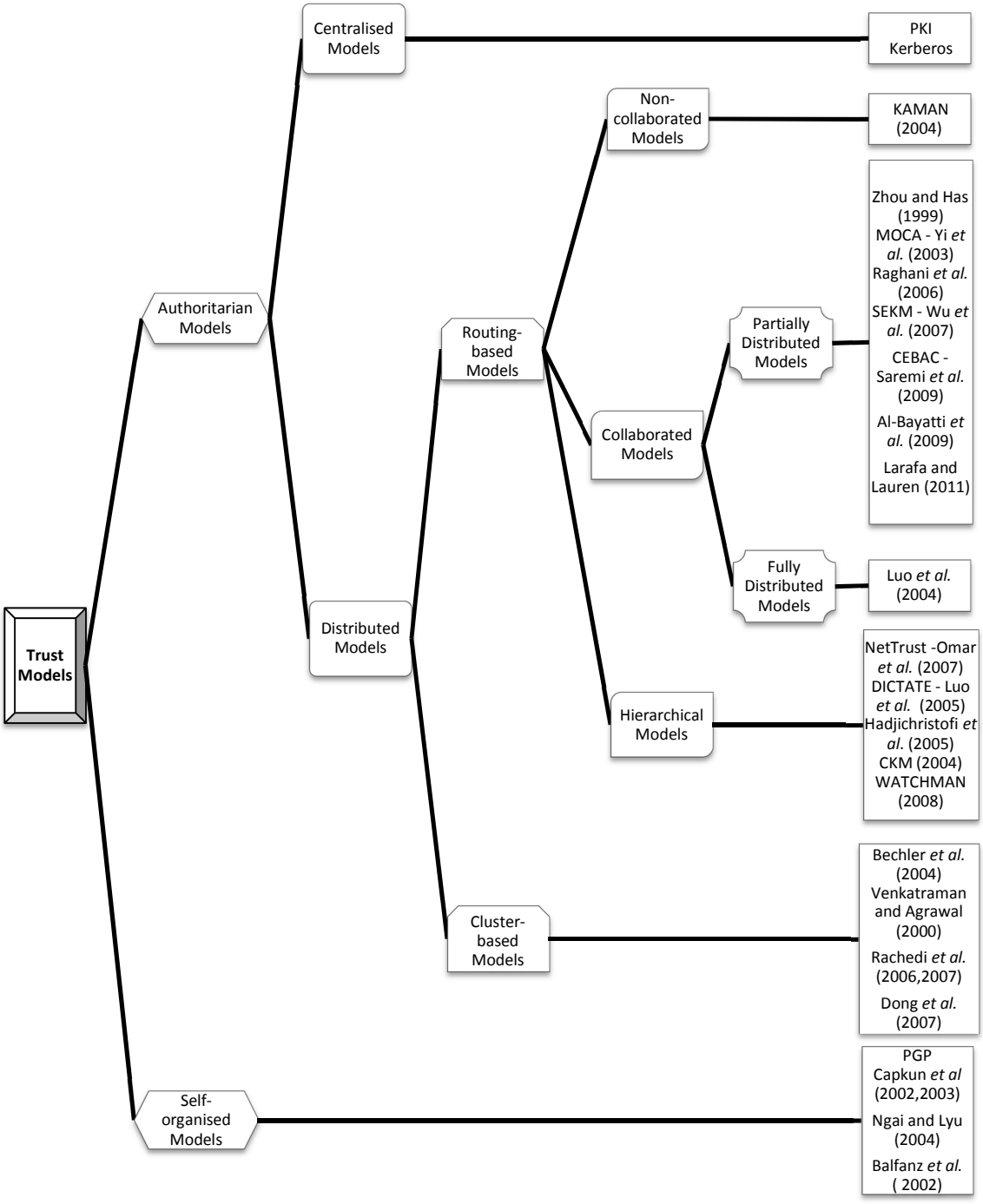


Figure 2-6: The categories of security trust models with their proposals

To recognise the differences of these models regarding credentials management they are split into two main classes, authoritarian and self-organised models and their related subclasses along with the state-of-art proposals as shown in Figure 2-6. These categories are defined according to certain criteria and characteristics such as Trust Third Party (TTP) reliance, a security architecture type, connectivity infrastructure and a cryptographic technique. All categories will be described in more detail in the following sections.

2.5.1 The Authoritarian Models

In an authoritarian trust model, every node generally relies on a special trusted entity a so-called Trust Third Party (TTP) (Menezes *et al.*, 1996) for establishing trust with other nodes within the network in order to enable nodes to interact with each other (i.e. accessible). In other words, a TTP is normally based on a certain authority-based infrastructure (e.g. CA, AA, KDC, AAA, etc.) which plays a role of organising trust among nodes through offering particular services of issuing, maintaining, updating and revoking credentials. As shown in Figure 2-7, there are three different approaches for the TTP involvement: **(a)** inline**, (b)** online and **(c)** offline. Both inline and online approaches require TTPs to be accessible to other nodes during a normal network operation phase whereas an offline one is opposite or may make use of another type of out-of-band channels for communication (e.g. physical contact, location-limited side channel (Balfanz *et al.*, 2002), etc.). However, an inline TTP involvement is distinguished from online one by the fact that TTP becomes an intermediary, facilitating communication between nodes. Correspondingly, Hoeper and Gong (2004) suggest four different cases for availability of TTP in MANETs according to the network life phases (i.e. network initialisation and running): TTP always available, TTP available at network initialisation stage and when a node joins, TTP only available at network initialisation stage, No TTP available at any network stage. Besides each approach of TTP involvement may allegedly entails new requirements and also may need additional configurations and installations before making the network fully

operational. To select the appropriate approach of TTP involvement in any network, it is important to carefully take into consideration some aspects of scalability, availability, reliability, conformity and localisation. In the current research of MANET trust, although MANETs have dynamic topologies and limited power resources, most authority-based trust models are still in favour of an online approach and few are adopted offline or both such as Martucci *et al.* (2004) Verma et al (2004), Luo *et al.* (2005) and Hadjichristofi *et al.* (2005a).

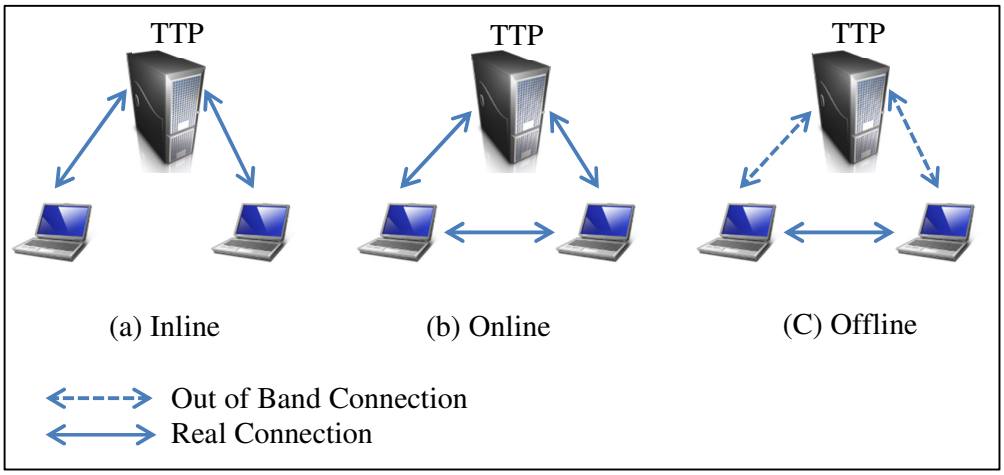


Figure 2-7: The approaches of TTP involvement

Eventually, depending upon the TTP architecture type, several authoritarian models for MANETs can be organised into two mainstreams: centralised and distributed models. In the former, TTP depends on only on a central node to manage trust of the whole network whereas the role of TTP must be distributed among a number of specific in the latter. Additionally, since the category of distributed models relies certainly on different techniques for applying distribution (collaboration, replication, hierarchy, dependency and routing infrastructure), there are four proposed types for the distributed models that can be recognised in this regard: non-collaborated, collaborated, hierarchical and clustered-based models. In the next subsections, as presented in Figure 2-6, all cases relevant to centralised and distributed models will be explained.

2.5.1.1 Centralised Models

Public Key Infrastructure (PKI) based on an asymmetric cryptosystem is still acknowledged as one of the most effective and well-known models for managing digital certificates (e.g. issuance, revocation and distribution) to trusted peers. Those certificates like X509v3 (Chokhani *et al.*, 2003; Cooper *et al.*, 2008) are essentially used for deploying security services, such as authentication, authorisation and digital signatures and encryption (Perlman, 1999). PKI relies on TTP, the trusted entity in the system like what is a so-called CA (Certification Authority) and it is considered a central element of a PKI. On one hand, the typical mission of CAs is to create certificates and digitally signs them using the CA's private key. On the other hand, by using the CA's public key, any entity demanding to validate a certificate's authenticity verifies the CA's digital signature, as well as confirms the integrity of the contents of the certificate.

In addition, PKI has originally been developed for wired networks and some infrastructure-based wireless networks where there is no serious problem in connectivity and availability, It is noticeable in the PKI domain that security and scalability issues of CA are considered very crucial as CA is required to handle a large number of requests effectively. On the other side, it is important to indicate that most of other trust models for MANETs discussed in the study obviously lean on PKI fundamentals.

Similar to PKI, Kerberos (Neuman and Ts'o, 1994; Neuman *et al.*, 2005) developed by MIT is a centralised TTP-based trust model and relies on a symmetric cryptosystem. The TTP is here referred to a KDC (Key Distribution Centre) which holds all shared secret keys for user and server principles. The role of a Kerberos service is to become the trust reference for supporting authentication and authorisation processes in the system. Kerberos shares different key-secrets with each entity in the network and it creates trust tokens (i.e. includes session key, a validity period thwarting replay attacks and the requesting node's

identity encrypted with the server's secret key) which may be used later between users and servers.

2.5.1.2 Distributed Trust Models

In this type of models for sake of fault tolerance, power saving and trust management facilitation, the role of TTP must be shared between different pre-determined nodes in order to manage trust either independently or dependently. However, since these models rely on several nodes to fulfil the purpose, it is required to have a form of a routing infrastructure to make those nodes accessible. Therefore, the distributed trust models in regards can be classified into two classes. The first one, a so-called routing-based model is based on a common MANETs unicast routing protocol (e.g. AODV, OLSR, DSR, etc.) while the second one takes advantage of some clustering algorithms to establish a particular connectivity infrastructure within intra- and inter- clusters for MANET along with handling trust (i.e. cluster heads may take the role of TTPs).

2.5.1.2.1 Routing-based Models

This section presents the specific distributed trust models whose implementations depend on a typical routing infrastructure in MANETs like using well-known MANET unicast routing protocols (e.g. proactive or reactive routing protocols). Besides, most models of this type intend apparently to separate data routing from data handling of trust services. However, these models can be displayed in three different themes: (1) non-collaborated, (2) collaborated and (3) hierarchical models. Firstly, non-collaborated models refer to distributing a single TTP into a number of independent replicas of the original TTP and each replica carries out the same duties as the original one. Secondly, collaborated models differ from non-collaborated ones in that a group of specific nodes work together to perform as a single TTP. These models normally rely on the (*k* out of *n)* scheme in

Threshold Cryptography (TC) to achieve that cooperative distribution. Finally, in contrast to the two types of models aforementioned, hierarchical models take into account multiple levels of different TTPs; they may exploit different collaborated and non-collaborated security techniques (e.g. threshold cryptography, trust or certificate chain, replication, etc.).

2.5.1.2.1.1 Non-collaborated Models

To avoid a single point of failure, most non-collaborated models attempt to replicate a homogeneous TTP (e.g. CA, KDC, etc.) in a number of nodes which can independently act the same as the original TTP. However, those TTP replicas need to be well protected as they will become a single point of compromise. In other words, if anyone of those TTP replicas is compromised, the whole trust in the network will be jeopardised.

As an example, for adapting centralised Kerberos (Neuman and Ts'o, 1994; Neuman *et al.*, 2005) to a dynamic network like MANETs, Pirzada and McDonald (2004) have presented certain modifications to the original Kerberos protocol. In their proposed model called KAMAN (Kerberos assisted Authentication in Mobile Ad-hoc Networks), multiple Kerberos servers have been allocated in order to provide distributed authentication and load distribution. Those servers periodically and securely synchronise their databases (i.e. having session keys) with each other. If a node *N1* is interested in communicating with another node *N2*, it begins to send a request to one of the Kerberos servers. The server then generates a token to send back to node *N1*. Node *N1* makes use of the token by sending to node *N2*, which must supposedly admit the token. At that point, a secure session can be created between node *N1* and *N2*. Although this model mainly caters for distributing a load to multiple servers, it has not discussed issues like the availability of servers and the number of servers.

2.5.1.2.1.2 Collaborated Models

As indicated, this type of model should rely on a particular technique which enables a set of special homogeneous nodes to cooperatively play a role of a single TTP. This collaboration can be implemented through using threshold cryptography (TC) which is described before in section (2.3.3.6). TC is basically based on the *k* out of *n* scheme which it is enough for *k* specific nodes out of *n* predetermined nodes to do normal tasks of a signal TTP (e.g. issuing, revoking, etc.). It is important to recognise that collaborative models in this regard is characterised with fault tolerant and protection enhancement. This stems from the fact that it sounds hard for adversaries to compromise multiple TTP nodes at least *k* and on the other hand, having several cooperative TTP nodes keeps the system up running even there are some failures(<*k*). According to the way of distributing TTP, there are two types can be distinguished, partially distributed or fully distributed models.

In models where TTPs is partially distributed, only a specific number of nodes from total network nodes can be in charge to run TTP services. This entails crucial management issues about the node selection criteria (e.g. resource capacity, scalability, location, mobility, etc.). There are several proposals that are in favour of this model type. For example, Zhou and Haas (1999) have first proposed a partially distributed certification authority (CA) based on the (*k* out of *n)* scheme of Threshold Cryptography (TC). The role of CA (i.e. TTP) is distributed among specific nodes: servers, combiners, and a dealer. Servers and combiners perform signing public key certificates for users. The dealer is a particular server which holds the completely private-key certification authority. For any joining node, if all partial signatures are collected, it can then compute the complete signature locally to obtain the complete public key certificate. It is worth indicating that this model becomes a reference to other approaches adopting threshold cryptography for deploying their security service in MANETs.

The approach of Yi and Kravets (2003) is based on Zhou and Haas's (1999) solution but with a small modification by making the requesting node to be a combiner instead and also considering revocation. This approach describes how to select particular nodes according to their best physical security and capability so as to be MOCA servers (MObile Certification Authority). The communication overhead in this solution is also apparently reduced by using the technique of caching routes to MOCA servers. The system utilises unicast instead of flooding when sufficient cached routes exist.

On the other hand, in SEKM (Secure and Efficient Key Management in MANETs) proposed by Wu *et al.* (2007b), the CA trust is distributed to a group of nodes, which could be nodes with normal or better hardware in a mesh-based topology. SEKM, much the same as MOCA, is designed to offer efficient share updating among servers and to quickly reply to certificate updating. For efficiency, only a subset of the server nodes initiates the share update phase in each round. A ticket-based scheme is developed for efficient certificate updating.

Raghani *et al.* (2006) suggest a similar Zhou and Haas (1999) solution but their solution introduces an approach for how to dynamically adjust the value of the threshold when required, and by this means decreases the certification delays. Also, the proposals of Larafa and Lauren (2009, 2011), Al-Bayatti *et al.* (2009) and Saremi *et al.* (2009) which are primarily based on Zhou and Haas (1999), aims to present a trust model of distributed access control using an Attributes Authority (AA) or Authentication Authorisation Accounting (AAA) services as TTPs.

Alternatively, in the category of fully distributed models, all nodes, instead of a number of special selected nodes as in partially distributed models, participate cooperatively in establishing and managing trust within the network (i.e. act as a TTP). Even though this category shows better efficiency than the other category of partially distributed models, it encounters other concerns of self-initialisation, self-configuration, and capabilities of all nodes, security robustness and requirements of nodes density. Luo *et al.* (2004) proposed, for MANETs, a

completely distributed certification authority model, based on threshold cryptography. The model distributes the share of the authority private key among all nodes at the time when they join the network. When a new node wants to access to get its certificate, it sends a request to its k neighbouring nodes for partial certificates. If the coalition decides that the requesting node is a "well-behaved" node, they issue their partial certificates. These partial certificates are then combined together by the target node to create the complete certificate using an interpolation function. On the other hand, trust is maintained by the notion that all the nodes must observe the direct neighbours behaviour and maintain their own CRL (Certificate Revocation List). In case a node discovers one of its neighbours is dishonest, it adds its certificate to the list of revocations and disseminates through the network an accusation. If the certificate of accusatory is revoked, the accusation is ignored. Otherwise, the node is marked suspect by all the nodes receiving the accusation.

2.5.1.2.1.3 Hierarchical Models

In this subcategory of trust models, trust is implemented by a hierarchy of several homogeneous TTPs (e.g. CAs, etc.). In other words, these models offer different levels of TTPs which can be exploited in order to improve availability and scalability of trust services. However, due to the hierarchical arrangement, these models entail much more management overheads in sense of selection, maintenance, and complexity. Luo *et al.* (2005) propose DICTATE (Distributed CerTification Authority with probabilisTic frEshness for ad hoc networks). The DICTATE architecture presents a hierarchical CA between one mCA (mother CA) in wired network, and a group of dCAs (distributed CAs) in MANETs. The group of dCAs relies on the TC scheme for signing a certificate of a joining node. Nodes in MANETs can cooperatively be isolated from the mCA, but always have the need for CA's services. The mCA delegates a group of dCAs in order to increase the availability of security services when mCA is offline or out of reach.

Hadjichristofi *et al.* (2005a; 2005b) develop a key management framework that provides redundancy and robustness in trust establishment (i.e. creating security association between pairs of nodes for IPsec). Their proposal of a key management system (KMS) depends upon an adapted hierarchical Public Key Infrastructure (PKI) model where nodes can dynamically play roles of trust management. KMS can be realised by a three levels of hierarchy: Root Certificate Authority (RCA) (the first level), Delegated Certificate Authority (DCA) (the second level) and Temporary Certificate Authority (TCA) (the third level). The model aims to provide high service availability based on trust-based SA among nodes.

Omar *et al.* (2007), present a hybrid trust solution called NetTRUST (mixed NETworks Trust infrastRUcture baSed on Threshold cryptography). NetTRUST exploits two sets of particular CAs for managing PKI: central CAs (CCA) in wired network and mobile CAs (MCA) in ad hoc network. MCA servers emulate the CA role by using the TC-based (k out of n) scheme, and the CCA servers delegate the CA role to MCA servers by using the same TC-based scheme. The system leverages decentralisation, supports nodes mobility, and resists against MCA failures. This solution also introduces the usage of the standard X509-v3 certificate issued by CCA or MCA and demonstrates how to get benefits from the powerful attributes of X509-v3 in this context, for more details in Omar *et al.* (2007).

Composite Key Management (CKM) devised by Yi and Kravets (2004) consists of a multiple key management approach which incorporates TC-based distributed certificate authority (i.e. TTP) with certificate chaining based on PGP (Abdulrahman, 1997). This model relies on two principles: (1) key management can be deployed between multiple nodes and (2) a distributed TTP must be an anchor of trust for other nodes when using trust chains. As in Capkun et al (2003), issued certificates in this proposal are stored and disseminated in self-organised way. Furthermore, by using both approaches (distributed CA and certificate

chaining) side by side, this enhances the availability of the certificate management service through offering two options to acquire certificates.

To improve the access control in MANETs using the AAA (Authentication, Authorization, and Accounting) infrastructure, Khakpour *et al.* (2008) come up with WATCHMAN, a hierarchical distributed AAA architecture using OLSR, a proactive routing protocol. This server-based AAA architecture also considers resource and location awareness for a mechanism of server election. This proposal, essentially based on an overlay, caters for a lightweight and secure authentication and authorization model for MANET nodes. However, the design of this architecture intends to reduce communication overhead and computation cost. In fact, this proposal shows that different tasks are fairly distributed among distributed AAA servers. The computation cost and overhead communication is insignificant compared to OLSR signalling and routing overheads.

2.5.1.2.2 Cluster-based Models

The category of cluster-based models relies on a cluster-based routing infrastructure to group nodes according to a particular clustering algorithm (i.e. selecting cluster heads (CHs) and members in clusters). On the other side, this particular category may take advantage of that infrastructure to handle trust efficiently by, for example, distributing a TTP among CHs. In fact, it is usually identified as a special case of hierarchical trust and also it can be deployed by a form of group authentication, where clustered groups of nodes are considered as single trust entities and authenticated as a group. Even though these models achieve better scalability and performance comparing to the other models in the same classification, they exhibit a number of crucial challenges such as strict dependency between routing and trust services and cluster management (i.e. clustering algorithms and the selection, configuration, maintenance, and replacement of CHs, merging clusters, etc.) (Merwe *et al.*, 2007). There are several approaches adopting cluster-based architecture for managing trust.

Venkatraman and Agrawal (2000) develop a cluster-based authentication model for ad hoc networks. This model is established upon a cluster architecture, where the network is divided into clusters. Each of these clusters has an elected CH holding cluster membership information and acting as the certificate authority (CA) for its cluster. For key distribution, the model requires that when a node joins a network, it is provided with public and private system key pair as all the nodes in the network must share this key pair. Additionally, each node must also obtain a cluster key, created by the CH and shared by all the nodes within a cluster. For exchanging session keys between different communicating nodes, CHs get involved in facilitating that connection by exploiting a unique public/private key pair. However, mutual trust among network nodes is the main assumption of this model.

Bechler *et al.*(2004) introduce a model of a Key Management System (KMS) applied on a cluster-oriented MANET where CHs are basically assigned the role of signing certificates for other nodes (i.e. called warrantors). In order to enable new nodes to become full members of the network, nodes in this model are required to present a certain number of warranty certificates to a CH in order to obtain membership certificates in a similar manner to the approach of Zhou and Haas (1999). Those warranty certificates are initially issued by warrantors (i.e. existing full members of the network) as a resulted of verifying nodes' identities.

Rachedi and Benslimane (2006) and Rachedi *et al.* (2007), present a similar cluster-based model which is based on trust values metric and behaviour monitoring. In the cluster a CH also becomes a CA. The trust can be established by CAs among clusters by means of recommending nodes with certain trust level from CA to another. However, this model also leverages the new term of Dynamic Demilitarised Zone (DDMZ) for enhancing the protection of CAs in each cluster by using a set of redundant confident nodes, called registration authorities (RAs) surrounding a cluster CA. These RAs take role of a check point CA, by handling and filtering receiving requests of certification before forwarding them to the CA.

Alternatively, Dong *et al.* (2007) propose also a CA cluster-based architecture where the system caters for dividing the network into clusters. Each cluster head (CH) has a CA information table (CIT), which typically includes a list of CA nodes in its local cluster and in the other clusters. The CA information is distributed and managed among CHs, which enables a decrease in service response delay and system overhead.

2.5.2 Self-organised Models

In contrast to authoritarian models discussed broadly in pervious section, self-organised models cater mainly for allowing network nodes to manage trust by themselves (i.e. issuing, maintaining, updating and revoking). This means that there is no entity like TTP trusted by all nodes within the network, instead nodes which trust each other generate "credentials" based on local trust as shown in Figure 2-8.

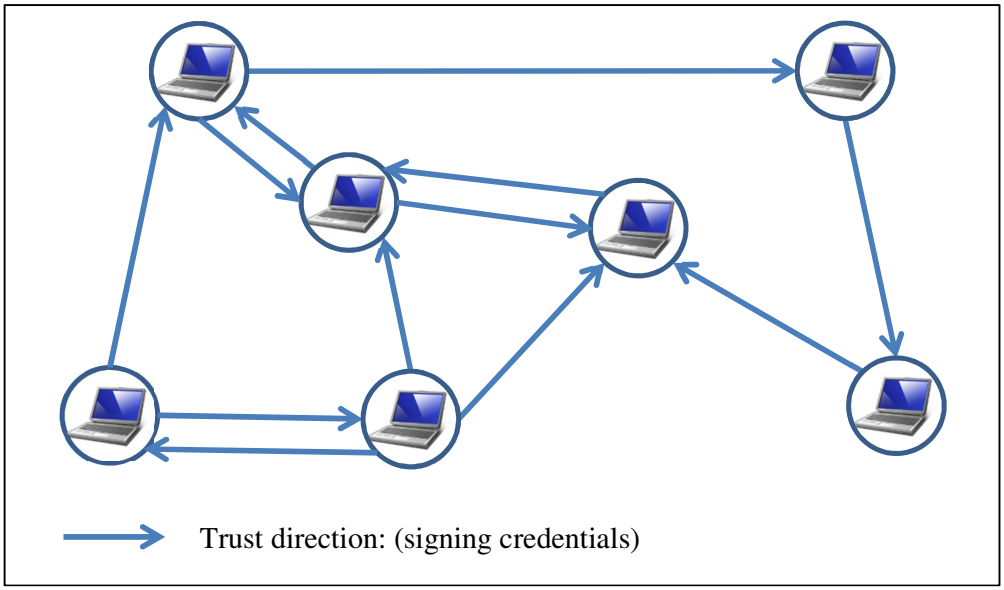


Figure 2-8: The General Self-organised Trust Model

PGP (Pretty Good Privacy) (Abdulrahman, 1997) is a completely distributed trust model which was developed by Phil Zimmermann in 1992. PGP is developed to be a substitute to the conventional PKI based on trusted authorities in order to offer a free practical security solution to protect low value communications, such as emails. The fundamental part of PGP is referral certification which enables multiple users to "recommend" a certain user (i.e. working as an introducer) by signing certificates of its public-key. PGP relies on a system, called the "Web of Trust" which mainly facilitates and manages key distribution. However, this scheme has a drawback making the system vulnerable because, for example, dishonest users may issue false certificates to cheat other users. Therefore, the level of confidence for the certificate is relatively low.

Alternatively, taking advantage of the same approach of PGP, Capkun *et al.* (2002; 2003) offer a self-organised trust model for MANETs, in which trust among nodes is maintained through a physical contact. Every node in this model issues public key certificates to those who it trusts from its own domain. Regardless of the network partitions and without any centralised services, nodes are able to authenticate each other with chains of trust. However, in case a trust chain cannot be found, a node can solicit neighbouring nodes in one or two hops (called helper nodes). This model also includes all required algorithms (e.g. the shortcut hunter algorithm) to facilitate the initialization and authentication processes, and nodes are expected to store as many certificates as possible by means of merging certificate repositories. In this model, trust is established from "offline trust relationships", which are created from general "social relationships".

Based on the same approach as Capkun *et al.* (2003), Ngai and Lyu (2004) apply their trust model on a cluster-based network model constructed with the zonal algorithm. They essentially develop a secure public key authentication service to thwart malicious nodes from disseminating false public keys in the network. Besides, trust metrics in this model are evaluated according to direct monitoring as well as recommendation. However, this work neglects several

issues such as the maximum length of trust chains and their effect on the performance of their trust management model.

In contrast to other aforementioned models, Balfanz *et al.* (2002) describe a security solution, called demonstrative identification, which enables nodes to establish initial trust relationships with each other without prior relationship or the existence of an offline TTP. This proposal normally exploits Location Limited Side Channels (LLSC) to support initial bootstrapping and leverages a basis for more complex trust establishment. The LLSC is also considered a secure proximity-based channel to exchange critical information (e.g. keys or hashes of keys for authentication) and thereby it is difficult for adversaries to gain physical access to the channel (e.g. to read or inject messages). However, the demonstrative identification approach is developed for specifically targeting spontaneous, small, and localised ad hoc networks.

2.6 Conclusion

This chapter has introduced MANETs and issues of security. Several critical problem and approaches have been identified. Unlike traditional networks, MANETs have many exceptional characteristics which have classified into main sets, network-related and node-related attributes as presented in Figure 2-2. Additionally, many feasible applications from a wide range of domains such as military, civilian and space can take advantage of MANET capabilities. However, security is still very crucial subject in any network system like MANETs as this issue is dealing with how to protect a system from any damage caused by security attacks. Therefore, the most vital security dimensions that need to be taken into consideration when securing any network are security requirements, threats and techniques. Alternatively, trust management in MANETs can be realised by two directions, credential-based and monitored behavioural approaches. For a robust security system, both two directions are essential to be incorporated with each other. As the research scope of this study is credential-based approaches, they are analysed and categorised into a hierarchy of classes according to different criteria

as shown in Figure 2-6. The last level in the hierarchy represents the current proposals in theses subclasses.

In conclusion, it appears that securing MANETs is an intricate mission because of several factors affecting the security service development for MANETs. These factors stem from restrictions in MANET characteristics, a variety of MANET application domains, diverse threats and attacks, performance concerns, and sophistication in various security mechanisms and trust models being proposed. On the other side, some of these factors mentioned above are interrelated with each other. Therefore, developing and evaluating an appropriate security architecture for such a service in MANETs become very complicated, which needs to be simplified and well-defined. Also, there is a lack of systematic approaches tackling security from different perspectives cooperatively. the next chapter will describe in detail the research approach adopted to fulfil the aim of this study.

Chapter 3: The Research Approach

3.1 Overview

In this chapter, the methodology and techniques adopted to perform the research of this thesis will be discussed. This chapter is initiated with an overview of the research methodology for accomplishing the aim of this study and its corresponding objectives. Thereafter, the simulation technique and its methodology will be described and justified as this research method was used mainly for testing SSAM performance and communication.

3.2 Research Methodology

In order to achieve the aim of this study and its objectives, the workflow of the adopted research approach for this study is summarised in Figure 3-1. A major aspect of this work uses computer simulation (a simulation approach (Balci, 1990; Law and McComas, 1991; Balci, 1994; Nance, 1994)); the reason for adopting simulation is described later in this chapter. To summarise, (1) this research began with identifying the scope of the problem area which is in "security service realisation in MANETs". Then, (2) the investigation of this research was initiated by reviewing the literature in the topics of the MANET technology and its related security issues as presented in Chapter 2. The main purpose was to understand the unique MANET properties, potential MANET applications and different MANET-related security and trust fundamentals (i.e. security requirements, cryptosystems, MANET threats and attacks, trust models, etc.). (3) Because of the various features involved in the MANET system, the development of security solutions for this system becomes very challenging. On the other side, lack of practical approaches, for evaluating the different security models in MANETs

under different aspects (performance, an application context, etc.), characterised the problem of this research.

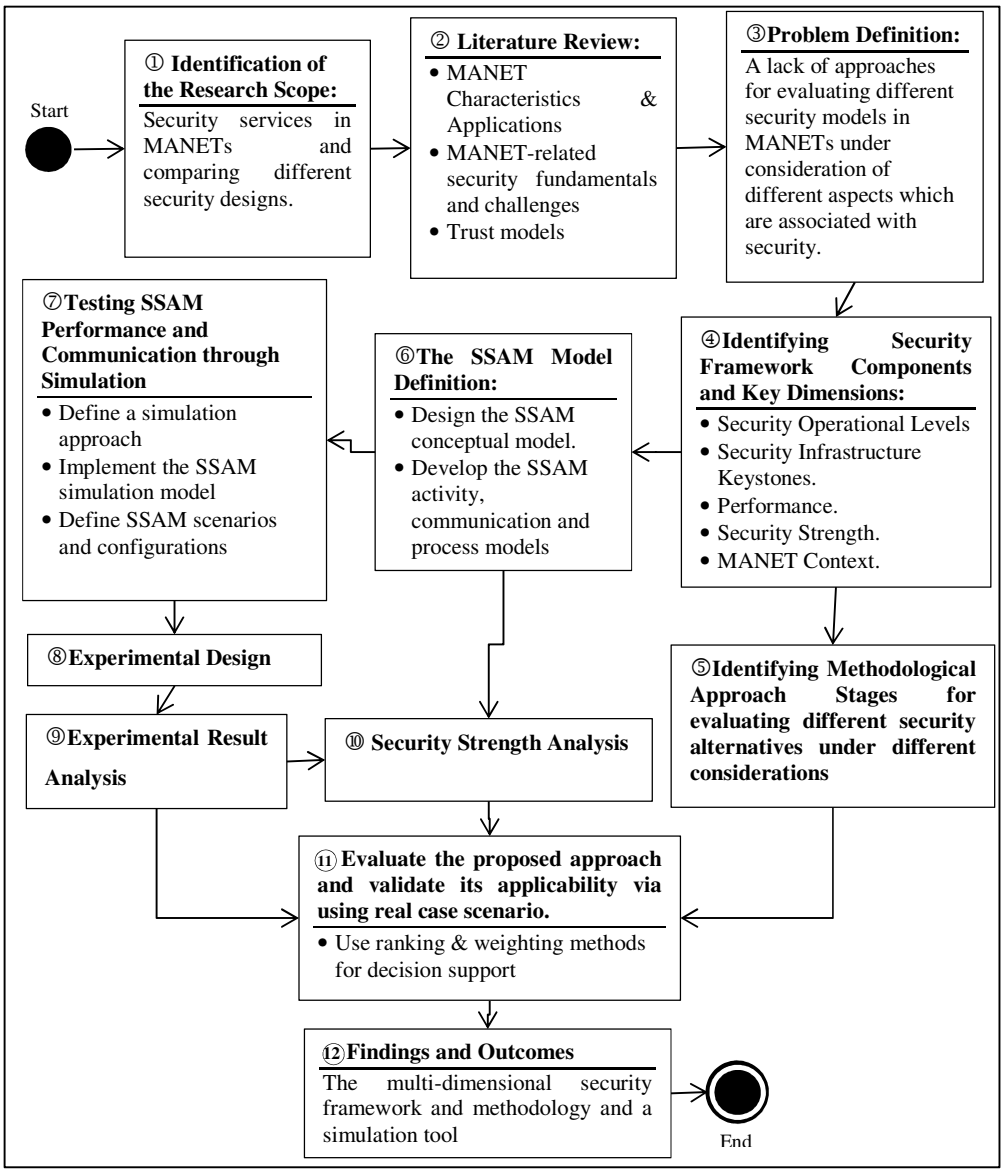


Figure 3-1: The flow of the research methodology for this study

Therefore, as discussed in Chapter 4 – Part A, (4) all relevant aspects (security- and non-security-related) that are associated with a security service (i.e. means a security/trust infrastructure) were investigated and analysed, such as a security operational level, security components, performance, and a MANET context. This

exploration of these aspects was in order to conceptualise and simplify them into a multi-dimensional design so that MANET developers can easily address all interdependencies of these aspects. Based on these identified aspects, (5) the methodological approach stages were established for the sake of evaluating different security service alternatives in a MANET context. (6) The SSAM model was derived from the simplification of the security keystones of security/trust infrastructure identified in the multi-dimensional design. It basically consisted of a group of different security architectures (i.e. different server architectures and different protocols) as detailed in Chapter 4 – Part B. This security model was to offer a basic approach of developing the security service in MANETs at the service level. Also, this model was intended to be used a case study for validating the defined methodological approach. The SSAM conceptual model was created and visualised to define all relevant elements and their dependencies existing in this model. A computer-aided modelling tool (e.g. MS Visio) was used to produce all activity, communication and process models for the SSAM model as presented in Chapter 4 – Part B. Then, (7) a simulation technique was used to test the performance of this model. Therefore, a simulation approach was defined by describing the system boundaries in the SSAM context, the provided services and main simulation scenarios. Relying on the OMNeT++ simulator, the SSAM prototype was implemented using C++ and NED languages as presented in Chapter 5. In addition, all types of necessary configurations (i.e. referred to the data input model) are established for the SSAM simulation model and its predefined scenarios. Thereafter, (8) the simulation experimentation for SSAM was arranged through determining the key metrics, test cases and the number of replications. (9) The results of experimentation were analysed and then were presented in combined bar charts as displayed in Chapter 6. With incorporating some experimental performance results, (10) the evaluation of security strength for each security architecture in SSAM was performed in qualitative manner, based on the predefined threat model (e.g. single point of compromise (SPC), etc.). (11) The results of performance, communication and security strength evaluation were employed in the phase of substantiating the feasibility of the

proposed methodology through using three real case scenarios. Also, simple ranking and weighting systems were used in this methodology to sort security alternatives of SSAM based on different criteria for decision support. (12) The outcome of this research was to provide the multi-dimensional security framework and methodology for evaluation security solutions and the SSAM simulation tool.

3.3 Simulation Technique

A major part of this research approach followed the simulation methodology (problem definition, model development and decision support stages, based on Balci (1990) and Nance (1994)). This stemmed from the fact that the SSAM model was required to be implemented and experimented in order to evaluate performance and communication of different security architectures in MANETs. These evaluations were primarily incorporated in the proposed methodological approach in this study. Therefore, simulation was an appropriate scientific methodology for this research. The concept of simulation is defined as the imitation of some real activities, state of affairs, or processes in any particular system whether it can be an abstract or physical system (Nelson *et al.*, 2001). Simulation can often be exploited to model natural, machine or human systems in order to gain insight into the operation of those modelled systems. Furthermore, it is considered as a very important tool for understanding interactions between various systems where there may be a lack of physical implementations for those systems or it may be difficult to control them in the real world (Sokolowski and Banks, 2009). In the domain of technology, researchers widely take advantage of simulation tools for testing, performance optimisation and measurement, etc.

However, simulation is specifically indispensable in the context of this study for several reasons. This is because this study is based on using a particular wireless network (i.e. a MANET) targeting a large-scale manner (scalability) for testing the performance of the SSAM Model. It is almost impossible in terms of cost to make use of real wireless devices to demonstrate this research idea since there is a scarcity of the real networking systems and applications. Also, it is

almost impossible to reproduce the whole wireless propagation environment in this particular type of network. Therefore, using a simulator can overcome all these challenges. Most network simulators, especially for MANETs, aim to create near accurate reproductions of most features in the environment, such as noise, probability of loss, routing, queuing, traffic or mobility. In addition, they mostly provide a layered-view model of wireless devices for the sake of simplifying the network simulation modelling. Hence, using network simulators allows the author to focus on the research aim of this study rather than the physical implementation details. On the other hand, those simulators facilitate building a valid model whose behaviour and operation can be repeated and measured (i.e. leading to a statistically verifiable proof of concept). Eventually, for the reasons mentioned above, the **O**bjective **M**odular Network **T**est-bed in C**++** network simulator (OMNeT++), which a MANETs simulation tool, was selected for developing the SSAM model. The justifications for selecting this particular network simulator rather than other simulators will be detailed in Section 5.3.1. The next section presents the description of standard simulation approach steps along with this study reflection on those steps. In other words, it is to show how the research approach of this study complied with the simulation approach.

3.3.1 Simulation Approach

This approach consists of three key stages, a problem definition, a model development and a decision support as proposed in Balci (1990) and Nance (1994) (see Figure 3-2). Each stage has a specific number of phases (steps) which are recommended to be followed by simulation researchers to successfully achieve the aim of each stage as displayed by ovals in Figure 3-2. The dashed pointers represent the process being performed to associate the phases to each other. However, the solid pointers indicate iterative attempts of evaluation among phases for leveraging more accuracy and credibility in phases outcomes by conducting a series of verification, validation and testing (VV&T) as emphasised in Balci (1994). A VV&T activity is exploited to discover any deficiencies (e.g. errors, bugs, etc.) in the phases. As there are numerous techniques and principles

for verification, validation and testing (VV&T) for different simulation contexts, only techniques relevant to this study will be briefly highlighted, for further details, refer to Balci (1994). In the next sections, the problem definition, model development and decision support stages along with their own processes and phases shown in Figure 3-2, will be explained and related to this research. (Note that not all phases and processes were applicable to this study as this methodology targets a wide range of different simulations)

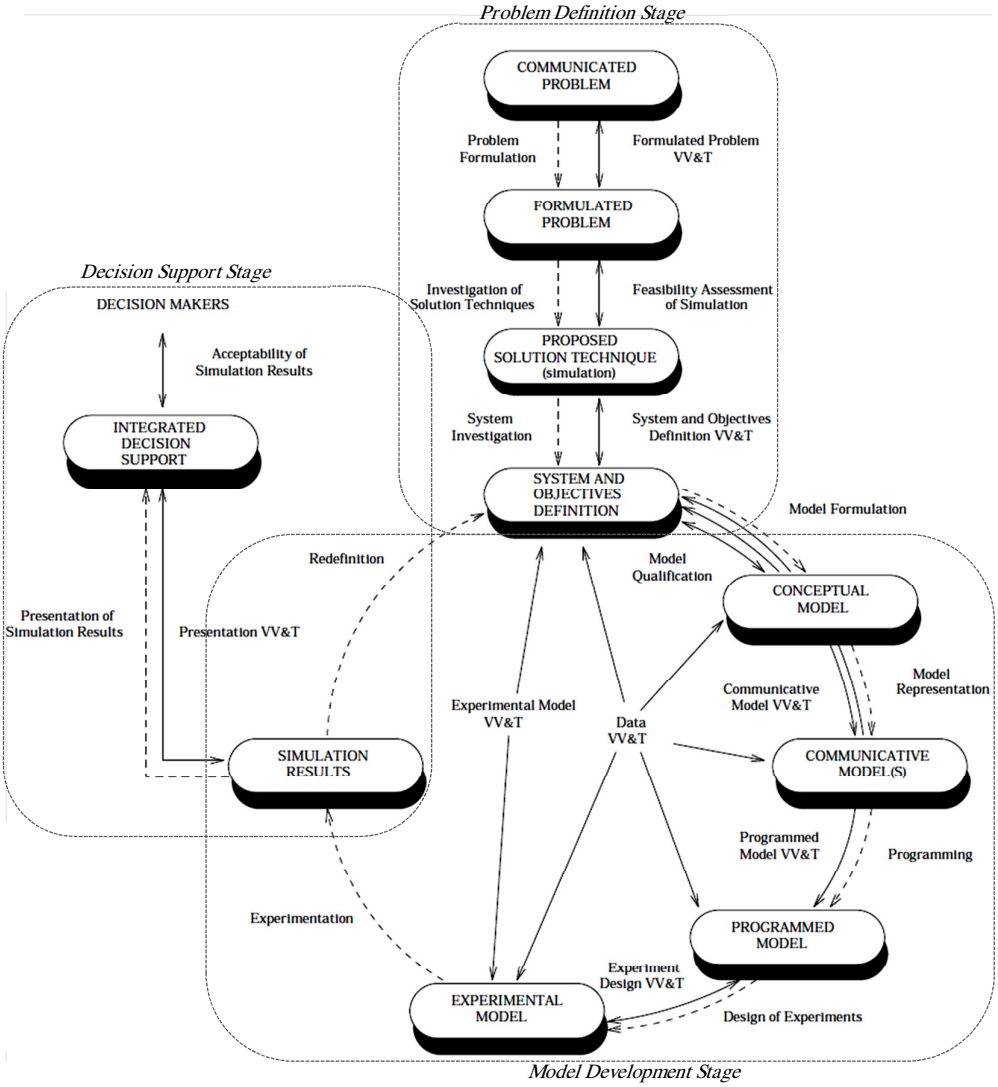


Figure 3-2: The Simulation Approach - Life Cycle (Balci, 1990, pp26)

3.3.1.1 The Problem Definition Stage

This stage is intended to have a well-defined problem required to be solved using a simulation technique. It is composed of *communicated problem* (i.e. a primitive form), *formulated problem* (i.e. an inclusive form), and *proposed solution technique*, and *system and objectives definition* phases as displayed in Figure 3-2. The stage begins with the process "problem formulation" of transforming a *communicated problem* into a *formulated problem* that is well-defined and well-structured to initiate an investigation to find the solution for this problem. Then, all potential techniques for solving the formulated problem should be researched to select the best cost-effective one (i.e. referring to the process of "investigation of solution techniques"). However, since this approach is dedicated to simulation studies, simulation is presumed to be the most cost-benefit proposed method for this context. Thereafter, the properties of the system in which the formulated problem is being tackled, need to be examined and characterised, such as system environment settings, interdependency and organisation. This is because this helps researchers to model this particular system appropriately using a simulation tool. Eventually, at the end of this stage, the *system and objectives definitions* are established.

Briefly, the main problem within this study was to identify and select the most suitable security architectures for developing and deploying the security service in MANETs that satisfy a set of requirements, such as lower communication and better performance. In order to solve his problem, this necessitated experimentation for assessing performance and communication for a given security architecture in MANETs. The simulation technique was selected as this appeared to be appropriate for this context (the justifications of this choice are mentioned). However, there were different security architectures proposed in the SSAM model for large-scale MANETs. Each security architecture relied on a different server architecture and authentication protocol as described in Section 4.6. The system boundaries were delineated by defining system parameters which were related to the SSAM model and MANETs, such as a routing protocol in use,

a wireless technology, etc. Most of these issues are discussed in detail in Section 5.3.3 and 5.3.4. As a result, the goal of simulation was to compare all alternatives of these architectures in term of performance and communication so that the results would be incorporated in the proposed approach of this study for decision support.

3.3.1.2 The Model Development Stage

This stage introduces distinct developments to a simulation model throughout the simulation study as shown in Figure 3-2. These developments respectively are a *conceptual model*, a *communicative model*, a *programmed model*, an *experimental model*, and *model results*. The stage is initiated by the process "model formulation" of constructing a *conceptual model* which is an abstracted visualisation of the system under study. Then, this particular *conceptual model* should be converted into *communicative model* (i.e. a design specification of a simulation model) by using for example documentation and computer-assisted graphic tools. This particular model is intended to represent all proposed activities, processes and interactions in the simulation model before implementation. The specification generated from the previous model should be implemented using a high-level standard programming language (e.g. Java, C++, etc.) or a specific proprietary simulation language so that a simulation model can be compiled and executed on a computer (i.e. referring to the *programmed model*). The design of experiments is the process of creating a plan on how to conduct simulation experiments (i.e. an *experimental model* as shown in Figure 3-2). This is in order to efficiently obtain the necessary results on which the analyst relies, to come up with valid inferences. Various methods can be recognised in the design of experiments, such as variance reduction methods, factorial designs, replications, etc. After that, the simulation model should be experimented according to the predefined *experimental model* for a specific target such as comparison, optimisation, etc., so as to generate *model results* of simulation. At the end of this stage, two processes can be acknowledged, "redefinition" and

"presentation of model results". "Redefinition" is to modify a simulation model to adapt new settings of the system in order to study this system from another perspective or investigate other different experimental conditions (e.g. amending control variables). "Presentation of model results" is the procedure of analysing the obtained experimental results and then these results will be presented to decision makers to get their approval. Also, the analysed results can be exploited by decision makers in their applications.

For this study, the conceptual model of SSAM was defined which represented a new node joining the network to utilise a given security service in order to obtain its membership certificate as shown in Figure 4-6, Section 4.6. Also, this conceptual model consisted of different server architectures, different authentication protocols, and a specific cryptosystem and credential. The *communicative model* was generated as presented in Section 4.7 and 4.7.1. This model included the SSAM activity, communication and process models (Figures 4-10, 4-11, 4-12…and 4-19). The SSAM prototype was implemented in the OMNeT++ simulation environment using C++ and NED languages as illustrated in Section 5.3.2. This prototype was considered as a *programmed model* for the simulation model of this study. The SSAM experiments was designed (i.e. the *experimental model*) for the SSAM simulation model by defining the important performance and communication metrics, test cases and the number of replications (using the confidence interval method) for simulation experimentation as described in Section 5.4. The results of simulation experiments were interpreted using the simple standard estimation (i.e. averages). Then these analysed results were plotted in the combined bar charts, representing different scenarios (i.e. *Churn* and *No-Churn* node behaviours), node populations and different security architectures as shown in Section 6.2. Finally, the results were summarised and incorporated in the evaluation stage of the proposed approach as demonstrated in Section 6.5 and 6.6.

3.3.1.3 The Decision Support Stage

This stage is relevant to this study as the generated results were used in a simple decision making method (i.e. a ranking approach) to compare the security alternatives in SSAM based on particular criteria and then to find the best suited security architecture for a given application in MANETs, as detailed in Section 6.5 and 6.6. This particular method consists of achievement rating, reciprocal ranking weighting and weighted averaging. Therefore, the phase of *integrated decision support* was involved in this context in accordance with this methodology. The outcome of this phase is assumed to enable decision makers to take an appropriate action depending on the model results acquired from simulation experimentation.

Last but not least, the Validation and Verification and Testing (VV&T) techniques were used through this research approach to substantiate the proposed simulation model developed based on the formulated problem as shown in Table 3-1.

VV& T Activities	Techniques used in this approach
Formulated Problem VV&T	*Walkthrough, Structural Analysis*
System and Objectives Definition VV&T	*Walkthrough*
Communicative Model VV&T	*Walkthrough, Graphical Comparison,* and *Graph-based Analysis*
Programmed Model VV&T	*Walkthrough, Code Inspection Visualisation, Debugging* and *Graph-based Analysis*
Data VV&T	*Walkthrough, Statistical Techniques (Confidence Intervals)* and *Data Flow Analysis*
Experiment Design VV&T	*Walkthrough*
Experimental Model VV&T	*Walkthrough, Inspection, Visualisation, Graphical Comparison* (i.e. using time-based series charts) and *Graph-based analysis*
Presentation VV& T	*Walkthrough, Inspection*

Table 3-1: Validation and Verification and Testing (VV&T) activities with applicable techniques for this study

3.4 Conclusion

The research approach of this study is discussed in this chapter with key steps as shown in Figure 3-1. Also, a simulation method was employed for this research to evaluate the performance and communication of the SSAM model. The description of the simulation approach and its main stages and processes for conducting simulation experimentation is presented in addition to The VV&T activities and techniques. Eventually, the aim of this chapter is to demonstrate how the research approach of this study is consistent with the simulation approach. The next chapter will discuss the conceptual multi-dimensional security framework, the prototypical design of the proposed methodological approach and the proposed SSAM model.

Chapter 4: Conceptual Security Framework, Approach and SSAM Design

4.1 Overview

Chapter 2 established the background for MANETs, the fundamentals of MANET security, and the MANET-related trust models. To leverage a better understanding on how to realise and evaluate a security service in MANETs, this chapter mainly presents three key proposals: a conceptual security framework for MANETs, an initial proposal of the methodological approach and a model of the Server-based Security Architectures for MANETs (SSAM). In this context, the proposed security framework is intended to address the key security elements (e.g. security roles, security communication protocols, etc.) and the affecting dimensions (e.g. an operational level, performance, etc.) in the design and development of a trust/security infrastructure in MANET. Based on this framework, a set of well-organised steps are formed into a methodological approach in order to enable security developers to come up with an effective security solution satisfying a given MANET context. On the other hand, as will be seen, the SSAM model is proposed to offer a new security design for MANETs relying on the service operational level and also to validate the suggested multi-dimensional methodological approach stemmed from the framework.

This chapter is divided into two main parts (i.e. Part A and B). In Part A**,** Section 4.2 describes the operational view and security design of MANETs through introducing two key operational levels ("Network" and "Service") in order to handle security issues independently and effectively. In Section 4.3, the primary the building blocks of the MANET security/trust infrastructure are formalised into

a conceptual security framework by analysing prior works: a security role, security-server architecture, security communication protocol, security mechanism, and security credentials. In Section 4.4 a multi-dimensional framework identifies the important dimensions to be considered for producing a well-defined and -designed security/trust infrastructure tailored to the requirements of a given MANET context. These dimensions represent the security strength (Section 4.4.1), performance (Section 4.4.2), and the MANET context (Section 4.4.3). The MANET context consists of MANET constraints (Section 4.4.3.1) and settings of a given application (Section 4.1.3.2). Section 4.4 presents an outline of the methodological approach which is developed to investigate a suitable security/trust infrastructure for a certain MANET context.

In Part B**,** Section 4.6 conceptualises all relevant elements incorporated in the SSAM model: initialisation, server architecture, cryptosystem, credentials authentication protocols, strategy of calling, MANET settings. Some of these proposed elements are designed producing different alternatives. The server architecture may be established on different servers (i.e. *CAS*, *TAS* and *DAS*) and there are different architectures which can be recognised (i.e. *CAS*, *TAS*, *D\CAS*, *CAS_TAS*, *TAS_DAS* and *CAS_TAS_DAS*). Three standard X.509 authentication protocols are utilised in SSAM. In the "strategy of calling" element, *AAO* and *IPS* are only employed in the hybrid and hierarchical architectures where different types of servers participate in providing a security service. Finally, Section 4.7 presents the SSAM activity model which represents the lifecycle of a new joining node utilising a security service in SSAM. In addition, the communication and process models in Section 4.7.1 are intended to display the core security-related SSAM activities. The communication model (Section 4.7.1.1) specifically concerns with describing the sequence flow of creating a single connection between a user and server nodes when a particular authentication protocol is applied. However, the process model for both user and server nodes (Section 4.7.1.2) is defined to point out to the flow of internal processes for utilising a security service (i.e. calling and then obtaining a membership certificate) in distinctive server architectures within SSAM.

4.2 MANET Views – PART A

A typical MANET is composed of wireless mobile nodes forming a dynamic network without relying on centralised routing infrastructure where nodes communicate through multi-hops as depicted in the MANET physical view in Figure 4-1. However, similar to the other conventional networking systems, this particular network can be realised by a layer-based architecture complying with the standard OSI model as demonstrated in the MANET operational view in Figure 4-1. Two given end nodes, representing a user and a service provider, interact at the transport and application layers while other network nodes become routers for facilitating communication between those two ends in the network layer following the OSI reference model.

This design approach (i.e. layering) aims to facilitate controlled interaction among layers so that developing and maintaining single layers can be attained independently of the rest of the stack. In other words, this design organises protocol and network tasks in layers, splitting the networking system into modules in order to enable transparent and efficient improvements of these single modules. Thus, in a strict-layered system, protocols can be deployed independently of each other and interact through predefined layer interfaces (i.e. each layer implementation relies on the interfaces accessible from the lower and upper layers). The layer-based design approach supports flexibility to a system's architecture as any amendments introduced into any single level do not affect the rest of the system. Furthermore, the separation of networking system activities in this regard, would lead to cut down development costs through re-using existing code. As presented in Figure 4.1, this design approach leverages "horizontal" communication between peer protocol layers on the sender and receiver devices. However, the drawback of this particular design is that high overhead communications are incurred from interaction of the multiple layers and this may be very critical especially in limited bandwidth networks.

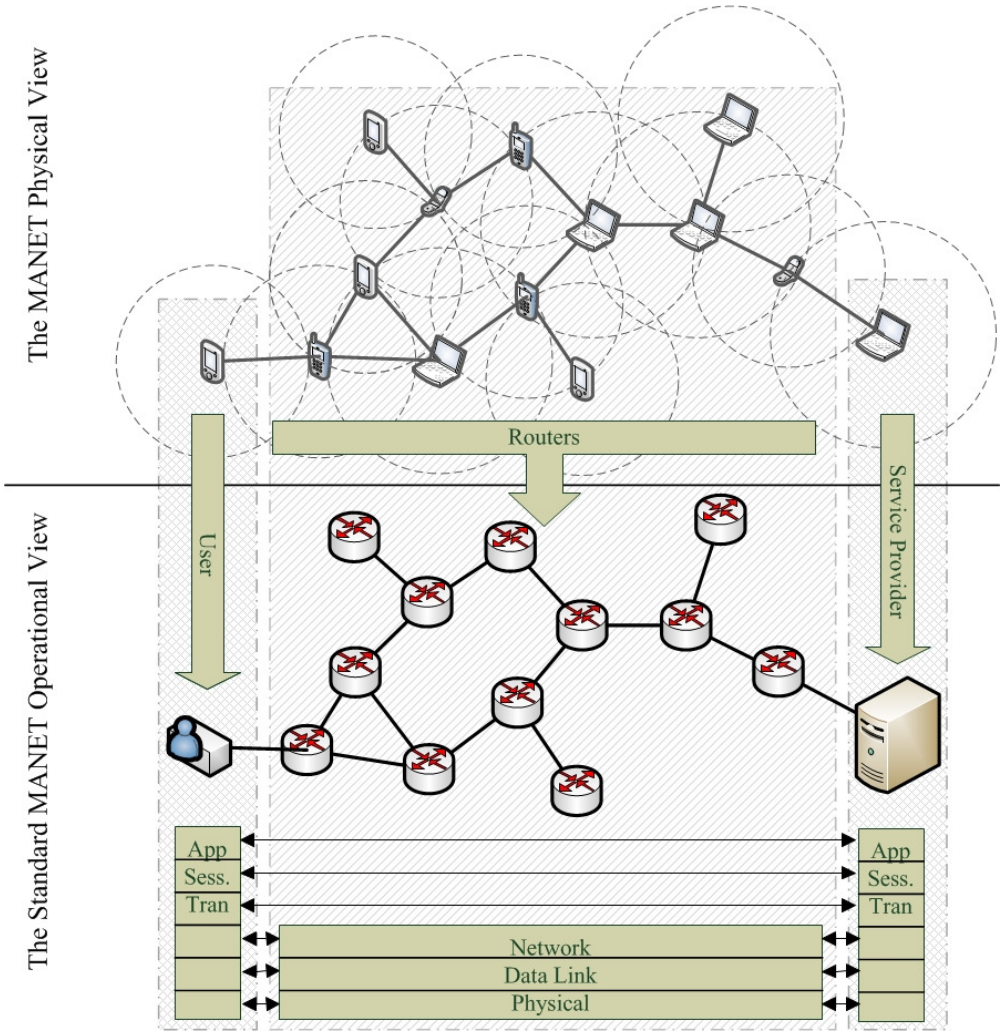


Figure 4-1: The MANET OSI-based Operational View (the Physical and Logical Views)

However, other design approaches can be found in literature. For example, Corson *et al.* (1999) have suggested a design approach for Internet-based MANETs in a tactical domain by vertically integrating certain layers (e.g. routing network and link layers) in the OSI model. This is to minimise the problem of limited bandwidth sacrificing flexibility and interoperation aspects, but they have emphasised that it is important to make trade-offs between the traditional layered model and the integrated one according to domain context requisites. Also, the Conti *et al.* (2004) and Messerges *et al.* (2003) proposals take advantage of the

cross-layer design approach in order to enhance network performance. This specific cross-layer design can allow protocols in different layers to collaborate in sharing network-status information or security materials while still keeping the layers separated. However, an unrestricted cross-layer design can lead to overlapped code which is very difficult to maintain efficiently since every amendment must be spread out across all protocols. Also, cross-layer designs could trigger unintended interactions among protocols, such as adaptation loops, that may cause performance degradation.

As it is concluded from the discussion of different design approaches, several key motivations can be acknowledged to encourage the adoption of the strict-layer approach for MANETs. Some of these motivations include (1) the standard "IP-based" view of MANETS for internet interoperation purposes; and (2) the flexibility, (3) modularity and (4) simplicity provided by independent layers, which support reuse of existing software. On the other hand, the choice of the layered approach is supported by the fact that MANETs offer mobile extensions of the Internet, and thereby the standard layer-based stack – the OSI (ISO/IEC 7498-1) appears to be appropriate.

Therefore, the next two sections discuss this thesis proposal of the level abstraction in the layered operational model for MANETs without violating the reference OSI or internet model. This proposal presents the simplification and differentiation of two key generic operational levels (Network and Service) each of which involves distinct activities and responsibilities. The main purpose of this proposal is arguably to offer a guide especially for MANET security designers so that they are able to address security issues independently and effectively as illustrated in Section 4.2.2.

4.2.1 Proposed Two-Level-Based MANET Design

According to the above discussion, it is important that a MANET system should abide by the standard layer-based architecture for the sake of maintaining

modularity, flexibility and interoperation in networking system design. However, having several different layers in a system design, especially for MANETs, may introduce more complexity and redundancy in the system development and deployment. As a result, without overlooking the layered-based design, the operational levels of the MANET system can be simplified and distinguished into two key general levels; the Network and Service Levels as presented in Figure 4-2. The proposed two-level design which represents the abstraction of the OSI layers, is justified by the fact that most activities in the "Network Level" address only the networking issues, such as node addressing, routing and node-to-node communication whereas the "Service Level" concerns with data resources and applications (i.e. services available to users) shared within the MANET system. Based on the OSI model, the "Network Level" typically represents the network (i.e. *IP*) and MAC layers while the "Service Level" indicates to the transport and application layers.

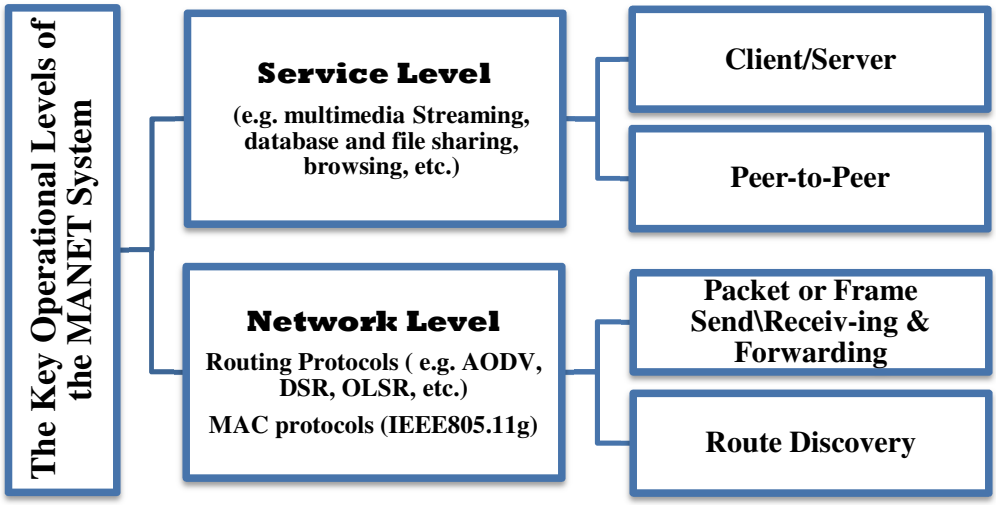


Figure 4-2: The generic MANET operational levels with their typical interactive activities (the two-level-based model)

The "Network Level" represents the networking infrastructure of MANETs which is different from other similar conventional networks, in the way of

handling routing between mobile hosts (i.e. packet sending, receiving and forwarding and route discovery). There are various types of reactive and proactive routing protocols, such as AODV, DSR, OLSR, etc., developed for this reason (for more details, see Section 5.3.4.4). In the "Service Level", regardless of network types being used, the client/server (C/S) and peer-to-peer (P2P) models are the most well-known computing paradigms that can be acknowledged for running various applications and sharing different data resources (Umar and Fraser, 1993; Umar, 1997; Taylor and Harrison, 2006; Tanenbaum and Van Steen, 2007). The client-server model is realised by the fact that a single or a group of servers (where data resources and services reside) process and handle invocations initiated from certain clients (e.g. MANET nodes in this study) on the network. In this particular model, servers are presumed to be powerful devices with the adequate bandwidth network connectivity whereas clients are considered weak end points. This model is widely adopted by many internet applications (e.g. Grids, Web Services, Clouds, Database systems, etc.). This is due to its generality feature (most other paradigms are derived from this model), simplicity of the client design, and direct support by single or multiple servers. However, the main drawback is that when more clients require service, this may entail more overhead on a server, creating a bottleneck (i.e. degradation in system scalability). The peer–to–peer (P2P) model, an alternative to the client/server model, was first introduced to describe point–to–point communication between two nodes sharing equal status, such as a telephone conversation (Taylor and Harrison, 2006). Peers still implicitly exploit the client/server paradigm since whenever one peer contacts another; it becomes the client node, while the other becomes a server node. In this model, every peer takes the role of both a client and a server. Nowadays, this model becomes a building block of what is called P2P networks through supporting a unique communication paradigm for large numbers of heterogeneous nodes and building "virtual" nodal connectivity (Moore and Hebeler, 2002) at the application layer. Most P2P networks are intended to use the resources of the peers, for example content, CPU cycles, and bandwidth to be shared with other peers for sake of augmenting efficient utilisation and decentralisation. For this reason, several famous P2P applications, such as Napster, KaZaA, Gnutella and

BitTorrent, have emerged to allow for nodes to share information over the wired Internet. Therefore, the main advantages of P2P are the exploitation of unused distributed peer resources, enhancement in information delivery and system scalability due to eliminating the server bottleneck, cost savings stemming from minimising the demand of centralised management, storage, and other related resources, and improved network fault tolerance. Yet, P2P like other systems encounters many challenges which inclusively include, interoperation, questionability on availability and performance, lack of network control, potential violation of privacy and security (Steinmetz and Wehrle, 2005; Taylor and Harrison, 2006).

Applying P2P over limited MANETs following this proposed level-based model would create a burden on the mobile nodes in terms of network traffic handling and energy consumption apart from their ad-hoc routing roles. Therefore, the majority of proposed P2P-MANET projects, such as Klemm *et al.* (2003), Gruber *et al.* (2004), Tang *et al.* (2005), Choi *et al.* (2006), Boukerche *et al.* (2010), etc., leverage the cross-layer design for adapting P2P applications in MANETs in order to reduce the communication overhead and enhance P2P delivery. Due to the similarities between P2P networks and MANETs and also having specific P2P applications (e.g. commonly file sharing and VoIP streaming), the P2P over MANETs has received a great attention by the research community studying the advantages of MANET potential in the context of P2P implementations. However, this domain is out of this study's scope. Nevertheless, only a specific case of a so-called pure P2P architecture, where only two peer ends interact independently (Schollmeier, 2001), can be adapted with the "Service Level" in this proposed two-level-based model for MANETs.

Alternatively, despite the fact that the client/server model may suffer from a server bottleneck which can be resolved by using distributed servers, this model is still acknowledged as a predominant basic computing paradigm adopted by numerous types of applications/services even some hybrid P2P applications. In addition, most implementations of this model can be typically found in the

transport and application layers based on the OSI model and this is consistent with the design of the "Service Level".

On the other side, MANETs are usually characterised as a self-organised and infrastructure-less networking system (i.e. no need to a pre-established infrastructure for deployment). However, these types of system features are realistically unachievable in all different levels of MANET operation especially in large-scale MANETs. Besides, it is important to realise that being infrastructure-less appears to represent only an extreme case for MANETs as stressed by Asokan and Ginzboorg (2000). They also stated that it is feasible in these networks to have a sort of partial infrastructure to facilitate and support several activities, such as security, management, etc. In this regards, in MANETs, two different infrastructures can be distinguished: the routing and service (server) infrastructures. These two particular infrastructures clearly adhere to two operational levels proposed in the two-level-based model as shown in Figure 4-2. The routing infrastructure, which intends to establish interconnection between network nodes, is *self-organised* in MANETs since every node has a dual role; a host role to handle its own traffic and a router role to route other node traffic using a specific ad-hoc routing protocol. The service (server) infrastructure typically consists of on-line servers where certain application/services are deployed, such as DNS, CA, Database and Web application servers, offering services to network user nodes. Furthermore, the presence of this infrastructure depends on the type of the application or service being targeted (e.g. Website hosting or P2P VoIP streaming).

4.2.2 Security Level Design for MANETs

Security is a very demanding element for MANETs. At the same time, it is a quite problematic issue. Security MANETs designers still come across many technical challenges and complications to develop appropriate security solutions for protecting MANETs. This stems from the fact that MANETs, unlike the other

typical networks, have unique features, such as infrastructure-less-ness, dynamic topology, limited resources and bandwidth, and an open and unreliable communication medium (Chlamtac *et al.*, 2003; Djenouri *et al.*, 2005). Furthermore, in terms of security, MANETs also show no clear line of defence (Yang *et al.*, 2004) and encounter many different vulnerabilities and threats as described in Section 2.3.2.

On the other hand, security is still treated as an add-on feature considered after developing the whole networking system. This approach in developing security solutions renders more complexities and discrepancies which need to be resolved before any security solution can be implemented in the system. Therefore, in the first place, security must be handled as a main element involved in the design and development of networks like MANETs.

Although many security solutions, such as trust models shown in Figure 2-6 and a survey of secure MANET routing protocols in Abusalah *et al.* (2008), have been developed to improve the security of MANETs, none of them takes into account designing security from a system architectural view. In addition, there is a lack of a clear systematic approach which can efficiently deal with the complexity of security requirements in different levels and circumstances. As a result, security solutions are prone to suffer from several problems of misplacement and overlapping of security mechanisms being applied in those solutions.

Due to the great success of the standard OSI model, the layer-based design in any networking system is still the most effective and flexible design on which security designers can rely to develop their security solutions. Furthermore, considering different layers from the perspective of security design allows various security operations and requirements to be addressed independently and this arguably leads to increase robustness, manageability, flexibility, simplicity and interoperation in the development of network security. International Telecommunication Union (ITU) in ITU-T recommendation X.805 (ITU, 2003) has proposed a global standard security architecture which is primarily based on a layered design (i.e. *network infrastructure*, *network service* and *network-based*

application security layers) for telecommunication systems (i.e. cellular and wired networks). This particular security architecture logically arranges end-to-end security-related network activities into separate layers and planes taking into consideration the important security dimensions (i.e. security requirements) that need to be fulfilled. In addition, this architecture is intended to offer a comprehensive, top-down, and end-to-end approach of networking system security which can be applied to network elements, services, and applications for the sake of improving protection from attacks and threats and also efficiently handling potential vulnerabilities. This approach aims to facilitate the planning, development and deployment phases of new security solutions and it can be exploited to evaluate the security of the existing networks for maintenance purposes (for more details, see ITU-T recommendation X.805 (2003)). However, this particular security architecture is developed for networks which rely on specific infrastructures for operation, unlike infrastructure-less and dynamic MANETs.

There are few attempts in proposing a fully layered security design to a MANET system aiming to attain complete protection, such as Yu *et al.* (2003), Yang *et al.* (2004), Al-Bayatti et al. (2009) and Sehgal *et al.* (2011). These references have provided a similar layered security design which comparatively complies with the standard security architecture proposed in ITU-T recommendation X.805 (ITU, 2003). These proposed architectural designs include application, networking, routing, and node-to-node security layers along with a trust infrastructure layer. Alternatively, some MANET security approaches have leveraged layers partially in the implementation of their security solutions (i.e. do not adopt all layers in the security design). For example, Verma *et al.* (2004) have considered both application and data-link layers in the implementation of trust negotiation between two ends for facilitating confidentiality and authentication. Komninos *et al.* (2006, 2007) have presented a security framework of handling security in the data-link and network layers by using suitable authentication protocols. Aljnidi and Leneutre (2007) have suggested an access control model for securing virtual domains over MANETs and this model is only applicable to the

application layer. Obviously, the concept of security layer can still assist security designers to easily identify the specific security issues associated with key layer activities and to deal with those activities independently and effectively during the development of MANET security system.

The works discussed above suggest the layer-based security architecture with several security layers for modularity and flexibility purposes. Those several layers however may become a source of undesirable complexity in tackling security especially in MANETs. Therefore, in the light of the proposed two-level-based operational model described in the previous section, the notion of a security level can also be introduced for MANETs in order to handle important MANET security issues effectively and to facilitate the development of proposed MANET security solutions. The two-level-based security design consists of the "Service and Network Security Levels". Having two security levels for MANETs is justified by the fact that both levels show different security concerns and requirements which can be tackled separately in each level. Also, various threats in MANETs according to the OSI layers (as illustrated in Section 2.3.2), can be accommodated this security-level-oriented model. The "Network Security Level" represents networking activities and procedures that need to be protected, such as route discovery, address location, packet forwarding, frame delivery, etc. For "Service Security Level", service-based security is in contrast intended to secure service\application infrastructures (e.g. users and service providers, application-based protocols, etc.) already deployed in this operational level of MANETs. Furthermore, most common security services in this level are associated with end-to-end security, authentication and authorisation (user's roles and identities, access policies, and security authority). However, this does not mean that there is no need for network-level security. The main reason is that it is much more seamless to reconfigure the security infrastructure at the service level than at lower-level (network) especially in MANETs where no fixed routing infrastructure typically exists for establishing connections (i.e. infrastructure-less).

In conclusion, this two-security-level-based architecture arguably presents a pragmatic approach of streamlining the essential security operations in the development of MANETs into two key security levels, "Network" and "Service" levels. This can be considered as a key enabler to security designers for developing their security solutions seamlessly with a balanced view in MANETs. In addition, the security architecture aims to instigate more interest in the "Service Security Level" to the research community as there is a lack of research in this topic and also many published studies in MANET security have only focused on routing and networking security (Abusalah *et al.*, 2008; Cho *et al.*, 2011).

4.3 Conceptual Security Framework

As described in Section 2.4, irrespective of the operational level being considered, establishing trust associations among MANET nodes is very important to enable secure interaction and utilisation within MANETs. This stems from the fact that almost all types of security services (authentication, authorisation, etc.) rely on trust for their deployment and implementation. Various categories of trust models can be acknowledged for MANETs (as illustrated in Section 2.4 and 2.5) according to the trust form, the architectural type of a trust service, the authority reliance and security mechanisms being used, etc. However, this study takes into account only trust which is represented by certain credentials (e.g. digital certificates) issued by single or multiple authorities (e.g. CA, AA, KDC etc.).

Involving an authoritarian trust model, especially a centralised one in MANETs, may compromise the MANET common desire of having self-organisation and self-administration. However, for several reasons, this particular model is still an effective solution of trust establishment for various applications which are required to be under control of a certain body (administration, government, etc.). Firstly, using an authority can leverage a high-level assurance to secure high-value communications particularly in large-scale networks as every node in the network typically trusts the authority with high confidence for managing their security issues (Yi and Kravets, 2004; Luo *et al.*, 2005). Also, the

authority often has necessary policies for regulating access, admission and membership in the network (Jansen *et al.*, 2003; Keoh *et al.*, 2004). On the other side, the self-organised (i.e. non-authority-based) trust models for MANETs, such as Capkun *et al.* (2002; 2003) and Ngai and Lyu (2004) usually rely on PGP (as described in Section 2.5.5) which has some shortcomings, such as questionable scalability, low-level assurance and poor and costly certification management. The questionable scalability is caused by the fact that finding a complete trust chain between a pair of interacting nodes is not all the time possible, especially in networks with a large number of nodes. The low-level assurance is instigated from the risk of having a malicious introducer within a trust chain, which jeopardises the whole trust. The process of certificate revocation in PGP is complicated as disseminating revocation lists among all nodes is problematic specifically in large-scale networks. Yet, a transitive trust used in PGP can entail a computational overhead (i.e. verifying a long trust chain) which is not applicable to limited MANETs (Luo *et al.*, 2005). Therefore, the authority-based trust model still presents great promises for MANETs and their applications as this is appeared from a number of studies in this research field, as shown in Figure 2-6. On the other side, an "authority" can play different roles for supporting specific security service (e.g. authentication, authorisation, policy enforcement, etc.).

Generally, a "security infrastructure" is defined to represent all important underlying security elements which are utilised to protect networking systems like MANETs. However, the intended security infrastructure, in this study, mainly concerns with facilitating trust distribution, especially in the service level (i.e. a security trust service), using specific security credentials. Therefore, a "security/trust infrastructure" is decided to be termed for covering a broad security meaning. This is because a trust management system (as explained in Section 2.4) is considered a fundamental part of any protection system. Also, some security measures (e.g. authentication protocols, intrusion detections, access controls, etc.) need to incorporate trust services for achieving security robustness.

Catering for more coherent design of a security/trust infrastructure in MANETs, a security framework is suggested, as shown in Figure 4-3. This framework is composed of five security building blocks: (1) a security role, (2) a security-server architecture, (3) a security communication protocol, (4) a security mechanism, and (5) a security credential. Arguably, these design elements can assist security MANET designers to speculate about relevant aspects and challenges of the design of security/trust infrastructure so as to develop a well-designed and feasible security solution for MANETs. Therefore, the elements of a security/trust infrastructure are explained as follows:

1. The ***Security Role***: refers to the type of security service that is provided by a security server in the security/trust infrastructure. Different types of security services can be recognised, such as certificate authority (CA), attribute authority (AA), policy certificate authority (PCA), key distribution centre (KDC), authentication, etc. Typically, the CA server is used to issue, update and revoke name or Id certificates for network users. Similar to the CA server, the AA server deals with generating and revoking attribute or authorisation certificates. The PCA server is presumed to provide users in the network with their policy certificates in order to manage security policy in users' devices (Jansen *et al.*, 2003). The KDC server is used specifically in the Kerberos protocol to secure secret keys to network users. The authentication server, such as an AAA server is used to handle users' admission in the network using a particular token.

2. The ***Security-Server Architecture***: describes the distribution model to security servers which typically run a particular security role in the security/trust infrastructure, such as central, distributed and hierarchical models. The central model is where only one server is responsible for providing a security service to users (see Section 2.5.1.1). The distributed model involves multiple servers which collaborate dependently (see Section 2.5.1.2.1.1) or independently (see Section 2.5.1.2.1.2) to offer a security service. The hierarchical model is similar to distributed model but with different types of servers arranged in a

particular hierarchy (see Section 2.5.1.2.1.3). However, several reasons can be perceived behind proposing distributed and hierarchical models in the security/trust infrastructure, such as providing fault tolerance, availability, more security robustness, etc.

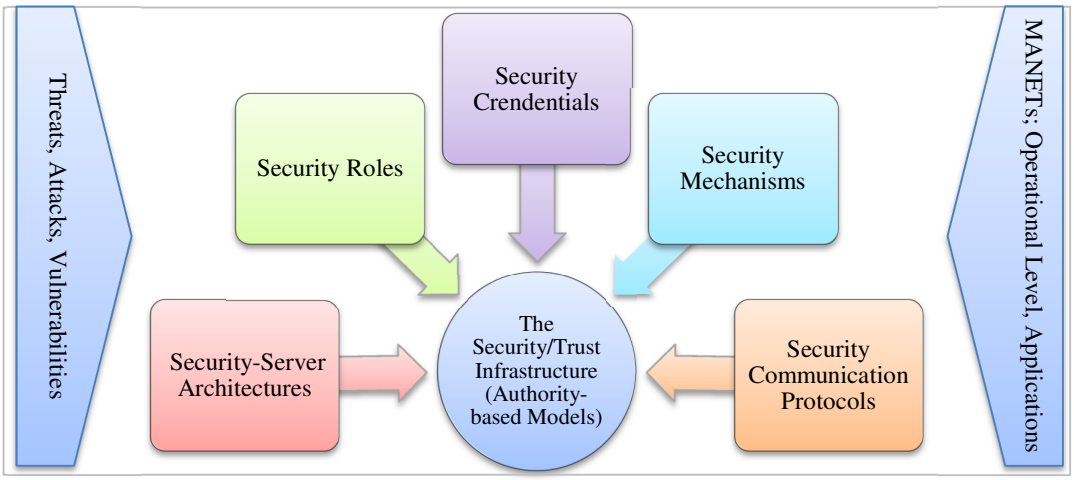


Figure 4-3: Security/Trust Infrastructure building blocks

3. The ***Security Communication Protocol***: denotes to the particular protocol which is used to establish connection (i.e. a trusted channel) between user and security server nodes in the security/trust infrastructure. This is in order to enable user nodes to obtain security credentials from certain security servers. However, two distinct communication channels are commonly used in any security/trust infrastructure, peer connection and out-of-band channels. The peer connection channel relies on MANET routing infrastructure involving one-hop or multi-hop networking. The out-of-band channel refers to any communication medium other than MANET, such as physical contact, and location-limited side channel (Balfanz et al., 2002). This channel has also its own interaction approaches for utilisation, but these approaches are out of this study's scope as the focus is on using MANET medium. Therefore, the particular security communication protocol can be represented simply by one call followed by one successful rely (i.e. one-way protocol), or by a series of different handshakes (i.e. two- and three-way pass protocols, etc.) using a peer

connection channel. Besides, the protocol (if required) can involve a specific authentication protocol (i.e. called "authentication exchange" described in Section 2.3.3.4) which aim to validate identities and to create a protected link among communicating nodes, for example Extensible Authentication Protocol (EAP) (Aboba *et al.*, 2004). Eventually, the protocols can vary from proprietary to standard protocols.

4. The ***Security Mechanism***: represents the security cryptographic function employed in the security/trust infrastructure to generate and validate trust (i.e. security credentials), such as asymmetric, symmetric and threshold cryptography. Therefore, this particular security element appears primarily to be associated with the type of the security credential and also the specific security-server architecture being utilised. For example, using secret keys should rely on a symmetric cryptosystem whereas a pair of different keys (public and private keys) is typically generated via using asymmetric cryptosystem. Also, both symmetric and asymmetric cryptosystems are distinguished with different configurations, key lengths and algorithms (e.g. symmetric algorithms: AES, Blowfish, DES, etc.; asymmetric algorithms: RSA, ECC, ElGamal, etc.). It is worth pointing out that applying a symmetric cryptosystem often takes less computation cost than an asymmetric one. However, as the asymmetric cryptosystem uses a public key, it is much more scalable than the symmetric cryptosystem (Smart, 2003; Stallings, 2010). Alternatively, threshold cryptography -$TC(k , n)$ (Desmedt and Frankel, 1990), as detailed in Section 2.3.3.6, is usually used to allow n multiple trusted entities (signers) to generate partial signatures relying on an asymmetric cryptosystem using a particular public-key signing algorithm (e.g. RSA, DSA, Schnorr, etc. (Saxena *et al.*, 2003)). Obtaining k partial signatures can be enough to produce a valid complete signature on a particular credential. Therefore, threshold cryptography is intended to create a collaboratively or dependently distributed model of a signing service (i.e. deployed in multiple security servers) for more security and availability. This technique however experiences a number of challenges: (1) adjusting adequate threshold values k

and *n* to provide strong security and low delay, (2) the complexity of sharing management (i.e. distributing and updating and verifying secret shares among active security servers).

5. The ***Security Credential*****:** indicates a type of a credential being used in the security/trust infrastructure to facilitate the building of trust associations and secure utilisation. However, each credential type is associated with a particular security-server role involving in the infrastructure. For example, a CA server is typically used to provide users with digital certificates (as described in Section 2.3.8). Various formats of credentials can be acknowledged, such as public and secret keys, certificates, and access tokens and tickets. Furthermore, the same credential type, like a X.509v3 digital certificate (Chokhani et al., 2003; Cooper et al., 2008), can be exploited for different purposes, identification (i.e. using name or Public-Key certificates), authorisation (i.e. using attribute or role-based certificates) and security policy management (using a policy certificate). Access tokens and tickets can be created in particular Kerberos (Neuman and Ts'o, 1994; Neuman et al., 2005) and AAA security systems (Larafa and Laurent, 2009), respectively. Different security credentials differ in terms of size, generation and distribution overheads and security robustness. Therefore, this may instigate some challenges in using particular credentials, especially in limited and vulnerable networking systems like MANETs.

A number of existing relevant security proposals for MANETs (discussed mostly in Section 2.5) are selected to be analysed against the proposed building blocks in the security/trust infrastructure, as presented in Table 4-1. Some authority-based security proposals, such as cluster-based solutions are out of scope of this design. The reason for this is as follows. These specific proposals show a strict interdependency between routing and trust infrastructures leading to apparent violation the proposed two-level design described in Section 4.2. On the other side, the clustering domain for these security proposals also entails more management and operation complexities (i.e. clustering algorithms and the

selection, configuration, maintenance, and replacement of CHs, merging clusters, node cluster membership management, etc.). This indicates the necessity for this domain to be tackled in its own for an effective evaluation.

In Table 4-1, various security roles can be realised in these presented proposals, such CA, AAA, TGS, coordinator & PCA, etc. Almost all security proposals are established on a distributed architecture irrespective of a distribution type (duplicate, hierarchical, etc.), excluding the centralised proposals, Keoh *et al.* (2004) and Jansen *et al.* (2003). Regarding the security communication protocol being used, many proposals leverage the simple one-way communication protocol, such as Zhou and Haas (1999), Luo *et al.* (2004), Pirzada and McDonald (2004), Keoh *et al.* (2004) & Jansen *et al.* (2003), Luo *et al.* (2005), Raghani *et al.* (2006), Omar *et al.* (2007), and Al-Bayatti *et al.* (2009). Some other proposals incorporate their own proprietary protocol for facilitating communication between user and security servers, such as Yi and Kravets (2003, 2004), Hadjichristofi *et al.* (2005a; 2005b), Luo *et al.* (2005), Wu *et al.* (2007b), Saremi *et al.* (2009) and Khakpour *et al.* (2008). Only two proposals make use of a standard authentication protocol, the (EAP) protocol in Larafa and Lauren (2009, 2011) and Kerberos protocol in Pirzada and McDonald (2004). Few proposals have proposed the out-of-band channel as the main means to obtain security credentials in their security infrastructure, for example Luo *et al.* (2005) and Hadjichristofi *et al.* (2005a; 2005b). It is noticed that 11 out of 16 presented security proposals take advantage of asymmetric cryptography and threshold cryptography (*TC*), for the sake of more protection robustness and better availability and scalability. Several proposals make use of a certificate credential with different purposes (name, authorisation, policy, etc.), and sometimes adopt the standard X.509 certificate (Chokhani *et al.*, 2003; Cooper *et al.*, 2008). Some proposals rely on a particular standard security system which generates its own unique credential, such as a Kerberos ticket in the Kerberos system and AVP access token in the AAA system.

Security Infrastructure Proposal	Security Role "Server Task"	Security-Server Architecture	Security Communication Protocol	Security Mechanism	Security Credential
DICTATE - Luo *et al.* (2005)	- Revocation authority servers (i.e. dCAs). - Identification authority servers (i.e. mCA).	Distributed & Hierarchal	- dCAS: a simple one-way communication protocol within the range of a dCA & PILOT, a proprietary group communication protocol. - mCAS: an out-of-band mechanism.	- Asymmetric cryptography (RSA) - Threshold cryptography (TC)	Name certificates
MOCA - Yi and Kravets (2003)	Certification authority servers (i.e. MOCA):	Distributed	A proprietary MOCA certification protocol relying on its own routing table for MOCAs apart from a standard routing MANET protocol.	- Asymmetric cryptography - Threshold cryptography (TC)	Public-key certificates
KAMAN - Pirzada and McDonald (2004)	KDC - Ticket Granting Server (TGS)	Distributed - server duplicates	A Kerberos-assisted protocol (similar to a one-way protocol)	- Symmetric cryptography	Kerberos tickets
Zhou and Haas (1999)	CA & combiner servers	Distributed	A Simple one-way communication protocol	- Asymmetric cryptography - Threshold cryptography (TC)	Certificates
SEKM - Wu *et al.* (2007b)	CA servers	Distributed	A proprietary Flooding protocol by broadcasting the request at the beginning (server overlay)	- Asymmetric cryptography (RSA) - Threshold cryptography (TC)	X.509 certificates
Hadjichristofi *et al.* (2005a; 2005b)	Different CA servers (i.e. RCA, DCA and TCA)	Distributed & Hierarchal	Two types of communications : - Out-of-band (RCA, DCA &TCA) - Peer connections only for DCAs using a proprietary protocol (one call for a DCA and *n* replies from other accessible DCAs.	- Asymmetric cryptography	X.509 certificate chain
Saremi *et al.* (2009)	Authoriser Servers	Distributed	A proprietary four-way communication protocol	- Asymmetric cryptography - Threshold cryptography (TC)	Authorisation certificates
Larafa and Lauren (2009,	AAA Servers	Distributed	An Extensible Authentication	- Asymmetric cryptography	Access Token -

2011)			Protocol (EAP) is used by authenticator and is based on a three-way communication protocol.	- Threshold cryptography (TC)	Attribute Value Pair (AVP)
Al-Bayatti *et al.* (2009)	Backbone Nodes (BBN) work as authorisation servers within different third parties.	Distributed	A simple One-way communication protocol	- Asymmetric cryptography - Threshold cryptography (TC)	Authorisation certificates "security clearance"
Raghani et al. (2006)	CA servers	Dynamic Distributed	A simple One-way communication protocol	- Asymmetric cryptography (RSA) - Threshold cryptography (TC)	Certificates
NetTRUST - Omar *et al.* (2007)	Hierarchy of CA Servers (i.e. CCA, MCA)	Distributed & Hierarchal	A simple one-Way communication protocol	- Asymmetric cryptography (RSA) - Threshold cryptography (TC)	X.509 certificates
CKM - Yi and Kravets (2004)	Apart from available MOCA, Some trusted users associated with MOCAs are allowed to issue certificates.	Distributed	A proprietary MOCA certification protocol relying on its own routing table for MOCAs apart from a standard routing MANET protocol.	- Asymmetric cryptography - Threshold cryptography (TC)	A X.509 certificate chain starting with MOCA vouching
URSA - Luo *et al.* (2004)	All networks nodes do a CA job	Fully Distributed	A simple one-way communication protocol	- Asymmetric cryptography - Threshold cryptography (TC)	URSA tickets
Peace - Keoh *et al.* (2004) and Jansen *et al* (2003)	The admission coordinator and policy server (i.e. PCA)	The single central	A simple one-way communication protocol (joining and enrolment transaction)	Asymmetric cryptography	An admission list and a policy certificate
Khakpour *et al.* (2008)	Hierarchy of AAA servers (i.e. a MAS, SASs)	Distributed & Hierarchal	A Proprietary six-way communication protocol	Asymmetric cryptography	A *Inf1* ticket

Table 4-1: Security proposals against the building blocks of the proposed security/trust infrastructure

Eventually, since the proposed security/trust infrastructure is primarily designed for the MANET and its applications, building blocks aforementioned in the security/trust infrastructure should arguably be tailored to the demands of this network and its applications. These demands can stem from the unique MANET

characteristics (i.e. limitations), the focused operational level (Section 4.2.1) and requirements of the application domain. On the other hand, some of the building blocks, discussed above (Figure 4-3), may be susceptible to different security threats and attacks (e.g. Man-In-The-Middle, credential forging, Denial of Service (DoS), etc.). For this reason, it is important to take this security concern into consideration when dealing within these particular building blocks for an effective infrastructural design. The next section discusses the key dimensions each of which has an influence on the design and implementation for a certain security/trust infrastructure. These dimensions will be involved in the proposed approach of this study for facilitating the development of an applicable security/trust infrastructure matching context requirements. Prototype

4.4 Multi-Dimensional Framework for MANET Security

The design and development of the security/trust infrastructure appear to significantly be affected by a number of key dimensions: *security strength*, *performance* and *context*, as shown in Figure 4-4. Especially, some related building blocks of the security/trust infrastructure described in the previous section such as security-server architecture, a security mechanism, a security credential and a security communication protocol can be shaped by these particular dimensions as this will be addressed in Section 4.4.1 , 4.4..2 and 4.4.3. Generally, even though most security solutions are proposed originally to solve particular security problems (e.g. trust facilitation), these solutions may still experience security weaknesses (i.e. vulnerabilities, etc.) as there is no security panacea. Hence, a security solution should allegedly show a sufficient level of robustness against potential threats and attacks. However, any enhancement in security will definitely lead to increase the cost of computation, communication, and management, as a result of extra measures that must be applied for supporting protection. Also, as many existing proposals look at their security solutions from the cryptographic angle, they unintentionally neglect the aspect of performance

(Yang *et al.*, 2004). Therefore, security performance becomes an important factor, especially for developing a security solution for resource-constrained MANETs. Thus, both dimensions of security strength and performance are very crucial to be equally considered for better security design of MANETs. This can be typically realised by making an appropriate trade-off between them, based on the requirements of a certain context being tackled. In this study, the dimension of the MANET context is proposed primarily to describe MANET capabilities (e.g. constrained resources) and application settings (e.g. requirements, environment, etc.). This is due to the fact that information extracted from these two scopes can arguably assist to identify the importance of some related issues (i.e. security and performance dimensions) to the security design of MANETs.

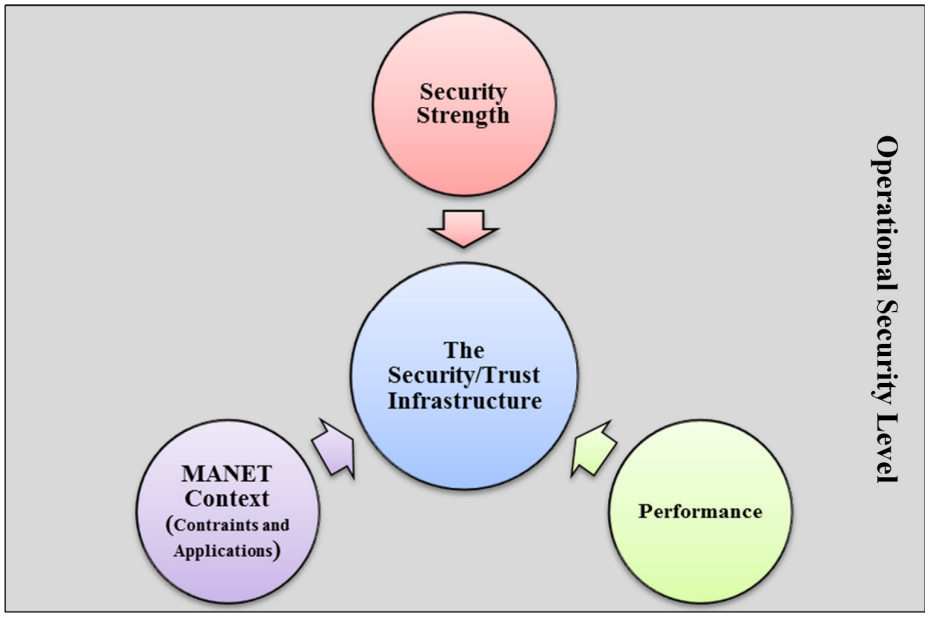


Figure 4-4: Key dimensions related to the security/trust infrastructure

Developing an effective and efficient security solution for MANETs is still a challenging task since several dimensions, operational levels and restrictions, which have a significant impact on the security design of MANETs, are involved. Few attempts have addressed some of these issues, for example, in Balakrishnan and Varadharajan (2005), three dimensions (i.e. cryptographic function, and behavioural-trust and resources management), are identified and must be taken

into account when securing MANETs. Only two of these proposed dimensions, cryptographic function and resources management are relevant to the proposed multi-dimensional framework of this study. The aspect of cryptographic function can be involved in the proposed building blocks of the security/trust infrastructure whereas the aspect of resource management can be considered in the context of MANET constraints. Cayirci and Rong (2008) only highlight the five key attributes (i.e. applicability, security, robustness, scalability, usability) which are used specifically for analysing different existing key management schemes for MANETs, not for design purposes. Similarly, Merwe *et al.* (2007) present a list of detailed security attributes relevant to a key management system (e.g. confidentiality, key freshness, forward secrecy, availability, robustness, etc.). Some of these attributes (applicability, security, robustness, and availability) in both works are selected to be integrated in the dimensions of this proposed framework. It appears that there is a lack of a systematic approach to design a security solution from different perspectives (i.e. an operational level, performance and context). The perspective of the operational level particularly shows an absence in the design view of MANET security even thought this perspective is crucial in design. A different operational level can usually introduce different security concerns and performance assessment (i.e. different metrics). For example, security threats and attacks vary in the service level (application & transport layers) and the network level (MAC & network layers), as shown in Figure 2-3. Also performance assessment in the network layer (different protocols, secured routing overheads, bandwidth utilisation, etc.) differs from performance assessment in the application layer (a response time, security server availability etc.). However, in this study, the proposed service operational level is the main focus on applying this framework. The next subsections will discuss in detail each of the proposed dimensions in this multi-dimensional framework.

4.4.1 Security Strength

The dimension of security strength presents all security concerns and protection issues that the security/trust infrastructure (considering its building blocks discussed in Section 4.3) may encounter. This dimension is intended to examine the resistance of the security/trust infrastructure against any potential threats and failures (e.g. denial of service (DoS), single-point-of-compromise, man-in-the-middle, etc.). This stems from the fact that security building blocks involved in the security/trust infrastructure are expected to suffer from weaknesses. For example, adopting distributed servers in the infrastructure may alleviate the disruption caused by a DoS attack. Alternatively, security strength of the security/trust infrastructure can also reflect the security requirements (i.e. confidentiality, non-repudiation, mutual authentication, etc.) being already satisfied in this infrastructure for more protection. For example, taking advantage of a different security communication protocol can show different security features applied, such as the one-way pass protocol attains unilateral authentication and a two-way pass protocol can fulfil mutual authentication (Boyd and Mathuria, 2003).

However, in contrast to other networks, MANETs are vulnerable to different threats and attacks, especially in the network layer, as detailed in section 2.3.2. Therefore, the security/trust infrastructure which is originally developed for MANETs, could inherit most of security problems from this type of network. As this study proposes two levels of operations (i.e. service and network levels) for MANETs, security problems can be tackled independently, especially when designing the security/trust infrastructure for MANETs. Eventually, the security/trust infrastructure should demonstrate enough security strength through withstanding any potential threats and attacks, and satisfying most important security requirements. However, enhancing "security strength" may result in a performance overhead due to additional measures that are required for more protection.

4.4.2 Performance

Performance is a very important dimension for the design of a security/trust infrastructure for MANETs since any security enhancement (e.g. using security end-to-end protocols, asymmetric cryptography, etc.) in this infrastructure would entail a performance overhead (more processing and communications). This particular dimension is fundamentally associated with security service availability, rapidity, reliability, scalability, etc. (Yang *et al.*, 2004; Merwe *et al.*, 2007; Cayirci and Rong, 2008). High-availability and good-reliability attributes are very desirable in order to avoid any degradation in security service and to ensure that a credential is provided to user nodes in the network when anticipated. Also, satisfying a low-delay feature in the response time of security services is very crucial as in some situations, such as rescue mission and battle field applications time is very critical for particular applications,. High scalability can be realised when the security/trust infrastructure is able to seamlessly handle many nodes invocations regardless of the varying network sizes and node densities.

However, unlike the other conventional networks, MANETs experience unique performance challenges which may affect the service security utilisation. This is due to the fact that bandwidth in MANETs is shared for routing and utilisation purposes at the same time, in other words, each MANET node has a dual role (i.e. a router and host). Hence, it is vital to keep network traffic as low as possible. This is because any increase in traffic leads to limit the available bandwidth for loading data and consequently this will have an impact on the whole performance of MANETs including security services. Furthermore, node mobility and churning in MANETs introduce more network partitioning and broken links which may cause performance disruption in the network. MANETs are typically characterised with limited resources (i.e. a battery, small memory and CPU, etc.) which results in more restriction on the network performance (i.e. long delays, etc.) as well. Therefore, the MANET attributes need to be considered carefully for achieving desired performance.

Eventually, performance of a security/trust infrastructure can be affected by a number of factors in MANETs, for example MANET specifications, an application scope, and the strength of security measures being involved in the service\trust infrastructure. Therefore, the so-called security performance is arguably still a very key element that must be taken into consideration in order to enable security designers to produce a successful and effective security design in MANETs.

4.4.3 MANET Context

Generally, the term "*context*" in computing can refer to many aspects, such as environment, location, computing capabilities, application types, time, etc. (Dey, 2001). Accordingly, a MANET context can be perceived in different ways. In this study, the context boils down to two principal scopes: MANET constraints and applications settings. This is due to the fact that these two scopes can arguably shape most of features in the MANET networking system (i.e. hardware and software specifications). Hence, in the design phase, these features can be taken into account for selection an applicable design of a security/trust infrastructure for MANETs. The first scope ("MANET constraints") describes the substantial physical and networking limitations in MANETs, whereas the second scope ("MANET application settings") mainly highlights MANET application requirements (complexity and functionality) and environment settings that need to be satisfied. The next subsections discuss these two scopes in more details.

4.4.3.1 MANET Constraints

As explained in Section 2.2.2, several attributes of MANETs can be acknowledged, such as infrastructure-less-ness, dynamic topologies, poor physical security, limited resources, etc. (Chlamtac *et al.*, 2003; Djenouri *et al.*, 2005; Savola and Uusitalo, 2006; Merwe *et al.*, 2007; Chadha and Kant, 2008).

However, some of these attributes imply some critical limitations (e.g. limited resources, etc.) in MANETs which instigate more considerations especially when a security/trust infrastructure is being designed for MANETs. For example, as MANETs usually have limited resources in terms of power and capabilities, light-weighted cryptographic functions should be proposed. Therefore, in this section, these particular limitations are identified to create a model of constrains for MANETs. This constraint model can be incorporated in this study approach to support designing an effective security/trust infrastructure which matches actual MANET potentials. Table 4-2 summarises the key MANET constraints and possible consequences if these constraints are not taken into account in the design. It is worth mentioning that most of these constraints presented herein appear to mainly affect the availability of connectivity in MANET (i.e. disrupting routing and networking activities).

Constraints	**Possible Impacts/Consequences**
Limited resources (i.e. battery power, small memory and CPU, etc.)	Having frequent node failures, and experiencing long latency
An unstable topology (i.e. stemming from node mobility, joining and leaving common behaviours	Incurring frequent network partitioning
A limited wireless bandwidth	Creating unreliable and broken links
Poor physical security	Easy to be compromised especially in hostile environments.
Routing protocol shortcomings	Leading to lower packet delivery rates

Table 4-2: MANET constraints and their potential consequences

However, for a security/trust infrastructure design, the model constrains mentioned above introduce different security design challenges. It is crucial to search for inexpensive-computation cryptographic mechanisms and lower-communication-overhead reliable security protocols as to efficiently manage power, capacity and bandwidth restrictions in MANETs. Taking advantage of

well-distributive adaptable security architectures is desirable so as to overcome topology instability in MANETs. A robust device security technique, for example using biometric identification, can alleviate serious consequences of poor physical security in MANETs. To conclude, it is important to involve MANET constraints in developing the security/trust infrastructure for MANETs as these constraints become MANET boundaries to which each security component in the infrastructure should adhere.

4.4.3.2 MANET Application Settings

Many potential MANET applications in different domains can be acknowledged, as presented in Section 2.2.3. This stems from the fact that this particular network has an elegant way for providing nodes with interconnection support where no communication infrastructure is available, or where the installation of a permanent infrastructure is economically not workable. However, MANET applications differ in terms of their usage requirements (e.g. third-party-configuration, self-initialisation, self-management, etc.) and their domain settings (e.g. small- or large-scale MANETs, hostility, urgency, etc.). These two factors (i.e. requirements and domain settings) in the application context define the complexity and cost of a MANET initialisation, operation, and management.

Furthermore, the scope of the application context is very influential in this study. This is due to the fact that the well-defined requirements of a certain application enable security developers to characterise the importance of the other related dimensions, such as performance and security. Therefore, the different choices of security/trust infrastructures for that MANET application can be prioritised accordingly. For example, depending upon the network application purpose, its security and performance requirements in the military settings vary according to the encountered circumstance. Confidentiality and availability are the most important issues in a battlefield, whereas in a humanitarian rescue mission scenario, availability is more important than confidentiality. Hence, the security

and performance requirements in every case can be formulated from the tackled application context. For better understanding the demand of applications, several general properties can be acknowledged for applications over MANETs, such as *dependency*, *lifetime*, *capacity*, and *environment* settings. These particular properties described in this section are representative, but not exhaustive as other new application properties can be considered, if relevant. In addition, The properties of dependency, lifetime, and capacity are based on Hoeper and Gong (2004) and Dawoud *et al.* (2011).

The network application *dependency* (i.e. planned or self-organised) indicates whether the application relies on an independent governing body (i.e. an administration or authority) or not (i.e. a self-organised application) for configuration and management before the nodes start participating or joining the network. The planned MANET application is proposed to be pre-configured by a certain body with necessary materials, such as, certificates, policy documents, shared secret keys, etc. However, in the self-organised network application, nodes are allowed to use the application without any assumptions of prior configuration or security associations. This feature can be realised usually in applications for pure ad-hoc networks as the nodes, in these applications, can join and use the network freely without complicated initialisation. Also, it is worth pointing out that in the planned applications, different authority involvements (i.e. initialisation, operation and management) and different types of authority structures (i.e. flat or hierarchical authorities) can be distinguished. This presents more complexity and overhead to the initialisation, operation and management of applications in MANETs.

The network application *lifetime* describes a time period that a certain type of a MANET application lasts (i.e. in operation until the end of the application mission). It can be short time or long time. For the short time application, all nodes with a simple initialisation are expected to establish a MANET and then interact with each other on a specific purpose for a relatively short period of time, for example few hours. When these nodes meet the target of this application, all

information about these nodes and their relations (e.g. certificates, session ids, etc.) will be eliminated. However, in the long-time applications, all nodes intend to stay longer (e.g. days or months) in MANETs and they may join or leave the network frequently. Therefore, these nodes need to retain previous associations and relevant information for future interactions in particular resources, for example directories, databases, etc. The long-time application, unlike the short-time application, shows much more complexity in initialisation (e.g. trust establishment) and introduces more concerns in resource consumption (i.e. resource limitation).

The network application *capacity* (i.e. small-scale and large-scale) stands for a MANET space and node population which a particular application can serve. This is because the MANET depends on all nodes for handling the routing activities as well as for running its existing applications. On the other hand, some applications require nodes to be in close vicinity (i.e. a localised area) in order to be functional (e.g. applications for meetings or classes). Also, these applications are typically involved in small-scale manner (i.e. a small number of nodes and a limited space) working with short range communication (e.g. one-hop or physical interactions). In the large-scale applications, several MANET nodes are distributed within a fairly large area. Hence, these applications, as opposed to the small-scale application, need a multi-hop routing infrastructure for allowing far dispersed nodes to interact with each other. Also, in these applications, the MANET must have sufficient distributed resources for performing application's activities effectively. Nevertheless, the initialisation cost and complexity in the large-scale application are often greater than in the small-scale as a result of dealing with numerous nodes.

The application *environment* settings refer to specific characteristics a certain application may require to suit domain demands, or describe specific conditions a certain application may experience. For example, a particular application, working in a *hostile* environment (contrary to a *non-hostile* one), poses much more risk to this application to be violated by adversaries. Hence, this entails much more

considerations (security issues) to the application protection measures that are necessary to be implemented in order to avoid such breaches within this particular environment. On the other side, some environments, such as a rescue mission and ambulance service and law enforcement scenarios, are time-sensitive (i.e. *urgency*), so any applications running in these environments should take into account the time factor seriously. Therefore, the MANET applications must be responsive and reliable in all times. In conclusion, the highlighted properties of a MANET application can be a cornerstone for establishing a model of application settings for MANETs. This model can be incorporated into the MANETs design for more effective and efficient solutions, especially in designing a security/trust infrastructure as summarised in Table 4-3.

Application Settings	Description
Dependency	Describing whether a particular MANET application requires a third party (a governed body) for configuration and management. Two different settings can be distinguished, planned and self-organised features ones.
Lifetime	Indicating a time duration in which a certain type of a MANET application is on operation. Two types of lifetime can be identified, *short-time* and *long–time*.
Capacity	Representing a MANET space and node population in which a particular application can serve over this network. The *small-scale* and *large-scale* settings can be recognised.
Environment	Reflecting specific characteristics and conditions a certain application may necessitate for achieving demands of this application domain such as *urgent* and *hostile* environment.

Table 4-3: Application settings and their summary descriptions

4.5 Methodological Approach Discussion, Justification and Motivations

An approach, which is established on the relevant elements and dimensions of security/trust infrastructure clarified in previous sections, is developed in this study in order to guide MANET security designers to effectively design the security/trust infrastructures. This proposed approach, as shown in Figure 4-5, consists of well-structured stages allowing for exploring all possible security alternatives and evaluating those nominated security alternatives against the criteria of security strength, performance and MANET context. Firstly, in this approach, **(1)** security designers should decide about the operational level being targeted in MANETs (as discussed in Section 4.2.1 and 4.2.2). This helps them to consider only specific security/trust infrastructures and their related security elements that are linked to this particular chosen level. Also, the possible security/trust infrastructures should be narrowed down according to the suitability for a domain study. Therefore, **(2)** a list of all potential security/trust infrastructures characterised in previous stage will be created and prepared for the next assessment stage. In the assessment stage, **(3)** all listed infrastructure alternatives should be examined regarding their security strength and performance and the MANET context being involved. **(4)** The outcome of infrastructures' evaluation will be compared to find the best suited security design satisfying the MANET context requirements by using a certain decision making mechanism. However, during applying this approach, security designers can develop their own security/trust infrastructure which be involved in this approach in case there is a lack of a feasible infrastructure matching MANET context demands.

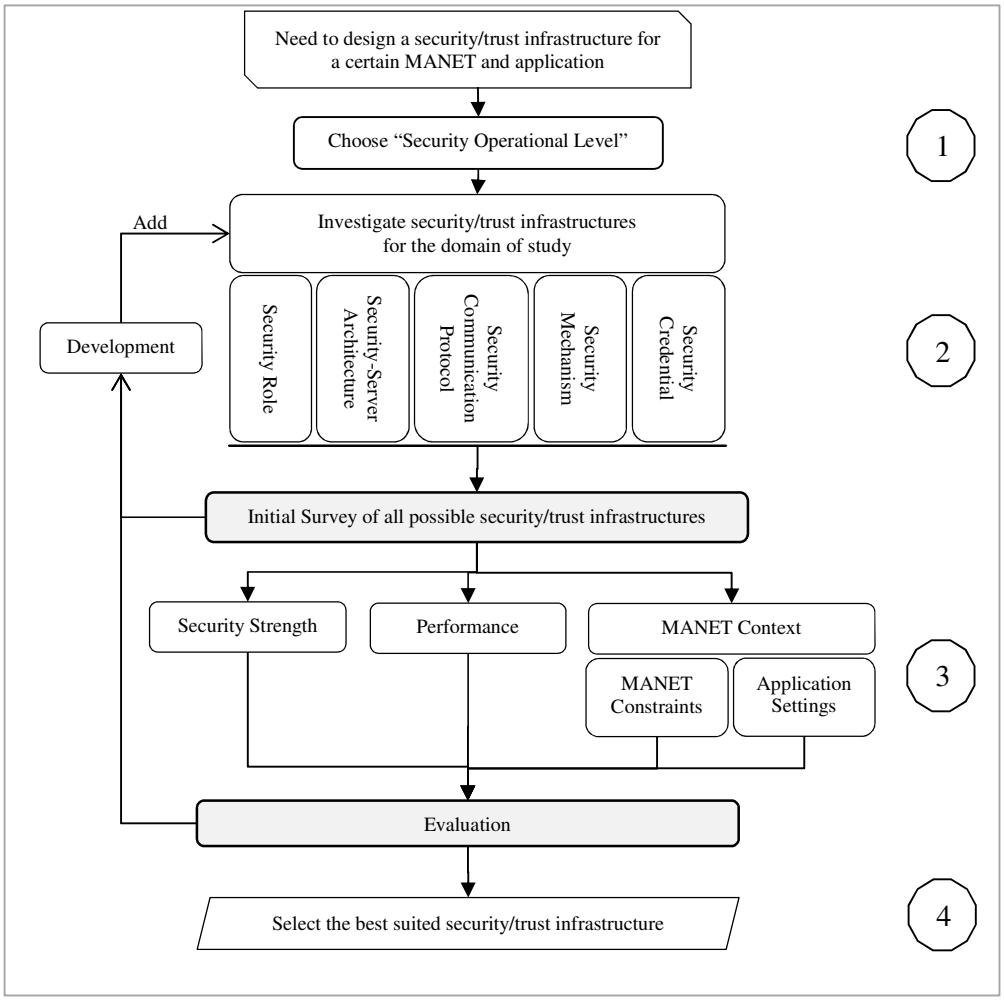


Figure 4-5: The initial proposal of methodological approach for designing the security/trust infrastructure for MANETs

Eventually, most of security proposals in this domain appear to focus on cryptographic and protocol-based viewpoints and ignore other aspects, performance, context, level which may have a significant impact on those particular solutions when implemented and deployed over the network. Hence, this approach aims to establish a balanced view of how to design and develop a multi-objective security solution (i.e. a security/trust infrastructure) for MANETs taking into consideration (i.e. a right trade-off) other dimensions associated with security such as an operational level, performance, and a MANET application context. In addition, this approach arguably becomes a key enabler for producing

an effective and efficient MANET security design through providing a set of guidelines to which security designers can adhere. Eventually, this approach will be applied to the proposed model of SSAM as will be seen in the next section and the following chapters throughout implementation and evaluation phases.

4.6 Server-based Security Architectures for MANETs (SSAM) – PART B

In order to put all aspects discussed before into practice, it is essential to conduct a case study to prove the feasibility of the proposed approach. Therefore, a novel suite of standard security/trust architectures, known as the ***S***erver-based ***S***ecurity ***A***rchitectures for ***M***ANETs (SSAM), is proposed in this study for two main objectives. Initially, (1) the SSAM model will offer a new security model for the "Service Level" in large-scale MANETs where there should be hundreds of nodes joining the network in order to make use of available offered services (e.g. video streaming, internet gateway etc.). Secondly, (2) this model can be exploited to perform the evaluation and validation of the proposed methodological approach in this thesis. However, proposing this particular model in this context is as a result of the complexities and unresolved overlapping interdependencies in the current security solutions for MANETs, such as using proprietary protocols, integrated in different layers, etc. Also, there is a lack of a standard and organised model shaping most of those solutions for effective design and evaluation.

Therefore, the model of SSAM is fundamentally derived from the exploration of different trust models for MANETs discussed in Section 2.5. In addition, this model is designed and established based on the key specific security building blocks which are illustrated in detail in Section 4.3. These building blocks are briefly suggested for formalising any security/trust authority-based infrastructure for MANETs to handle trust and authentication. On the other side, the "Service Level" is considered as the main design aspect of a security level for SSAM. This level indicates to where the components of SSAM are assumed to be implemented and utilised. The reason is that this particular design level appears to be neglected

by a MANET research community and the focus is more on the "Network level", especially the routing issue (Cho *et al.*, 2011). Thus, the client-server model is adopted for performing security services proposed in SSAM as this particular model is very popular in this design level (as discussed in 4.2.1). This solution practically describes how to make use of available different alternatives of trust infrastructures under particular settings of MANETs for the sake of fulfilling the following properties:

- Maintains higher availability and better scalability to security services.
- Improves performance and efficiency of security services (an optimal round trip time or delay).
- Provides a standard level of protection through securing nodes and their communications.
- Take into consideration MANET resources limitations and application requirements.
- Enhances manageability in certification (issuance, renewal and revocation).
- Be more flexible in managing security policy.

Therefore, SSAM is dedicated to offer an appropriate security service infrastructure (i.e. security servers) which takes the role of issuing and distributing membership certificates to user nodes in large-scale MANETs. These generated certificates are intended to be a general-purpose credential which can be used for different security purposes, such as proofs of authenticity, admission and authorisation. In addition, having these membership certificates would arguably allow user MANET nodes to utilise available services (e.g. streaming, downloading, browsing, etc.) like in a civilian context, or to take a decision when receiving orders like in a battle field (e.g. initiate an air strike). However, in this study, SSAM only concerns with the issues of security service providers (i.e. generating and delivering credentials) whereas the utilisation of other service providers (i.e. establishing associations between service users and service providers) is out of the SSAM scope in this study.

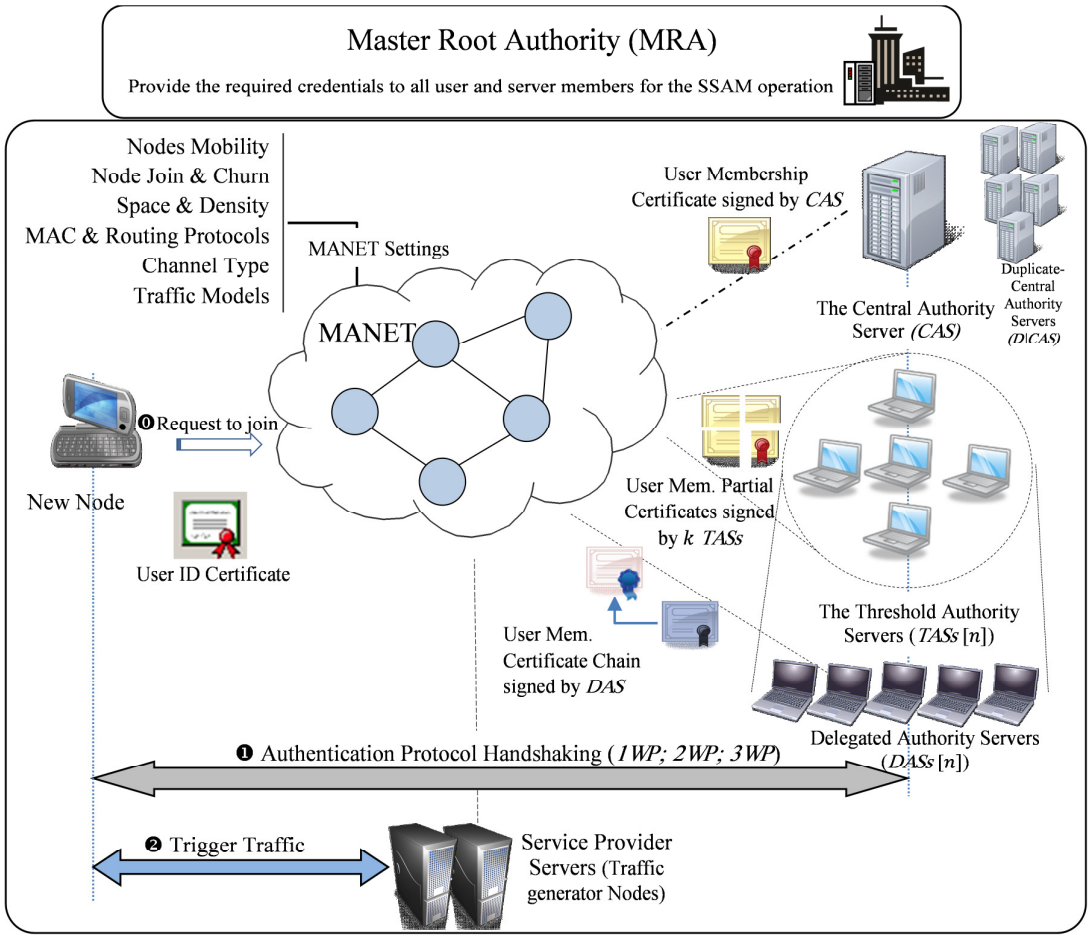


Figure 4-6: The SSAM Conceptual Model

Complying with the taxonomy of a security/trust infrastructure highlighted in Section 4.3, SSAM consists of different security architectures each of which is represented by the types of the server architecture and the authentication protocol being used. However, the term "Security Architecture" is intended to express the partial security design of the whole security infrastructure. The server architecture presents the form of security servers involved to run security services, such as central and distributed server architecture. The authentication protocol denotes to a specific security communication protocol which is used to secure a connection between a calling user node and a particular security server over MANETs, for instance, a three-way pass authentication protocol, etc. In addition, in the SSAM model, a standard digital certificate (i.e. X.509-*v3* (Chokhani et al., 2003; Cooper

et al., 2008)) and asymmetric and threshold cryptograph (e.g. RSA) (Gennaro *et al.*, 2008; Dossogne *et al.*, 2013) are incorporated. The next subsections will explain the details of system model in SSAM, as shown in Figure 4-6.

• Initialisation (Configuration):

At the enrolment stage, a Master Root Authority (MRA) is devoted to manage the admission of a node (either a server or a client) to the network (i.e. MANETs) following the predetermined authority rules and policies. This is through issuing a node's certificate (i.e. user or server ID certificates) which confirms the binding between the node identity and its initial public key. In addition, MRA is presumed to provide all enrolled user nodes with required certificates in order to enable those nodes to use available security services in SSAM, such as membership validation certificates and ID server certificates. On the other side, all management processes in all types of existing authority servers should be managed by a MRA, similar to Luo *et al.* (2005), for example, generating and updating secret keys used in issuing, refreshing secret key shares in threshold authority servers (*TASs*) and checking servers still in operation.

• Server architectures

As presented in Table 4-1, different security-server architectures can be acknowledged to deploy and provide security services in a MANET. It is noted that most of them obviously have certain common themes, as shown in Figure 4-7. These themes can be identified by four key types of security servers: (1) a single standalone server, (2) multiple collaborated servers, (3) multiple standalone duplicated servers and (4) multiple standalone delegated servers. For consistency and avoiding confusion, a "Security Server" is intended to be described in SSAM as an "Authority Server" to imply both authority and security roles at the same time because the SSAM model is established on the authoritarian security schemes in MANETs. On the other hand, this term is believed to implicate a broad meaning (i.e. including authority and security tasks). Therefore, these key authority servers (shown in Figure 4-7) on which different proposed server architectures rely, are described as follows:

1. A Central Authority Server (*CAS*) is a single standalone server whose mission is to offer a security service (i.e. issuing, renewing and revoking credentials) exclusively for all nodes within the network. Despite the fact that the *CAS* architecture often suffers from a single point of failure; this architecture still presents promising potentials to exploit. These potentials can be realised by administration simplicity (i.e. manage only one server) and easy integration in other security solutions (e.g. hierarchical and combined architectures). On the other side, as *CAS* is typically working a standalone system, this server is usually assumed to be fully protected and well-equipped.

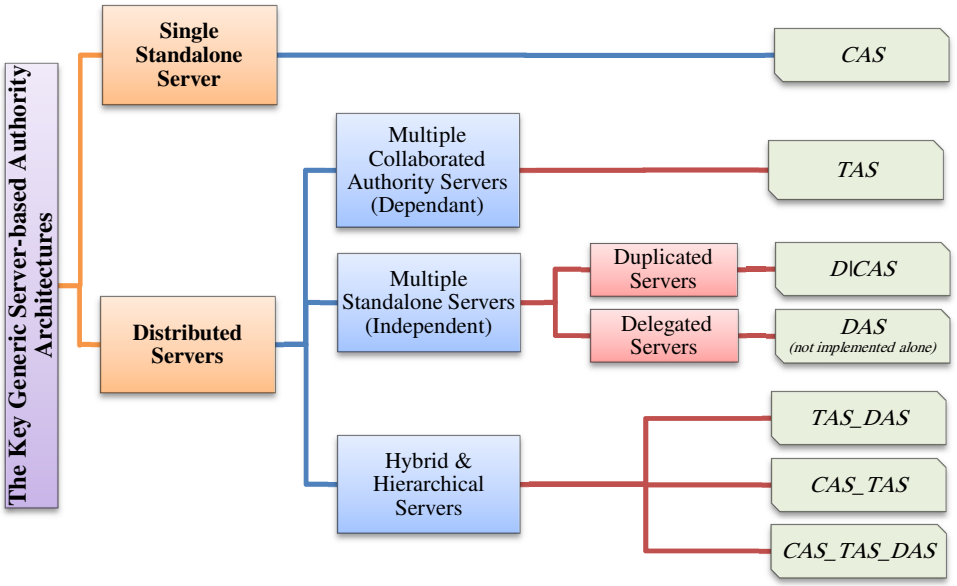


Figure 4-7: The common authority server architectures in SSAM along with corresponding examples for MANETs

2. Threshold Authority Servers (*TASs*) are considered multiple collaborated servers which count on the Threshold Cryptography (k , n) scheme (illustrated in Section 2.3.3.6). The architecture, involving this particular scheme, requires a coalition of certain authority servers (i.e. generating k partial certificate from different servers) to provide a fully-functioning security service. A user node can produce a valid membership certificate in this server architecture through combining k partial certificates received from reachable *TASs*. The *TAS*

architecture demonstrates better fault tolerance and security robustness because of depending on the distributed dependant servers.

3. Duplicated Central Authority Servers (*D\CASs*) are multiple standalone duplicated servers which perform the same job as *CAS* with distributed independent instances. The *D\CAS* architecture is proposed to avoid single point of failure and to increase security service availability through having several identical server copies within the architecture. However, this architecture is vulnerable to a single point of compromise.

4. Delegated Authority Servers (*DASs*) are realised as multiple standalone delegated servers which are similar to *D\CAS* but with a small difference. Each of these servers must have their own delegation certificate (i.e. as a proof of authority permission) which allows these servers to issue their own certificate to users on behalf of the delegator (i.e. the Master Root Authority (MRA)). Hence, both delegation and generated user certificates should be piggybacked in the reply back to user nodes when *DAS* is invoked. It is important to mention that the *DAS* architecture is intended to be incorporated, particularly with the *TAS* architecture, and be used only in hybrid and hierarchical server architectures for a number of reasons. Initially, the *DAS* architecture alone appears to be similar to the *D\CAS* architecture in terms of functionality and independency of security services but with a little difference, an extra delegation certificate being shared with user nodes. On the other hand, for reducing the overhead of performing server election (*TASs* are typically elected with high confidence by *MRA*), each *TAS* is proposed to work as a *DAS* when required, in the *TAS_DAS* and *CAS_TAS_DAS* architectures. In other words, both security services of the *TAS* and *DAS* architectures are suggested to be deployed in one server as these two types of servers have the same distribution feature (i.e. a number of servers), which facilitates server integration in the combined architectures.

In the hybrid and hierarchical server architectures, various types of servers discussed before can participate jointly to offer security services to users, such as

the *TAS_DAS*, *CAS_TAS* and *CAS_TAS_DAS* architectures. There are two categories of the hybrid and hierarchical server architectures which can be recognised, two-level and three-level server architectures. The first one involves different two security services (i.e. *TAS&DAS* and *CAS&TAS*) whereas the second one involves three security services (i.e. *CAS&TAS&DAS*).

However, in SSAM, the degree of security service distribution, especially in the collaborated distributed server architectures, is proposed to be partial among MANETs node. This means that only a specific group of nodes (i.e. a predefined number of security servers) take the role of offering security services, unlike the other approach (i.e. fully distributed architectures) which gets all nodes to take part in providing security service, such as in Luo *et al.* (2004). Even though involving all nodes in the network would increase security service availability, this approach, especially in large-scale MANETs, entails more overhead and complexity in node configuration and management as a result of potentially handling a large number of nodes. In addition, allowing all nodes which usually have a physical vulnerability (i.e. a common MANET characteristic) to participate in the core security service would introduce security robustness concerns. For example, compromising few nodes may lead to compromise the whole security service and also this approach is prone to suffer from Sybil attacks (i.e. node has multiple forged identities) (Yi and Kravets, 2004).

• Cryptosystem and Credentials

Along with using digital certificates, most security proposals for MANETs shown in Table 4-1 make use of public-key cryptography, specifically the RSA cryptosystem (Rivest *et al.*, 1978), for managing credentials and securing communication and utilisation. Even though this particular cryptosystem is computationally expensive, especially in the case of limited MANETs, it still demonstrates substantial promise that compensates that particular drawback. This stems from the fact that public-key cryptography unlike the symmetric one shows effective and robust trust handling and provides the non-repudiation (using digitals signatures) and anonymity in communication (Boyd and Mathuria, 1998).

On the other hand, the type of cryptosystem plays a key role in defining the type of a credential that should be utilised, such as public-key cryptosystem typically involves public keys or digital certificates.

Therefore, the RSA cryptosystem (Rivest *et al.*, 1978) (i.e. an asymmetric cryptographic function) and the standard X.509 version 3 digital certificate (i.e. a security credential) (Chokhani *et al.*, 2003; Cooper *et al.*, 2008) are adopted for SSAM. This is due to the fact that these two security elements are standardised and broadly used in telecommunication security systems too (Stallings, 2010). In addition, due to the standard format of the X.509-v3 certificate, using this type of credential will leverage seamless interoperation between different systems. Also, the X.509-v3 digital certificate is part of the standard which promotes particular standard authentication protocols (i.e. X.509 One-Way-Pass (*1WP*), Two-Way-Pass (*2WP*) and Three-Way-Pass (*3WP*) protocols (ITU-T, 2008)).

The X.509-v3 digital certificate:

According to Chokhani *et al.* (2003) and Cooper *et al.* (2008), the standard format of the X.509-v3 certificate, as presented in Figure 4-8, includes the following main fields: version, serial number, signature algorithm name, authority name, validity period, user identity, public-key information (e.g. a public key value), extensions (e.g. user attributes and roles, policy, revocation list, etc.) and the certification authority signature. All types of certificates which are involved in the SSAM model are generated based on this particular standard format. On the other side, a "membership user certificate" (e.g. full or partial certificates) generated and updated by different proposed authority servers (i.e. *CAS*, *TASs* and *DASs*) in SSAM are presumed to comply with this standard.

In addition, this certificate is termed as the "membership user certificate" aiming to indicate a certificate which can be used for general purposes (e.g. admission, authorisation, authentication, etc.). This is because a membership credential often may contain a holder identify, holder attributed and roles and a membership granted authority. Alternatively, this certificate is also planned to be

a *short-lived* certificate available to a succeeded joined user node for utilising offered services in MANETs. This is intended to reduce the overhead of managing certificate revocation. In other words, there is no requirement to proactively refresh certificate revocation lists among nodes as a certain certificate will expire shortly and needs to be renewed by available authority servers. Finally, the other standard identity certificates which are pre-installed in network nodes (initialisation phase) are used as initial credentials only for facilitating SSAM authentication (i.e. employed in authentication protocols) and for securing SSAM utilisation.

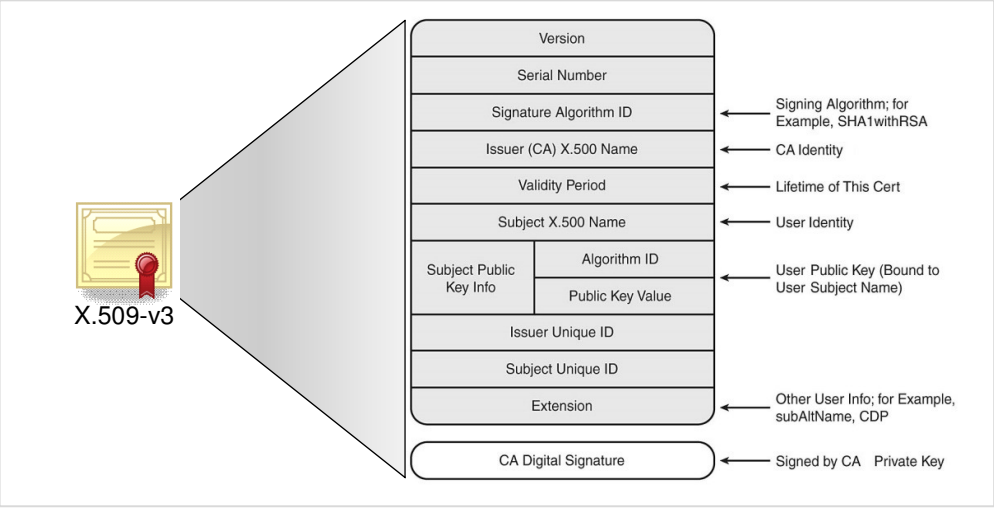


Figure 4-8: The X.509-v3 digital certificate format

The RAS and Threshold RSA cryptosystems

The RSA cryptosystem relies on the hardness of factoring large integers for leveraging its security robustness. These integers are defined as the product of two large prime numbers. However, extracting the original prime numbers from the result of multiplying these two numbers (using total factoring) is very difficult because of the long time being consumed even with using powerful machines. RSA computation begins with selecting two random secret prime numbers p and q to compute modulus integer $n = p \times q$. The defined public key exponent e along with n represents the public key, and the associated private key is estimated by d

$= e^{-1} \bmod ((p-1)\times(q-1))$. For encrypting a message m, the cipher text c is generated as follows: $c = m^e \pmod{n}$. To decrypt the cipher text, the message can be obtained from using the private key d as follows: $m = c^d \pmod{n} = m^{e.d} \pmod{n}$. The key length in RSA is very crucial as indicating the level of encryption strength. However, increasing the RSA key length would entail an overhead. RSA keys can usually be 1024 or 2048 bits long. For more details about using RSA cryptography, see Rivest *et al.* (1978), Menezes et al. (1996), Smart (2003) and Stallings (2010).

MRA and all types of authority servers take advantage of RSA cryptographic procedures for generating digital signature on ID and membership certificates being used. However, in the *TAS* architecture unlike the other architectures, a group of servers collaboratively create partial certificates based on "Threshold Cryptography" and these partial certificates can be combined to produce one full valid certificate accordingly. The "Threshold RSA algorithm" is chosen to be used by *TASs* because of two vital properties: (1) standard signature generation and (2) non-interaction between participants for generating signatures as stated in Gennaro *et al.*(2008). Hence, two phases of performing this algorithm (i.e. *TC*(k, m) scheme, $k << m$), based on Luo *et al.* (2004), Raghani *et al.* (2006) and Omar *et al.* (2007) are proposed for the *TAS* architecture, as presented in Table 4-4 (Note that the formulas below are taken from Luo *et al.* (2004)):

Phase1: Secret Share Generation (Initialisation)
(1) During SSAM initialisation, MRA initially creates the corresponding RSA keys for the threshold authority. Public key: (e, n) Private key: (d, n) where e is Public exponent, d is Private exponent, n is the modulus
(2) MRA determines a k-1 degree polynomial which is considered a fundamental part for sharing the threshold authority private key as: $f(X) = \sum_{i=0}^{k-1} a_i X^i$ where $a_0 = d.$ and $a_1, a_2, .., a_{k-1}$ are random values over a finite field.

(3) The value k is the threshold value of the *TAS* architecture. The MRA initialises m servers (*TASs*) via providing them with the private-key shares of the threshold authority. Each *TAS* obtains its private share P_{vi} from MRA as:

$$P_{vi} = f(v_i) \bmod n \quad \text{where} \quad v_i \text{ is a unique identifier of } TAS_i$$

The private share can be calculated from a coalition of k initialised *TAS* nodes $\{v_1, v_2, ..., v_k\}$ using Lagrange's Interpolation formula as:

$$P_{vi} = f(v_i) \bmod n = (\sum_{j=1}^{k} l_{vj}(v_i) P_{vj}) \bmod n \quad \text{where} \quad l_{vi}(x) = \prod_{j=1, j\neq i}^{k} \frac{(x - v_j)}{(v_i - v_j)}$$

Therefore, the private key of the threshold authority can be recovered by:

$$d = f(0) = (\sum_{i=1}^{k} l_{vi}(0) P_{vi}) \bmod n$$

Phase2: Threshold RSA Signature (Operation)

(1) In this phase, in order to create a valid signature of the threshold authority's private key, it requires a coalition of at least k *TASs* (k Threshold value). Therefore, when a user node demands to obtain its certificate, this node should send requests (called *cert*) to all accessible *TASs* in order to get signature on those *certs* accordingly. A *cert* may include general certificate information relevant to the certain calling node and coalition *TASs*. Typically, without sharing the private key, a certificate of the calling node *cert* can be signed using the private key threshold authority d as follows:

$$CERT = cert^d \bmod n$$

(2) Once *TASs* receive the *cert* value, they start to produce a corresponding partial certificate by signing it using their additive private shares. Additive share SK_{vj} of *TAS* v_j can be computed as:

$$SK_{vj} = P_{vj} l_{vj}(0) \bmod n$$

(3) *TAS* v_j constructs the partial certificate for the calling user node taking advantage of the additive share value determined above, based on the following method:

$$CERT_{vj} = cert^{SK_{vj}} \bmod n$$

(4) Afterward, *TAS* v_j sends the partial certificate back to the calling user node. When this user node receives all required partial certificates, it begins to combine them by product so as to generate an initial certificate *CERT´* as: $$CERT' = \prod_{j=1}^{k} CERT_{vj}$$
(5) In certain cases, *CERT´* may vary from actual certificate *CERT* as a result of an additional exponent representing *k*-bounded multiple of *n*. *CERT* can be restored from *CERT´* via employing the "k-bounded coalition offsetting algorithm" which is developed by Kong *et al.* (2001) and Luo *et al.* (2004).

Table 4-4: Threshold RSA Cryptosystem (phase 1: Secret Share Generation Initialisation, phase 2: Threshold RSA Signature Operation)

- **Security communication protocols**

As illustrated in Section 4.3, a particular security communication protocol in a security/trust infrastructure indicates a facility of securing connections between user and security server nodes, for the sake of allowing user nodes for utilising security services in the side of server nodes (providing security credentials). This protocol may involve exchanging of few or several specific messages (e.g. one-, two-, and three-way pass handshaking possibly for authentication purposes) for creating those connections. Furthermore, some protocols could rely on a specific propriety interconnection infrastructure (e.g. an overlay of servers, or multicast protocols) for accessing security services in servers, such as Yi and Kravets (2003) and Wu *et al.* (2007b) and Luo *et al.* (2005). In SSAM, a security communication protocol is proposed to be generic (i.e. a call follows by a reply similar to calling services on a server) and not reliant on any propriety interconnection infrastructure since this is believed to be common theme of calling for interoperation purposes.

However, since different server architectures in SSAM can be requested differently especially in hybrid and hierarchical architectures, a security communication protocol is defined here to represent all procedures that incorporate in establishing and maintaining connections. Three main procedures can be identified, an authentication protocol, a re-authentication scheme and a

strategy of calling. An authentication protocol is considered an important element as this protocol is typically used to confirm identities, and to create a secure individual link, between user and server nodes. A re-authentication scheme is exploited to alleviate potential connection failures by allowing a server invocation to be systematically renewed. A re-authentication scheme includes a waiting time interval and a number of re-try; however this issue will discussed in detail in Section 5.3.3.2. A strategy of calling is proposed specifically for hybrid and hierarchical architectures to denote to approach how to call different servers in those architectures.

The Authentication Protocol

Although a variety of authentication protocol types, as shown in Boyd and Mathuria (2003), could be involved in SSAM, The three standard X.509 ISO/IEC 9594-8 authentication protocols (ITU-T, 1989, 2008) are adopted for a number of reasons. Initially, these protocols are standardised and non-strict-layered protocols. Also, for coherent comparison, these three protocols belong to the same pool of authentication protocols (i.e. asymmetric key transport authentication protocols). However, other authentication protocols can be considered, if it is appropriate to a study. This is because this component of the authentication protocol is suggested to be a placeholder which can be altered freely (if needed). These protocols are the "One-Way-Pass" (*1WP*), "Two-Way-Pass" (*2WP*) and "Three-Way-Pass" (*3WP*) protocols. These protocols mainly are used to provide entity authentication (i.e. unilateral or mutual authentication) between two ends with optional key transport for a confidentiality purpose. Each protocol as its name implies is distinguished with different number of passes (message exchanges) involving timestamps and nonce values. These protocols are developed to satisfy a number of requirements: identity confirmation, message freshness, non-repudiation, and key transport secrecy. However, each protocol has different techniques to meet these requirements (e.g. using timestamps in *1WP* and *2WP* protocols for freshness unlike *3WP* which uses nonce values). In addition, the protocols are characterised with different robustness, for example,

the *2WP* and *3WP* protocols achieve mutual authentication whereas the *1WP* protocol only ensures unilateral authentication. Table 4-5 addresses all relevant issues of initialisation and operation for these three authentication protocols.

Notation:

$Pub_X(m)$: the cryptographic function of encrypting m using public key to data y.

$Sec_X(m)$: the cryptographic function of singing m using the X's private key.

r_A, r_B: nonce values are used for preventing from impersonation and replay.

ts_A, ts_B: timestamps are obtained from synchronised time clock.

$cert_X$: a certificate linking entity X with a public key is to facilitate encryption and signature verification.

A: a user node*, B:* a server node*, k:* an optional private transport key, *: refers to items that are optional.

Initialisation:

(I) Each entity (i.e. A or B) requires its pair of public key and its certificate typically issued by an authority (MRA) for signatures, authentication and encryption.

(II) (b) A should have encryption B public key by obtaining $Cert_B$ from the authority using an out-of-band method, for example.

The *1WP* protocol in operation:

One message (**I**) is exchanged in this protocol

data $D_A = (ts_A, r_A, B, data_1^*, Pub_B(k_1)^*)$

$A \rightarrow B : cert_A, D_A, Sec_A(D_A)$ **(I)**

- A generates ts_A and r_A used for checking an expiration time and message freshness. A secret key k_1 and $data_1$ can optionally be used in this message-I if required. Message-I, which contains $cert_A$ and D_A along with its signature, is sent to B.
- B validates $cert_A$ (checking the signature, expiry date, etc.) and verifies A's signature on D_A using A's public key. Then, the items within D_A will be checked (e.g. the identifier in message, intended recipient B, the validity of timestamp and nonce to avoid any replay. The secret key (if included) will be extracted for furthered usage. If all checks are approved, B confirms that A has successful authentication (unilateral authentication).

The *2WP* protocol in operation:

Two messages (**I** & **II**) are exchanged in this protocol,

Let $D_B = (ts_B, r_B, A, r_A, data_2^*, Pub_A(k_2)^*)$

$A \leftarrow B : cert_B, D_B, Sec_B(D_B)$ **(II)**

- Following the same process of sending and checking message-I shown in the *1WP*

protocol, B continues to do the same of creating ts_B, r_B, and the D_B's signature and sending message-II to A. • Similarly, A performs the same checks carried out by B. If all those checks are successful, A grants B successful authentication (i.e. in this case, mutual authentication is satisfied). In case, k_2 is provided, both A and B have mutual secrets k_1 and k_2 for securing interaction.
The *3WP* protocol in operation: Three messages (**I** & **II & III**) are exchanged in this protocol. $A \rightarrow B$: (r_B,B), $Sec_A(r_B,B)$ **(III)** In this protocol, there are few differences from the other two protocols described above: • Timestamps ts_A and ts_B should be ignored (set to zero). • On receiving message-II, A detects the r_A value matches that value included in message-I. Then, message-III, which contains r_B, B and the A's signature, is generated and sent to B. • B validates the signature and then checks identifier B and r_B received is similar to that in message-II.

Table 4-5: The three standard X.509 ISO/IEC 9594-8 authentication protocols

Alternatively, the SSL/TLS protocol (Dierks, 2008) is considered a standard and widely used protocol developed for the service level (i.e. transport and application layers). This is by allowing two parties to authenticate each other and to set up a secret key which is used to protect the communication session. However, it is argued that this protocol may not be feasible to apply in the SSAM model because of a number of reasons. The SSL/TLS protocol is strictly TCP-related and also several handshakes (more than three) required in this protocol would entail more communication overheads which become very problematic in limited MANETs. Therefore, the three standard X.509 protocols still appear to be the best candidates which can be involved in the security communication protocols of SSAM for this study.

Strategy of Calling & Trust Pattern

There are two types of strategies that are suggested for requesting heterogeneous servers in the hybrid and hierarchical security architectures: "***All at Once***" (*AAO)* and "***In priority sequence***" (*IPS).* The *AAO* strategy means that node begins to simultaneously create multiple connections for calling all types of server architectures that are pre-deployed in MANETs. For example, in the *CAS_TAS* architecture, all connections for each different individual server architecture will be created concurrently as follows: 1 connection for *CAS* and *n* connections for available *TASs.* However, calling *DAS* in the *CAS_TAS_DAS* security architecture is proposed not to be concurrent with calling other servers (*CAS* and *TAS*).This is because, according to SSAM specifications, this particular calling should be initiated as a result of failure on calling the *TAS* architecture.

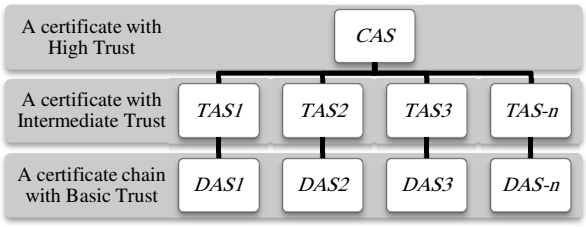


Figure 4-9: A trust policy for utilising different servers

Alternatively, the *IPS* strategy relies on a priority to request different server architectures. A certain priority sequence: *(1)CAS* ➔ *(2)TAS* ➔ *(3)DAS*, follows the proposed trust level model for SSAM, as shown in Figure 4-9. The high priority is appointed to the *CAS* architecture due to the highest trust and confident architecture among user nodes. The second priority is given to the *TAS* architecture as certificates obtained from this architecture are characterised with intermediate trust. The *DAS* architecture has the third priority in calling because of the *TAS* architecture reliance and having basic trusted credentials. Finally, this priority sequence is applicable for any predefine two- or three-server architectures (i.e. *CAS_TAS*, *TAS_DAS*, and *CAS_TAS_DAS*). On the other side, using this proposed trust policy along with different servers enables service provider to generate their own flexible access control policies, similar to Martucci *et al.*

(2004). For example, a certain service provider offers its service only for user nodes which possess their certificates from the *CAS* architecture.

However, it is important to highlight that the strategy of calling a group of servers specifically in the homogenous multi-server architectures (e.g. the *D\CAS* and *TAS* architectures) is intended to be only all at once. The reason is to avoid complexities in prioritising calling these multiple servers and making decision about a desired number of servers that should be requested especially in the *TAS* architecture.

• MANETs Settings

Unlike other conventional networks, a MANET is distinguished with different characteristics, specific constraints (as detailed in Section 2.2.2 and 4.4.3), and various settings, for example, mobility, churning, MAC and routing protocols, a channel type, space, node density, and a traffic model being involved. As shown in Figure 4-6, these features can have a significant impact on the SSAM model when applied. This stems from the fact that all these settings would contribute either positively or negatively to the communication medium of MANETs (establishing connected routing infrastructure). Most MANET settings, which are defined for this study, are discussed exhaustively in Section 5.3.4. On the other side, this part of the SSAM is considered very crucial as it defines the majority of issues which are relevant to the part of MANET context in the proposed approach for this study.

Ultimately, the SSAM model with its components is developed to provide an open solution which meets various security objectives for the operational service level of MANETs. As distributed security architecture is quite demanding for enhancing performance and robustness, especially in MANETs, the SSAM model presents a pool of distributed server architectures (e.g. *CAS_TAS*, *TAS_DAS* and *CAS_TAS_DAS*). On the other hand, involving standard authentication protocols (i.e. *1WP*, *2WP*, and *3WP* protocols) and a standard credentials (i.e. X509-v3 certificate) in SSAM would facilitate interoperation with other different systems

using those security standards as this feature becomes a key incentive towards widespread employment (Corson et al., 1999). SSAM flexibility can be realised from offering different security alternatives which can meet different requirements of several applications for MANETs. For example, an authentication protocol which meets node and MANET capabilities can be easily selected, a proper trade-off between different server architectures for a rescue mission, in terms of availability and security strength, can be simply made in SSAM.

However, it is important to mention that the SSAM operation consists of two main sides, usage and management sides. All the above description of the SSAM model represents the usage side only. However, the management side deals with all issues that are related to server management (e.g. choosing a number of servers, defining a threshold value for the *TAS* architecture, updating secret shares and checking server availability). Considering the management side in SSAM is out of this study's scope. The next section presents SSAM activities model and relevant workflow diagrams.

4.7 SSAM Activity Model

This section describes the basic activity model which represents the lifecycle of a new node joining MANETs when a certain security architecture is applied in SSAM, as shown in Figure 4-10. The activities in this model represent different stages, such as node search-for-connection, node authenticating and node churning. As this study's focus is the usage of security architectures for authentication in SSAM, some authentication-related activities in this model will be detailed and mapped with appropriate diagrams (i.e. sequence diagrams and flowcharts) in the next section.

Initially, **(1)** a user node joins the playground, based on pre-determined time arrival model being applied (e.g. Poisson- arrival rate λ). Then, **(2)** this node starts to roam according to a certain mobility pattern, aiming to find in its range at least one node in order to establish a connection in MANETs (i.e. access-to-network).

When the node becomes a part of the network (i.e. to activate its routing role), **(3)** it initiates the necessary server requests according to the predefined server architecture in SSAM by opening a connection to each server accordingly. In each server connection, **(4)** the node starts handshaking with accessible security servers using a certain security communication protocol (i.e. *1WP*, *2WP* or *3WP* authentication protocol) in order to create an authenticated channel and to obtain its own membership certificate (i.e. this certificate can be full or partial). However, in SSAM, each server architecture has different requirements in terms of the requisite number of successful server connections for attaining full successful authentication. This matter will be discussed in detail in the next section. If the calling node satisfies the requirements of the predefined server architecture, **(5)** this node is considered successfully authenticated, and then **(6)** it triggers traffic from available traffic generator nodes (i.e. used to keep MANETs always in operation) in the playground. Otherwise, this node is considered unsuccessful in authenticating and it fails to receive its certificates. Hence, **(7)** the node terminates all current active server connections.

At this point in the activity model, the node is offered to stay or leave based on its states of authentication and Churn flag. The Churn flag is used to indicate whether the node must leave or stay. If the flag is set off, regardless of the node's state of authentication, **(8)** a node is allowed to stay and take part of operating MANETs. If the flag is set on, **(9)** the node with unsuccessful authentication must leave the playground, whereas the node with successful authentication has two distinct circumstances (i.e. staying with/out node lifetime). One of these two circumstances are proposed to present node churning for some nodes with successful authentication as this issue will be elaborated in Section 5.3.4.6. Relying on a random selection (e.g. 50% population of successfully authenticated nodes are marked to stay), the node is either allowed to **(8)** stay, similar to the case of without churning, or **(10)** forced to leave after finishing its predefined lifetime.

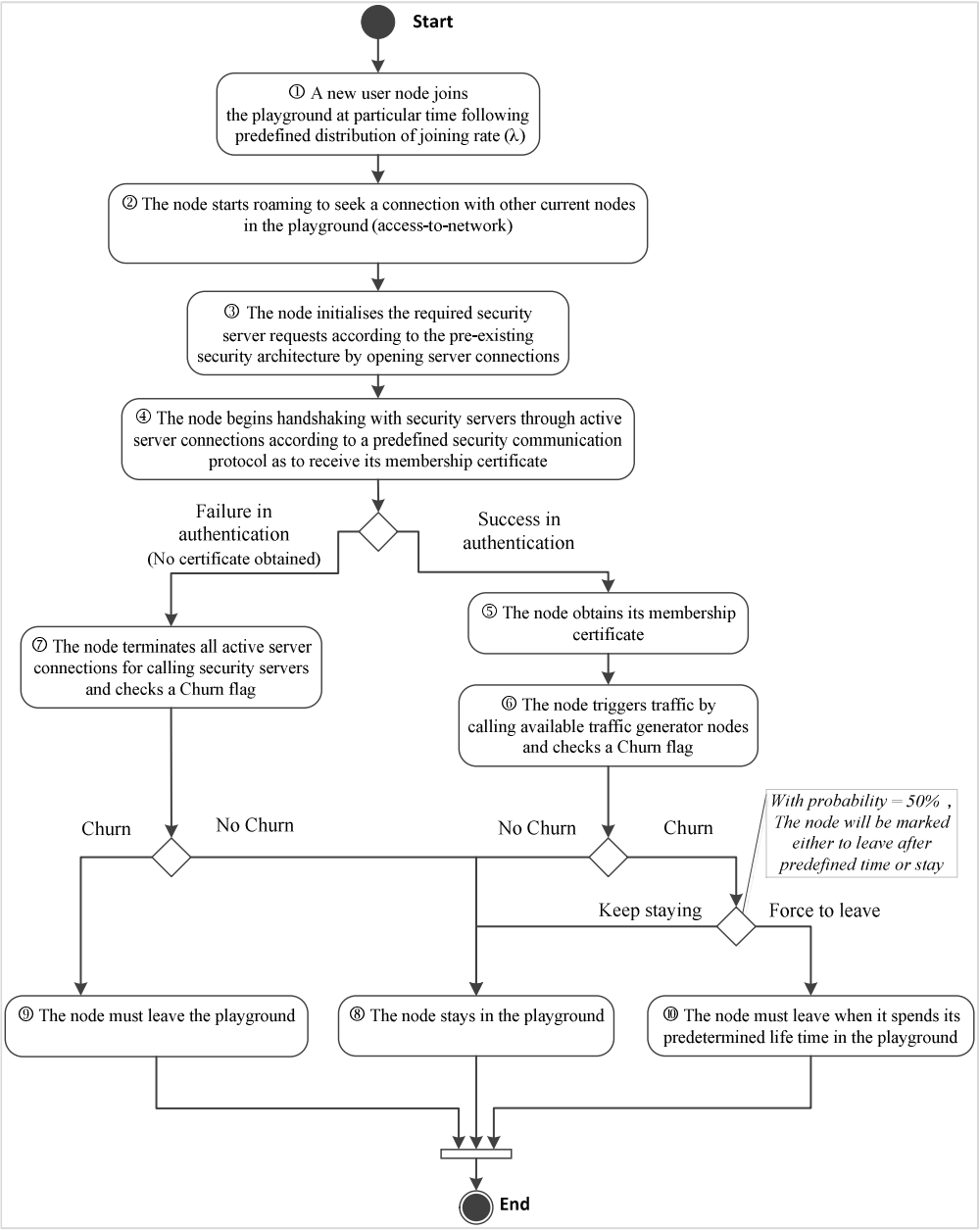


Figure 4-10: The Node Activity Model in SSAM

As mentioned earlier, the activities described above represent three stages, "search-to-access" 1-2, "authentication process" 3-7, and "Churning" 8-10. The core components of the SSAM model appear to be realised implicitly in the "authentication process" stage, as shown in Figure 4-10**.** Hence**,** the next section illustrates in detail these activities which typically represent the processing and

communication occurring between node and server nodes with different SSAM case scenarios (i.e. different authentication protocols, different server architectures, different calling strategies and traffic generation).

4.7.1 Node and Server Communicative Models

This section discusses the workflows of key security operations in SSAM. Two different level models (i.e. the communication and process level models) are proposed to describe these workflows. The communication level model specifically is intended to present the flow of establishing a single connection between user and server nodes through using a particular authentication protocol (i.e. performing a sequence of particular communications), irrespective of the server type being involved. The process level model for both user and server nodes represents the internal processes which handle a security service (i.e. providing and obtaining a membership certificate) in different server architectures within SSAM.

4.7.1.1 Communication Level Model

For the communication level model, the *1WP*, *2WP* and *3WP* authentication protocols which are based on the X.509 standard - ISO/IEC 9594-8 (ITU-T, 1989, 2008) are adopted in SSAM to facilitate authentication between user and server nodes. The sequence diagrams for each of authentication protocols are shown in Figure 4-11, 4-12 and 4-13 following operation section in Table 4-4. The goal is to build a secure channel enabling a user node to obtain its membership certificate from a single server node in a secure manner (integrity, authentication, etc.). Each authentication protocol consists of different control and data messages being exchanged. The required control messages are expected to be the main body of the proposed authentication protocols. This is because control messages aim to prepare a secure connection whereas data messages are used to

carry membership certificates back to user nodes from specific servers according to the server architecture in use. In the *1WP* authentication protocol, within the state of the "Protocol Msg State 1", a user node begins to generate a request for a certain server by sending the first control message (CMsg1). This control message mainly includes $timestamp_{user}$, $nonce_{user}$ and $Id\ certificate_{user}$ fields along with the CMsg1 signature created by user node private keys for authentication and integrity purposes. This message, in this state in the protocol, is used to achieve a unilateral authentication (i.e. a user to be authenticated by a server). When CMsg1is received and verified by a certain server, the server generates and sends a membership certificate to the calling user node within a data message (DMsg). A state of "Data Msg" in the protocol is used to denote to a particular state where DMsg is generated or validated by server and user ends, accordingly as shown in Figure 4-11.

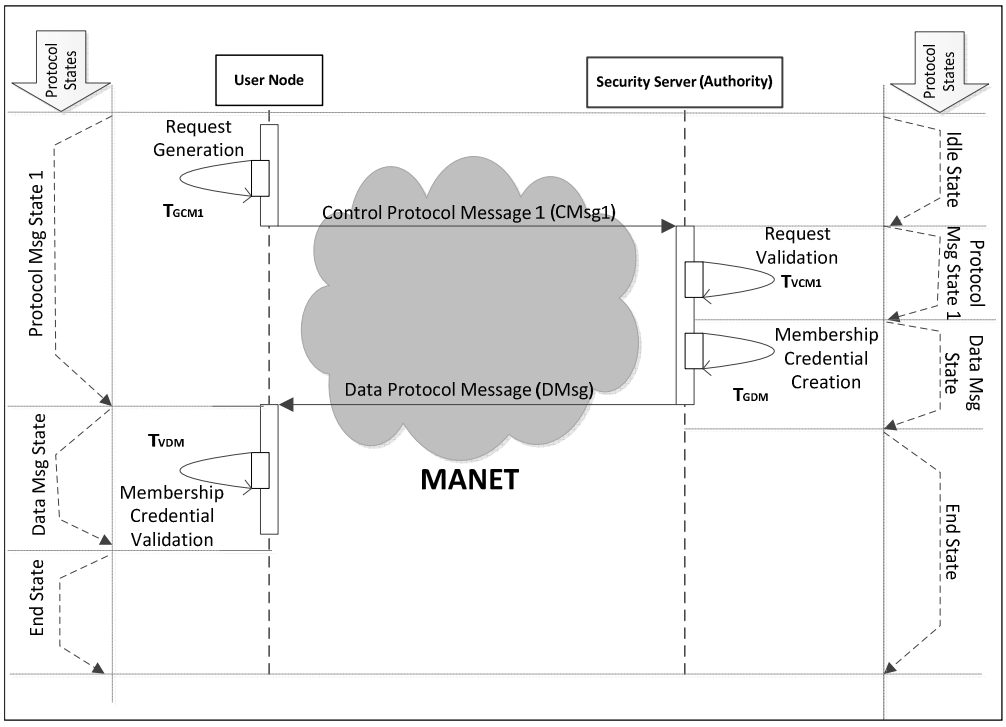


Figure 4-11: The sequence diagram of the One Way-Pass authentication protocol (*1WP*)

As shown in Figure 4-12, the *2WP* authentication protocol varies from *1WP* with one extra control message (CMsg2) and an additional protocol state, "Protocol Msg State 2" in the user node side, for handling this particular control message. CMsg2 is exploited in this protocol to complete the mutual authentication between user and user nodes. Also, this control message comprises $timestamp_{server}$, $nonce_{server}$, $nonce_{user}$ and $Id\ certificate_{server}$ with the signed CMsg2 field.

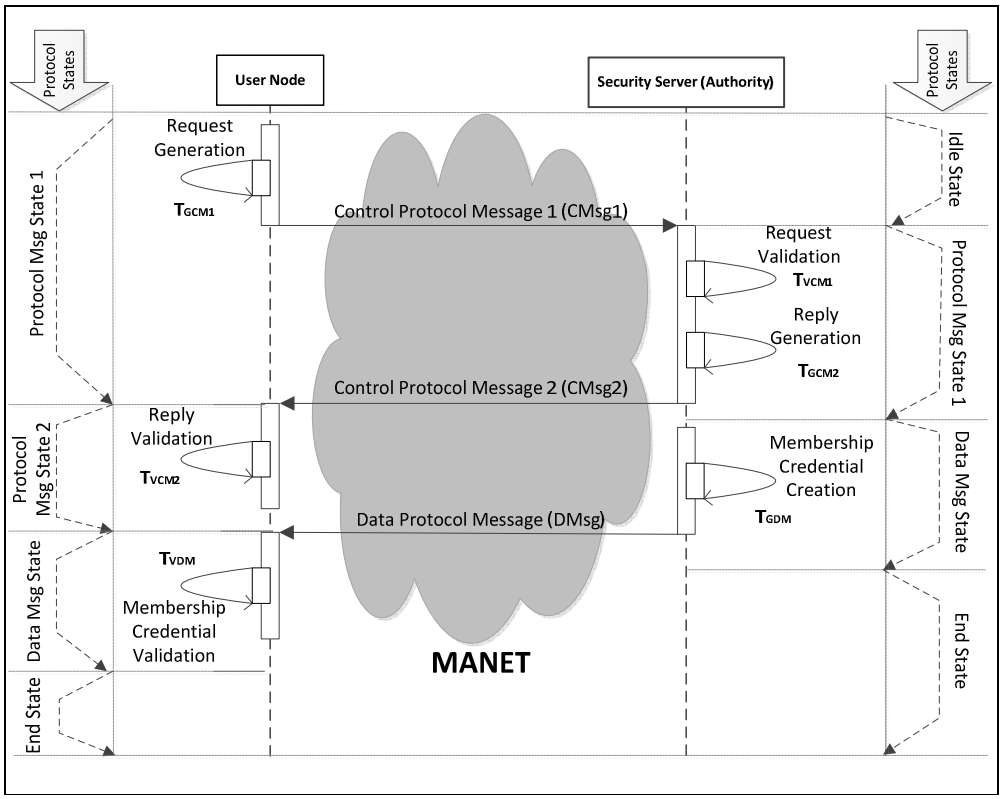


Figure 4-12: The sequence diagram of Two Way-Pass authentication Protocol (*2WP*)

The *3WP* authentication protocol, as shown Figure 4-13, makes use of three different control messages unlike the other authentication protocols discussed above. Furthermore, the third new message (CMsg3) contains $nonce_{server}$, $server_{address}$ and signature of CMsg3 and it is considered a confirmation reply in this protocol for making this authentication more robust against certain threats, such as Man-in-the-Middle (Burrows *et al.*, 1990; I'Anson and Mitchell, 1990;

Boyd and Mathuria, 2003). Similar to *2WP*, the state of "Protocol Msg 2" in *3WP* is added to the server node side in order to represent the processing of CMsg3.

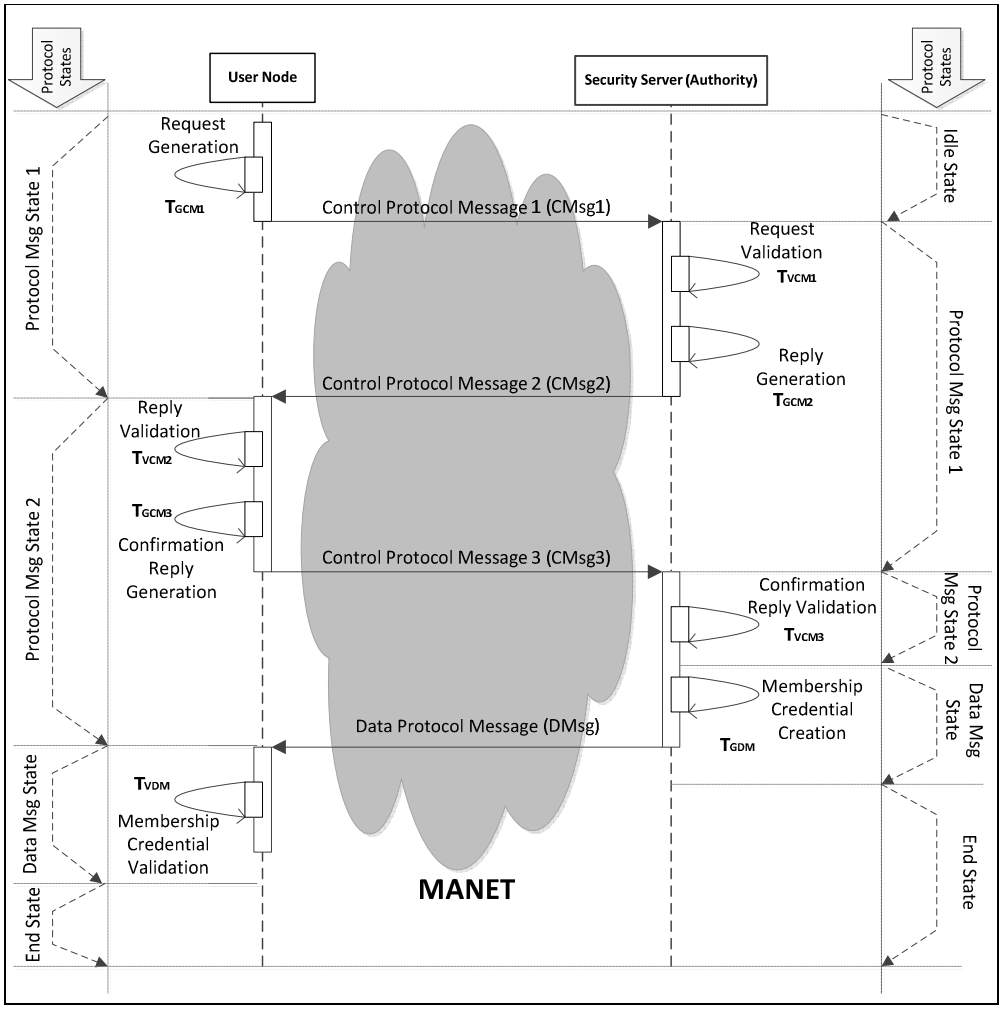


Figure 4-13: The sequence diagram of Three Way-Pass authentication Protocol (*3WP*)

Eventually, as shown in the sequences diagrams, Figure 4-11, 4-12 and 4-13, using the suggested states in authentication protocols enables both sides to describe a current message that needs to be exchanged or processed within the protocol. However, it is important to point out that an entire sequence of control messages in any particular authentication protocol is intended to be seen as a join request (i.e. called a certificate request), while a received data message represents

an admission response for this request. This particular vision will be used to simplify the process model which is presented in different flowcharts below.

4.7.1.2 Process Level Model

For the process model, apart from the authentication protocols, eight different server architectures can be recognised, for which the process model needs to map out their internal processes in both server and node sides for better understanding of the SSAM logic and behaviour. Also, this process model can arguably facilitate the SSAM implementation and deployment. As all different server architectures in SSAM rely on certain primitive servers (*CAS*, *TAS* and *DAS*), the process models of these primitive server architectures should be presented initially. Most process models shown in this section, involve user and server node instances. Server instances differ according to a certain server architecture being involved.

In **the *CAS* architecture,** as shown in Figure 4-14, a new user instance starts initialisation of necessary elements (e.g. current and maximum attempt variables (CNA and MNA), a session ID, waiting time (Timeout) etc.) for the process of calling *CAS*. A join request with a new session ID is initiated according to a specific authentication protocol being pre-configured. At the same time, a timeout timer is activated for re-authentication purpose and the variable of a current number of attempts (CNA) is increased. If this timer expires without completing the control messages' handshaking of the authentication protocol to receive an admission answer "*DataMsg*", another join request with a new session ID will be generated unless the current number of attempts (CNA) reaches the max number of attempts (MNA). In this case, the user node instance ends up with *unsuccessful authentication*. On the other hand, if the admission response "*DataMsg*" is received successfully, the session ID of "*DataMsg*" will be checked to verify whether this response is fresh or expired. If the answer is "expired", the message will be dropped. Otherwise, the current timeout timer will be cancelled and the

received "*DataMsg*" will be validated by checking the extracted membership certificate. Then, the user node instance is marked with *successful authentication*.

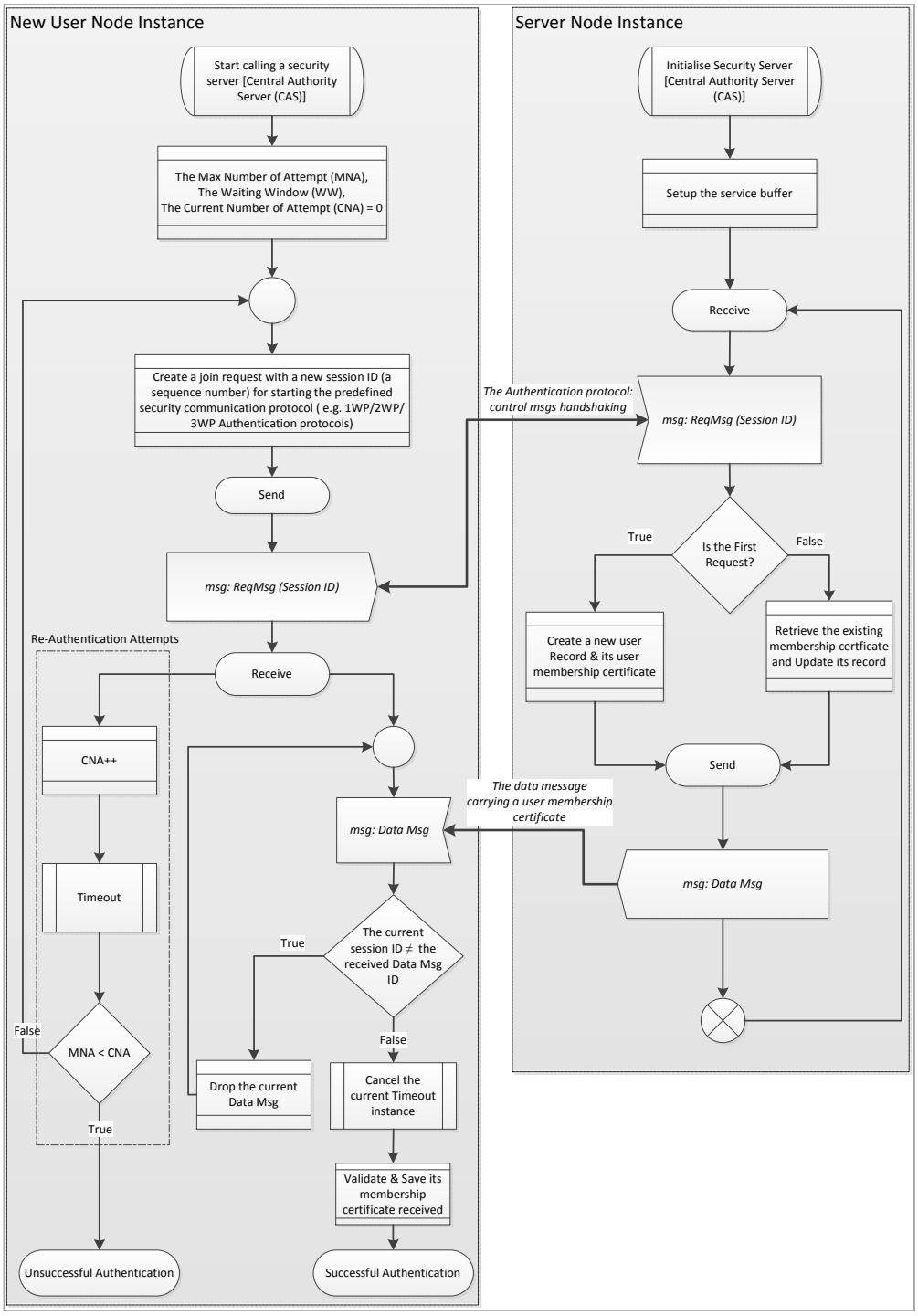


Figure 4-14: The flowchart of user node and *CAS* process models

In the server side, the server node instance starts to set up the service buffer and then this instance waits for new join requests. In case of receiving a new request, a control message handshaking will be handled for the particular request. Afterwards, this processed request made by a calling user node will be checked to verify whether it is the first or not. If this request is the first, a record for the specific user node will be created and user own membership certificate will be issued, accordingly. Otherwise, the membership certificate will be retrieved and the user node own log record will be updated. Finally, "*DataMsg*", including the membership certificate, will be generated and will be sent back to the user node. For avoiding repetition, almost all process models in the server side are quite similar regardless of the server architecture, as shown in all flowcharts presented in this section. However, the only noticeable difference is the type of a membership certificate which is generated based on the server type being used, usually a full certificate, a partial certificate or a certificate chain.

The *TAS* architecture relies on (n) number of security servers (i.e. *TASs*) to generate a valid certificate from received partial certificates using the *TC* (n,k) scheme. Therefore, a user node instance simultaneously creates a corresponding number of similar server-connection instances (i.e. called "calling *TAS*[i] instance" $\forall$ i= 1,2,..,n, $n>1$), as shown in Figure 4-15. At the same time, all needed variables are initialised, such as the number of received partial certificates (NPC), the number of calling instances in progress (NIP), and (k) necessary number of partial certificates for a successful certificate generation. Along with relevant variables, the procedures of handshaking for the authentication protocol, setting re-authentication timer and handling "*DataMsg*" in the *TAS* architecture are similar to those procedures in the *CAS* architecture. However, a few variations can be acknowledged. Firstly, in the case of expiring timeout timer without allowing for more attempts, the active calling instance will check whether (NIP - 1) is less than k as this instance fails to obtain its own partial certificate for the corresponding *TAS*.

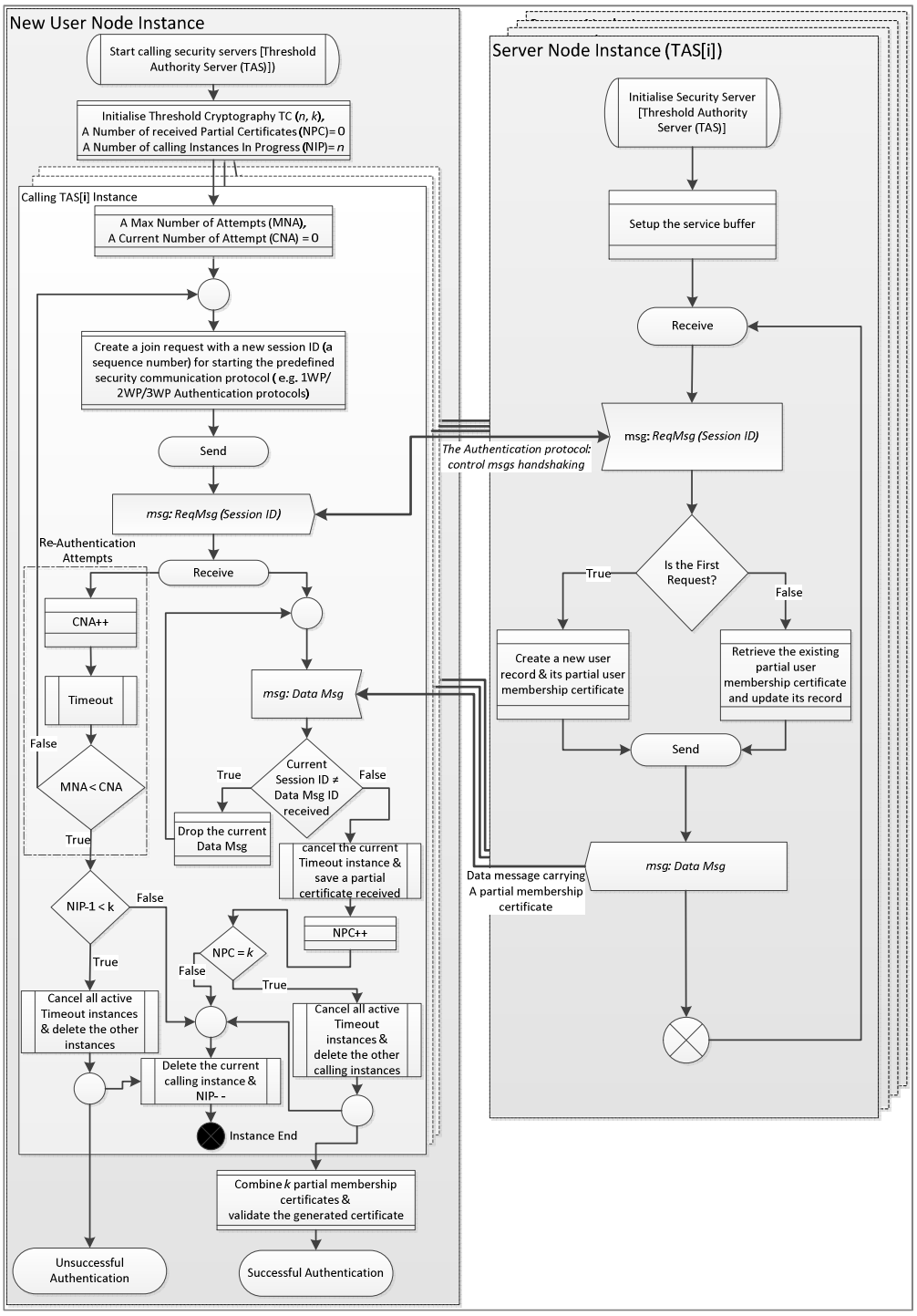


Figure 4-15 : The flowchart of user node and *TASs* process models

If the condition (NIP-1< k) is not satisfied, NIP will be updated (i.e. NIP--) and the current calling instance will be ended up. Otherwise, if this condition (NIP-1< k) is satisfied, this leads to cancel all active timeout timers in the other calling instances and also to terminate all existing calling instances including the current calling instance. Then, this user node instance becomes in the state of *unsuccessful authentication*. Secondly, after successfully receiving "*DataMsg*" and cancelling timeout timer for this calling instance, the NPC variable has an increment. Then, this variable will be checked. If it is less than *k*, this means that the parent user node instance still expects more partial certificates to be received from other active calling instances. Therefore, NIP will be decreased by one as this calling stance will be closed. Otherwise, if NPC is equal to k, this causes the cancellation of all active timeout timers in other existing calling instances and also the termination of all remaining calling instances including the current calling instance. Finally, the user node instance combines all partial certificates delivered to produce a valid membership certificate in order to be marked with *successful authentication*.

In the ***D\CAS* architecture**, a number of security-server copies (i.e. *CAS*s) are involved in providing a security service to user nodes in MANETs similar to the *CAS* architecture. After initialising NIP with the number (*n*) of available security servers, a user node instance begins to form the same number of calling instances accordingly (i.e. "calling *D\CAS*[*i*] Instance", ∀ *i*= 1,2,..,*n*) for the sake of broadcasting join requests. Like the other server architectures discussed above, each calling stance includes similar process flows for re-authentication, authentication protocol and "*DataMsg*" delivery, as shown in Figure 4-16. However, the user node instance is considered *unsuccessfully authenticated*, if all calling instances fail to receive "*DataMsg*" from their associated *D\CASs*. This can be realised by checking the number of calling that are still in progress (i.e. testing the value of NIP). When "*DataMsg*" is delivered successfully and then a user membership certificate extracted from this message is verified, this enables the user node instance to have *successful authentication*.

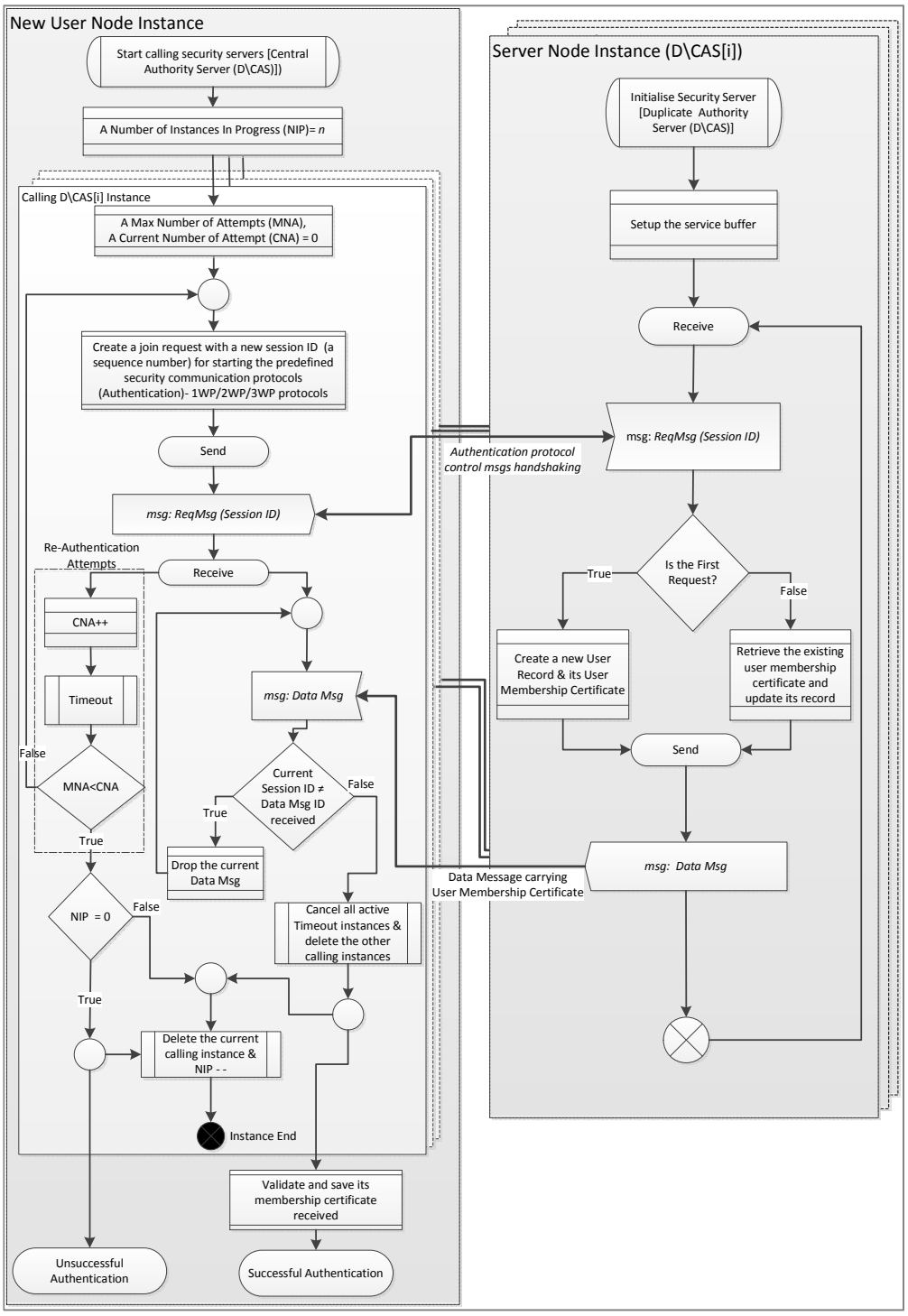


Figure 4-16 : The flowchart of user node and *D\CASs* process models

It is important to point out that it is proposed that the *DAS* architecture in SSAM always exists along with the *TAS* architecture in the compound or multi-level server hierarchical architectures. The reason is that when the *TAS* architecture fails to provide a certain user node with its membership certificate, the *DAS* architecture will take over (i.e. *TASs* become *DASs*). Clearly, the separate *DAS* architecture appears to be quite similar to the *D\CAS* architecture in terms of single-server dependency since a user node depends only on one admission response (i.e. containing membership certificate) to announce this node successfully authenticated. However, in the integrated *DAS* architecture, after *TAS* authentication failure, one of reachable *TAS*s included in a defined list will be randomly selected to be a *DAS*, as shown in Figure 4-17. This is in order to initiate a request ("*DAS Request*") for calling this particular *DAS*. Since the user node and the particular *DAS* are already associated from previous interaction (between the user and *TAS* nodes), it is decided to skip the part of handshaking protocols from this process model in order to avoid more communication overhead. Therefore, "*DAS Request*" will be sent directly and then a timeout timer for the request re-try will be managed similar to the other architectures of handling re-authentication timers. If a message of "*DAS Data Msg*" is received successfully, the re-try timer will be cancelled and a certificate chain (two user membership and server delegation certificates) extracted from "*DAS Data Msg*" will be validated. Then, *successful authentication* can be realised for the user node instance. If the list of accessible *TASs* at the beginning is empty or the (*MNA*<*CNA*) condition for timer is met, as presented in Figure 4-17; the user node instance will be *unsuccessfully authenticated*.

All process models of the server architectures discussed previously are established only on one server type (i.e. *CAS*, *TAS*, *D\CAS* and *DAS*). However, other multi-level server hierarchical architectures in this study are developed to involve different server types in one compound server architecture, such as *TAS-DAS*, *CAS-TAS*, and *CAS-TAS-DAS*. Furthermore, the two different calling strategies (i.e. In Priority Sequence (*IPS*) and All At Once (*AAO*)) are proposed to represent potential approaches of utilising the distinct fundamental servers within

these specific compound architectures. In *IPS*, a user node sequentially calls principal server architectures according to a particular priority whereas in *AAO* a user node broadcasts invocations simultaneously for all types of servers in the architectures.

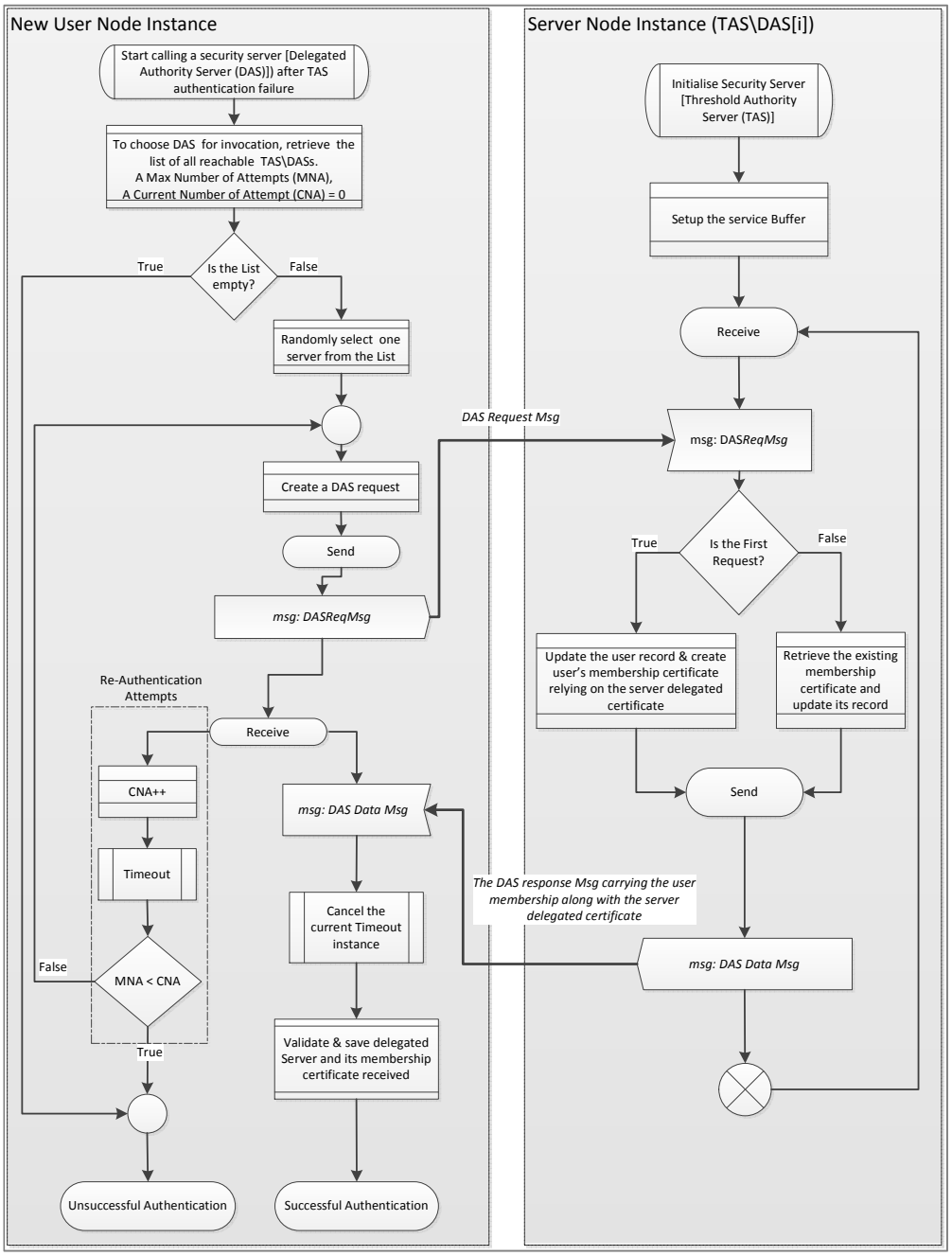


Figure 4-17 : The flowchart of user node and *DAS* process models

Therefore, all user node process models of the two- and three-level server hierarchical architectures are presented collectively in Figure 4-18. However, their server process models are intended to be ignored as these models are similar to the other architectures discussed before. The process flows in these compound architectures vary based on the different server types being involved and the calling strategies being adopted. Thus, Figure 4-18 is divided into two main sections relevant to the type of calling strategies: Part (A) is for *IPS* and Part (B) is for *AAO*. Also, in this figure, the different colour dashed lines indicate the flow of processes for the two-level server hierarchical architectures, such as *CAS_TAS* and *TAS_DAS*, and the solid line indicates the flow of processes for the three-level server hierarchical architectures, such as *CAS_TAS_DAS*. It is important to mention that each rectangle notation in these process models, in Figure 4-18, typically embodies the complete user process model for one of those basic server architectures (e.g. *CAS*, *TAS* or *DAS* architectures), as described in Figure 4-14, 4-15, and 4-16. Accordingly, the output of this notation denotes to the current state of authentication (i.e. whether successful or unsuccessful) from calling a particular architecture. In the *TAS_DAS* architecture, a new user node instance starts to initialise and activate the calling process of the *TAS* architecture initially. Then, if this process fails to get successful authentication, the calling process of the *DAS* architecture will be initiated accordingly, as shown in Figure 4-18. In the *CAS_TAS* architecture, two calling cases can be recognised, *IPS* and *AAO*. A new user node instance with *IPS* begins to call the *CAS* architecture. When this call is not fulfilled, the calling procedure of the *TAS* architecture will be triggered. However, in the case of using *AAO*, both *CAS* and *TAS* architectures are invoked simultaneously and then the new user node instance waits until receiving their replies. If calling *CAS* is completed successfully, the *TASs'* calling, which is still in progress of handshaking, will be cancelled. In both cases of the *IPS* and *AAO* calling strategy, the process models of the *CAS_TAS_DAS* architecture, like the *CAS_TAS* architecture, consist of requesting the *CAS* and *TAS* architectures either all at the same time (i.e. the *AAO* calling strategy) or one after another (i.e. the *IPS* calling strategy). On the other hand, the *DAS* architecture is proposed to be utilised after a complete failure in calling both *CAS* and *TAS* architectures.

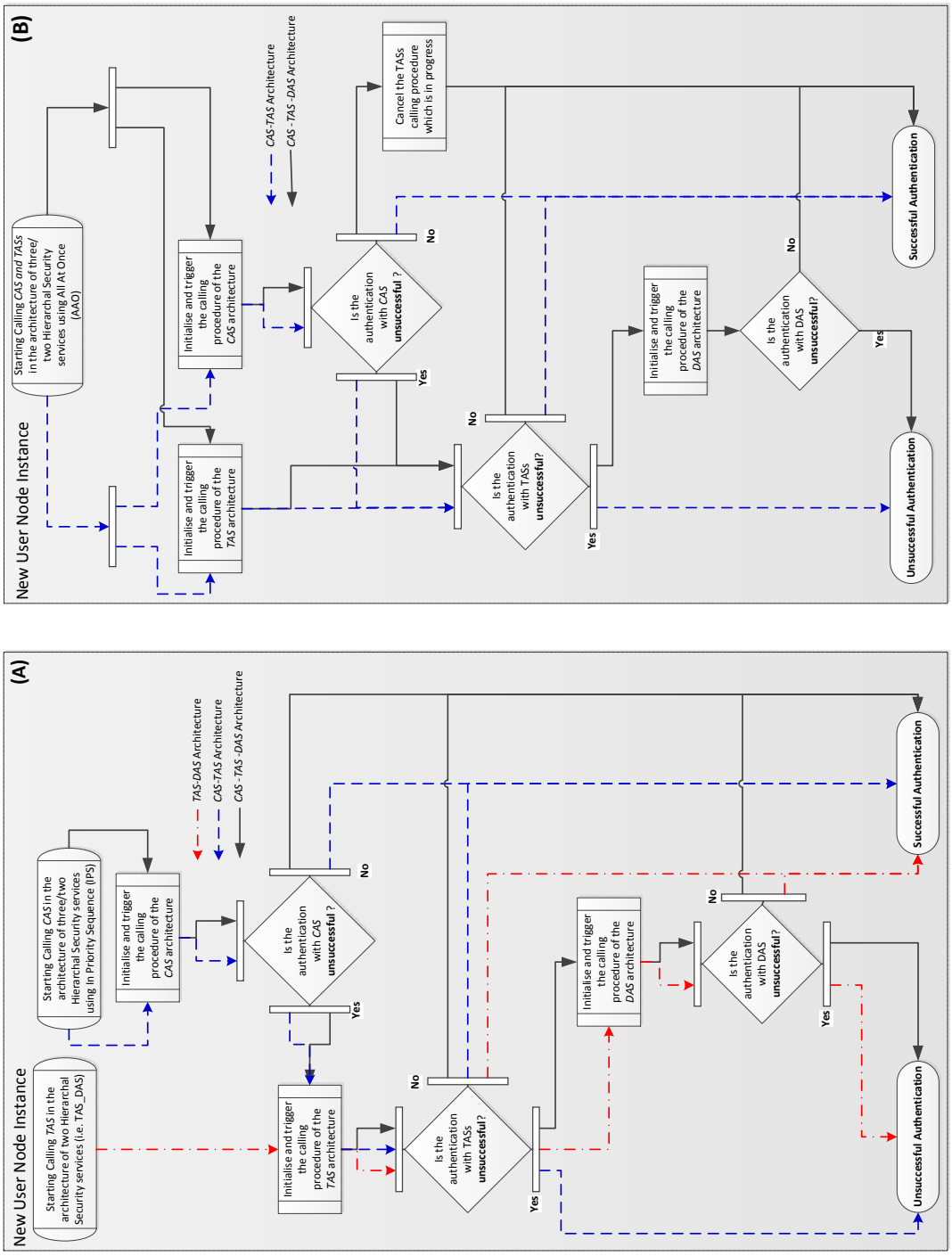


Figure 4-18: The user node process flowchart for the two- or three- server hierarchical architectures: (A) In Priority Sequence - *IPS* and (B) All At Once - *AAO* calling strategies

According to the node activity models, shown in Figure 4-10, if a user node succeeds in authentication, this node is proposed to instigate traffic in the MANETs. This is performed by calling (i.e. sending "*Traffic Request Msg*") a particular traffic generator node (i.e. *Traffic Source Server - TSS*). When "*Traffic Request Msg*" is received by the traffic generator node, this node becomes a traffic sources generating traffic and the corresponding calling user node works as a traffic sink, as presented in Figure 4-19. A timeout timer is used for allowing a user node to produce several attempts of traffic requests (if required) as to ensure the "*Traffic Request Msg*" delivery since MANETs usually suffer from broken links and network partitioning. In this process model, the traffic generation design is proposed to be generic in order to fit most of popular traffic models (i.e. using the rectangle notation of processing traffic as placeholder). It is worth pointing out that there are different ways of processing and sending traffic in the traffic generator nodes, which normally relies on a certain traffic model being adopted (e.g. ON/OFF Pareto, ON/OFF Exponential, Constant Bit Rate, etc. (Giannoulis *et al.*, 2009; Pal *et al.*, 2011)). The amount of traffic created by a traffic generator node is supposed to be delivered to the corresponding requesting user node. When the user node is marked with "Churn" (i.e. leaving), the calling instance of this node will be terminated. Also, the new node instance stops once it fails to reach any available traffic source servers in the playground.

Eventually, the process of traffic generation shows no direct involvement in the security services of SSAM. However, this part is still very crucial for a number of reasons. Initially, more traffic in MANETs, unlike other networks, would entail much more disruption for utilising security servers in SSAM since the majority of MANETs' nodes play a dual role (i.e. both router and host). On the other hand, this process may be perceived differently as a user node is calling a service provider to get served although this particular interaction is out of this study scope. This interaction can also be a vital point of furthered research for supporting SSAM in terms of security and performance.

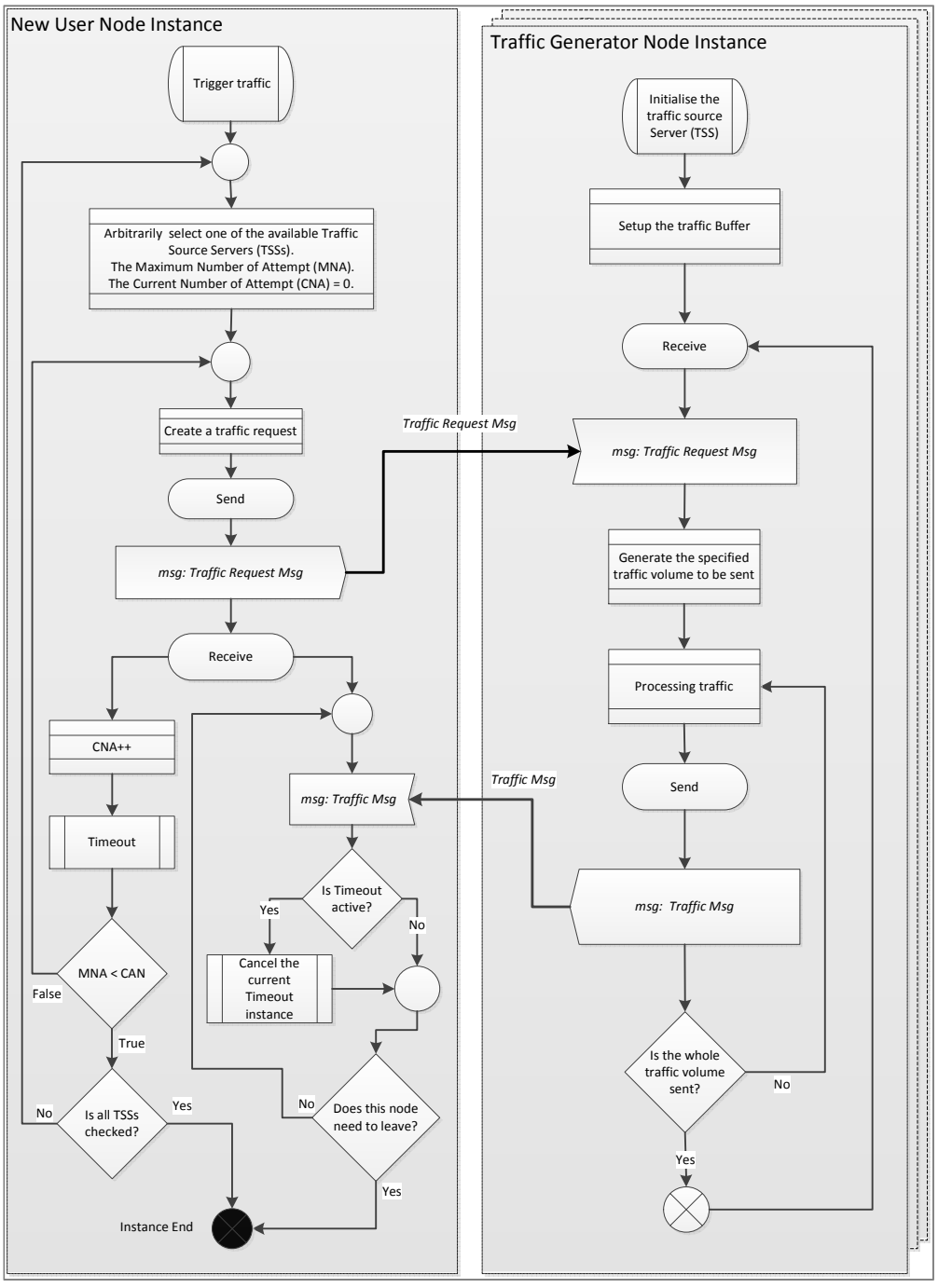


Figure 4-19: The flowchart of user node and traffic generator node process models

4.8 Conclusion

This chapter is split into two main parts to present (A) the proposed multi-dimensional security framework along with a systematic approach for evaluating the security/trust infrastructure in MANETs and (B) the proposed SSAM model. The first part highlights the design of two different operational levels for MANET security operation, the building blocks of MANET security/trust infrastructure and the key dimensions affecting the design of the security/trust infrastructure. The second part presents the proposal of SSAM with its relevant elements and the SSAM activities, process and communication models.

Part A: The two-security-level-based architecture (i.e. Network" and "Service" levels) derived from the MANET operation view is suggested to manage crucial MANET security issues effectively. This architecture is intended to be a key motivation to security designers to develop their security solutions smoothly, with a balanced view in MANETs. To perceive the design of the security/trust infrastructure, a security framework is introduced with five security building blocks: (1) a security role, (2) a security-server architecture, (3) a security communication protocol, (4) a security mechanism, (5) a security credential. Tackling these particular building blocks would facilitate an effective and efficient design of security/trust infrastructure. However, a security/trust infrastructure can be affected by a number of dimensions: security strength, performance and the MANET context. The MANET context describes MANET constraints and application settings. Therefore, the multi-dimensional framework is created to shed light on those dimensions in order to build a methodological approach which can leverage for well-designed security/trust infrastructures, as shown in Figure 4-5.

Part B: The SSAM model is developed to offer different security architectures which can be exploited in the service level in MANETs and also to become a case study for evaluating and validating the proposed methodological approach discussed in Part A. The SSAM conceptual model, as shown in Figure 4-6, presents its core related components. These components are server architectures,

cryptosystem, digital certificates, authentication protocols, "strategy of calling", and MANET settings. Some of these components are characterised with different properties. The server architecture component proposes different servers (i.e. *CAS*, *TAS*, *DAS*) to generate different architectures, *CAS*, *TAS*, *D\CAS*, *CAS_TAS*, *TAS_DAS* and *CAS_TAS_DAS*. The authentication protocol component involves three standard X.509 authentication protocols (*1WP*, *2WP* and *3WP* protocols). The two strategies of calling (*AAO* and *IPS*) are suggested to be used in the hybrid and hierarchical security architectures. In the SSAM design phase, the SSAM activity model is generated to demonstrate the lifecycle of a new joining node when a certain security architecture is deployed in MANETs. The Two different level models (i.e. the communication and process level models) are recognised to define the workflows of particular activities in the activity model. The communication level model specifically is used to show the flow of creating a single connection between a user and server nodes via using a particular authentication protocol. The process level model for both user and server nodes refers to the internal processes which occur while performing a security service (i.e. providing and obtaining a membership certificate) in different server architectures within SSAM.

The next chapter illustrates the implementation of the SSAM prototype (i.e. the OMNeT++ simulation model) and describes the necessary configurations and design of experiment for conducting performance testing.

Chapter 5: Implementation and Experimentation

5.1 Overview

This chapter initially discusses the proposal of model assumptions for the sake of facilitating the implementation and experimentation of the SSAM model demonstrated in the previous chapter. As the simulation approach is adopted for the SSAM implementation in this study, the OMNeT++ simulation tool has been appropriately selected for creating the SSAM prototype. This prototype consists of defining the necessary network and node structures and creating the related C++ classes. Furthermore, all important configurations and initialisations for this prototype are thoroughly addressed in order to properly conduct performance testing. Two types of configurations can be recognised in this chapter; the security and network configurations. The security configuration concerns with the SSAM server architecture, massage sizes in different authentication protocols, processing time in authentication process and the re-authentication scheme being used. The network configuration refers to various related MANET components, such as mobility, traffic, transport, routing and MAC protocols and churn mode. Eventually, the experimental design for running SSAM simulation is well illustrated. This includes selecting performance and communication measurements, defining test cases & experimental parameters and deciding the required number of replications for simulation.

5.2 Model Assumptions of SSAM Experimentation

The proposed SSAM model relies on the following assumptions so that the design of this model can be implemented and then be experimented with to study performance and communication cost using simulation. A number of these assumptions are derived from Venkatraman and Agrawal (2000), Messerges *et al.* (2003), Hadjichristofi *et al.* (2005a), Hadjichristofi *et al.*(2005b), Luo *et al.* (2005) and Yang *et al.* (2006):

- Each node before joining the network is presumed to be initialised with all required information and documents (e.g. an identity certificate, a membership validation certificate and self-signed root CA certificate) by using reliable out-of-band methods, such as biometrics. This is because services deployed in MANET in this study are controlled by some independent governing body (e.g. a company, university, army, medical association, etc.) to manage a security policy for participation and also facilitate registration and initialisation. This is the task of MRA in the SSAM model

- Regardless of a security architecture being investigated in SSAM, all security servers are predefined and configured with necessary information and certificates to deliver the security service to other user nodes.

- At the beginning of simulation, all security servers are supposed to be static at predetermined locations along with the other source traffic nodes (i.e. service provider servers). This is in order to create and maintain a central point for the network since server mobility is not taken into account in this study.

- All nodes that join the network trust all security servers. Also, in certain cases, this trust is presumed to be mutual.

- The MANET is operated in a non-hostile environment. In other words, no malicious node exists.

- Each network node, especially a server node, has adequate energy and bandwidth to take part in MANET operation and has enough computational power to run the encryption algorithms and key generation algorithms.

- Each node has a sufficient storage resource to store security materials like certificates.

- Each node has a unique non zero ID or address (e.g. a real IP address); dynamic address allocation issue is outside of the scope of this study.

5.3 The SSAM Simulation and Implementation

A simulation approach has been applied for this study, in order to test the SSAM model and investigating its potentials. This section illustrates all key subjects of the SSAM prototype and its key configurations. This prototype is based on the SSAM conceptual model discussed in Section 4.6 and 4.6. The OMNeT++ simulation tool is used for realisation of the prototype. The OMNeT++ simulator appears to have versatile features and capabilities (e.g. strong GUI, acceptance with the research community, etc.) which will be detailed in the next subsection. The key configurations, which are required for applying SSAM over MANETs, are highlighted and justified.

This section is structured as follows: The choice of OMNeT++ simulator is justified in Section 5.3.1. The OMNeT++-based components of the SSAM prototype are defined in Section 5.3.2. The SSAM and network configurations for experimentation are described in Section 5.3.3 and Section 5.3.4 accordingly.

5.3.1 Choice of Simulation Tool

There are several simulation tools that can be used to simulate MANETs, such as "NS2", "GloMoSim" and "OMNeT++". The OMNeT++ simulation tool has been selected to implement the SSAM model. This is due to the fact that this particular simulator is one of the best MANET simulation tools as emphasised in Hogie *et al.* (2006), Lessmann *et al.* (2008) and Mallapur and Patil (2012).Also, several reasons of this choice will be highlighted.

OMNeT++ (***O***bjective ***M***odular ***N***etwork ***T***est-bed in C***++)*** is an open source and open architecture discrete-event simulator whose kernel is written in C++. It is used mainly for academic and educational purposes. It is also regarded as a general-purpose tool that can simulate any system consisting of devices communicating with each other. OMNeT++ exhibits many features that support wireless and mobile networking simulation. Its components are primarily established by nested hierarchical extensible modules using a simple text-based language, Network Description (NED) which can be easy to learn, while being very easy to read. All components and modules must be coded in C++ using C++ class libraries which includes the simulation kernel and utility classes for random number generation, topology discovery, statistics collection, etc. In OMNeT++, there are main object classes, such as "module", "gate", "connection" that any new modules can be derived from. NED is used to compose individual components into larger components and models. In addition to the simulation kernel library, OMNeT++ offers a number of useful tools for setting up the simulation environment, such as a Graphical Network Editor (GNED), a NED compiler, graphical (Tkenv) or command line (Cmdenv) interfaces for simulation execution, graphical tools for simulation result analysis (Eventlog, Scave), and a model documentation tool.

Therefore, it is worth pointing out that unlike most common simulators, OMNeT++ apparently offers a user friendly extensive GUI supporting graphical network editing, animation, configuration and analysis of simulation runs as

shown in Figure 5-1. GNED normally facilitates the construction of network topologies graphically so as to provide an easy way of defining network simulation scenarios. GNED uses (*.ned) files to store all settings of the predefined topologies. The Tkenv environment alternatively leverages interactive execution of the simulation, tracing and debugging. Also, simulation execution can be easily managed in Tkenv (i.e. using start/stop buttons or changing variables or objects inside the model at runtime). Accordingly, Tkenv allows the user to have detailed view of the simulation state at any point of execution.

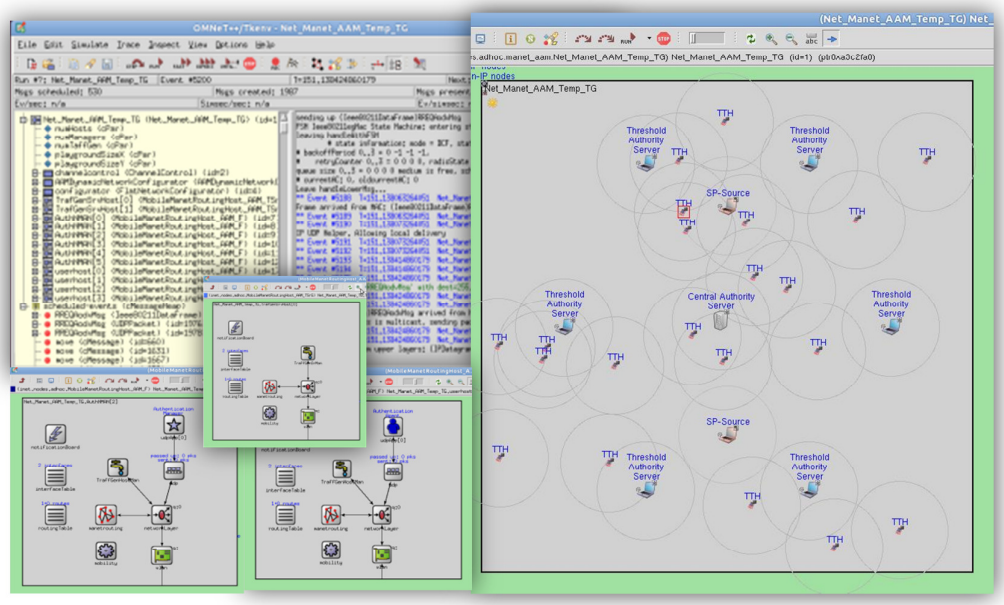


Figure 5-1: OMNeT++ GUI during SSAM simulation execution

A building block in OMNeT++ is a module that can be identified by one of the two available types (simple or compound). The simple single module is normally used to capture and process simulated behaviour whereas the compound modules can consist of the other linked sub-modules whether those sub-modules are simple or compound modules. Modules can be connected through their gates so they can interact with each other by sending and receiving messages. OMNeT++ also allows the user to associate gates with propagation delay, error rate and data rate. Also, these gates can promote only one-to-one communication. Consequently, any

simulation model can be realised as an instance of a compound module type (i.e. hierarchically nested modules). All the details of components and topology can be found in (NED) files. During simulation execution, each module can generate, read or react to messages (i.e. events). The messages can be for internal use (i.e. a self-timer) or for external use (i.e. communication with other modules).

Regarding MANET simulation, OMNeT++, through its INETMANET extension, provides a number of specific modules that can simulate some layers of the OSI model (e.g. Application, Network, MAC, etc.) and also few mobility models (e.g. Random Waypoint, GaussMarkov, and Manhattan Grid mobility models). However, not many OSI or mobility relevant models are complete in OMNeT++ (Mallapur and Patil, 2012). Nonetheless, the OMNeT++ framework is still very extensible and amendable. This compensates to a certain extent the unavailability of some models. Hence, OMNeT++ appears to be a good choice when a lot of customisation or development is expected for the simulation.

Release version 4.1 of OMNeT++ is used in this work. After searching the literature, there are many legitimate reasons for taking advantage of this particular simulator. These reasons are listed below:

1. As opposed to the other simulators mentioned earlier in this section, OMNeT++ offers the richest set of GUI features with extensive GUI support (e.g. Tkenv, GEND, Eventlog, Scave, etc.). This feature enables much more efficiency in editing, verifying, tracing, debugging and analysing the simulation.

2. OMNeT++ is open source software, and its source code written in C++ can be amended easily. Also, it is regarded as a scalable simulator dealing with large scale simulation (Mallapur and Patil, 2012) .

3. OMNeT++ is distinguished with its flexible hierarchical and modular architecture for defining its components and modules (e.g. nodes, networks,

MAC Layer) using an easy-to-learn text-based language. This helps to speed up the modelling process and reduce the level of unnecessary details during research development by focusing only on the targeted component or module.

4. OMNeT++ supports some standard modules, queues, channel control etc., which can be easily extended too, using basic C++ notation. Also, OMNeT++ includes the INETMANET package which includes the most accurate models necessary to MANETs (e.g. the AODV routing protocol, IEEE802.11g, etc.).

5. OMNeT++ offers a mobility suite with a sufficient number of models. This was the main motivation (or reason) for using OMNeT++ in this project. In addition to providing basic mobility models, OMNeT++ can generate a dynamic infrastructure to deploy links between nodes that can be dynamically established at runtime, and also to implement node churning.

6. OMNeT++ has very convenient documentation (available on the internet) and has active discussion forums for learning and solving problems. Additionally, it has a good level of acceptance in research and academia.

The next section explains SSAM implementation. This implementation includes the network infrastructure definition and the development of related MANET and SSAM C++ classes used in this simulation experiment.

5.3.2 SSAM Prototype

This section discusses the SSAM implementation using the OMNeT++ simulator. This SSAM tool, which is based on the conceptual model of SSAM (see Section 4.6 and 4.7), facilitates the simulation of MANETs. Furthermore, this tool makes use of the OMNeT++ package ("INETMANET") which includes the essential C++ classes to build and simulate MANETs (e.g. MAC and routing protocols, etc.). This "INETMANET" package has been amended to suit SSAM

requirements. To find the source-code of the SSAM tool, visit http://code.google.com/p/ssam/.

There are three key components, that can be identified (the MANET infrastructure, message types and SSAM implementation classes) to illustrate the prototype of SSAM tool in this study. The MANET infrastructure refers to network elements and their links following a top-down scheme (i.e. from the playground and channel control type to transport and application protocols being used in any node). The message types represent SSAM messages (i.e. authentication protocol messages) since other types of messages possibly generated are outside of scope of this study (e.g. routing discovery messages, MAC frames, etc.). The SSAM C++ classes represent the active elements (i.e. C++ classes) which are created to fulfil all SSAM specifications (i.e. procedures, protocols, a client-server architecture, etc.).

5.3.2.1 MANET Infrastructure

As described in Section 5.3.1, OMNeT++ modelling using the NED language typically relies on the hierarchical structure to design network simulation (compound and simple modules). Therefore, two main infrastructural levels (network and node levels) can be used to describe the MANET structure where SSAM is deployed. The network level deals with a network topology in the simulation playground (i.e. simulation area) and nodes' connections and positions. The node level presents the modular structure of a MANET node that is primarily tailored to the OSI model (e.g. MAC, "manetrouting", IP, TCP). In addition, this particular level also includes other non-OSI modules such as mobility. However, in terms of the SSAM simulation model, three different types of nodes can be distinguished in the network infrastructure (security server, traffic server and user nodes). There are presented as follows:

1. Network Level Topology

By using GNED editor in OMNeT++, the SSAM network level is described in the following NED file as shown in Figure 5-2.

```
network Net_Manet_AAM_Temp_TG
{
    parameters:
        int numHosts; // Numbers of user nodes
        int numManagers; // Numbers of server nodes
        int numTaffGen; // Numbers of traffic nodes
        double playgroundSizeX;
        double playgroundSizeY;

    submodules:

        channelcontrol: ChannelControl {
            parameters:
                playgroundSizeX = playgroundSizeX;
                playgroundSizeY = playgroundSizeY;
                @display("p=31,63;i=misc/sun");
        }

        AAMDynamicNetworkConfigurator: AAMDynamicNetworkConfigurator {
            parameters:
                dyNetworkAddress = "145.236.0.0";
                netmask = "255.255.0.0";
                numHosts = numHosts;          @display("p=7,33");
        }
        configurator: FlatNetworkConfigurator {
            parameters:
                networkAddress = "145.236.5.0";
                netmask = "255.255.0.0";     @display("p=7,6");
        }
        TrafGenSrvHost[numTaffGen]: MobileManetRoutingHost_AAM_TSrG {
            parameters:
                @display("r=50,,grey,1;is=l");
        }

        AuthNMAN[numManagers]: MobileManetRoutingHost_AAM_F {
            parameters:
                @display("i=device/wifilaptop_l;r=50,,grey71,1");
        }

   connections allowunconnected:

}
```

Figure 5-2: The NED file for defining the SSAM network level

The main definition of the network infrastructure in OMNeT++ begins with the key word "**network** *Net_Manet_AAM_Temp_TG*" which contains all necessary elements to generate the network. The "numHosts", "numManagers" and "numTaffGen" parameters indicate the actual number of different nodes that need to be created at the simulation runtime. "numHosts" denotes the total number of user nodes whereas "numManagers" and "numTaffGen" refer to the

number of security and traffic server nodes. However, as shown in Figure 5-2, each of these declared parameters are involved in a specific module, such as "`AuthNMAN`" (i.e. a security server node module) and "`TrafGenSrvHost`" (i.e. a traffic server node module). This is to enable the network model to have a flexible number of node instances in initialisation phase (e.g. `numHosts`= 100, `numManagers`= 7, `numTaffGen`= 2).

The "`control channel`" module is used to define the size (i.e. the "`playgroundSizeX`" and "`playgroundSizeY`" parameters) of a geographical simulation area ("Playground") where all generated network nodes exist and also where they are allowed to roam according to their mobility model. Furthermore, this module handles issues of radio channel transmission (e.g. signal fading and loss, a signal range, etc.). The `AAMDynamicNetworkConfigurator` module is specifically developed to dynamically create and delete user nodes in the playground and update the results of simulation. Also, this module allocates IP addresses to those generated user nodes relying on the value of its "`networkAddress`" and "`netmask`" parameters. All functionalities of this module will be explained in Section 5.4.2.3. Finally, the "`configurator`" module manages the IP addresses only for particular server nodes since those servers are assumed to already exist in the playground before starting the OMNeT++ simulation.

Figure 5-3 shows the initial state of simulation playground in OMNeT++ which includes configuration and channel control modules along with the initial network topology of most servers in SSAM (*CAS*, *TAS*, etc.). Figure 5-4 depicts a snapshot of a current state of MANET infrastructure after some nodes have joined the playground.

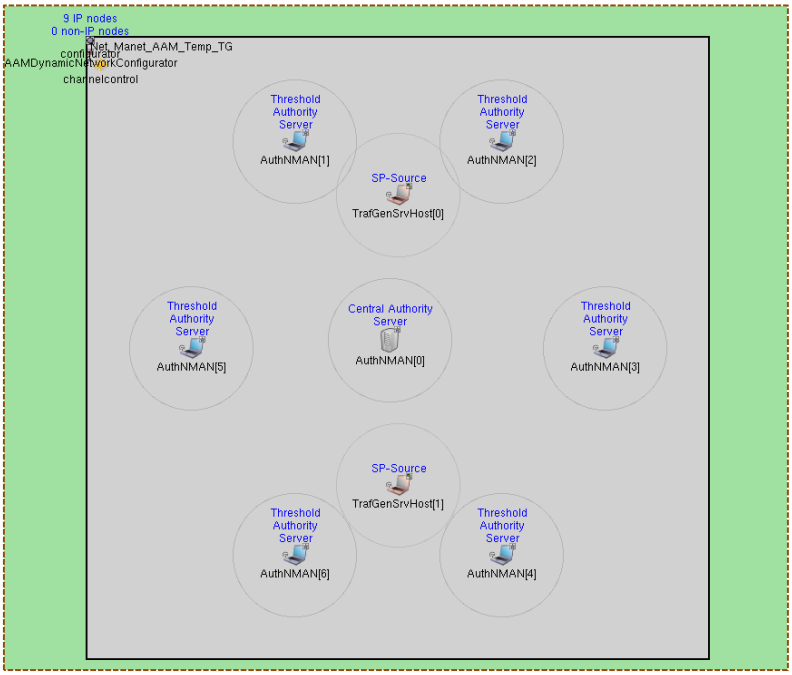


Figure 5-3: The simulation playground with the initial topology of all types of servers being used in SSAM.

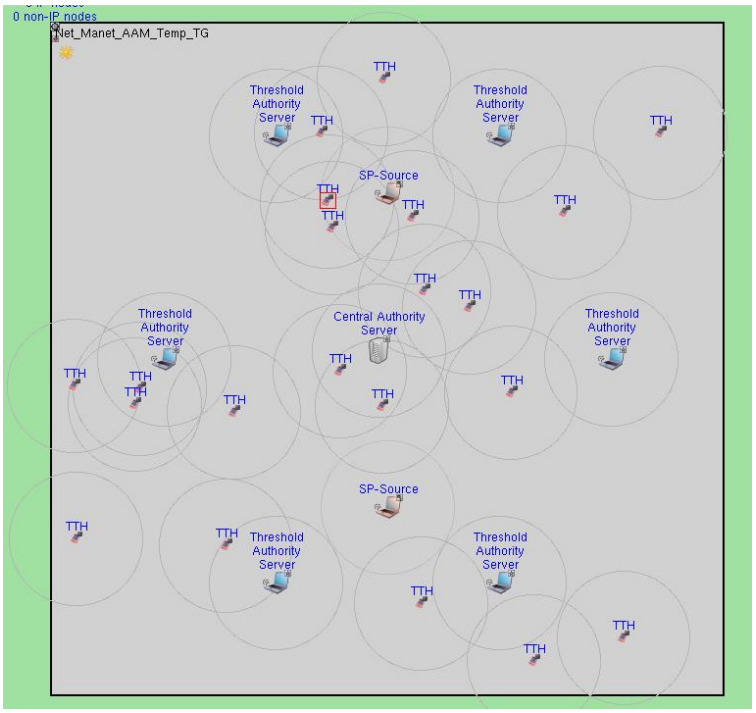


Figure 5-4: The screenshot of a SSAM simulation run in OMNeT++

2. Node Level Structure

The MANET node structure in OMNeT++ involves several modules (e.g. mobility, Wlan, network layer, etc.), each of which has its own role and connections with other modules. Some of these modules are organised and linked to match the layers of the OSI model, for example, the transport layer (a UDP module), the network layer (routing table, IP & "manetrouting" modules), data link layer (the WLAN module contains radio and MAC sub-modules), as shown in Figure 5-5. The NED file, defining the structure of a MANET node, can be found in Section A.2 (*Appendix A*).

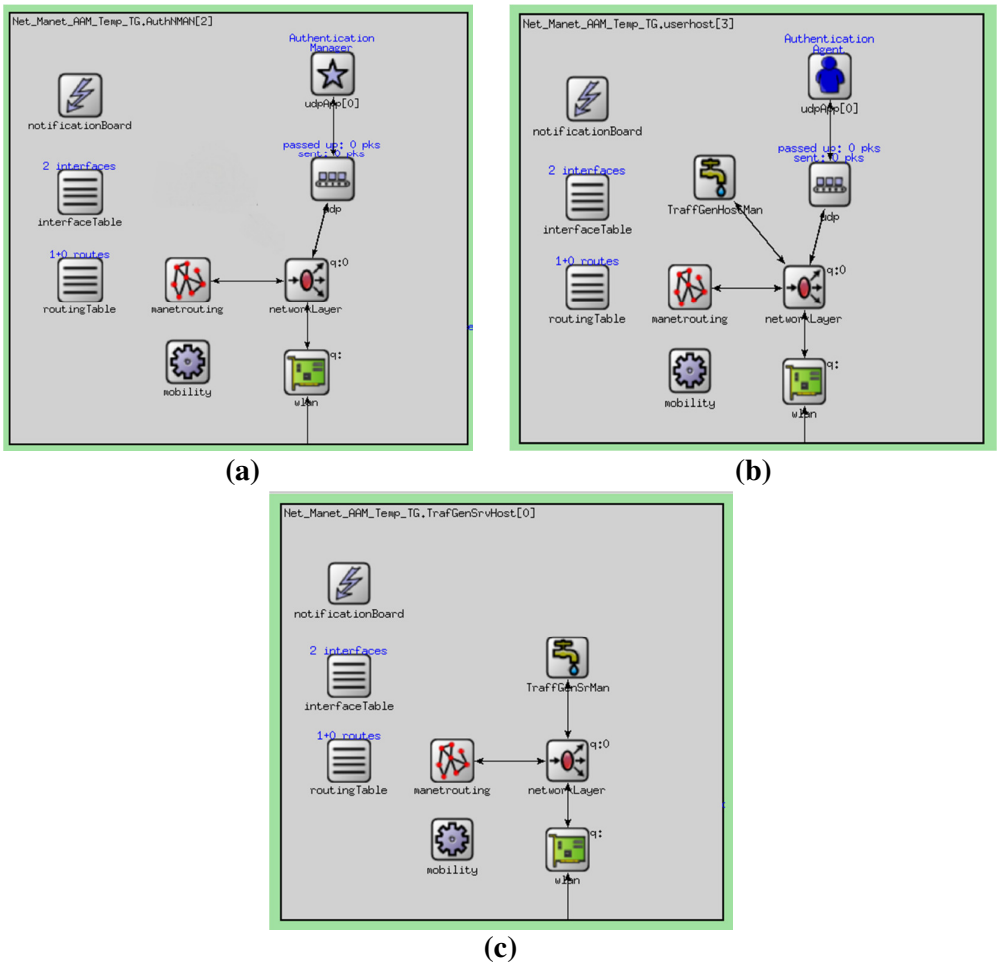


Figure 5-5: The node level structures in SSAM tool - (a) Security server node, (b) User node, (c) Traffic server node.

The mobility module (i.e. a non-OSI module) is used to enable the node to mobilise according to the mobility pattern being defined in initialisation (e.g. random walk). The "Wlan" module represents the physical and data link layers in the OSI model where signal transmission in radio channels and MAC protocols can be handled in wireless networks (e.g. IEEE 802.11 MAC protocol). The "manetrouting" module, along with the "network layer" and "routing table" modules, manages routing between nodes in MANETs (i.e. includes the implementation of a particular MANET routing protocol). The "udp" module is defined to implement the UDP transport protocol.

In SSAM, three distinct nodes can be identified (user, security server, and traffic server nodes) as presented in Figure 5-5. Every node is distinguished from each other by the type of the application module being used. In user node, the "updApp" module (i.e. called an authentication agent) handles user invocations and servers' replies (i.e. processing according to a specific authentication protocol and a server architecture). Also, a user node contains the "TraffGentHostMan" module which is used to interact with traffic server nodes to trigger traffic in the network. While, in the security server node, the "updApp" module (i.e. called an authentication Manager) to take a role of security service which provides calling user nodes with authentication (i.e. distributes membership certificates). In the traffic server node, the "TraffGentSrMan" module is used to generate traffic in the network following particular traffic patterns (e.g. CBR, etc.).

5.3.2.2 Message Declaration

Each of the authentication protocols being used in SSAM depends on the handshaking of different messages (i.e. control and data messages) between user and security server nodes. Also, it is important in simulation to differentiate between those messages as they are the key part of the authentication process in SSAM. OMNeT++ offers a message file editor which is used to create the "msg" file. This text-based file, which uses C++ notations, normally contains the

definition of requisite messages (along with their fields) for OMNeT++ simulation. After compiling this "msg" file, a number of specific C++ class files are automatically generated. These particular class files will be used to simulate a message object at simulation runtime.

```
enum AAuthNProtoType // Authentication Protocol type
{
    X509_One_Pass   = 1;
    X509_Two_Pass   = 2;
    X509_Three_Pass = 3;
}
enum AAMMessageType
{
    // Authentication Protocol Control Messages type
    AAM_AUTHN_PROT_MSG1 = 1;
    AAM_AUTHN_PROT_MSG2 = 2;
    AAM_AUTHN_PROT_MSG3 = 3;

    // Authentication Protocol Data Messages type
    AAM_AUTHN_ACK = 5; // Successful Authentication
    AAM_AUTHN_NOACK =6; // Unsuccessful Authentication (Not in use)
};
packet CommonAAM  // Common Auhority Architecture for MANETs Message
{
    int MsgType enum (AAMMessageType);
    int indxSrv ;  // Server Index in for user use
    int SeqNum ;  // call Freshness
}
packet AuthenProtocolMsg extends CommonAAM  // a control message
{
    int Nonce;
    string Timestamp ;
    string IdCertificate;
    string Signature;
    int AuthNType enum (AAuthNProtoType);
}
packet DataMsg extends CommonAAM  // a data message
{
    string AttrbCert; // User Membership Certificate
    string ThreshAttrCert ; // Partial Membership Certificate
    string AuthorityDeleCert; // Authority delegation Authority
}
```

Figure 5-6: The message file ("*.msg") for declaring messages for authentication protocols in SSAM

As shown in the SSAM "msg" file (see Figure 5-6), three authentication protocols are identified by the "*AAuthNProtoType*" variable while the "*AAMMessageType*" variable refers to different types of messages available for use. Those two variables are also acknowledged as a variable type that will be exploited in the body of any defined message. "*CommonAAM*" is a generic message (i.e. packet superclass) which contains all shared fields among other messages

(e.g. "*MsgType*", "*SeqNum*" and "*indxSrv*"). Both "*AuthenProtocolMsg*" and "*DataMsg*" messages, which are derived from "*CommonAAM*", denote the body of control and data messages. "*AuthenProtocolMsg*" has a number of distinct control fields based on the authentication protocol being used, such as "*Nonce*", "*Timestamp*", "*IdCertificate*", "*Signature*" and "*AuthNType*". "*DataMsg*" is used to carry a user membership certificate and other relevant credentials according to the server architecture in use. By using the OMNeT++ message compiler (opp_msgc), the "*CommonAAM*", "*AuthenProtocolMsg*", and "*DataMsg*" C++ classes are automatically generated with all necessary variables and methods from the SSAM "msg" file described above. These specific classes are also considered a part of SSAM implementation classes as shown in Figure 5-7. This figure illustrates the class diagram of SSAM. Both "*AuthenProtocol-Msg*" and "*DataMsg*" classes are derived from superclass "*CommonAAM*" classes. All classes in this regard include specific methods which are used to manage message fields (i.e. obtain or set fields in the message object). For example, the "*setMsgType()*" and "*getMsgType()*" methods are used to obtain and set the "*MsgType*" field in the "*CommonAAM*" class.

5.3.2.3 SSAM C++ Classes

After defining the network, the node structures, and the messages in the previous sections, this section demonstrates the main classes (written in C++) of SSAM application so as to be integrated with their predefined modules or messages. These particular classes, which include all necessary data and methods, are utilised to implement and simulate the related SSAM activities, processes and workflows (see Section 4.7) in OMNeT++ environment. As shown in the class diagram in Figure 5-7, various classes along with their associations, can be recognised based on their own roles in SSAM, such as "*AuthNAgent*" (user node), "*AuthNMAN*" (security server), etc. Furthermore, a number of these classes are derived from particular built-in superclasses (e.g. "*cMessage*",

"`BasicModule`" "`UDPAppBase`", etc). To understand the classes' roles in SSAM, key functionalities and dependencies of each class are highlighted in the following sections.

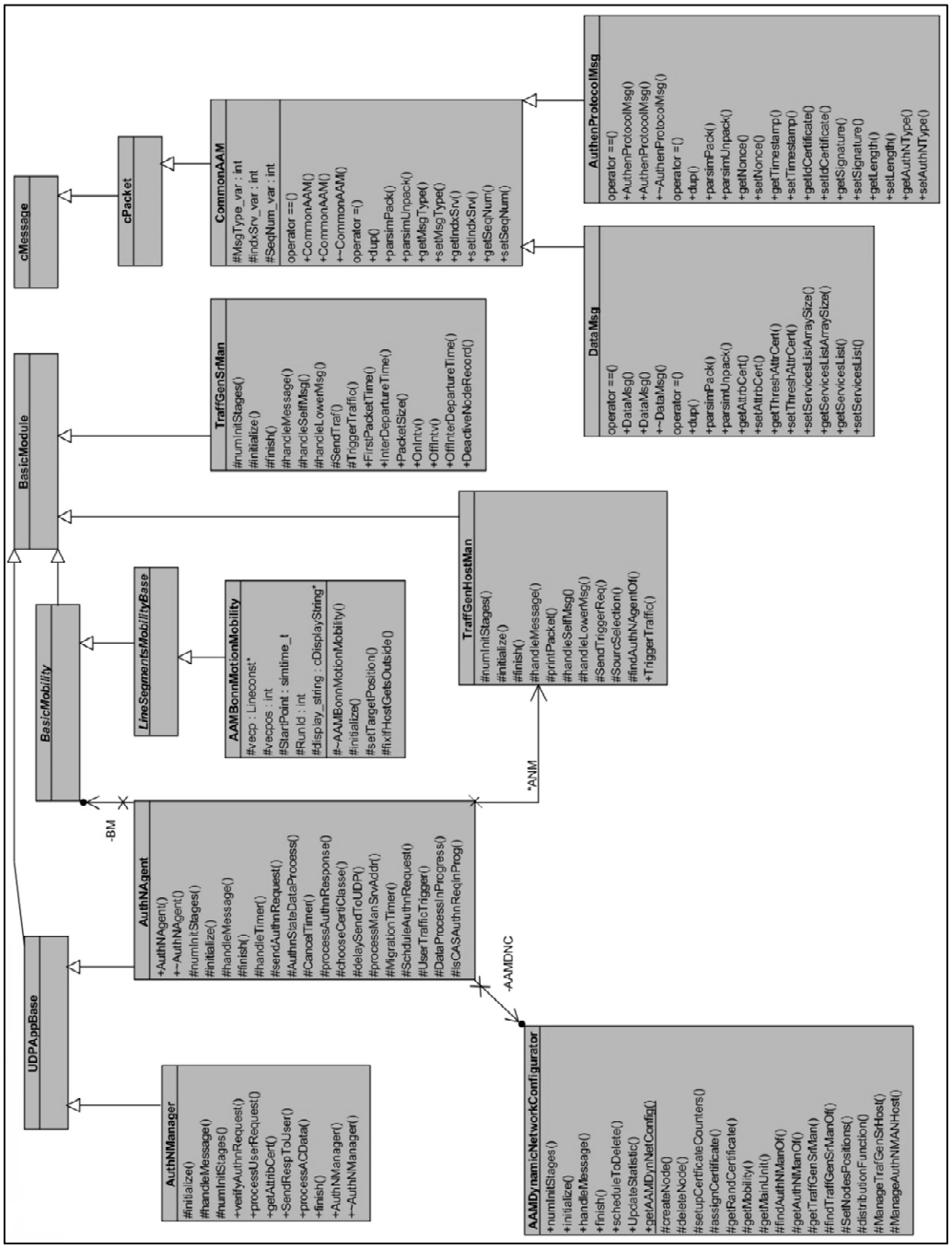


Figure 5-7: The SSAM Class Diagram

1. The "*AAMDynamicNetworkConfigurator*" Class

This class mainly deals with user node creation, stay and deletion (i.e. joining, staying and leaving) dynamically at the simulation runtime. Those events can be simulated by using a number of methods in this class. Initially, it is essential to schedule joining times for potential nodes and this is implemented by setting up arrival timers in the "*initialize()*" method, as shown in Figure 5-8. These timers, which are self-messages (*msgTimer*) marked with "*CREATE_NODE_MSG*", are created according to the joining model being adopted, such as the Poisson arrival model.

```
void AAMDynamicNetworkConfigurator:: initialize ()
{
 . . . . . .
 for (int i= 0; i <nH; i++)
      {
          int * xp = new int(); // Number ID
          *xp = i ;

          std::stringstream msgName ;
          msgName <<"CreateNode["<<i<<"]";
          cMessage *msgTimer = new cMessage ((msgName.str()).c_str(),CREATE_NODE_MSG);
          msgTimer->setContextPointer(xp);

          // The Type of arrival 1 . Uniform , 2. Possion Process
          switch(indxs)
          {
              case 1 :     // Uniform Arrival
              . . . . . .; break;

              case 2 :  // Poisson Arrival

              x = x+ exponential (BlockUnit/(double)par("JoinRate"));
              break;

          default: opp_error("Error in Arrival Type");
          }
       }
 . . . . . .
}
```

Figure 5-8: The lines of code to generate user node arrivals in the "*initialize()*" method

On the other hand, the "*scheduleToDelete()*" method is used to initiate a timer for a user node which needs to be removed from the playground (i.e. a leaving event). This particular timer is triggered based on certain conditions (i.e. an authentication failure or lifetime) and also it is marked with

"*DELETE_NODE_MSG*" for identification purposes. As the "*handleMessage()*" method typically handles messages (i.e. timers or inbound packets) in any classe in OMNeT++, both "*DELETE_NODE_MSG*" and "*CREATE_NODE_MSG*" markers are utilised to indicate the method "*createNode()*" or "*deleteNode()*" that needs to be called accordingly within the "*handleMessage()*". The "*createNode()*" method concerns with creating a new node in the playground and initialising with all necessary configurations, such as an initial node position, a node IP address, display features, etc. Whereas, the "*deleteNode()*" method is used to remove a particular user node from the playground. The "*UpdateStatistic()*" method is used to support result collection for specific metrics, such as success ratio, failure frequency, etc.

2. The "*AuthNManager*" Class (Security Server Node)

The "*AuthNManager*" class takes the role of managing the security service in the server side. This is by handling the calls of user nodes through completing authentication protocol handshaking in order to provide the nodes with their membership certificates, according to the SSAM flowcharts and sequence diagrams, see Section 4.7.1. This class is also derived from the main "*UDPAppBase*" class, in order to be embedded in the "udpApp" module (i.e. application layer) in security server node as described in the node level structure in Section 5.4.1. In initialisation phase, various parameters should be initialised in this class, such as "*AuthorityServerType*", "*respMsgLengthAK*", "*localAuthnManPort*", "*MaxServicesCapacityBuffer*", etc. Different types of servers (*CAS*, *TAS* and *DAS*) can be distinguished in this particular class. This is by setting up the "*AuthorityServerType*" parameter. In addition to handling self-messages (i.e. timers), the "*handleMessage()*" method controls the flow of user messages that are received from the gate of this module using a queue (i.e. a service buffer) as it is supposed to handle many invocations from various nodes. On the other hand, there are a number of timers with distinct markers (e.g. "*Verification_Timer*", "*Protocol_Process_Timer*") along with a number of methods (e.g. "*verifyAuthnRequest()*","*processUser-*

Request()" and "*SendRespToUser()*") which incorporate in this class in order to simulate the processing of user requests, as shown in Figure 5-9. Also those particular timers can be initiated based on the processing time model being adopted as this issue will be elaborated in Section 5.4.3.

```
if (msg->arrivedOn("udpIn"))
     {
       if(reqBuffer.length()< MaxServicesCapacityBuffer)
         {
           // buffering a user node request if thesecurity service is busy

             if(reqBuffer.empty() && currentAtrrCreaThreads <MaxThreads)
             {
                 verifyAuthnRequest(PK(msg));
             }
             else
             {
                 reqBuffer.insert(msg);
             }
         }
         else
         {
           delete msg; // drop the request due to full buffer
         }
     }
     else if (msg->isSelfMessage())
     {
       switch(msg->getKind())
       {
         case Verification_Timer     : processUserRequest(msg); break;

         case Protocol_Process_Timer : SendRespToUser(msg); break;

         case Req_Release_Timer      ://for releasing a request from buffer

             if (reqBuffer.empty())
             {
                 ev<< "No Request to serve --------->  " ; return;
             }
             else
             {
               ev<< "RELEASE REQUEST FROM REQUEST BUFFER)(:)**** "<<endl ;
               verifyAuthnRequest(PK(reqBuffer.pop()));
             }
             break;
         }
     }
 }
```

Figure 5-9: The body of "*handleMessage()*" for handling user messages and self-message timers

The "*verifyAuthnRequest()*" method is used to verify a user request released from the service buffer, whereas the "*processUserRequest()*" method is used to process a user request after verification (e.g. updating a user record and preparing credentials for authentication). The "*SendRespToUser()*"

method is used to generate a server reply and send it back to the relevant user node (i.e. may carry a user membership certificate).

3. The "*AuthNAgent*" Class (User Node)

The "*AuthNAgent*" class, similar to "*AuthNManager*", as shown in Figure 5-7, is derived from the main "*UDPAppBase*" class, to be integrated in the "udpApp" module (i.e. application layer). In the "*initialize()*" method, many parameters in this class should be initialised according to the particular security architecture being applied. An example of these important parameters and the value that they can take are presented in Table 5-1.

Parameters	Description
AuthenticationType	To indicate the authentication protocol being used 1 ➔ *1WP* ; 2➔ *2WP* ; 3➔ *3WP*
SrvTypTestCase	Refer to server architecture being deployed 1 ➔ *CAS*, *D\CAS*; 2➔ *TAS*; 3 ➔ *CAS_TAS*, *TAS_DAS* (disableCAS= true); 4 ➔ *CAS_TAS_DAS*
strategyType	Indicate to the strategy with calling when considering two- or three-level server architectures. 0➔ All at Once (*AAO*); 1➔ In Priority Sequence (*IPS*)
fix_Exp_WT	The re-authentication scheme being used 0 ➔ the fixed model; 1 ➔ the exponential model
WTime	Waiting time Unit between two consecutive authentication attempts (e.g. T_0=30s)
MaxReAuthNCounter	A number of Attempt for authentication (e.g. 3 tries)

Table 5-1: indicative parameters in "*initialize()*"

The "*handleMessage()*" method in this class includes two main sub-methods, "*handleTimer()*" and "*processAuthnResponse()*". "*handleTimer()*" manages timer messages when expired, whereas "*processAuthnResponse()*" processes authentication protocol messages,

received from security server nodes, such as "*AuthenProtocolMsg*" marked with "*AAM_AUTHN_PROT_MSG2*" and "*DataMsg*" marked with "*AAM_AUTHN_ACK*". For managing invocation, the "*ScheduleAuthn-Request()*" method aims to prepare necessary server calls related to the particular server architecture being used via using certain timers marked by "*MSGKIND_START_REQ*". Besides, the "*sendAuthnRequest()*" method generates different server invocations that comply with different authentication protocols as shown in the sequence diagram for each authentication protocol, see Section 4.7.1. There are other supporting methods in this class that handle authentication states, such as "*AuthnStateDataProcess()*", "*delaySendTo-UDP()*", "*DataProcessInProgress()*" and "*CancelTimer()*".

On the other hand, various timers, which are handled by "*handleTimer()*", are developed to model different conditions (e.g. waiting for server replies) and processes (e.g. verification and generation in authentication). For example, once a user node is created in the playground, this node should search for another node to connect (exploration). The "*SearcForConnection*" timer marked with "*MSGKIND_SRCH_FOR_CONN*" enables a new user node to explore other active existing nodes in its vicinity (i.e. radio range) while it is roaming. In other words, a user node cannot initiate a server call until initially this node can find at least another active node to start communicating. This can be implemented by the lines of code in "*initialize()*" and "*handleTimer()*" methods that are presented in Figure 5-10.

```
void AuthNAgent::initialize(int stage){
... . . . . .
if(par("SearchForConnection").boolValue() && !BM->isAnyNeighbourAround())
        {
         std::stringstream Natimer;
         Natimer<<"Serch4Conne-"<<getParentModule()->getFullName();
         cMessage * SearcForConnection =
        new cMessage(Natimer.str().c_str(), MSGKIND_SRCH_FOR_CONN);
         scheduleAt(simtime_t
        (simTime()+par("updateSearchTimer").doubleValue()),
        SearcForConnection);
        }
... . . . . .
}
void AuthNAgent::handleTimer(cMessage *msg)
{
 . . . . . .
 case MSGKIND_SRCH_FOR_CONN : // Search for nearby node to access

     if(!SetMobilMod) BM->Registration(true); turn on signal
     if (!BM->isAnyNeighbourAround())

scheduleAt(simtime_t(simTime()+par("updateSearchTimer").doubleValue()),msg);
             else
                 {
                     SchduleAuthnRequest(strategyType,TTimer); //call servers
                     delete msg;
                 }
             TotalSrcTime = simTime()- timeToStart;
     if (!SetMobilMod) BM->Registration(false); // turn off signal
         break;
. . . . . .
```

Figure 5-10: The search to connect implementation in SSAM

The "*BM->isAnyNeighbourAround()*" method is associated with the basic mobility module which collaborates with the control channel module in OMNeT++ to check any other nodes within the radio range for a certain user node. The other timers, which primarily represent a processing delay, are briefly described in Table 5-2:

Timer Marker	Description
MSGKIND_START_REQ	Processing an initial authentication request
MSGKIND_ReREQ	Processing authentication re-try
MSGKIND_PR_PROS_TIMER	Only used for validating a control message 2 (CMsg2) in 2WP authentication protocol.
MSGKIND_DATA_PROS_TIMER	Validating membership certificate obtained from security server.
MSGKIND_COMBINPR_TIMER	Combining partial certificate received

	from servers in the *TAS* architecture.
MSGKIND_DAS_REQ	Generating initial *DAS* request when DAS is involved.
MSGKIND_DAS_ReREQ	Processing authentication re-tries only for the case of *DAS*
MSGKIND_DAS_PROS_TIMER	Validating membership certificate (Chain) obtained from *DAS*.

Table 5-2: Timer markers used in the "*AuthNAgent*" class

The "*UserTrafficTrigger()*" method is used to enable the "TraffGenHostMan" module within the user node to call available traffic servers in order to generate traffic in the network.

4. The "*TraffGenHostMan*" Class (User Node)

This class is the implementation of the "TraffGenHostMan" module in the structure of user node. The role of this class is to trigger traffic by calling available traffic server nodes. This class includes a number of methods that support traffic handling in the user side. The "*TriggerTraffic()*" method is utilised to initiate the process of traffic generation between traffic server and user nodes. The "*handleSelfMsg()*" method is used to deal with recalling timers, whereas the "*handleLowerMsg()*" method processes traffic that is received from traffic server nodes.

5. The "*AAMBonnMotionMobility*" Class (User Node)

This class is the implementation of the mobility module in user node which task is to update node's movement (i.e. coordination) in the playground according to the mobility model being used. Also, this class is created based on the built-in superclass in "INETMANET" package for mobility, such as "*LineSegments-MobilityBase*" and "*BaiscMobility*". On the other hand, the trace file, generated from the "BonnMotion" tool (see Section 5.3.4.1), is imported into this class to create a node journey in the playground when a certain user node joins.

6. The "*TraffGenSrMan*" Class (Traffic Server Node)

The "*TraffGenSrMan*" class is developed to simulate a source of traffic in the network. Furthermore, this class typically interacts with the "*TraffGenHost-Man*" class in the user node to implement the particular traffic model being proposed (e.g. CBR, ON/OFF Pareto, etc.). A number of methods are coded to facilitate traffic generation, such as "*handleSelfMsg()*"(i.e. handling sending timers), "*SendTraf()*" (i.e. managing traffic dispatching), "*handleLowerMsg()*" (i.e. processing user traffic request), etc.

The next section addresses the issues of configuration and initialisation for SSAM simulation model. These issues concern with both the SSAM model and network characteristics.

5.3.3 The SSAM Security Configurations

The security model apparently relies on two elements for deployment: security servers and authentication protocols. However, these two elements can be associated with several settings and these certain settings may represent different situations where and how the model can be implemented and tested. To prepare the model for experimentation, all settings are therefore defined which are tailored to this study's goal in the next subsections.

5.3.3.1 The Server Architecture (*Security Servers*)

In accordance with the SSAM described in Section 4.6, four specific types of security servers (i.e. *CAS*, *TAS*, *DAS* and *D\CAS*) are used to give security services (i.e. generating membership credentials of user nodes when authenticated) to other user nodes to establish trust in MANETs. Each particular server type has its own properties in terms of the number of servers required, server placement, independency and mobility. In the light of assumptions made for this study, some

of these properties are vital that they are identified and justified for performance experimentation to avoid any misleading experimental results. Therefore, these properties will now be discussed for each security server type.

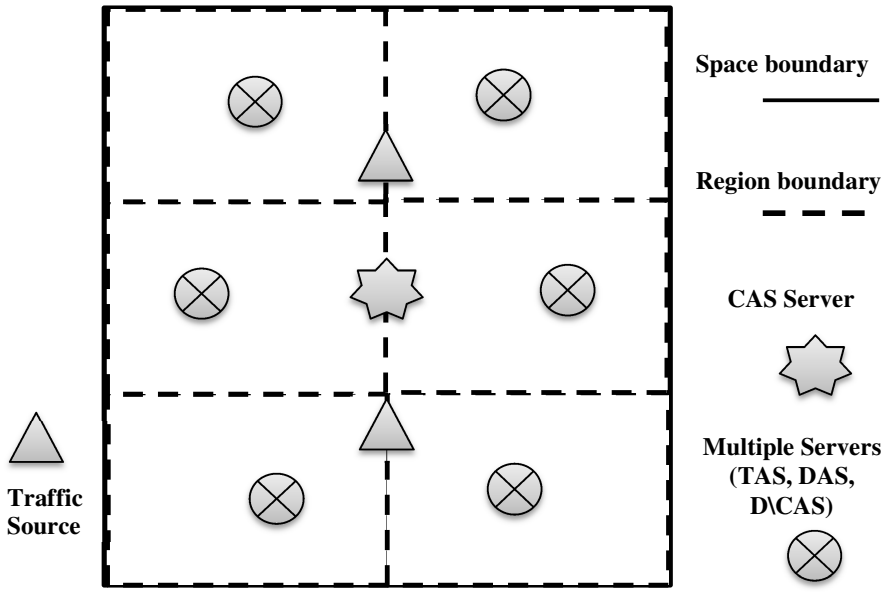


Figure 5-11: Simulation Area "Playground" and Server Location

Regarding the requirements of multiple servers, the *CAS* case needs only a single standalone server for providing a security service whereas the *TAS*, *DAS* and *D\CAS* cases are based on a scheme of multiple servers. However, the *TAS* case is distinguished from *DAS* and *D\CAS* because the *TAS* case relies on a group of particular servers cooperating together in the security service provision (i.e. dependant servers). In the cases of *DAS* and *D\CAS*, each server can independently provide the whole security service even though each one has different procedures for providing the security service as discussed in Section 4.6. Six server nodes for each case of *TAS*, *DAS*, and *D\CAS* are predetermined and uniformly distributed in six regions of the square space (i.e. a simulation area "playground") while one server in the *CAS* case is located in the centre of the space. Each server is presumed to belong to its region and may be responsible for offering services to nodes of its region. Figure 5-11 displays the all proposed topologies for locations of all server types in SSAM; depending upon the security

architecture being considered, a circle shape can take one or two of the available three forms (*TAS*, *DAS* and *D\CAS*). On the other hand, the *TAS* architecture is built on the *TC*(k = 3, n = 6) scheme (i.e. threshold value k is estimated based on Gennaro *et al.* (2000)), in other words, any 3 *TASs* is sufficient to provide a security service to user nodes.

In this study, the main target is to test the performance of different security architectures with a fixed number of servers. This is due to the fact that increase in the number of servers in these particular cases would entail proportional impact on output performance metrics. It is arguable that SSAM is developed in this study to offer different types of server architectures to user nodes (i.e. usage side) ignoring the different server counts for multiple-server-based cases as these can be further researched in the server management dimension. Eventually, all types of servers are uniformly located and fixed (i.e. static) in the middle of separate symmetrical regions. For server topology balancing, this particular distribution would enable those static server nodes to cover the most space of playground during simulation, especially in multi-server architectures.

5.3.3.2 The Authentication Protocols

As the three security communication protocols (i.e. One Way Pass, Two Way Pass and Three Way Pass protocols), as well as the threshold cryptosystem, are considered to be a means of performing authentication between user and server nodes, they have different settings that need to be identified. According to each protocol workflow shown in Section 4.7.1.1, the message size, processing time, and the re-authentication scheme (i.e. waiting time and a number of attempts) are the key parameters in configuring each protocol model for the simulation.

- **The Message size Model:**

Each protocol is normally distinguished with a particular count and type of messages exchanged between two ends (i.e. control and data messages). Also,

each certain message in every protocol includes different contents (e.g. timestamp, nonce, an encrypted session key, etc.). Due to the fact that the type of adopted authentication protocols involved in SSAM is asymmetric for processing authentication (i.e. unilateral or mutual authentication), X.509-v3 certificates must be piggybacked on certain protocol messages for facilitating authentication. Therefore, the size of protocol messages differs from one another according to their sequence and required contents and the type of server architecture in use. All message protocol sizes are summarised in Table 5-3.

		Authentication Protocols (Security Communication)		
		Three Way Pass (3WP)	***Two Way Pass (2WP)***	***One Way Pass (1WP)***
Data Message	*DMsg1*	**1024 Byte** (TAS,CAS,D\CAS) 2 ×1024 = **2,048 Byte** (DAS)	**1024 Byte** (TAS,CAS,D\CAS) 2 ×1024 = **2,048 Byte** (DAS)	**1024 Byte** (TAS,CAS,D\CAS) 2 ×1024 = **2,048 Byte** (DAS)
Control Messages	*CMsg1*	**1110 Byte**	**1110 Byte**	**1110 Byte**
	CMsg2	**1110 Byte**	**1110 Byte**	
	CMsg3	**512 Byte**		

Table 5-3: Message sizes in the authentication protocols

The average size of a standard certificate is proposed to be 1KB corresponding to the approximate size of actual X.509 digital certificate generated by the most famous toolkit, OpenSSL (Cox *et al.*, 2002) as certain protocol messages may contain a certificate like DMsg1, CMsg1 and CMsg2. However, in the particular case of using *DAS*, the size of the protocol data message (DMsg1) becomes double because the authentication protocol specifications, as explained in Section 4.7.1.1, indicate that message should include two types of certificates, a node membership certificate along with a server delegation certificate.

- **The Processing Times Model:**

Processing time (*T*) is typically defined as a delay or cost that each operation in the sequence flow of a particular authentication protocol needs in order to be completed (e.g. signature creation and validation, certificate issuing, partial

certificate combining, etc.), as shown in Figures 4-11, .. , 4-18. Processing time can vary because this cost depends upon three elements which are the device capability, the operating system utilisation, the cryptographic function in use (e.g. asymmetric or threshold algorithms).

In the literature, there are several studies that conduct primitive operation evaluation to the real implementation of different cryptographic algorithms on particular devices such as Luo *et al.* (2004), Raghani *et al.* (2006), Komninos *et al.* (2007) and Saxena *et al.* (2007). However, the outcomes from those cryptographic algorithm evaluations are restricted to the particular context and time. This is because these outcomes are realised from testing specific cryptographic algorithms in a specific hardware in the past. In recent times, there has however been rapid development of hardware and many efficient cryptographic algorithms. Therefore, considering processing time in certain cases possibly becomes insignificant. Nonetheless, This study makes use of *generic processing time* as in Salmanian *et al.* (2010) following a standard uniform distribution. Processing time in a real network system, where devices may run different operating systems, depends not only on the cost of running cryptographic functions of authentication protocols but also on other factors, for example, device capacities, pipelines in operating systems, reading/writing to storage resources, buffering, etc.. Therefore, generic processing time is considered for this context. Additionally, this study does not take into consideration any specific power and capacity model for the network nodes.

According to the design of these authentication protocols, presented in the sequence and workflow charts in Section 4.7.1, there are miscellaneous types of processing time for generating and validating protocol messages, creating new user record and combining received partial certificates. Table 5-4 outlines all types of processing time alongside with their value model. As can be seen from Table 5-4, uniformly distributed values between 0.3 and 3 seconds are used for modelling processing time in this study. It is obvious that, in the case of *DAS*, the time of validating data messages is two times greater than others server models

due to validating a certificate chain with a length of two certificates. On the other hand, the procedure of creating a data message including the membership certificate for a particular user arguably takes more time than the process validating the same message at the other end. In the control message processing times for the Three Way Pass (*3WP*) protocol, as shown in Table 5-4, the third message operations (i.e. generating and validating) are specified to take less than the half processing time of the other two messages within the protocol. This is due to the fact that according to the *3WP* design specifications, processing in this specific phase includes only the generation or verification of a comparatively small acknowledgment message (i.e. *CMsg3*) without having any extra cryptographic operations, such as certificate validation.

			Authentication protocols (Security Communication)		
			Three Way Pass (3WP)	***Two Way Pass (2WP)***	***One Way Pass (1WP)***
Data Message (*DM*) Processing Time [Sec]	T_{GDM} - *Generating* ***DM*** *(Server nodes)*	*CAS D\CAS*	uniform(2,3)	uniform(2,3)	uniform(2,3)
		TAS	uniform(2,3)	uniform(2,3)	uniform(2,3)
		DAS	uniform(2,3)	uniform(2,3)	uniform(2,3)
	T_{VDM} - *Validating* ***DM*** *(User nodes)*	*CAS D\CAS*	uniform(1,1.5)	uniform(1,1.5)	uniform(1,1.5)
		TAS	uniform(1,1.5)	uniform(1,1.5)	uniform(1,1.5)
		DAS	2 × uniform(1,1.5)	2 × uniform(1,1.5)	2 × uniform(1,1.5)
Control Messages (CM1/2/3) Processing Time [Sec]	T_{GCM1} – *Generating* ***CM1***		uniform(0.8,1)	uniform(0.8,1)	uniform(0.8,1)
	T_{VCM1} – *Validating* ***CM1***		uniform(0.8,1)	uniform(0.8,1)	uniform(0.8,1)
	T_{GCM2} - *Generating* ***CM2***		uniform(0.8,1)	uniform(0.8,1)	
	T_{VCM2} – *Validating* ***CM2***		uniform(0.8,1)	uniform(0.8,1)	
	T_{GCM3} - *Generating* ***CM3***		uniform(0.3,0.5)		
	T_{VCM3} – *Validating* ***CM3***		uniform(0.3,0.5)		
Server side - User Record Processing Time (T_{UsrRec}) = uniform(1,1.5) applied once at the first invocation					
• User Side – Processing Time of combining partial certificates (T_{Comb}) = uniform (1.5, 2) (combining partial certificates) in the TAS model. • *Uniform (x , y) is a standard uniform distribution function between lower-bound x and upper-bound y					

Table 5-4: The processing times model in SSAM for simulation

- **The Re-authentication Scheme:**

It is unrealistic to ensure that each distinctive attempt of a server invocation for authentication will lead to a successful authentication, especially in dynamic networks as MANETs. Therefore, it is important to consider the case of attempt failure by using a re-authentication scheme which enables a server invocation for authentication to be systematically renewed in order to overcome connection limitations. There are a few number of studies such as Yi and Kravets (2003), Luo *et al.* (2004), and Larafa and Laurent (2011), that have drawn attention to the re-authentication process.

Complying with the approach of Larafa and Laurent (2011), it is essential to consider the three settings that the proposed re-authentication scheme is based on. These settings are the total waiting time (*Timeout*) before giving up calling, time interval (*T*) for waiting before triggering a new attempt, and a maximum number of attempts (*MNA*). However, larger *Timeout* values, regardless of a maximum number of attempts, can improve the chance of authentication success because the node waits longer for completing the request via the predetermined message-based handshaking protocol with certain servers to obtain its membership certificate. Yet, if some of those protocol messages are dropped or lost in transmission, this will entail failure in the request and the larger *Timeout* values can lead the node to back off unnecessarily. In the case that the *Timeout* value is defined too small, even if there are sufficient replies on their way back to the user node for completing the request, the user node abandons too early resulting in being discarded. On the other side, with having large *MNA*, the chance of authentication success would typically be increased, however an increase in *MNA*, incurs extra traffic in the network. Hence, these parameters must be carefully defined for a more effective and efficient re-authentication scheme. There are different ways to model this scheme by using different trends in increasing waiting time (i.e. linear, exponential, etc.) and numbers of attempts.

Two common parameter models have been identified for this study, the fixed and the base-2 exponential interval models. In the fixed interval model, the time interval between attempts is constant during *Timeout*. The base-2 (2^x) exponential model uses a variable time interval (i.e. initial waiting time unit T_0 **= 30** seconds) as a time interval between two consecutive attempts differs exponentially. This allows more relaxed time for authentication comparing to the fixed model. Besides, **three** attempts are defined to be the maximum number of attempts for server invocation and accordingly the total *Timeout* in the base-2 exponential model is **210** seconds, as presented in Figure 5-12, whereas it is **90** seconds (i.e. 3 Tries × (T_0 = 30s)) in the fixed model.

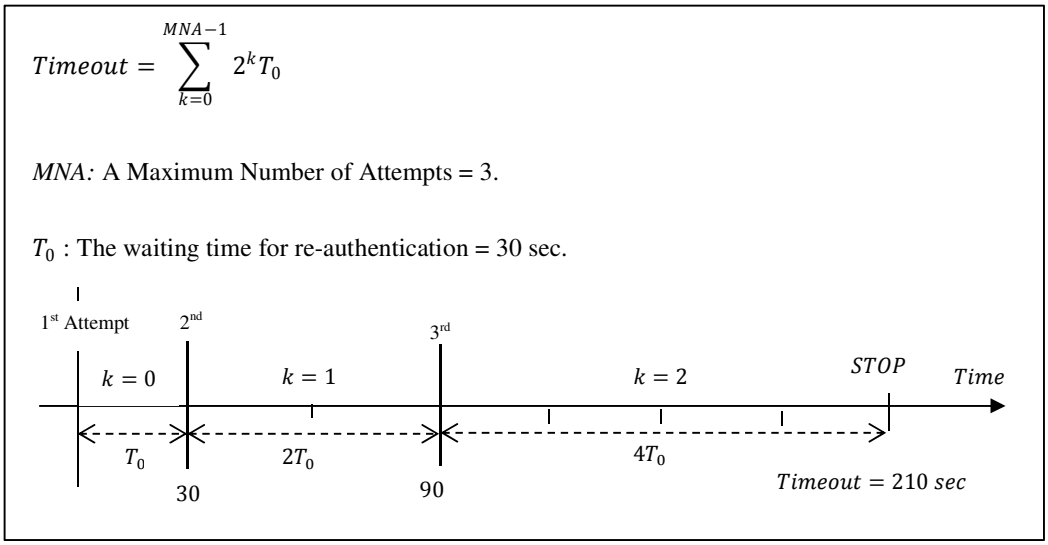


Figure 5-12: The base-2 exponential re-authentication scheme in SSAM

The base-2 exponential model is used in SSAM for facilitating the invocation of almost all server types (i.e. *CAS*, *TASs* and D\CASs) as this scheme arguably offers flexibility in the time interval within *Timeout*. Therefore, the time intervals vary to facilitate the connection demands so as to boost calling success.

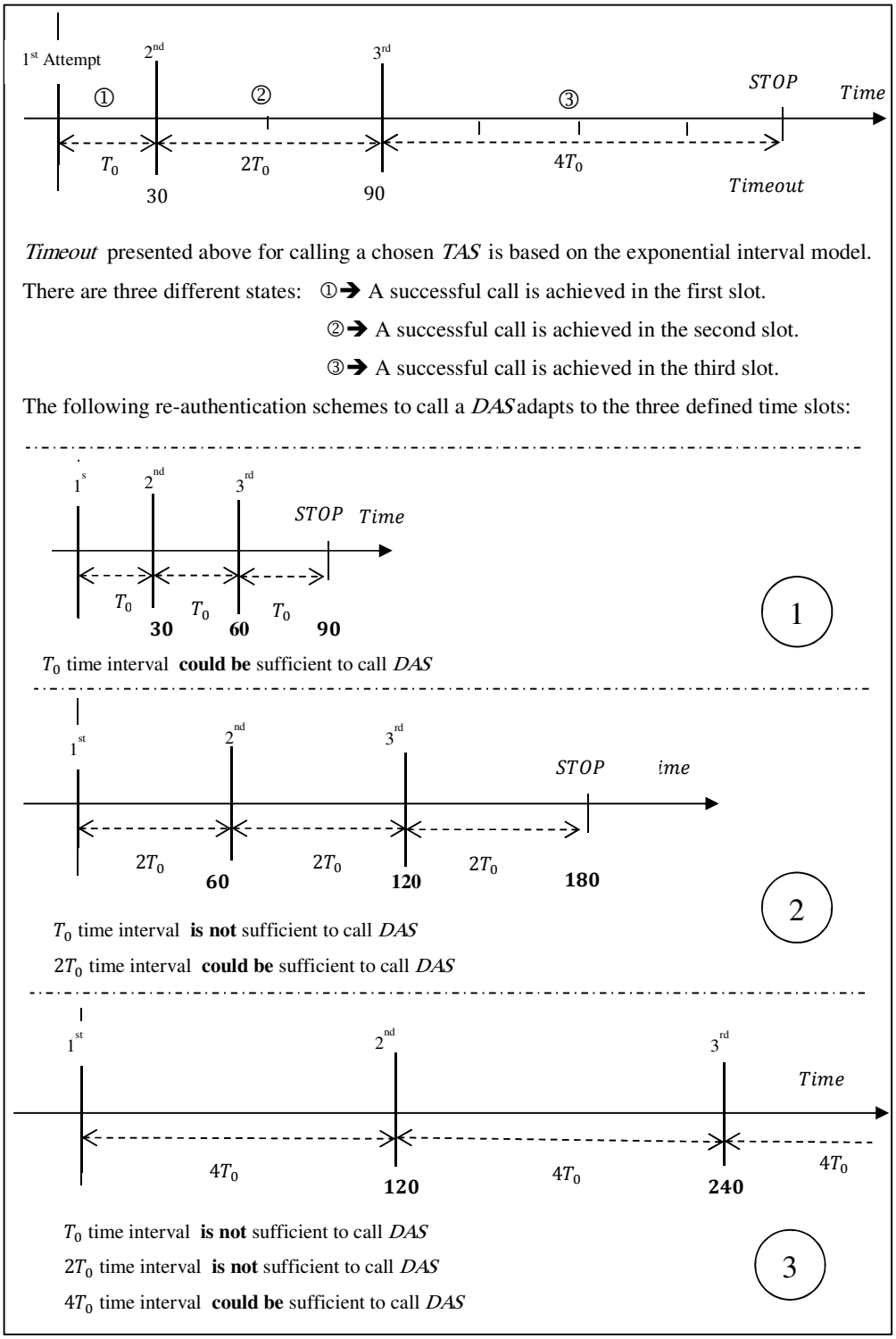


Figure 5-13: The description of different alternatives in the adaptable fixed interval model for calling *DAS*

Alternatively, in the multi-level security architectures where *DASs* are integrated, the adaptable fixed interval model is adopted for the sake of efficiency and optimisation since calling a DAS unlike other servers is associated with the prior *TAS* invocation according to SSAM specifications (i.e. *DASs* and *TASs* are deployed in the same server).The proposed adaptable fixed interval mode is similar to the normal fixed model but the time interval (T_x) can be amended, based on a state of prior successful calling for a particular *TAS*, every time a *DAS* invocation is initiated. This state in this regard indicates the waiting time slot within *Timeout* that a successful call of *TAS* occurs. Also, it is normally calculated based on the unit of waiting time (T_0) and the current number of attempts in the base-2 exponential interval model obtained from the previous successful connection to a certain *TAS*. The three different cases of adaptable re-authentication *DAS* schemes are clarified in Figure 5-13.

5.3.4 The Network Configurations

In addition to security-related SSAM configurations mentioned in the previous section, this section presents all necessary network components of MANETs that need to be defined and then justified in order to conduct performance testing of the SSAM model using simulation. Thus, the network model configuration mainly includes the mobility model, traffic model, transport, routing, and MAC protocols and churn model. Each has its own settings that will be detailed in the following subsection.

5.3.4.1 The Mobility Model

Node mobility is a very common feature in MANETs and this feature normally leads MANETs to suffer from network partitioning and unstable topologies. According to a MANET domain being tackled (e.g. human, vehicles, etc.), node mobility can also take several forms in terms of direction, speed, path shapes,

pause time, etc. There are various popular mobility models in MANETs such as Random Waypoint, GaussMarkov, Manhattan Grid and Reference Point Group Mobility models (Camp *et al.*, 2002). In spite of numerous existing mobility models in MANET literature, the Random Waypoint Mobility model (RWM) is still regarded as the most widely used mobility model. As stated in Kurkowski *et al.* (2005), the authors have found that the sixty four percent (64%) of total simulation papers were using the Random Waypoint Mobility model (RWM) in creating their simulation scenarios (see Figure 5-14).

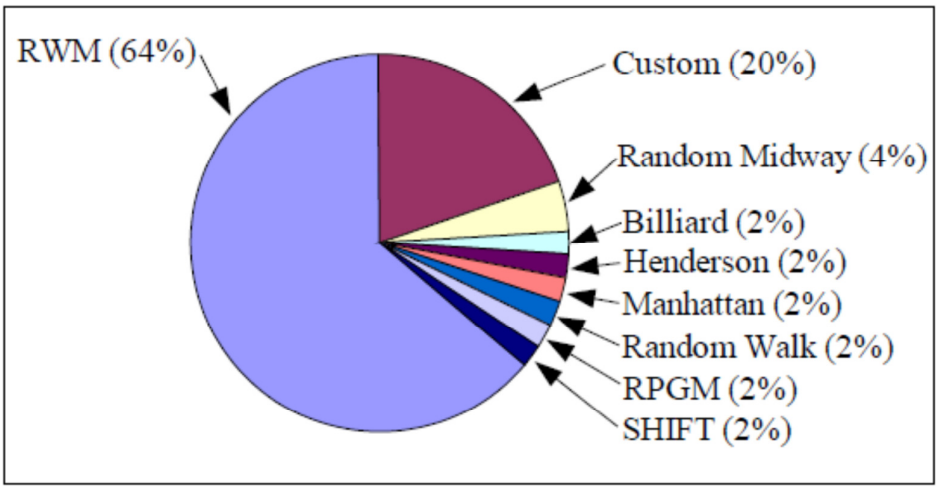


Figure 5-14: Mobility model usage from the study of Kurkowski (2005).

The RWM model is based on two key parameters, speed and a pause time between changes in direction. In this model, each node starts waiting in one location for a certain period of time (i.e. called a pause time). After pause time finishes, the node selects a random destination in the simulation area and a speed which is uniformly distributed between [MinSpeed, MaxSpeed]. The node herein moves towards the new destination at the predefined speed and then stops at destination for a period of pause time to repeat the whole scenario again. Therefore, the Steady-state Random Waypoint Mobility model (SRWM), is chosen for this study, which is a modified version of the normal RWM. This particular version enables nodes to begin in the steady-state distribution of the RWM (i.e. to resolve the problem of having zero speed, as addressed in Yoon *et*

al. (2003)) and it also facilitates analysing a simulation scenario from time zero, without considering initialisation bias incurred by initial node movement (Navidi and Camp, 2004; Navidi *et al.*, 2004).

Besides, "BonnMotion" (Aschenbruck *et al.*, 2010) is a Java-based package developed to generate mobility trace files to simulate and analyse many mobility models (e.g. Random Waypoint, Manhattan Grid, etc.). Also, those generated files can usually be exported easily into simulators. This is in order that each node can have its own trip (i.e. its positions and movements) during simulation runs. Therefore, this tool is used to create SRWM trace files for all nodes (see section A.1 in *Appendix A* for more details about the configuration and implementation of "BonnMotion" tool).

To conclude, the parameters of the selected SRWM model for this study are set to comply with human mobilisation (e.g. walking, jogging, etc.) especially in terms of speed and pause time. Thus, node speed can be between [0.5 m/sec, 1.5 m/sec] following the speed of human walking based on Carey (2005) and TranSafety (1997), while pause time is between 30 and 90 seconds. Since having different configurations for mobility is out of the scope, it is proposed that the this mobility scenario can be characterised as a base mobility scenario so that it can be compared with other new scenarios tackling higher values in their mobility parameters. Also, note that SSAM in this project can be adapted to any other mobility model that is considered appropriate by any researcher.

5.3.4.2 The Traffic Model

Apart from traffic stemming from interaction between server and user nodes in the SSAM model, the amount of additional traffic herein is intended to be generated in order to create a situation closer to the real network traffic during running the simulation of SSAM. Although, there are many traffic models (e.g. ON/OFF Pareto, ON/OFF Exponential, etc.) that can be exploited in MANETs (Giannoulis *et al.*, 2009; Pal *et al.*, 2011), the Constant Bit Rate (CBR) traffic model is

selected to produce traffic for this simulation similar implementations can be found in Larafa and Laurent (2011). This is because SSAM is developed to accommodate general-purpose context not only for a specific domain of MANET applications or service providers, such as multimedia streaming or data services. On the other hand, CBR traffic model is acknowledged as a very popular traffic model used in network simulation. It normally produces traffic at a predefined constant rate along with some randomising fluctuation in the inter-packet departure interval. The type of traffic generated by the CBR model can usually match the properties of real data service traffic. In this particular traffic, the data rate and the delay remain constant during the packet transmission (Pal *et al.*, 2011). For this simulation, two CBR sources are dedicated to cover the whole simulation area; each one is located in the middle of the half area (as shown as a triangle shape "Traffic Source" in Figure 5-2). Additionally, these traffic sources are configured with a rate 2.5 KB/sec and a packet size 512 Bytes and each one generates and sends a traffic volume of 25 Kbytes for every user node attempting to connect. Both CBR generators in this experiment are hypothetically regarded as service providers (i.e. traffic sources.) invoked or triggered only by successfully authenticated nodes.

5.3.4.3 The Transport Protocol

UDP (Postel, 1980) and TCP (Postel, 1981) are the most dominant transport protocols employed by networking applications. In contrast to UDP, TCP offers network-level services with reliable delivery. This reliability is implemented by TCP via sending packets and waiting for confirmation from destination about successful delivery of those packets. In case, no reply is acknowledged from the destination within a specific period of time at that point the packet must be resent. As such, in an environment like MANETs where packet loss is quite frequent having limited bandwidth or physical-layer congestion, using TCP can bring about a high increase in traffic owning to either the loss of the acknowledgement or original packet entailing more retransmission. This can clearly lead to

conditions where the MANET becomes jammed, making the TCP protocol present worst case performance. For that reason, it can be argued that by evaluating the effectiveness of jamming under each protocol, UDP rarely exhibits congestion comparing to TCP. On the other hand, UDP is lightweight and connectless and these features make it preferable in the light of MANET limitations. Besides, a number of works, for example Luo *et al.*(2004), Raghani *et al.*(2006) and Salmanian *et al.* (2010), make use of the UDP protocol for implementing their security model. Therefore, only the UDP transport protocol is adopted in this performance experimentation.

5.3.4.4 The Routing Protocol

As described in Section 2.2.2, there is a variety of routing protocols exploited mainly to establish a multi-hop routing infrastructure for MANETs. Proactive and reactive unicast routing protocols are the two large categories used in this field (Chlamtac *et al.*, 2003; Conti and Giordano, 2007). Proactive routing protocols are table-driven protocols which are based on legacy Internet distance-vector and link-state protocols such as the Optimized Link State Routing (OLSR) protocol (Clausen *et al.*, 2003) and the Destination-Sequenced Distance-Vector (DSDV) routing protocol (Perkins and Bhagwat, 1994). In this particular category, the routing table in every node is maintained and periodically updated to reflect the current state of links or paths with other nodes in the network. While, reactive routing protocols are distinguished with an on-demand attribute by finding the route to a destination only when required such as Ad hoc On-Demand Distance Vector (AODV) routing protocol (Perkins *et al.*, 2003) and Dynamic Source Routing (DSR) protocol (Johnson and Maltz, 1996). In the route discovery process, the route request is typically launched by the source node. Once a route is formed, it is retained by a route maintenance procedure until either the destination becomes unreachable or the route is no longer utilised.

The AODV protocol (Perkins *et al.*, 2003) is considered one of the most well-known on-demand ad hoc routing protocols used in the research community and also investigated extensively by the Internet Engineering Task Force (IETF) and National Institute of Standards and Technology (NIST) (Broch *et al.*, 1998; Royer and Perkins, 2000; Perkins *et al.*, 2003; Chakeres and Belding-Royer, 2004; Rahman and Zukarnain, 2009; Gupta *et al.*, 2010; Klein-Berndt, 2010). This routing protocol is identified by two main features: (1) using on-demand broadcast discovery mechanism to learn about the network topology and (2) using a sequence number to guarantee that routing information is up-to-date. The path discovery is performed whenever a node needs to communicate with another, and only if it has no routing information of the destination in its routing table. Path discovery begins by broadcasting a route request control message "RREQ" which disseminates in the forward path. In the case of a neighbour having information about the route to the destination, it responds with a route reply control message "RREP" that propagates in the backward path. Otherwise, the "RREQ" will be re-broadcasted by the neighbour. The process will be repeated by other intermediate nodes until finding the destination node or meeting a limit criterion on "RREQ" dissemination which is configured by a source node (i.e. a mechanism known as "Expanding Ring Search"). To detect that neighbours are still in range of connectivity, AODV makes use of heartbeat control messages, called "Hello" messages, in order to maintain up-to-date paths. When, for any reason, a link is broken, an unsolicited route error control messages "RERR" is generated and broadcasted to all active source nodes that are currently using this link. Once the "RERR" is delivered to a source node, the node may re-initiate the path discovery process if necessary.

Eventually, despite the fact that there are several routing protocols in MANET that could be examined for this study, it is decided to involve AODV in the routing infrastructure of MANETs. This routing protocol is broadly used and there are a number of security models taking advantage of this particular protocol , such as Hadjichristofi *et al.* (2005a) Raghani *et al.* (2006), Al-Bayatti *et al.* (2009) and Larafa and Laurent (2011). On the other hand, using this protocol is meant to be

an illustrative example of a routing element contributing to the overall proposed approach of this study. Note that SSAM would be configured to use other routing protocols if relevant to a study.

5.3.4.5 The MAC Protocol

For establishing node-to-node (one-hop) communication in MANETs, there is a variety of wireless technologies that can benefit, such as, Bluetooth, Wi-Fi, Infrared-IrDA. However, the IEEE 802.11 (IEEE802.11, 1997) current wireless standard (i.e. MAC protocol) for interconnection is acknowledged as the most extensively used standard in MANET research community. Also, this particular standard provides an ad hoc mode connection that can suit MANETs and it is characterised by high-speed (54Mbp) and medium-range attributes. For this reason, The MAC protocol is included in the simulation model (Cocorada, 2008; Bredel and Bergner, 2009) of the IEEE 802.11g standard (IEEE802.11g, 2003). The configurations of the MAC layer is 150 meter fixed radio range for each node using a path loss radio model for propagation.

5.3.4.6 The Churn Model and Network Scenarios

MANETs are distinguished by their dynamic topologies. These dynamic topologies clearly stem from the features of node mobility and churn which are discussed thoroughly in Chapter 2. However, it is noticeable that, in the MANETs simulation field, most security proposals, which are simulated for testing the performance, only consider mobility with a fixed node density (i.e. a specific number of nodes) and ignore the situation of having volatile (i.e. growing and shrinking) MANETs. In other words, the network scenarios of those proposals are normally generated from the assumption that all MANET nodes come to existence all at once and are randomly distributed in the simulation area where those nodes start roaming based on the predetermined mobility model (e.g. the

RWM, etc.). Also, there is no consideration for the state of a node that leaves its network even though node churning in particular circumstances is justified and important to be acknowledged. Unlike Peer-To-Peer (P2P) networks regarding churn modelling (Stutzbach and Rejaie, 2006; Herrera and Znati, 2007), there is a lack of churn models for MANETs because there is no real and fully installed MANET application to rely on and to derive a model of churn behaviour for a particular use of MANETs. The author believes that node churning would have an impact on the availability and performance of the whole networking system due to incurring more broken links. Hence, node churn should arguably be taken into account for more accuracy and validity of simulating the real environment of MANETs.

Besides, since this study targets large-scale MANETs, the SSAM model is investigated under three different sizes of node population, 100 nodes, 250 nodes and 500 nodes within a space of 1500 m^2. However, in contrast to other typical works, those different populations refer to the total number of nodes that would participate in the network during the network's lifetime (i.e. total simulation time). In other words, node density varies over the time through having few nodes at the beginning and evolving until reaching a total population predefined in the experiment.

Therefore, for this study two main distinct scenarios are proposed to describe and represent different levels of dynamism in MANETs: (1) *the moderately dynamic topology* (i.e. called "*No-Churn*" as the churn rate is zero) and (2*) the highly dynamic topology* (i.e. called "*Churn*"). The goal of the two particular scenarios is arguably to scrutinise and understand the basic effects of various node densities (i.e. instigated by dynamic population with/without churning) in MANETs on availability and performance of SSAM. On the other hand, to the best of the author's knowledge, it is considered that this is the first attempt to involve the notion of node churn (i.e. a churn behaviour model) in this context of MANET performance and security. The author believes that this churn model will open a new direction incorporating with other metrics of interest (e.g.

performance, security, management, etc.) in dynamic MANETs for future research. This churn model mainly consists of three parts, node joining, leaving and lifetime. Every part can be modelled and configured differently.

For the case of node joining, a join rate (λ) can be defined as a node arrival rate. This rate can be variable or fixed depending on certain characteristics of context being undertaken in MANETs (i.e. rescue mission, education environment, etc.). Since no empirical results are available to represent joining behaviour (i.e. arrivals) of nodes in MANETs, a joining rate (λ) can be modelled to typically follow a standard distribution. For example, stationary or non-stationary Poisson process distributions where the inter-arrival times between joining nodes are independent exponential are widely utilised by authors such as Yang *et al.* (2010) and Papadopouli *et al.* (2005). These two works have discussed employing these distributions in modelling user arrival to connect to access points of the Wi-Fi network. However, the communication models of MANETs and Wi-Fi networks have common features in term of wireless technologies and mobility; the author alleges that these distributions can be employed for modelling node arrivals for MANETs. This approach can be considered as a starting point for further amendments and also a new direction for future research. For model simplicity and result stability, only the Stationary Poisson Process (SPP) distribution is adopted in this study (i.e. fixed joining rate $\lambda = 1$ node/min). Figure 5-15 describes how to generate arrival times for nodes joining using SPP.

Pre-condition and Parameters:

t_0 = Starting Time

N = Total Number of Nodes

Arrival Rate = $\lambda > 0$

Inter-event times: $A_i = t_i - t_{i-1} \sim$ exponential with mean $\frac{1}{\lambda}$

-stationary Poisson Process Algorithm (t_0 , N)

1. $t_{i-1} = t_0$
2. For $i = 1$ To N Do
3. Generate $U \cong Uniform\ (0, 1)$ independently uniformly distributed.
4. Return $t_i = t_{i-1} - \frac{ln\, U}{\lambda}$
5. End For

Note that $-\frac{ln\, U}{\lambda}$ is the desired exponential variate with mean $\frac{1}{\lambda}$

Figure 5-15: Stationary Poisson Process (SPP) Generator

Correspondingly, in the case of node leaving or churning, a MANET node would leave the network for several reasons, for example service unavailability, power failure or willingness. In accordance to the model assumptions, power issues are out of the study scope. Hence, as the SSAM model generally offers security service model and needs to be tested under a particular churning condition, node churn can be realised by two scenarios so either (1) when a node fails to be authenticated or (2) when a node is willing to leave voluntarily even if it is authenticated. The former scenario of node leaving is different from the latter one by the type of a reason making a node leave the network (i.e. compulsory and voluntary conditions).

Additionally, node lifetime in the network is a very important aspect that incorporates with the leaving condition especially in the case of voluntary leaving.

Due to that fact that there are very scarce empirical studies in node lifetime schemes in dynamic MANETs, a lifetime scheme is proposed in this study to tailor the simple normal distribution function in order to alleviate the stochastic nature of simulation and facilitate result reproducibility. The average lifetime in any networks varies, however the empirical studies in Yung-Chih *et al.* (2012) and Hutchins and Zegura, (2002) are relied on to decide about the value as a point of reference. These two studies have investigated a session length (i.e. lifetime) for users connecting to the Wi-Fi network on a university campus and they have found that the average of session length is approximately 10 minutes (i.e. 600 sec). Accordingly, the standard normal distribution function with a mean average (600s ± △200s) is applied for the lifetime scheme in this study. In addition, the random selected the 50% of successfully authenticated nodes takes that lifetime scheme into consideration and the rest of nodes stay until the end of simulation in order to keep MANETs up running.

In conclusion, the two key scenarios that are taken into account in this experimentation study are to create different settings of a node density in MANETS so as to evaluate performance of SSAM under different conditions. With three different volumes of population, the first scenario, the so-called *moderately dynamic topology*, can be accomplished only by a growing MANETs without churning ("*No-Churn*"). While, the second one, the so-called *highly dynamic topology* along with the same three settings as in the first one can be applied by using growing MANETs with node leaving ("*Churn*"). This node leaving can be achieved by either having unsuccessful node authentication or having a certain lifetime scheme employed as explained above.

5.4 Experimental Design

This experimental design consists of defining (1) the key performance and communication cost metrics, (2) the number of experiments based on the number of factors, their levels and also the final experimentation parameters and (3) the number of replications of each experiment. This experimental study basically aims

to produce comparative results in order to determine the best alternative out of various scenarios according to different criteria (e.g. security, performance, communication cost, etc.). This will be discussed in the next chapter.

5.4.1 Performance & Communication Metrics

Since the main goal of this study is to validate the performance of the SSAM model with different settings and under distinctive scenarios, there are a number of common indicators that can be measured to reflect the performance of any networking system, for example latency, throughput, failure rate, etc. (Jain, 1991). Along with performance metrics, the cost measurement in terms of communication load and resource consumption must be acknowledged because the SSAM model is developed to be deployed in MANETs where the networks are often characterised with limited resources (i.e. bandwidth, power, storage and processing). Hence, it is important to take into consideration this measurement. All metrics identified in this regard are listed below:

- The ***Success Ratio*** measures the ratio of the number of successful authenticated node over the total size of node population that join the network during the simulation time.

$$\textbf{\textit{Success Ratio [\%]}} = \frac{Successful\ Authenticated\ Nodes}{Total\ Node\ Population} \times 100$$

- The ***Failure Frequency*** estimates the number of failures (i.e. unsuccessful authenticated nodes) that occur during a time unit.

$$\textbf{\textit{Failure Frequency [Node/ Time Unit]}} = \frac{Unsuccessful\ Authenticated\ Nodes}{Time\ Unit}$$

- The ***Round Trip Time (RTT)*** refers to the average latency for successfully getting the credentials for a user node. As the SSAM model has different

authentications and server architectures, RTT is chosen to reflect time cost. RTT is calculated as the average time when the first message of the authentication protocol calls a particular server architecture until the time of completing calling when a user node obtains its credential. Also the RTT standard deviation (***RTT-STDev***) is considered to demonstrate the variance of the RTT.

$$\textbf{\textit{Round Trip Time [Time Unit]}} = T_{Recieve} - T_{Send}$$

- The ***Communication Overhead*** expresses the total amount of traffic (i.e. security communication messages) that is required to enable a joining new node to have successful authentication and obtain its credentials (i.e. membership certificate).

$$\textbf{\textit{Communication Overhead [Byte]}} = \sum Msgs_{Recieve} + \sum Msgs_{Send}$$

There are other supporting metrics (e.g. certificate type and count Acquisitions) for evaluating effectiveness productivity and manageability of security architectures in SSAM that will elaborated in the next chapter.

5.4.2 Test Cases & Experimental Settings

As discussed previously (see Section 5.3.2), the model implementation of SSAM is established upon three key components (i.e. authentication protocols, types of security server and calling strategies in certain hierarchical server arrangements). These components can be considered as the main factors that need to be investigated in the performance testing experiment. Therefore, depending on levels of each factor, as shown in Table 5-5, there are 18 different security test cases that are proposed for carrying out effective performance experimentation and evaluation of this model under particular scenarios (i.e. three population sizes with\out churn). Also, these security test cases are grouped into three sets

according to the type of authentication protocol. On the other hand, each factor combination (i.e. a test case) in this context represents a distinctive security architecture. In the experiments, each security architecture is denoted with a unique code (e.g. *1WP_CAS*, etc., see Table 5-3) so that it can be identified in an easy manner throughout experimentation and evaluation phases.

The Set of Subject Cases *'+':Included , '–': Not Included, 'n': Numbers of Servers*						**The Test Case Code "Security Architecture"**
Authentication Protocol Types	**Server Security Architecture Types**			**Calling Strategies**		
	Central Authority Server (CAS) $[n = 1]$	**Threshold Authority Servers (TASs)** [k out of n n=3,n=6]	**Delegated / Distribute-Central Authority Servers – (DASs) / (D\CAS)** $[n = 6]$	**All At Once – AAO**	**In Priority Sequence – IPS**	
One Way Pass + (1WP) *1 control message 1 data message*	+	–	–			*1WP_CAS*
	–	+	–			*1WP_TAS*
	+	+	–	+	–	*1WP_CAS_TAS_AAO*
				–	+	*1WP_CAS_TAS_IPS*
	–	+	+			*1WP_TAS_DAS*
	+	+	+	+	–	*1WP_CAS_TAS_DAS_AAO*
				–	+	*1WP_CAS_TAS_DAS_IPS*
	–	–	+			*1WP_D\CAS*
Two Way Pass + (2WP) *2 control messages 1 data message*	+	–	–			*2WP_CAS*
	–	+	–			*2WP_TAS*
	+	+	–	+	–	*2WP_CAS_TAS_AAO*
				–	+	*2WP_CAS_TAS_IPS*
	–	+	+			*2WP_TAS_DAS*
	+	+	+	+	–	*2WP_CAS_TAS_DAS_AAO*
				–	+	*2WP_CAS_TAS_DAS_IPS*
	–	–	+			*2WP_D\CAS*
Three Way Pass + (3WP) *3 control messages 1 data message*	+	–	–			*3WP_CAS*
	–	+	–			*3WP_TAS*
	+	+	–	+	–	*3WP_CAS_TAS_AAO*
				–	+	*3WP_CAS_TAS_IPS*
	–	+	+			*3WP_TAS_DAS*

	+	+	+	+	–	*3WP_CAS_TAS_DAS_AAO*
				–	+	*3WP_CAS_TAS_DAS_IPS*
	–	–	+			*3WP_D\CAS*

Table 5-5: The security experiment test cases identified by their key codes.

Eventually, in this performance experimentation, all the 18 security test cases are tested under two distinct types of scenarios, "*No-Churn*" and "*Churn*". As addressed in the Churn model Section (5.3.4.6), each type consists of three different population settings (100, 250, 500 Nodes). Hence, there are 108 total test cases (18 security test cases × 2 scenario types × 3 population sizes) for the experimental design. Also, as the study uses simulation, each test case must be replicated for specific number of times (i.e. *nr* = 30) in order to obtain statistically reliable results. The issue of selecting the required number of replications will be elaborated in the next section. Furthermore, the final experimental settings (i.e. controlled variables) are summarised in Table 5-6 and are mostly based on the pervious discussion about model configurations of network and SSAM models. These settings will be used to initialise the simulation environment.

Parameter	Value
Simulation Area	1500 × 1500 m^2
Transport Protocol	UDP
Routing Protocol	AODV
MAC Protocol	IEEE805.11g
Range	150 m
Radio Propagation Model	Path Loss Model
Mobility Model	The Steady-state Random Waypoint Mobility
Node Speed	(Mean) 1 ± △ 0.5 mps
Node Pause time	(Mean) 60 ± △ 30 Sec
Traffic Model	CBR = 2.5KB/sec Two sources (Service Providers), Packet Size = 512Byte Total traffic volume for each connection = 25KB
Simulation Time	7200s, 16400s, 35000s

Table 5-6: The Experimental OMNeT++ Settings (system parameters)

5.4.3 Number of Replications

The simulation model of this study can be characterised as a terminating deterministic simulation, however this model is also expected to have some stochastic simulation outputs stemming from dynamic topologies and error-prone wireless medium of MANETs. Due to these reasons, it is important to decide the number of replications that is necessary for accuracy and reproducibility purposes. Based on Robinson's (2004) approach, the confidence interval method is chosen as a popular and simple statistical means which helps to find the number of replications required in the simulation to attain better accuracy in the experiment results. Normally, this particular method is exploited to indicate the degree of accuracy in the mean being estimated. For a more accurate estimate, the confidence interval must be small to achieve the desired level of accuracy. Therefore, the confidence interval can be applied to simulation output data by carrying out several replications (samples) until the interval becomes sufficiently narrow to satisfy the desired level of accuracy (i.e. the percentage deviation of the confidence interval on either side of the mean).

Through analysing simulation results, the confidence interval (*CI*) can be estimated as follows:

$$CI = \bar{X} \pm t_{n-1,\alpha/2} \frac{S}{\sqrt{n}}$$

Where:

$\bar{X}$ = the standard mean of the output data from the replications
S = the standard deviation of the output data from the replications (see equation below)
n = the number of replications
$t_{n-1,\alpha/2}$ = a value from Student's t-distribution with $n-1$ degree of freedom and a significance level of $\alpha/2$

The equation for the standard deviation is:

$$S = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}}$$

Where: X_i = the result from replication *i*

The equation for the percentage deviation of the confidence interval on either side of the mean is:

$$STDevPRC_i\ [\%] = \frac{\bar{X}_i - CI_{lower}}{\bar{X}_i}$$

Where:

$\bar{X}_i$ = the cumulative mean average from the first until the replication *i*

CI_{lower} = the lower bound of confidence interval

In this study, the significance level (α) has been set equal to 0.05 which is commonly used. Also, a MS EXCEL spread sheet is created to calculate all necessary values depending on the formulas described before and accordingly these values are then plotted to study the required number of replication for each test case. However, since this experimentation study comprises several test cases as described in the previous section, it is difficult to replicate all test cases to obtain output data which is used to calculate the desirable number of replications. Consequently, two test cases (*3WP_CAS_TAS_DAS_AAO_100_nodes_Churn OR _No-Churn*) are selected to be involved in the process of choosing the number of replication. The reasons for this choice are several. Initially, it is argued that the security test case of *3WP_CAS_TAS_DAS_AAO* is unlike the other security test cases and includes the most of activities in the SSAM model which needs to be simulated: **(1)** *3WP* for handshaking relies on a fairly larger number of messages (i.e. four initial messages) which is greater than the other *1WP & 2WP* protocols; **(2)** this particular test case covers most of proposed server architectures: *CAS*, *TAS* and *DAS*; **(3)** the *AAO* strategy of calling incurs more traffic than the other strategies. Additionally, for simulation, there are two main distinct scenarios (i.e. *No-Churn* and *Churn*) that are proposed to refer to the dynamic nature of a

MANET topology and each one has three cases of a population size (i.e. 100, 250 and 500 nodes). Therefore, the first case of each scenario (i.e. the case of population has 100 nodes) is chosen which clearly exhibits more network partitioning than the other cases of 250 and 500 nodes. Eventually, these two specific test cases arguably would entail more variance in the output data of simulation experiments which is required to be reduced in order to achieve a better estimate of mean performance.

As described before, the key metrics in this experimentation study are the Success Ratio, the Round Trip Time (RTT), the Communication Overhead, the Failure Frequency and the Standard Deviation of RTT (RTT STDev) which are suggested to measure the performance and cost of applying the SSAM model on MANETs. A pilot experiment is conducted for 30 replications for each of two test cases considered. It is noticed that each different output collected from the particular replications in both cases shows less than 10% of the level of deviation within the calculated confidence interval as summarised in Table 5-7. This level (10%) specified can be considered to be satisfactory in this study (i.e. sufficient narrowness in the confidence interval), so the decision is to make use of 30 replications for every other test case in this study.

Output Data (Metrics)	**STDevPRC$_{i=30}$**	
	3WP_CAS_TAS_DAS_AAO 100 Nodes - Churn	***3WP_CAS_TAS_DAS_AAO 100 Nodes – No-Churn***
Success Ratio	7.42%	3.24%
Round Trip Time (RTT)	6.65%	5.26%
RTT STDev	2.56%	3.54%
Failure Frequency	3.61%	6.32%
Communication Overhead	2.72%	1.78%

Table 5-7: The percentage deviation of the confidence interval for 30 replications

For more details, the value of each metric with its replications are presented in tables and diagrams based on the test case. In this section, as shown below, only the tables and charts (i.e. Table 5-8 and 5-9 and Figure 5-16 and 5-17) of the success ratio for 30 replications for the predetermined two case scenarios (Churn and No-Churn) are displayed. The rest of metrics for the both cases can be found in Section B.1 (*Appendix B*). Eventually, apart from a satisfactory confidence interval, it is essential that the cumulative mean line is fairly flat in the convergence diagram as presented later.

Replication	Result: *Success Ratio*	Cum. mean average	Standard deviation	**Confidence Interval (95%)** Lower Interval CI_{lower}	Upper Interval CI_{Upper}	[%] Deviation STDevPRC
1	0.69	0.69	n/a	n/a	n/a	n/a
2	0.67	0.68	0.014	0.55	0.81	18.69%
3	0.69	0.68	0.012	0.65	0.71	4.20%
4	0.64	0.67	0.024	0.63	0.71	5.59%
5	0.53	0.64	0.067	0.56	0.73	12.90%
6	0.58	0.63	0.065	0.56	0.70	10.82%
7	0.65	0.64	0.060	0.58	0.69	8.72%
8	0.59	0.63	0.058	0.58	0.68	7.67%
9	0.73	0.64	0.064	0.59	0.69	7.62%
10	0.67	0.64	0.061	0.60	0.69	6.73%
11	0.67	0.65	0.058	0.61	0.69	6.03%
12	0.51	0.64	0.068	0.59	0.68	6.79%
13	0.62	0.63	0.065	0.59	0.67	6.21%
14	0.62	0.63	0.063	0.60	0.67	5.72%
15	0.6	0.63	0.061	0.60	0.66	5.36%
16	0.62	0.63	0.059	0.60	0.66	4.99%
17	0.65	0.63	0.057	0.60	0.66	4.67%
18	0.56	0.63	0.058	0.60	0.66	4.61%
19	0.69	0.63	0.058	0.60	0.66	4.45%
20	0.64	0.63	0.057	0.60	0.66	4.21%
21	0.63	0.63	0.055	0.61	0.66	3.99%
22	0.67	0.63	0.055	0.61	0.66	3.83%
23	0.68	0.63	0.054	0.61	0.66	3.70%
24	0.64	0.64	0.053	0.61	0.66	3.53%
25	0.57	0.63	0.054	0.61	0.65	3.50%
26	0.57	0.63	0.054	0.61	0.65	3.45%
27	0.74	0.63	0.057	0.61	0.66	3.55%

28	0.67	0.64	0.056	0.61	0.66	3.43%
29	0.65	0.64	0.055	0.61	0.66	3.31%
30	0.69	0.64	0.055	0.62	0.66	**3.24%**

Table 5-8: The Confidence Interval Method: Results of Success Ratio in the case of *3WP_CAS_TAS_DAS_AAO* - 100 Nodes *No-Churn* for 30 replications.

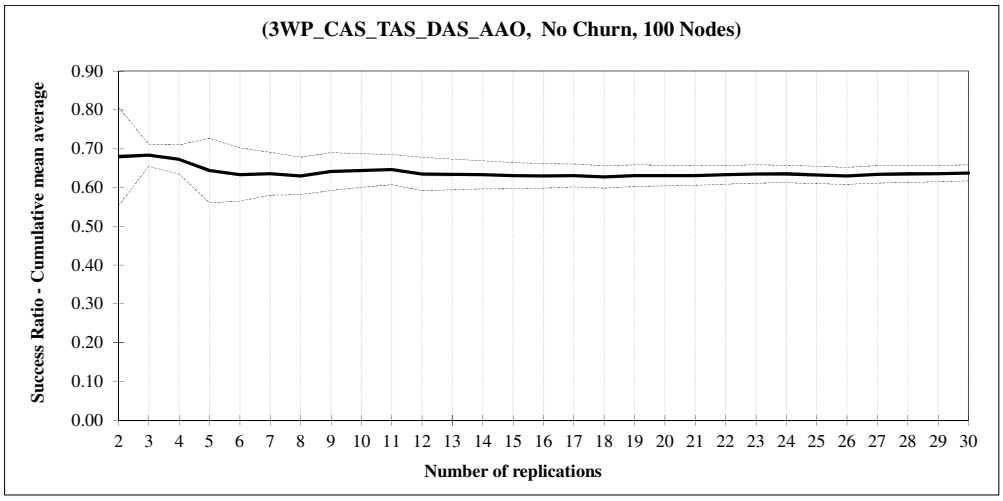


Figure 5-16: The plot represents the 95% confidence intervals and cumulative mean of success ratio metrics in the case of no node churning

Replication	Result: *Success Ratio*	Cum. mean average	Standard deviation	Confidence interval (95%) Lower Interval CI_{lower}	Confidence interval (95%) Upper Interval CI_{Upper}	Deviation *STDevPRC* [%]
1	0.37	0.37	n/a	n/a	n/a	n/a
2	0.42	0.40	0.035	0.08	0.71	80.42%
3	0.34	0.38	0.040	0.28	0.48	26.65%
4	0.37	0.38	0.033	0.32	0.43	14.07%
5	0.24	0.35	0.067	0.26	0.43	23.85%
6	0.27	0.34	0.068	0.26	0.41	21.22%
7	0.25	0.32	0.070	0.26	0.39	19.96%
8	0.33	0.32	0.065	0.27	0.38	16.67%
9	0.39	0.33	0.064	0.28	0.38	14.93%
10	0.25	0.32	0.066	0.28	0.37	14.58%
11	0.26	0.32	0.065	0.27	0.36	13.82%
12	0.2	0.31	0.071	0.26	0.35	14.64%

13	0.35	0.31	0.069	0.27	0.35	13.39%
14	0.29	0.31	0.066	0.27	0.35	12.39%
15	0.35	0.31	0.065	0.28	0.35	11.51%
16	0.3	0.31	0.063	0.28	0.34	10.73%
17	0.33	0.31	0.061	0.28	0.34	10.02%
18	0.29	0.31	0.059	0.28	0.34	9.48%
19	0.29	0.31	0.058	0.28	0.34	8.99%
20	0.25	0.31	0.058	0.28	0.33	8.82%
21	0.26	0.30	0.057	0.28	0.33	8.56%
22	0.25	0.30	0.057	0.28	0.33	8.38%
23	0.4	0.31	0.059	0.28	0.33	8.39%
24	0.3	0.31	0.058	0.28	0.33	8.02%
25	0.16	0.30	0.064	0.27	0.33	8.79%
26	0.33	0.30	0.063	0.28	0.33	8.44%
27	0.31	0.30	0.062	0.28	0.33	8.10%
28	0.35	0.30	0.061	0.28	0.33	7.83%
29	0.25	0.30	0.061	0.28	0.32	7.69%
30	0.32	0.30	0.060	0.28	0.32	**7.42%**

Table 5-9: The Confidence Interval Method: Results of Success Ratio in the case of *3WP_CAS_TAS_DAS_AAO* 100-Node Churn for 30 replications.

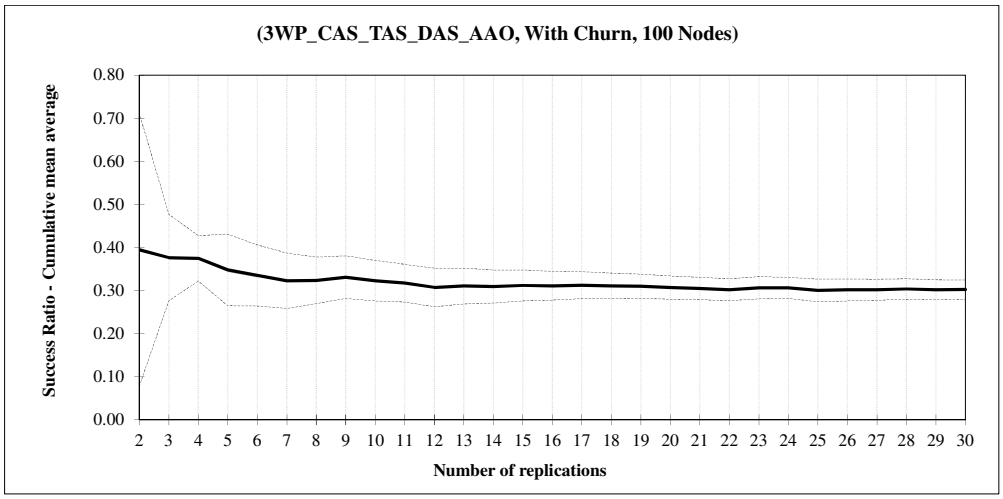


Figure 5-17: The plot represents the 95% confidence intervals and cumulative mean of success ratio in the case of having a node churning

5.5 Conclusion

This chapter first highlighted the model assumptions for this study. These assumptions were used to facilitate the implementation and experimentation of SSAM model designed in the previous chapter. A SSAM prototype was developed using the OMNET++ simulator which is considered one of the best simulation tools for MANETs. Furthermore, the development of SSAM prototype are completed by defining the MANET infrastructure and messages and creating the SSAM-relevant C++ classes. On the other hand, all vital SSAM and MANET –related configurations that is necessary for the prototype simulation were proposed in this chapter with their exhaustive explanation. The SSAM-related configuration deals with security settings for the SSAM model (e.g. security servers, authentication protocols), whereas the MANET–related configuration involves in the network features and settings (e.g. mobility, routing protocol, etc.). Finally, the experimental design was described by defining the main performance and communication metrics, the number of test cases and the required number of replications for each experiment. This approach can clearly help to properly prepare for simulation experiment.

The next chapter presents the analysis of results generated from conducting the required simulation experiments and security robustness assessments. It will underline the key findings of the whole proposed approach by applying it in three different real case scenarios.

Chapter 6: Result Analysis and Validation

6.1 Overview

This chapter discusses the analysis of experimental results generated from testing the performance and communication of the proposed security architectures in SSAM under different network scenarios (i.e. the *Churn* and *No-Churn* settings). The results of key measurements defined in the previous chapter for expressing performance and cost will be systematically interpreted and compared. These measurements concern with the success ratio, the failure frequency, the Round Trip Time (RTT), the standard deviation of RTT (RTT-STDev), the communication overhead and the multi-level security architecture productivity. In addition, the issue of security strength is evaluated for each security architecture, according to three security features: the single-point-of-failure (SPF) resilience and single-point-compromise (SPC) resilience and authentication protocol robustness. The employment of the methodological approach proposed in this study to MANET security evaluation is discussed for the SSAM model. On the other hand, three different scenarios (i.e. academia, rescue mission and military) which are suggested to validate the whole approach are described in this chapter. The "best" security architectures for each scenario will then be identified using a simple ranking approach using the achievement scoring (i.e. rating), reciprocal ranking weighting and weighted averaging methods. The results generated from this ranking system would facilitate the selection of the best alternatives suiting the demands of a given application for MANETs.

6.2 Experimental Results and Evaluation

After running the predefined simulation experiments using the OMNeT++ simulator with their required replications (i.e. 30 replications to obtain stable

result), output data was collected and analysed using the simple standard estimation (e.g. mean average, standard deviation, standard error, confidence interval, etc.). The evaluation presented in this chapter is based on this analysis. There are presented in full in *Appendix C*.

All necessary figures that illustrate these various results can be found in *Appendix B* apart from those figures used in this chapter. Based on the proposed security architectures tackling the different network scenarios, the following sections demonstrate the outcomes of the analysis of the performance metrics to examine the effectiveness and efficiency of the given architectures. These are: Section 6.2.1 presents the success ratio results; Section 6.2.2 presents failure frequency results; Section 6.2.3 presents the outcomes of Round Trip Time (RTT); Section 6.2.4 presents the results of the standard deviation of RTT ((RTT-STDev); Section 6.2.5 includes the result analysis of communication overhead, for each security architecture and network scenario being involved. Finally, Section 6.2.6 demonstrates the evaluation of certificate acquisition for the specific multi-level hierarchical server architectures (e.g. certificate types and certificate counts) to represent productivity and flexibility of those architectures.

6.2.1 Success Ratio

This section discusses the analysis of the success ratio for different security architectures performed under the two proposed MANETs scenarios (*Churn* and *No-Churn*). The higher success ratio a particular security architecture shows, the better the availability and robustness of the architecture. Availability improvement can contribute to the overall performance. Figure 6-1 shows overall success ratio results for the eight proposed server architectures under the *Churn* and *No-Churn* scenarios each of which includes the three distinct node populations (*100-Node*, *250-Node* and *500-Nodes,* etc.). In addition to this figure, there are a number of time-based charts in Section B.3.1, (*Appendix B*). These charts depict the success ratio changes over the simulation time in each case (i.e. categorised in three sets:

these are the One, Two and Three -Level secuirty architectures) to investigate and confirm differences among the tested architectures.

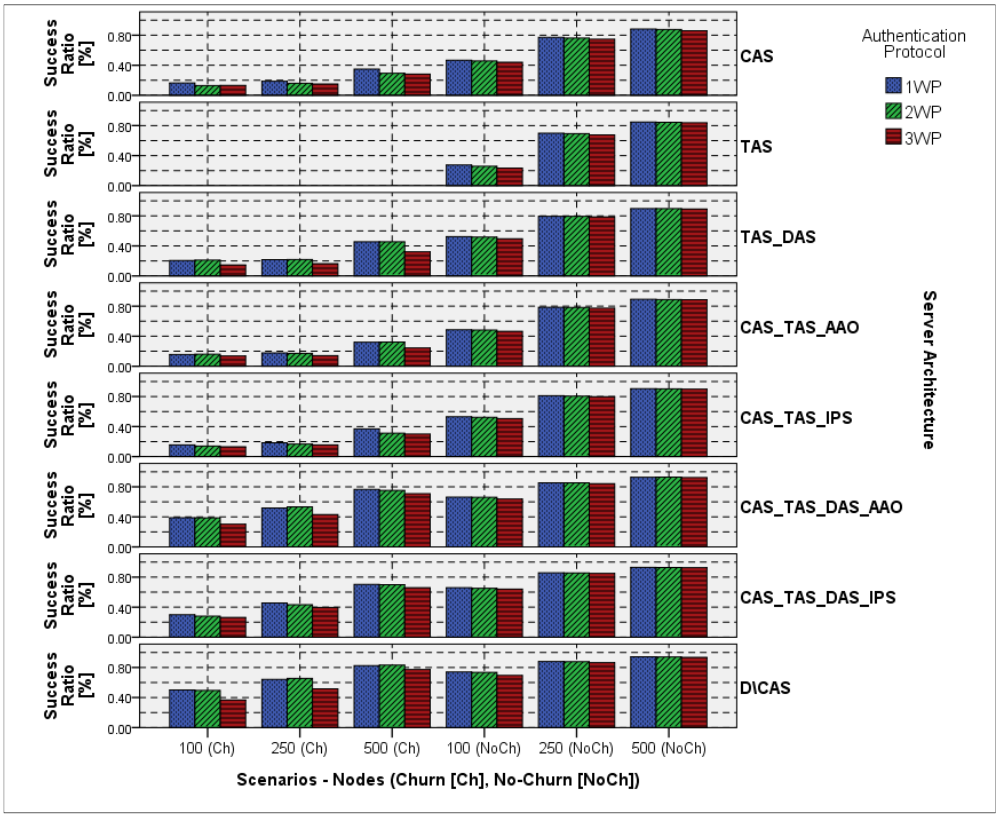


Figure 6-1: The Success Ratio Chart

Initially, in the *Churn* and *No-Churn* scenarios, most security architectures show a common pattern of growth in the success ratio when increasing the number of nodes joining the playground. This is due to the fact that as more nodes join, more connections can be established (i.e. increasing in node density) in the MANET routing infrastructure and therefore enhancing the server reachability. Also, it is important to remark that the success ratio of all security architectures in the *500-Nodes No-Churn* scenario takes the highest value without significant difference among the different architectures (approximately between 90%-94%) as the MANET nodes become fully connected.

On the other hand, regardless of the authentication protocol in use, the *TAS* architecture appears to be the worst server architecture in terms of availability in all considered scenarios. This is because every new joining node has only one option (a one-level server architecture) to secure multiple connections with at least three of the available security servers in order to be able to construct its membership certificate (i.e. using *TC* (*k =3* of *n=6*)), otherwise it fails. Thus, the availability of these connections is questionable, especially in a certain scenario where a MANET has a fairly high dynamic topology as displayed in Figure 6-1 (i.e. 0% ratio in the whole *Churn* scenarios as a result of all nodes leaving and lower rates in the other *No-Churn* scenarios). The cases of *D\CAS*, *CAS_TAS_DAS_AAO* and *CAS_TAS_DAS_IPS* reveal the greatest success rate throughout all network case scenarios. This due to the fact that these particular architectures offer more than three different alternatives of servers (i.e. either 6 distributed server replicas as in *D\CAS* or three-level hierarchical servers as in *CAS_TAS_DAS_AAO* and *CAS_TAS_DAS_IPS*) for new nodes to initiate authentication in order to obtain their membership certificates from accessible servers. Therefore, any node wishing to join is most likely to be authenticated in theses specific security architectures. However, the best availability under all case scenarios is shown by the *D\CAS* architecture which represents the case of having multiple server replicas for the same authority. Both *CAS_TAS_DAS_IPS* and *CAS_TAS_DAS_AAO* architectures come second in terms of their success ratio outcomes. *CAS_TAS_DAS_AAO* outperforms *CAS_TAS_DAS_IPS* with a 10%-5% increase particularly in the *Churn* scenario across various node populations. This makes it much more adaptable to the situation of having highly partitioned networks. The success ratios for the rest of cases (*CAS*, *CAS_TAS_AAO*, *CAS_TAS_IPS* and *TAS_DAS* are approximately the same (i.e. negligible differences less than 5%) throughout various network settings. Nevertheless, these particular architectures have different server hierarchies. Also, it is important to note that the *TAS_DAS* architecture shows higher authentication success among them, especially under the *Churn* scenario.

It is clear that the success ratio is not affected considerably by the authentication protocol being used except for few cases of the *3WP* protocol being exploited in certain server architectures (*TAS_DAS, CAS_TAS_AAO, CAS_TAS_DAS_AAO* and *D\CAS*), especially under the *Churn* scenario. This difference shows a decrease between 5% and 10% compared to the other authentication protocols, as seen in Figure 6-1 and it can be confirmed by the related time-based charts in Section B.3.1 (*Appendix B*). This exception can be explained by the fact that the *3WP* protocol needs to exchange several messages to establish an authentication connection (at least four messages per server connection). Besides, in those certain server architectures using *3WP* protocol, a new node often broadcasts multiple server calls simultaneously (open multiple connections). Hence, this would cause more traffic congestion and processing which leads to much more loss and long delays, especially in a highly partitioned network such as the *Churn* scenario. Consequently, in this condition, any new node would have a less chance of getting a successful authentication.

The success ratio is a very important criterion to evaluate the effectiveness and productivity of any given security architecture for MANETs. It is important to recognise that the security architecture which takes advantage of multiple servers and different levels (i.e. more than two-level) of a server hierarchy like *D\CAS*, *CAS_TAS_DAS_AAO* and *CAS_TAS_DAS_IPS*, demonstrates a better success ratio.

6.2.2 Failure Frequency

The failure frequency along with the success ratio can be incorporated to reflect both reliability and availability of the security architectures. The failure frequency measures the relative mean average failures occurring within a simulation time unit (i.e. minute). The lower the failure frequency, the more reliable the architecture. In the *Churn* and *No-Churn* network scenarios, it is recognisable, as shown in the Figure 6-2, that failure frequency for most server architectures at any

authentication protocol being used declines when increasing the size of node population. This is due to improving node density even though occasional node churning (in certain cases) and node mobility render disconnection between nodes.

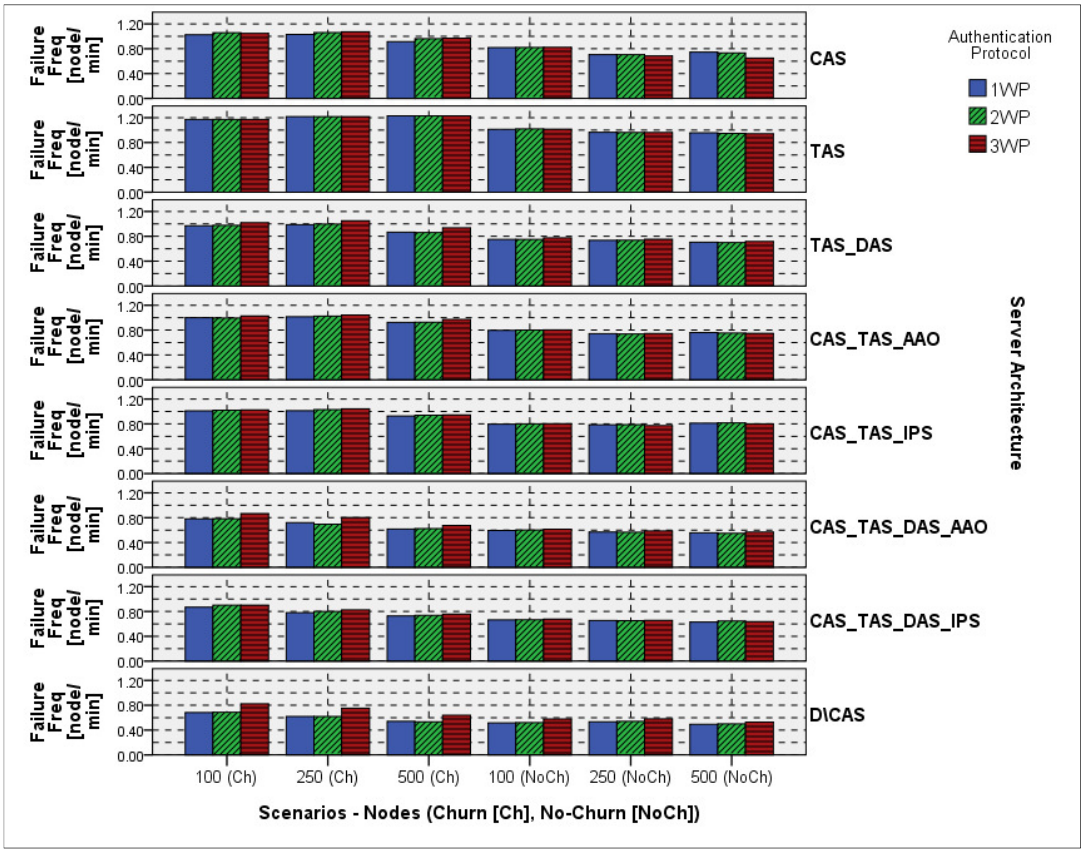


Figure 6-2: The Failure Frequency Chart

In addition to demonstrating worst success ratio, the *TAS* architecture also tends to have higher failure rate in both network scenarios (0.98 – 1.2 failures/min), especially in the *Churn* scenario. As discussed in Section 6.2.1., this is because the requirements of this architecture are restricted (i.e. only one option) and not easy to achieve particularly in an unpredictable MANET topology (securing multiple server connections) where failure frequency can be increased.

However, without considering the authentication protocol, specific architectures like *CAS*, *CAS_TAS_AAO*, *CAS_TAS_IPS* and *TAS_DAS* show

insignificant differences between each other in failure frequency (less than 0.04 failure/min) and follow the same trend, as shown in Figure 6-2. These results can be validated in the relevant time-based charts at the Section B.3.2. (*Appendix B*). This group of architectures show lower failure frequency than *TAS* with a variances 0.2 failure/min over the *Churn* scenario and 0.3 failure/min over the *No-Churn* scenario. The *CAS_TAS_AAO*, *CAS_TAS_IPS* or *TAS_DAS* architectures, unlike the *CAS* architecture, depend on a distributed server architecture (i.e. two alternatives) that aims to alleviate failures. However, they show approximately the same failure rate as the centralised *CAS* architecture. This may be justified by the fact that the *TAS* architecture which already suffers from higher rate of failures is already integrated in these architectures along with *CAS* or *DAS*. This would arguably restrict any improvements in failure frequency to the failure frequency of the *CAS* architecture.

The rest of the architectures, *D\CAS*, *CAS_TAS_DAS_AAO* and *CAS_TAS_DAS_IPS* can be considered as another group. This second group has the least failures regardless of the type of authentication protocol being utilised and the scenario being conducted .This due to a variety of multiple distributed server alternatives to connect. The *CAS_TAS_DAS_IPS* architecture among this group shows the higher rate of failure (0.66 – 0.82 failures/min), followed by the CAS_*TAS_DAS_AAO* architecture with a failure rate (0.55 – 0.7 failures/min). The *D\CAS* architecture is perceived as the best among them with the lowest failure frequency (0.5 – 0.65 failures/min).

Alternatively, similar to the success ratio results, the type of the authentication protocol being involved does not have a substantial role to contribute to the failure frequency except in a few cases. Most security architectures do not display any considerable differences among the three types of protocols mostly in the *No-Churn* scenario. However, there are few cases where the *3WP* protocol is used (e.g. *TAS_DAS*, *CAS_TAS_DAS_AAO* and *D\CAS*) and gives an increase (approximately 0.09 failures/min) in failure frequency when compared to the *1WP* and *2WP* protocols. This can be explained by the fact that those architectures

seem to overwhelm the network with large traffic originating from broadcasting several messages simultaneously (i.e. initial four messages like in *3WP*) and this would bring about more collision and loss in the network and cause more frequent failures.

6.2.3 Round Trip Time (RTT)

Round Trip Time (RTT) (i.e. sometime called a delay or response time) in any particular networking systems is important as this measurement is associated with system performance. In general, the lower the delay a network has, the better performance overall. As shown in Figure 6-3 regardless of the authentication protocol being applied, the different server architectures show different RTT across the two proposed scenarios. Some of the server architectures have to a certain extent a stable RTT in the *Churn* scenario like *CAS, D\CAS*, *CAS_TAS_AAO* and *CAS_TAS_IPS* while the rest display a moderate decline in RTT. Thus, an increase in the population size, in spite of node churning behaviour in some specific cases, plays an important role in improving RTT (i.e. reducing the time that should be spent for authentication) in most of the architectures by bringing more connections. Section B.3.3 and Section B.3.5 (*Appendix B*) show how RTT and node numbers change over time.

The *TAS* architecture obviously has no result for RTT in the *Churn* scenario because no successful authentication occurs during this particular scenario, (see Section 6.2.1). Additionally, this architecture which is based on a one-level server hierarchy reveals an average level of delays among other similar hierarchical architectures (e.g. *CAS* and *D\CAS* architectures) in the *No-Churn* scenario. This is due to the fact that every new joining node initially calls a specific number of available *TASs* and waits until it gets a sufficient number of partial certificates from those different servers (i.e. $k=3$ of $n=5$ servers). This is in order to produce a valid credential by threshold cryptography. However, this process may take much

longer time than other one-level hierarchical architectures that require only one response.

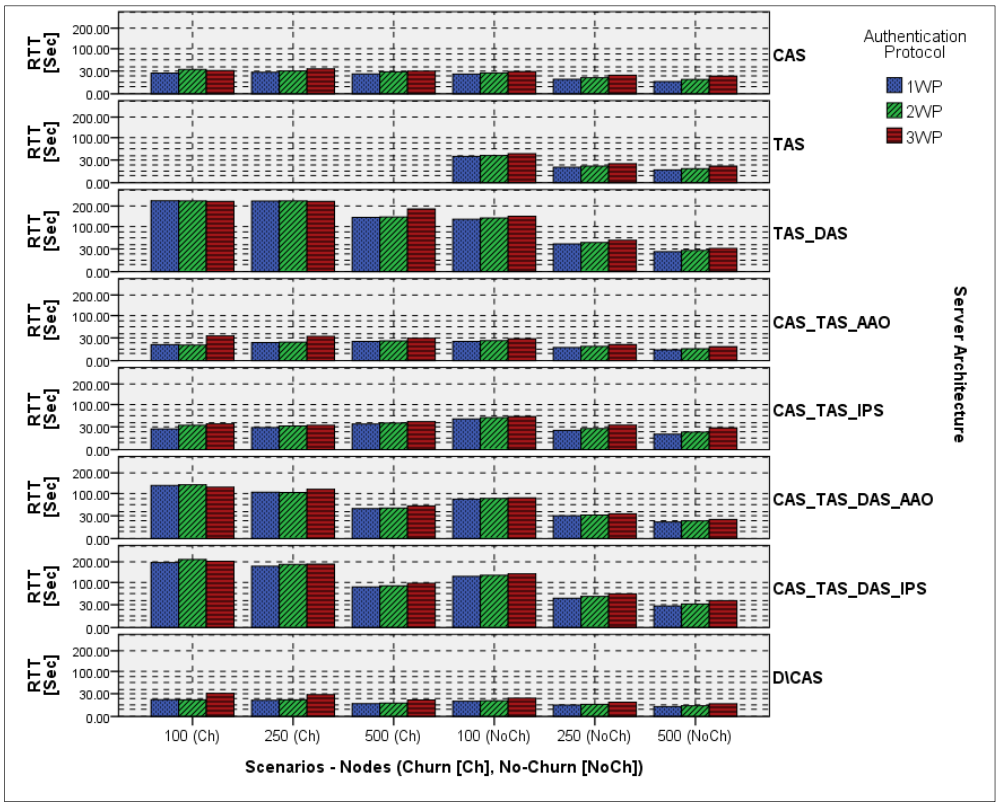


Figure 6-3: The Round Trip Time Chart

However, the *TAS_DAS* architecture, demonstrates longer response time (between 63 and 200 sec), especially in the *Churn* scenario since this specific architecture is established on a two-level server hierarchy. Also, this architecture relies initially on the *TASs*. However, if a required number of successful *TAS* connections cannot be satisfied, the architecture makes use of *DAS* by selecting one of reachable *TASs* (or *DAS*) in order to acquire credentials (e.g. membership and delegation certificates) from it. On the other side, most nodes using this architecture in the *Churn* case scenarios appear to be authenticated by a *DAS* and this clearly causes longer waiting times for *TAS_DAS*. The issue of certificate acquisition from different server architectures will be discussed in detail in Section 6.2.6.1.

On the other hand, the *CAS_TAS_DAS_IPS* architecture appears to have relatively high RTT similar to the *TAS_DAS* architecture but with a small decrease within the *Churn* scenario (refer to Figure 6-3). This stems from that fact that this architecture involves three different servers, *CAS*, *TASs* and *DASs* (i.e. the three-level server hierarchical architecture), for providing the security service to user nodes. Additionally, it exploits the *IPS* calling strategy which is built upon a priority sequence for calling three different types of servers, normally starting from invoking *CAS*, if it is not available, then calling *TASs* and finally calling *DAS* if a specific number of *TASs* are not reachable either. Hence, this strategy, especially in the case of invocation failures for certain server types, would need much more time to undertake successful authentication.

The *CAS_TAS_DAS_AAO* architecture, although it includes the same server hierarchy as the *CAS_TAS_DAS_IPS*, has better RTT by using a different strategy of calling the *CAS* and *TASs*. This strategy is called "*All At Once" (AAO)* (i.e. broadcasting invocations) and improves RTT especially in the *No-Churn* scenario, as presented in Figure 6-3. This type of calling strategy saves waiting time by checking accessibility of all servers simultaneously.

By observing the RTT outcomes of *CAS_TAS_IPS* across all types of scenarios at any authentication protocol in use, it can be recognised that this particular architecture takes a middle place among other architectures in terms of the RTT estimation. This particular architecture allows user nodes to depend only on two sequential choices of servers (*CAS* and then *TASs*) for authentication and this leads to an average value of RTT between *CAS* and *TASs* authentication cases. There is no significant difference (a variance between 0 and 4 seconds) between *CAS* and *CAS_TAS_AAO* although both are anticipated to be differed as the *CAS_TAS_AAO* architecture already includes a *CAS* in addition to *TASs.* This means that most authenticated nodes take more advantage of *CAS* rather than *TASs* in this architecture, especially in the *Churn* scenario, see Section 6.2.6.1. The *D\CAS* architecture, which is based on a one-level server hierarchy, demonstrates the best RTT by having a short and stable delay throughout most of

network scenarios regardless of the authentication protocol. This is because in this particular architecture a new joining node is allowed to connect with multiple independent servers (similar to *CAS*) and only one successful connection with one of those servers is sufficient to get the node full authentication.

Finally, the type of the authentication protocol can have an impact on RTT, as measuring RTT is required to complete handshaking and processing in both ends of user and server nodes. As processing and communication required by a given authentication protocol increase, so does RTT. There are noticeable differences (less than 10 sec) between the different protocols; the *3WP* protocol shows higher RTT compared to the *1WP* and *2WP* protocols. This is because the *3WP*, which consists of an exchange of four messages, clearly requires much more processing and communication against the other protocols.

6.2.4 RTT Standard Deviation (RTT-STDev)

The standard deviation values of the Round Trip Time (RTT-STDev) indicate the variation from its normal behaviour represented by the mean average. As shown in Figure 6-4, there are insignificant differences (less than 5 sec) among the three different authentication protocols (*1/2/3WP*) being used in any security architecture across most tackled network scenarios. Also, this is due to the same reasons discussed before in Section 6.2.3; the RTT-STDev for the *TAS* architecture in the *Churn* scenario has no results available.

On the other hand, throughout the *No-Churn* and *Churn* scenarios, the RTT-STDev takes distinct trends across different node populations for each of the architectures. The *CAS*, *D\CAS* and *CAS_TAS_AAO* architectures show to some extent a steady deviation across all scenarios regardless of the number of nodes being involved. This is because node authentication (credential possession) mostly originates from broadcasting all required server invocations and only one reply can be accepted from a certain server. This is in *CAS* or *D\CAS* server architectures. Even in the case of the *CAS_TAS_AAO* that has two different server

architectures, most authentications occur in *CAS* invocations, see Section 6.2.6.1. The other architectures, especially in the *No-Churn* scenario, have moderate decrease in RTT-STDev when increasing the population size. This is due to the fact that the network becomes more connected leading to an increase in the chance to get authentication by the first calling server architecture, like *CAS*. In contrast, in the *Churn* scenario, there are a number of particular architectures, like *TAS_DAS* and *CAS_TAS_IPS*, which display an increase in STDev while population is growing in size. Regardless of RTT-STDev values (i.e. levels or order), a better RTT-STDev trend for a particular architecture can be considered, one which demonstrates (1) a decrease or (2) stability in deviation when more nodes join the network at any scenario.

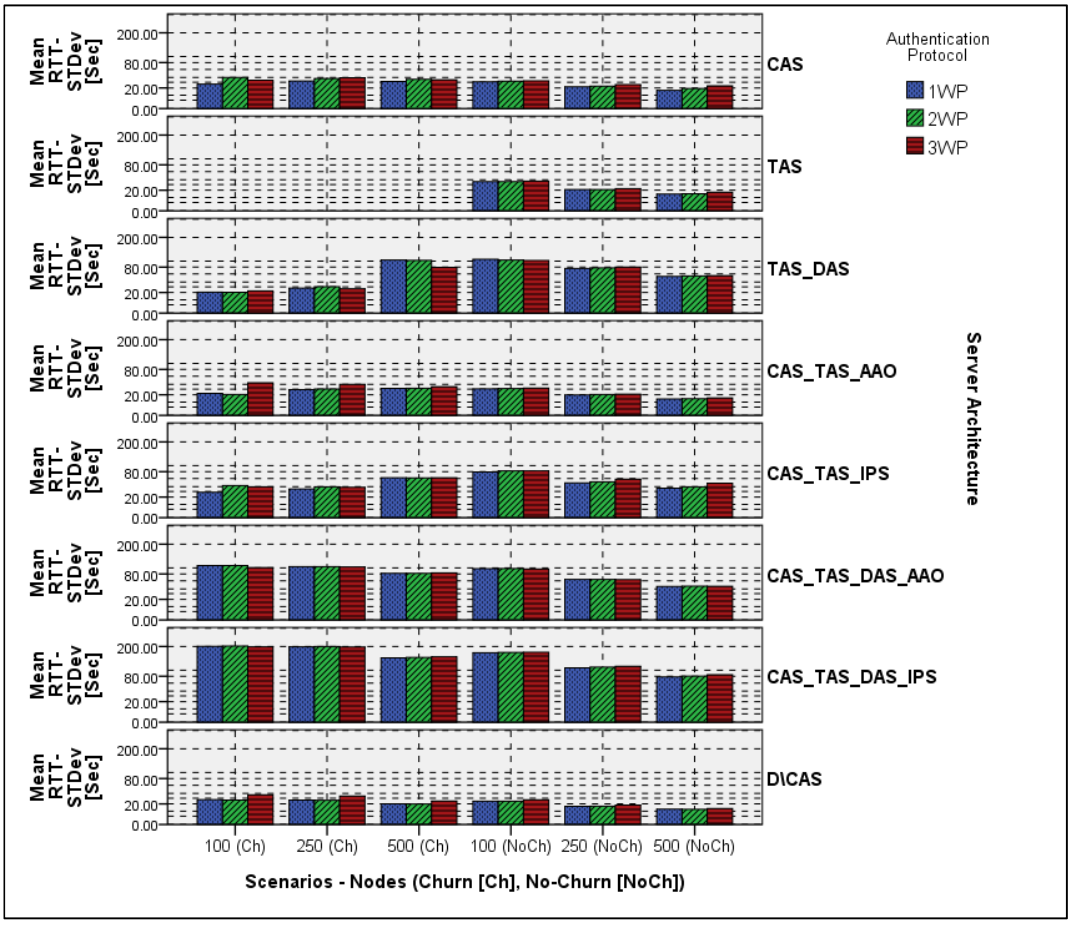


Figure 6-4: The RTT Standard Deviation Chart

However, comparing the actual RTT-STDev values of the available architectures, the *CAS_TAS_DAS_IPS* server architecture shows the largest RTT fluctuation in any scenario. This is because this architecture incorporates different alternative servers and calling such servers by a user node following a prioritised sequence entails various RTTs and longer latencies for a successful connection. Then, the *CAS_TAS_DAS_AAO* shows partly lower RTT-STDev than the *CAS_TAS_DAS_IPS* as the *CAS_TAS_DAS_AAO* utilises the *AAO* calling strategy for invoking *CAS* and *TASs,* which appears not to cause many differences in RTT unlike *IPS*. The *D\CAS* has the smallest RTT-STDev as it depends only on a one-level server architecture where a number of server replicas are distributed in the working space. The *CAS*, *CAS_TAS_AAO*, and *TAS* show a moderate level of variance as opposed to the other architectures throughout different scenarios. Finally, apart from having a unique trend in the *Churn* scenario, the *CAS_TAS_IPS* and *DAS_TAS* reveal a slightly high variation in RTT.

6.2.5 Communication Overhead

The communication overhead is acknowledged as a very important measurement in any particular networking systems since it mainly represents the cost of traffic generated by nodes in the network. In general, it is desirable to achieve lower communication overhead in any network. As a result, more communication would instigate more congestion and processing in the network which becomes more critical especially in limited networks like MANETs. This section discusses the results of total communication overhead produced by every successful authenticated node as a result of applying different security architectures under different network scenarios, as shown in Figure 6-5.

The amount of communication overhead can also be associated with various issues in the SSAM model. Those issues are relevant to the particular types of the authentication protocol (e.g. *1/2/3WP*) and the server architecture (e.g. *CAS*, *TAS*, etc.) being used. Each authentication protocol normally relies on a distinct number

of messages that need to be exchanged between two end points. Each server architecture may have a different number of active server connections and calling strategies (i.e. concurrent or on-demand invocations) which would clearly entail different amount of traffic required for accomplishing a successful connection. In addition to the authentication protocol and server architecture in use, a number of re-authentication attempts (e.g. a maximum of three attempts are proposed in this study) being part of the authentication protocol can also contribute to total communication.

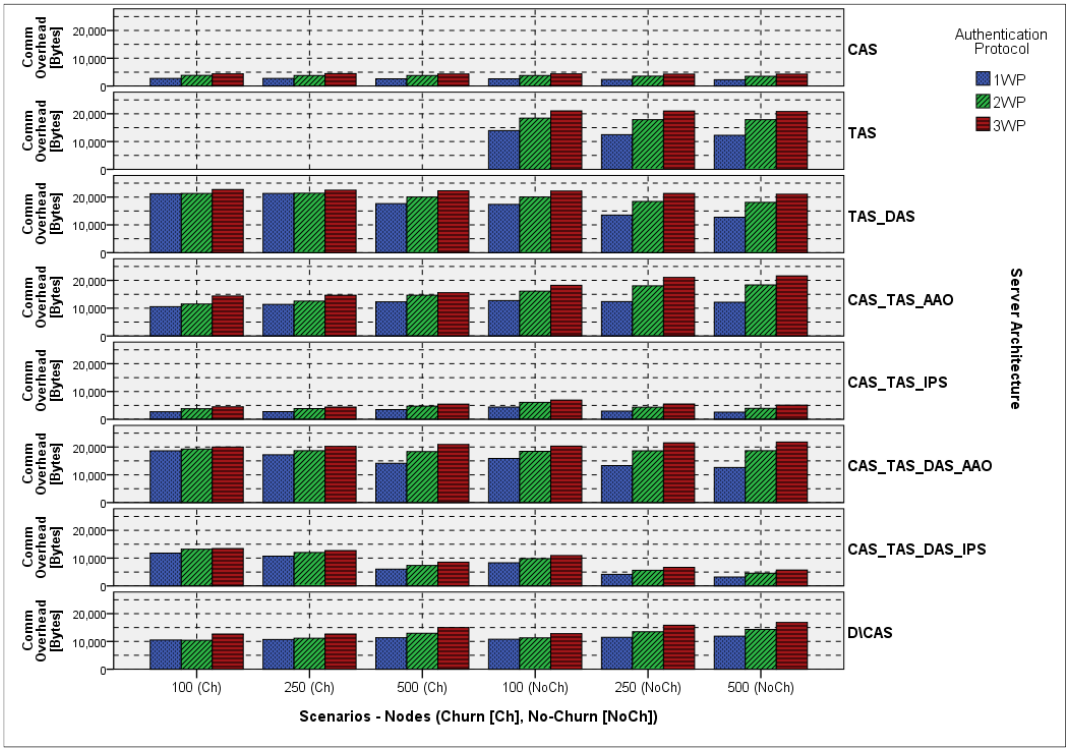


Figure 6-5: The Communication Overhead Chart

The three authentication protocols being exploited incur different communication overhead proportionate with the number of messages being exchanged. As presented in Figure 6-5, the *3WP* protocol causes the greatest communication overhead among the authentication protocols across the distinctive networks scenarios irrespective of the type of server architecture as this specific protocol primarily takes advantage of three control messages apart from a

data message. In addition, the *2WP* protocol obviously shows the middle overhead between the *1WP* and *3WP* throughout the different case scenarios whereas the *1WP* protocol has the lowest communication overhead as this has the fewest number of messages being involved in the protocol, see the sequence diagrams in Section 4.7.1.1. On the other side, based on the type of server architecture being used, the number of required server connections has a greater influence on the total communication because of an additional cost to establishing multiple new connections with different servers. Table 6-1 breaks down the details of the server connection within every architecture in SSAM to show differences.

Server Architecture	**The Number of Server Connections**			
	Connections Type (*m* Instant or *n* On-demand)	**Successful Connection Requirements** (connection from the server type: *Sn* → Server Type; *x,y,z,t:* distinct servers)	**Initial active connection Requirements** (Minimum Connections)	**Total Connections** (Maximum connections required in the worst case scenarios)
CAS	*1 Instant connection*	*(1 → CAS)*	1	1
TAS	*6 instant connections*	*(3 →$TAS_{x,y,z}$)*	6	6
TAS_DAS	*6 Instant connections* ***OR*** *1on-demand*	*(3 → $TAS_{x,y,z}$)* ***OR*** *(1→DAS_t)*	6	7
CAS_TAS_AAO	*7 Instant connections*	*(1 → CAS OR 3 → $TAS_{x,y,z}$)*	7	7
CAS_TAS_IPS	*(1 Instant connection* ***OR*** *6 on-demand connections)*	*(1 → CAS)* ***OR*** *(3→ $TAS_{x,y,z}$)*	1	7
CAS_TAS_DAS_AAO	*(7 Instant connections* ***OR*** *1 on-demand)*	*(1 → CAS* ***OR*** *3→ $TAS_{x,y,z}$)* ***OR*** *(1→ DAS_t)*	7	8
CAS_TAS_DAS_IPS	*(1 Instant connection* ***OR*** *7 on-demand)*	*(1 → CAS)* ***OR*** *(3→ $TAS_{x,y,z}$)* ***OR*** *(1→ DAS_t)*	1	8
D/CAS	*6 instant connections*	*(1 → CAS_x)*	6	6

Table 6-1: The Server Connection Evaluation

The *CAS*, *TAS* and *CAS_TAS_IPS* architectures irrespective of the authentication protocol in use, tend to have the lower cost of communication in each of the *Churn* and *No-Churn* scenarios. This is caused by these particular architectures requiring fewer active server connections (i.e. only 1 instant or 6 on-

demand connections, see Table 6-1) so as to accomplish node successful authentication. Most cases, where the *TAS_DAS* architecture is applied show the highest communication overhead. Additionally, some cases of *CAS_TAS_DAS_AAO* and *TAS,* especially when *3WP* protocol is involved, have high communication overheads, as those architectures need to simultaneously create multiple server connections (initially 6 or 7 connections). Alternatively, the *CAS_TAS_DAS_IPS* architecture has with fairly low communication overhead whereas the rest of the architectures, *D/CAS* and CAS_*TAS_AAO*, display an average cost of communication in comparison with the other architectures.

The communication overhead produced from employing a particular architecture presents a different trend throughout distinct population sizes within each different scenario being tackled (*Churn* and *No-Churn*). In the *CAS*, *TAS*, *CAS_TAS_IPS*, and *D\CAS*, the amount of communication comparatively stabilises over a growing population in both scenarios regardless of the authentication protocol being exploited. The *TAS_DAS*, *CAS_TAS_DAS_IPS* and *CAS_TAS_DAS_AAO* architectures demonstrate a small decrease in the communication cost when increasing the number of nodes in the cases of *1WP* and *2WP* protocols and stability in the case of the *3WP* protocol. On the contrary, the *CAS_TAS_AAO* architecture expresses a communication increase when the node population is gradually expanded. Eventually, stability in the communication overhead of any security architecture across different node densities would be desirable only when this particular communication overhead is not too high. Otherwise, a decline in the communication overhead would be much more recommendable as it is important to reduce communication as much as possible especially in limited MANETs.

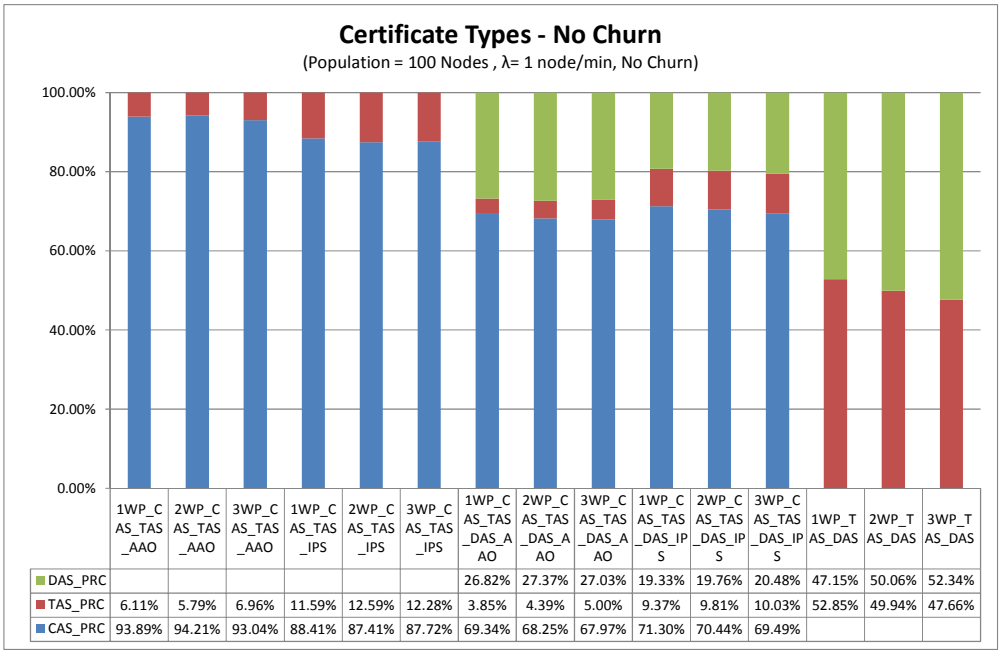
6.2.6 The Security Architecture Productivity in Certificate Acquisition

This section aims to discuss effectiveness and productivity of those architectures which make use of multiple different servers. These are two-level and three-level server hierarchies as there are three distinct server types involved in the SSAM model (i.e. *CAS*, *TAS* and *DAS*). The category of the two-level server hierarchical architecture includes the *CAS_TAS_AAO, CAS_TAS_IPS* and *TAS_DAS* architectures which make use of only two types of servers. While the three-level server hierarchical architecture includes the *CAS_TAS_DAS_AAO* and *CAS_TAS_DAS_IPS* architectures which employ three unique types of servers. However, it is very important to make sure that all server types are utilised and observe how practical they are to deliver security services to user nodes (i.e. membership certificate acquisition) when those server architectures are in operation under the network case scenarios (i.e. *100-Node Churn* or *No-Churn,* etc.). There are two indicators proposed in this study to represent the certificate acquisition evaluation: the certificate type and certificate count. The certificate type refers to the first type of a server which a user node successfully obtains its membership certificate (i.e. full and partial certificates or certificate chain) from. Whereas, the certificate count (i.e. having multiple varied certificates) determines how many different certificates a user node can acquire at the same time when this node tries to use different server architectures in the network. The next subsections will present the results of these two measurements and also address the key features which make those multi-level server architectures much more worthwhile in comparison with the single server type of architectures.

6.2.6.1 The Certificate Types

As aforementioned, the certificate type essentially concerns with the multi-level architectures where a new node can obtain its first membership certificate from different available sources (i.e. specific servers: *CAS, TASs*, or *DAS*) and can be

considered as an authenticated node. Therefore, the mean percentage of different certificate types, acquired by nodes in the particular proposed security architectures, are estimated under certain scenarios to understand security server effectiveness as shown in Figure 6-6 and Figure 6-7 for the *100 Nodes No-Churn* and *Churn* scenarios. Other scenarios (*250-Node No-Churn & Churn* and *500-Node No-Churn & Churn* scenarios) can be found in Section B.2.1 (*Appendix B*).

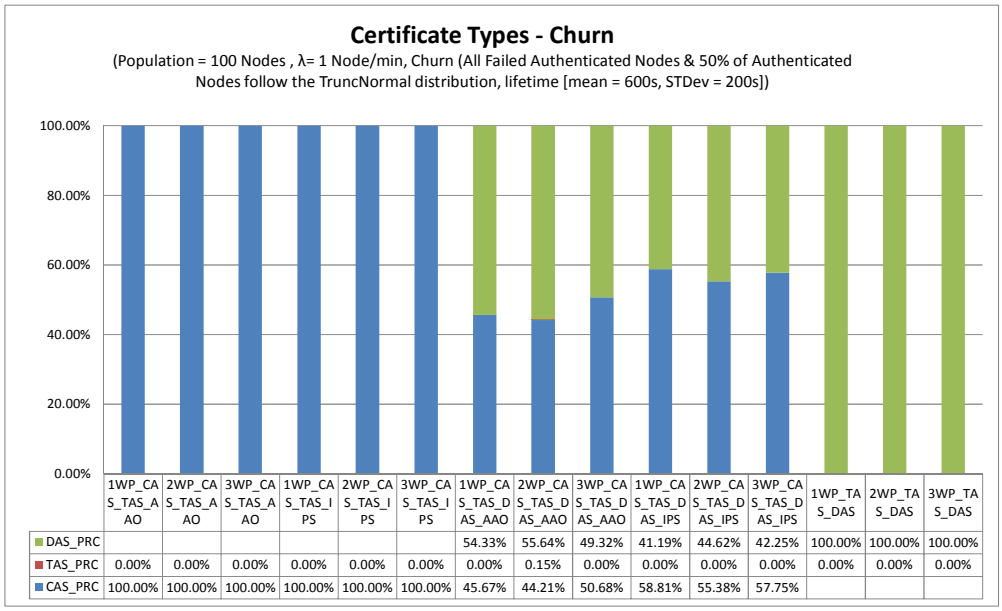


	1WP_CAS_TAS_AAO	2WP_CAS_TAS_AAO	3WP_CAS_TAS_AAO	1WP_CAS_TAS_IPS	2WP_CAS_TAS_IPS	3WP_CAS_TAS_IPS	1WP_CAS_TAS_DAS_AAO	2WP_CAS_TAS_DAS_AAO	3WP_CAS_TAS_DAS_AAO	1WP_CAS_TAS_DAS_IPS	2WP_CAS_TAS_DAS_IPS	3WP_CAS_TAS_DAS_IPS	1WP_TAS_DAS	2WP_TAS_DAS	3WP_TAS_DAS
DAS_PRC							26.82%	27.37%	27.03%	19.33%	19.76%	20.48%	47.15%	50.06%	52.34%
TAS_PRC	6.11%	5.79%	6.96%	11.59%	12.59%	12.28%	3.85%	4.39%	5.00%	9.37%	9.81%	10.03%	52.85%	49.94%	47.66%
CAS_PRC	93.89%	94.21%	93.04%	88.41%	87.41%	87.72%	69.34%	68.25%	67.97%	71.30%	70.44%	69.49%			

Figure 6-6: The percentage of different certificate type obtainability in the case of the "*100 Nodes No-Churn*" scenario.

In the three-level server hierarchical architectures at any authentication protocols being applied, the results indicate that all three types of servers (*CAS*, *TASs* and *DASs*) are being utilised under all network scenarios except the *100-Node Churn* case. In the *100-Node Churn*, the integrated *TAS* architecture is apparently not fully functional in this specific scenario which is explained by the higher network partitioning (i.e. lower node density). On the other side, as shown in the plots (Figure 6-6, Figure 67, and the figures in B.2.1 *Appendix B*), it can be recognised that the percentage of nodes obtaining their certificates from *CAS* is often higher (approximately between 69% - 95% in the *No-Churn* case, 47% -

82% in the *Churn* case) than the other nodes acquiring their certificates from different servers, such as *TASs* or *DASs*. Also, this server shows more productivity in terms of service delivery under the *No-Churn* network settings than under the *Churn* settings. It appears that group of *DASs* (approximately between 3% - 27% in the *No-Churn* case and 13% - 53% in the *Churn* case) are more operational in the case of having a highly dynamic network. There are fairly few numbers of nodes which have taken advantage of the *TASs* model under both scenarios (approximately between 3% - 10% in the *No-Churn* case, 0% - 9% in *Churn* case).



	1WP_CAS_TAS_AAO	2WP_CAS_TAS_AAO	3WP_CAS_TAS_AAO	1WP_CAS_TAS_IPS	2WP_CAS_TAS_IPS	3WP_CAS_TAS_IPS	1WP_CAS_TAS_DAS_AAO	2WP_CAS_TAS_DAS_AAO	3WP_CAS_TAS_DAS_AAO	1WP_CAS_TAS_DAS_IPS	2WP_CAS_TAS_DAS_IPS	3WP_CAS_TAS_DAS_IPS	1WP_TAS_DAS	2WP_TAS_DAS	3WP_TAS_DAS
DAS_PRC							54.33%	55.64%	49.32%	41.19%	44.62%	42.25%	100.00%	100.00%	100.00%
TAS_PRC	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.15%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
CAS_PRC	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	45.67%	44.21%	50.68%	58.81%	55.38%	57.75%			

Figure 6-7: The percentage of different certificate type obtainability in the case of the "*100 Nodes Churn*" scenario.

Having a higher proportion of nodes' population obtaining their credentials from the *CAS* or *DAS* models stems from the fact that, in the particular situation, any node relying on one specific server like *CAS* or *DAS* has much more likelihood of attaining successful authentication than a node that needs to secure multiple connections of different distributed servers like *TASs,* especially under a high dynamic network topology like in the *Churn* case scenarios. On the other hand, a number of *CAS* authenticated nodes often exceeds the total of *DAS* ones in

all scenarios except the case of *100-Node Churn*. This is arguably originated from the nature of connections that is required to be created to call *CAS* or *DAS*. In other words, in calling *CAS*, every joining node at first establishes an instant connection with *CAS*, whereas, in calling *DAS*, a connection is created whenever required after invocation failures from *CAS* and *TAS* based on the SSAM proposal to compensate those failures. Therefore, the number of successful *CAS* calls is more likely greater than the number of successful *DAS* calls unless there exists quite high partitioning in the network which causes *DAS* to outperform *CAS* because of the high rate of *CAS* failures, as presented in Figure 6-7.

As such, in the two-level server hierarchical architectures, irrespective to the authentication protocols being used, the results (Figure 6-7) clearly reveal that only one particular server type can be exploited when those architectures are tested under the *Churn* network scenarios, like *100-* and *250-Node Churn*, otherwise both servers (*CAS_TAS* or *TAS_ DAS*) are in operation under the other case scenarios. Also, in these two particular scenarios, almost all nodes in both the *CAS_TAS_AAO* and *CAS_TAS_IPS* architectures can attain successful authentication only from *CAS* while in the *TAS_DAS* architecture, only from *DAS*.

Results show that in the *CAS_TAS_AAO* and *CAS_TAS_IPS* architectures, the number of nodes having their certificates from *CAS* is always much greater (approximately between 88% - 93% in the *No-Churn* case, 94% - 100% in the *Churn* case) than other nodes acquiring their certificates from *TASs* (approximately between 3% - 10% in the *No-Churn* case, 0% - 5% in the *Churn* case) because of better *CAS* accessibility than *TASs*. However, in the *TAS_DAS* architecture, *TASs* appear to be more effective over the *No-Churn* scenario (approximately between 50%-94%) than *DAS* (approximately between 6%-50%). This is because the MANET in that scenario becomes sufficiently connected, especially in the *250-Node* and *500-Node* conditions, which enables the *TASs* to be more accessible to new nodes wishing to join. On the contrary, *DAS* becomes better in terms of providing the security service (approximately between 61% - 100%) than *TASs* (approximately between 0% - 39%) over the *Churn* scenario.

Therefore, the *TAS_DAS* architecture, where *CAS* is absent, shows a balance of its servers' utilisation throughout the two different scenarios.

With consideration for the type of authentication protocols being used, it is noted that using the *3WP* protocol in some particular server architectures such as *CAS_TAS_AAO, CAS_TAS_DAS_AAO, TAS_DAS* architectures under certain network case scenarios, presents a small impact on the *TAS* certificate being acquired by calling nodes compared to the other, *1WP* and *2WP,* protocols. This is by making this particular certificate less obtainable than the other certificates within those server architectures, (as shown in Figure 6-6, Figure 6-7, and the figures in B.2.1 *Appendix B*).

In conclusion, even though there are clearly significant variances between different types of certificates being obtained across various scenarios (i.e. having a certificate from *CAS* or *DAS* is often more possible than having a certificate from *TAS*), it is important to note that the server types (*CAS*, *TAS* and *DAS*) in the proposed three-level hierarchical security architectures compensate potential authentication failures (i.e. boost availability). The least effective obtainability comes from using *TASs* because these demand multiple responses from servers in order to generate the required membership certificate.

6.2.6.2 The Certificate Count

This section primarily investigates certificate count in the multi-level server hierarchical architectures. The aim is to show the possibility of a new joining node that can obtain more than one type of a membership certificate once it completes its authentication. This can be considered as a sort of certificate redundancy. For example, obtaining more than one type of certificate can be a certificate backup for a node so this backup can be exploited, when required, to avoid the cases of certificate loss or corruption. Therefore, the average number of authenticated nodes that acquire a different count of certificates irrespective of their actual types in the particular proposed security architectures is calculated under certain

scenarios to perceive security server effectiveness. The results are presented in Figure 6-8 and Figure 6-9 for the *100-Nodes No-Churn* and *Churn* scenarios as example and other scenarios (*250-Node No-Churn & Churn* and *500-Node No-Churn & Churn* scenarios) can be found in Section B.3.2 (*Appendix B*).

Two categories have been identified: (1) obtaining two types of certificates, and (2) obtaining one type of certificate. By disregarding the authentication protocol, results reveal that the only the server architectures (i.e. *CAS_TAS_AAO and CAS_TAS_DAS_AAO*) which take advantage of the *AAO* calling strategy can allow a user node to have two different certificates. This due to the fact that this particular strategy depends on opening concurrent connections with servers by a calling node for the sake of getting successful authentication as much quickly as possible. Whereas, the outcomes of applying the *IPS* calling strategy (i.e. used in the *CAS_TAS_IPS* and *CAS_TAS_DAS_IPS* architectures) demonstrates that this specific strategy is unable to allow a joining node to gain more than one certificate. This is because *IPS* is based on creating the server connections on demand and also in sequence (see Table 6-1) when a certain server connection failure occurs. As such, the *TAS_DAS* architecture shows the same results as architectures which make use of *IPS* because of the same reason discussed above.

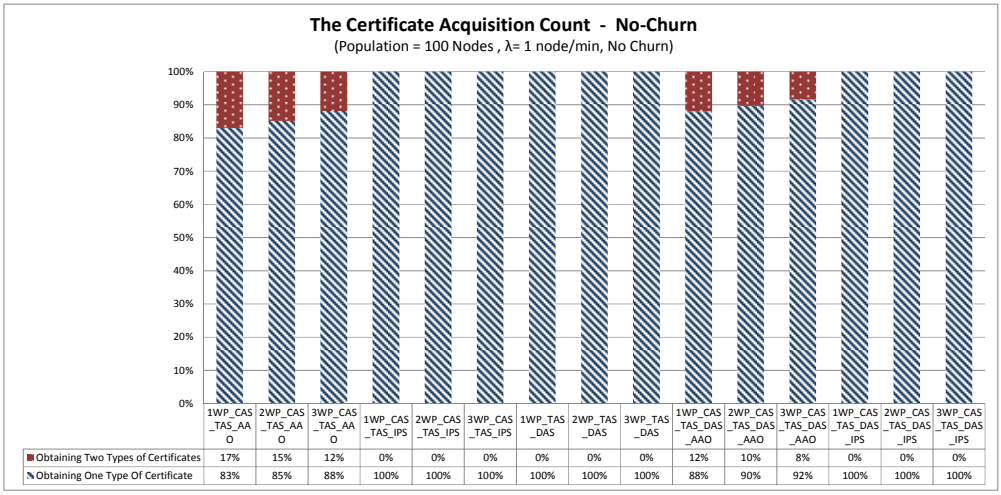


Figure 6-8: The percentage of certificate count obtainability in the case of the *"100 Nodes No-Churn"* scenario.

On the other hand, as shown in Figure 6-8, Figure 6-9, Figure B3.3 (*Appendix B*), etc., the scenario also plays an important role in the opportunity of a given node having more than one certificate. The more nodes that get involved in the network (i.e. by increase in the node population and decrease in churn rate) to make less partitioning, the more chances a node has to acquire more than one certificate as a result of improving server accessibility. For example, in the *CAS_TAS_AAO* and *CAS_TAS_DAS_AAO* architectures over the *250-Node* and *500-Node No-Churn* case scenarios, there are more than 50% of authenticated nodes that can have two certificates, whereas in the same architectures but over the *100-Node* and *250-Node Churn* case scenario, the possibility of an authenticated node to acquire two certificates becomes very low with less than 5% or even null in some cases.

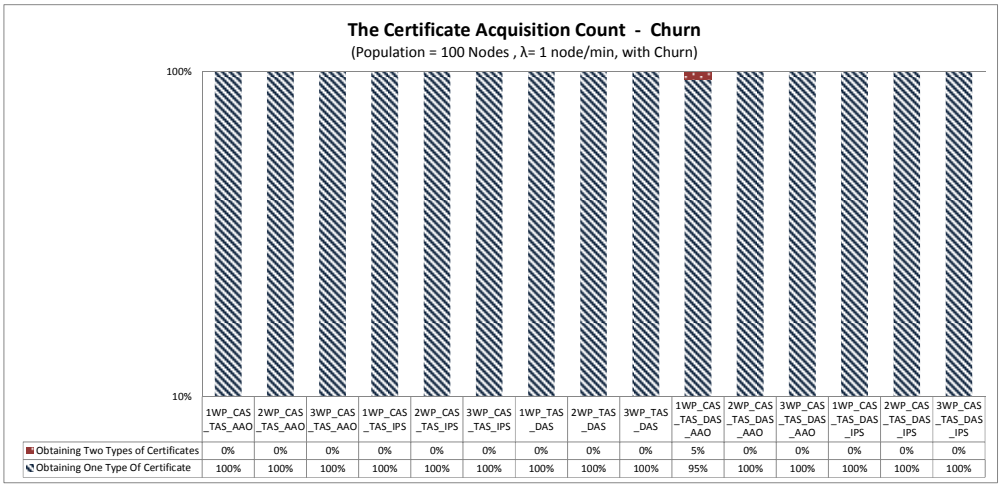


Figure 6-9: The percentage of certificate count obtainability in the case of the *"100 Nodes Churn"* scenario.

Alternatively, the type of the authentication protocol being involved in the particular architectures (e.g. *CAS_TAS_AAO* and *CAS_TAS_DAS_AAO*) appears to have a clear influence on certificate acquisition opportunities. The authentication protocol, that is characterised with processing and exchanging several messages for achieving success in authentication, is expected to have a negative impact on attaining more than one certificate as it evidently shows in the

results, especially under the *100-, 250- and 500-Node No-Churn* and *500-Node Churn* case scenarios. This is as a result of more delays and message losses that protocol can entail in the network. Thus, the *1WP* protocol across most case scenarios demonstrates the best results in the number of cases, that a node could have two certificates. While the *3WP* protocol shows the least chance, the *2WP* protocol clearly takes the mid-point of acquiring additional certificates between *1WP* and *3WP* protocols in most cases.

To sum up, it is important to realise from the results that any two- or three-level hierarchical server architectures incorporating the *AAO* calling strategy will gain an extra feature of a certificate backup (i.e. potentially having multiple certificates for the same usage) which could be used to overcome certificate corruption and loss. However, this feature could instigate a storage resource concern where it is required to have an additional space to store those certificates especially in the low-duty nodes within the network.

6.3 Security Strength Implications

As highlighted in Section 4.4.1, the security strength is considered a very important dimension studying the robustness for a certain security/trust infrastructure against potential failures or attacks. Several threats, such as Denial of Service (DoS), repudiation, etc., which any trust infrastructure may encounter in different operational level, can be recognised (see Section 2.3.2).

This section addresses the security analysis of the proposed security architectures in SSAM in order to determine the level of security strength for each architecture against specific challenges (e.g. single-point-of-failure, man-in-the-middle, etc.). This particular security strength primarily points out to the measures being used for securing utilisation (i.e. availability) and communication (i.e. confidentiality, authentication, integrity, etc.) in the security architectures. Each security architecture consists of two key elements: a security server architecture and an authentication communication protocol and both elements may have some

protection weaknesses. Therefore, both available server architectures and authentication communication protocols in SSAM need to be assessed so as to decide how robust they are against relevant threats. Similar to the approach used in Section 6.5, two simple rankings (i.e. 1,.., 6 for server architectures and 1,.., 3 for authentication protocols) are incorporated for comparing these elements in each security architecture. Also, these two rankings present the security strength from the lower to the higher in this analysis.

In SSAM, there are eight proposed server architectures for providing a security/trust service to the new joining node so that these nodes can obtain their membership certificate. These architectures may suffer from one or two of common security problems, Single-Point-of-Failure (SPF) and Single-Point-of-Compromise (SPC). Single-Point-of-Failure (SPF) in any server architecture means that server architecture relies on single servers when these servers break down, the whole server architecture will fail to provide the security service. This particular threat usually is caused by a denial of service (DoS) attack. Single-Point-of-Compromise (SPC) refers to the situation. When an attacker compromises or hacks a certain security server in the server architecture; the whole security architecture and all nodes that are associated with the compromised server, will become compromised. This can be as a result of security intrusions (e.g. brute-force attacks, privilege escalation attacks, Trojan, backdoors, etc.). However, this type of threat appears to be much more serious than SPF since an attacker can take full control of the whole system, leading to disclose confidential information and fabricate credentials.

Based on the SPF evaluation for all server architectures, the *CAS* architecture is affected the most by this problem unlike the other architectures, as shown in Table 6-2. This is due to the fact that this specific architecture depends on one single central server for providing the security/trust services to user nodes; when this server has a failure, the whole security architecture becomes completely non-functional. The rest of the architectures may not have this problem because they are built on multiple distributed servers (i.e. threshold, delegated or duplicated

servers) for the sake of overcoming failures. Furthermore, some of these distributed architectures (e.g. *TAS_DAS*, *CAS_TAS_IPS, CAS_DAS_TAS_AAO,* etc.) include different server hierarchies which make these particular architectures strongly resilient to SPF and manage server failures in operation effectively. Although all these architectures appear not to undergo SPF, there is still a possibility to experience a complete failure (i.e. related to SPF) especially in situations of multiple servers' halts. Therefore, it is very crucial to evaluate these architectures against this problem, for comparing and ranking purposes. On the other hand, the SPF problem is associated with system availability, so the general characteristics of system performance (i.e. success ratio, failure frequency) can assist to reveal the system's tendency to this problem. In other words, a system with low availability is likely to become unavailable if this system is under a Distributed Denial of Service (DDoS) attack, even though this system has measures to avoid SPF. As such, these distributed server architectures with/out a server hierarchy can be analysed and their SPF rank, shown in Table 6-2, can be justified through incorporating the success ratio and failure frequency criteria (i.e. reflecting availability) in ranking. Therefore, the following list represents the analyses of a complete failure for each of these architectures starting from the second rank onward:

Security Threats / **Server architecture**	**Single point of failure (SPF)**	**SPF Rank**	**Single Point of Compromise (SPC)**	**SPC Rank**
CAS	**Yes**	1	Yes – almost certainly	2
TAS	No →(SR8-FF8)	2	**No** – quite unlikely	6
D\CAS	No →(SR1-FF1)	6	Yes – certainly	1
TAS_DAS	No →(SR4-FF4)	3	Yes – unlikely	5
CAS_TAS_AAO	No →(SR6-FF6)	4	Yes – quite likely	3
CAS_TAS_IPS	No →(SR5-FF7)	4	Yes – quite likely	3
CAS_TAS_DAS_AAO	No →(SR2-FF3)	5	Yes – likely	4
CAS_TAS_DAS_IPS	No →(SR3-FF2)	5	Yes – likely	4

Table 6-2: SPF and SPC Comparison and Ranking (“SRi” refers to Success Ratio rank, “FFi” refers to Failure Frequency, rank, i= 1, 2,.., 6, the SPC probability scale: quite unlikely,…, certainly)

- As the *TAS* architecture involves *n* dependant servers relying on the *TC(k, n)* scheme for providing a security service, this architecture completely collapses if the servers' failure makes a number of existing servers less than *k* available servers required to satisfy the *TC(k, n)* scheme. In addition, this architecture shows poor availability based on performance evaluation (i.e. success ratio and failure frequency).

- Both *CAS_TAS_AAO* and *CAS_TAS_IPS* architectures show more resistance to a server failure than the detached *CAS* and *TAS* architectures, since in these two-level combined architectures, the architecture breakdown occurs if both *CAS* and *TASs* fail. Also, these two architectures show a small enhancement in performance, a little better than the *TAS* architecture which is sufficient to withstand potential failures.

- In the *TAS_DAS* architecture, servers play two roles (called *TAS* and *DAS* services) unlike the other pervious discussed architectures. Therefore, this architecture may only encounter a total failure when all servers break down. Even though the breakdown of this architecture seems to be hard to be achieved, this architecture however presents fair availability which may lead to more failures under an attack, such as a DDoS attack.

- Both *CAS_DAS_TAS_AAO* and *CAS_DAS_TAS_IPS* architectures involving different distributed servers (*CAS*, *TASs* and *DASs*) appear to have a better resilience against a complete failure. This stems from the fact that in order to break down these architectures entirely, all three different servers existing in these architectures must fail. Furthermore, these architectures also demonstrate better availability which may alleviate failures when a DDoS attack takes place.

- The *D\CAS* architecture is acknowledged as the most robust architecture against SPF because of two particular reasons. Firstly, this architecture uses

the approach of generating several copies of the *CAS*, so every server copy in this architecture can handle its invocation independently. Therefore, this architecture becomes in the state of a complete failure just in the circumstance that all copy servers are destroyed. Secondly, the *D\CAS* architecture shows the best availability among the other architectures decreasing the chance of a SPF occurrence.

Alternatively, by examining the server architectures in terms of Single-Point-of-Compromise (SPC), the *D\CAS* and *CAS* architectures are undoubtedly considered the most vulnerable architectures to SPC. This is due to the fact that compromising one server in these architectures will render all nodes compromised in the network. However, the *D\CAS* architecture is expected to be on the top of the SPC rank list before the *CAS* architecture. The reason is that compromising one of the server nodes in the *D\CAS* architecture does not only bring damaging consequences (e.g. security keys' disclosure and exploitation) to user nodes but also to the other available non-compromised server nodes in the whole architecture as well. The *TAS* architecture, in comparison with the other architectures, demonstrates the most robust architecture against SPC (stands on the bottom of the SPC rank list as show in Table 6-2). This is because this architecture relies on the *TC (t,n)* scheme which is originally developed to avoid such a problem. In other words, to compromise this architecture completely, this requires an attacker to compromise several threshold servers (t, $t >> 2$) which seems to be difficult to achieve.

The SPC scale (i.e. its scale represents: (1) quite unlikely, unlikely,…, (6) certainly) is suggested specifically to facilitate comparing and ranking the particular two- and three-level hybrid hierarchical architectures, based on the likelihood of those architectures being compromised totally under a certain attack. A ranking starts from the high probability of compromise in Rank 1 until the low probability of compromise in Rank 6, based on the "Dense Ranking" (see *Wikipedia* on ranking). The compromise likelihood is defined and interpreted by two guidelines: (1) *the level of individual SPC resilience* and (2) *the number of*

different user associations (i.e. obtaining different certificates). Initially, for the level of individual SPC resilience, this scope is to evaluate each single server type in the combined architectures against SPC. Apart from the *D\CAS* architecture, there are three main single-server-type architectures (*CAS*, *TASs* and *DASs)* that can be found in these compound architectures. As mentioned earlier, the *CAS* architecture is prone to be easily compromised, whereas the *TAS* architecture is quite adaptable to the problem of SPC. However, the *DASs* architecture can be anticipated to be fairly resistant to full compromise (i.e. similar to the situation of SPC). If one of *DASs* is compromised, only the user nodes associated with this compromised server can be compromised without affecting the other non-compromised *DASs* in the architecture. This is because each *DAS* relies on its own delegated certificate to prove its issuance authority to its calling user nodes.

Furthermore, to realise the whole view of the impact of compromise on those compound architectures, it is important to investigate the number of different user associations that can be generated by different type of servers, *CAS*, *TASs* or *DASs* (referring to the certificate type acquisition in Section 6.2.6.1). This information will indicate to the size of the impact (i.e. compromise likelihood) on the architectures when a certain server type is compromised. Therefore, based on the full summary of all cases (different authentication protocols, populations and churn) for each server architecture in Section 6.2.6.1, the number of *CAS* associations appears to be far greater than the *TAS* associations in the *CAS_TAS_AAO* and *CAS_TAS_IPS* architectures (approximately 90% difference). Hence, these two architectures are quite likely to be fully compromised. The *TAS_DAS* architecture is unlikely to be in a state of full compromise because both *TASs* and *DASs* are not easy to be compromised as discussed above in the first guidelines. Also, there is a relative balance in the number of different associations' types generated for this architecture (approximately 22% difference between *TASs* and *DASs*). Eventually, the *CAS_TAS_DAS_AAO* and *CAS_TAS_DAS_IPS* architectures are likely to have a complete compromise when they are under an attack. This stems from that fact that both particular architectures contain a *CAS* which is considered the weakest point in these

architectures comparing to the other servers, *TASs* and *DASs*. However, these *CAS_TAS_DAS_AAO* and *CAS_TAS_DAS_IPS* architectures show better resilient against SPC than the *CAS_TAS_AAO* and *CAS_TAS_IPS* architectures without *DASs*. This is due to the fact that these particular architectures are expected to have a number of *DAS* associations (approximately 22% of all associations) which result in reducing *CAS* associations and improving to some extent the security strength of these particular architectures.

Each security architecture in SSAM makes use of a certain end-to-end authentication protocol (i.e. One-Way-Pass (*1WP)*, Two-Way-Pass (*2WP)* and three-Way-Pass (*3WP)* in the standard of X.509 The Directory Authentication Framework (ITU-T, 1989, 2008)) to facilitate security communication between server and user nodes. However, this authentication protocol may suffer from security problems too, (e.g. Man-In-The-Middle attack, replay attacks, etc.) and also cannot satisfy desired protection (e.g. mutual authentication, integrity, non-repudiation, etc.). Based on the evaluation of the three adopted handshaking authentication protocols in Boyd and Mathuria (2003), I'Anson and Mitchell (1990), and Burrows *et al.* (1990), the *3WP* protocol is the most robust protocol among the other two protocols because this particular protocol uses a challenge technique (i.e. third message *CMsg3*) to avoid man-in-the-middle problem apart from apparently satisfying most of mutual authentication, integrity and confidentiality requirements. The *WP2* protocol demonstrates higher security robustness than the *1WP* protocol as this specific protocol ensures mutual authentication between two ends avoiding masquerading, forge of a certificate, etc. Hence, the three authentication protocols are ranked according to the level of security strength as it follows: (**1** Weak) *1WP*; (**2** Middle) *2WP*; (**3** Strong) *3WP*.

Eventually, this dimension of security strength is not only limited specifically to these criteria and threats (SPC, SPF, mutual authentication, etc.) discussed above. However other threats and related security features can be investigated and involved in this dimension if they are relevant to a study.

6.4 The Proposed Methodological Approach in Practice (Summary)

This section summarises the proposed methodological approach which is applied to the SSAM model. This approach, as presented in Figure 4-5, is established on well-structured stages which enable to investigate all alternatives of security architectures in SSAM and then to assess those security architectures based on their security strength, performance and MANET context as follows:

Stage 1- As the SSAM model is developed originally for the "Security Service Level", all security architectures should be applicable to this particular operational level (the first step in this approach is satisfied).

Stage 2- Based on the component analysis of the security/trust infrastructure described in Section 4.3, the elements of the SSAM model can be interpreted, as shown in Table 6-3.

Security Roles	Similar to a typical CA, a security service in SSAM is intended to provide membership certificates to users for the network admission (issuing and retrieving) using authority servers.
Security-server Architectures	The SSAM relies on four authority servers (*CAS*, *TAS*, *DAS* and *D\CAS*) for establishing its different server architectures. Three different architectures can be distinguished (one , two and three server level architectures (Section 4.7 and 4.7.1)): One-Level Architectures: (*CAS*; *TAS*;D\CAS) Two-Level Architectures: (*TAS_DAS*; *CAS_TAS*) Three-Level Architectures: (*CAS_TAS_DAS*)
Security Communication Protocols	Authentication Protocols (Section 4.7.1.1): - The X509 standard one-way-pass protocol (*1WP*) - The X509 standard two-way-pass protocol (*2WP*) - The X509 standard three-way-pass protocol (*3WP*)

	Strategies of Calling for hybrid server architectures (Section 4.7.1.2): - All at Once (*AAO*) - In Priority Sequence (*IPS*) Re-authentication schemes (Section 5.3.3.2): - The exponential interval model is used for calling all types of servers except *DAS* - The adaptable fixed interval model used in calling *DAS*
Security Mechanisms	RAS cryptosystems & RSA-Threshold cryptosystem (Section 4.6)
Security Credentials	The X.509-v3 standard certificates (Section 4.6)

Table 6-3: The SSAM based on the component analysis of a security/trust infrastructure

Accordingly, an initial survey of all security alternatives in SSAM is generated as an outcome of this stage. This survey includes 24 different security architectures representing different server architectures, authentication protocols, and calling strategies, as shown in Table 5-5.

Stage 3- All security architectures identified from the previous stage should be evaluated, according to the security and performance dimensions with incorporating the context of particular MANET settings (illustrated in Section 5.3.4). In the performance dimension, all security architecture candidates are implemented using the OMNeT++ simulation tool (Section 5.3.2) and then a performance testing is conducted for these candidates (Section 5.4). The performance results are analysed for comparison purposes (Section 6.2). However, in the security strength dimension, these candidates are examined against specific security challenges (e.g. single-point-of-failure, man-in-the-middle, etc.) and authentication protocol robustness (e.g. mutual authentication, confidentiality, etc.), as demonstrated in Section 6.4. The contexts of MANET constraints and application settings can be assessed only by studying and defining the application domain properties. This can be tackled separately from security

architecture evaluations, as illustrated in Section 6.6 (validation and scenarios). In the context of application settings, a given domain application should be studied in accordance with the proposed settings in this study, such as dependency, lifetime, capacity, and environment. However, as this study of MANETs has already targeted the planned, large-scale MANET, and long-term application, all features of dependency, lifetime, and capacity are already characterised. Only the environment settings need to be described since these settings are associated with the real case scenario being involved (Section 6.6). Figure 6-10 displays all relevant criteria that are intended to be evaluated in this approach when a particular real case scenario is considered. However, it is worth pointing out that the "node churn" criterion, which indicates the levels of dynamism in the MANET topology as detailed in Section 5.3.4.6, is proposed to be under the MANET constraint context. Also, this issue may affect the performance criterion since there are two separate different node churning scenarios (i.e. the *Churn* and *No-Churn* cases).

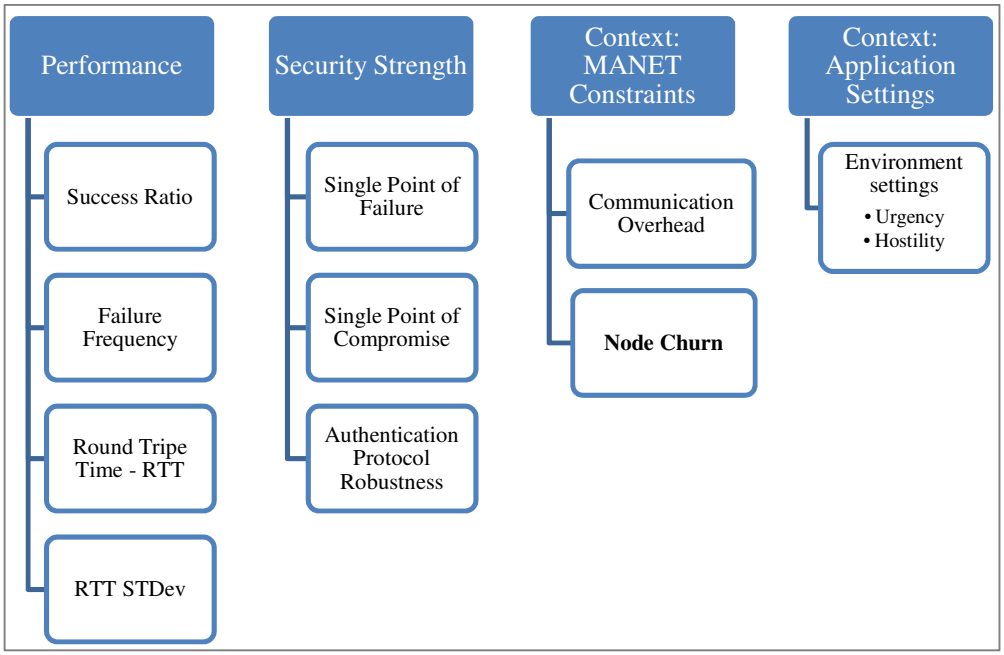


Figure 6-10: The elements (Multi-criteria) of the dimensions which are required to be assessed for the evaluation stage of the proposed methodology

Stage 4- A simple ranking and selection method (i.e. scoring, reciprocal weighting, weighted averaging, etc.) is employed in order to facilitate finding the best suited architecture matching context requirements. This can be realised by ranking all possible security alternatives in SSAM, based on two principles. The first principle refers to the desirable requirements (e.g. low traffic, high availability, strong protection, etc.) that a certain security alternative can satisfy. The second principle stems from the importance of criteria (security, performance, communication, etc.) derived from the context of the application being studied (e.g. academia, health, and military). Finally, the way of applying these ranking and weighting methods will be demonstrated in the next sections.

6.5 The Ranking Approach (Evaluation Stage 4)

The ranking approach proposed in this study consists of three main steps: (1) result scoring, (2) importance weighting, and (3) weighted averaging. Initially, in order to order a group of alternatives, every alternative within this group should have a particular score generated from a predefined fixed range of scores (scale) for coherent and consistent ranking. In this study, an Achievement Score (AcSc) is proposed to represent the degree of the preferable requirement fulfilment for a particular alternative. This score is defined as an achievement percentage ranged from 0% to 100%. Score 0% indicates to the worst scenario of the achievement (i.e. no requirement is met) whereas Score 100% is considered the ideal achievement by satisfying almost all desired requirements. As the results of performance and communication criteria are numerical with different metrics, the alternative AcSc can be estimated for every metric as follows:

$$AcSc_{altern.}\,[\%] = 1 - \frac{X}{W} \times 100$$

Where:

- X is an average value of a specific metric for a given alternative (*altern.*)
- W is the values of the worst case scenario for a specific metrics among all alternatives being involved.

Therefore, the above formula above is applied on the experimental results of performance and communication metrics. In addition, for each metric, the value of the worst case scenario (*W*) is identified as the maximum of the metric maximum average. For simplicity, all performance measurement results of three different node populations are summarised by taking the average of those three cases so as to represent the two main result cases, the *Churn* and *No-Churn* scenarios. In addition, in each scenario, an average value of each server architecture is taken regardless of the authentication protocol being used because of no significant differences among these protocols. Accordingly, the total scores for every performance metric in both *Churn* and *No-Churn* cases are calculated based on the AcSc formula, as shown in Table 6-4.

Server Architecture	The Churn Scenario (AcSc between 0 and 100%) for each measurement				The No-Churn Scenario (AcSc between 0 and 100%) for each measurement			
	Success Ratio	Failure Frequency	Round Trip Time RTT	RTT – STDev.	Success Ratio	Failure Frequency	Round Trip Time RTT	RTT – STDev.
CAS	20.56	71.73	88.57	84.82	69.64	73.30	92.08	87.64
TAS	0.00	66.42	0.00	0.00	59.60	64.91	89.90	87.97
D\CAS	62.25	81.75	92.58	88.23	84.52	80.85	94.79	90.95
TAS_DAS	26.62	73.20	20.52	79.00	73.32	73.55	71.31	61.12
CAS_TAS_AAO	20.42	72.38	90.57	86.28	71.61	72.49	93.19	89.15
CAS_TAS_IPS	21.31	72.32	86.68	80.18	74.21	71.32	86.17	71.22
CAS_TAS_DAS_AAO	53.17	79.70	62.00	57.44	81.01	79.22	82.07	66.05
CAS_TAS_DAS_IPS	46.61	77.42	38.45	20.16	81.16	76.49	70.75	40.22

Table 6-4: The scoring (AcSc) of server architectures for each performance metric in *Churn* and *No-Churn* scenarios (0% the worst → 100% the best)

On the other hand, the communication overhead results are intended to be tackled separately as these results implicitly indicate the cost of bandwidth (i.e. number of messages) and resources consumption (i.e. computation, power, etc.). Since MANETs are usually recognised with power, computation and bandwidth

limitations, communication overhead becomes a key factor in describing the feasibility of different security architectures (i.e. the context of MANET constraints in the proposed approach). Therefore, AcSc values are calculated for each security architecture, as shown in Table 6-5.

Authen. Protocol / Server Architecture	The *Churn* Scenario (Communication Overheads) (AcSc between 0 and 100%)			The No-Churn Scenario (Communication Overheads) (AcSc between 0 and 100%)		
	1WP	***2WP***	***3WP***	***1WP***	***2WP***	***3WP***
CAS	89.91	85.81	83.56	91.72	87.59	85.19
TAS	0.00	0.00	0.00	55.72	38.01	28.17
D\CAS	59.54	57.05	49.85	60.96	55.32	47.97
TAS_DAS	25.13	21.92	15.98	50.07	35.36	26.20
CAS_TAS_AAO	57.44	51.79	44.43	57.39	39.88	30.30
CAS_TAS_IPS	88.84	84.57	82.13	88.69	83.57	79.99
CAS_TAS_DAS_AAO	37.84	29.93	23.88	52.10	36.25	27.22
CAS_TAS_DAS_IPS	64.52	59.43	56.80	82.07	77.13	73.30

Table 6-5: The scoring (AcSc) of security architectures for communication metric in *Churn* and *No-Churn* scenarios (0% the worst → 100% the best)

(Note that the higher value of AcSc means a lower communication overhead as this metric is preferable with lower values (i.e. a lower traffic)).

For the security strength aspect, three main security features, discussed in Section 6.3, are considered to represent the robustness of a given security architecture alternative: Single-Point-of-Failure (SPF) and Single-Point-of-Compromise (SPC) resilience and authentication protocol robustness. However, the analysis of security strength is conducted in qualitative manner with ranking all alternatives, as shown Table 6-2. As a result, the AcSc values should be evaluated differently from the formula above. The rank of each alternative needs to be normalised within the predefined range of AcSc, as presented in Table 6-6. However, the results in this table are presumed to be applicable to both *Churn* and *No-Churn* scenarios, based on the security strength analysis presented in Section 6.3.

Security Strength / Security architecture	Authentication Protocol Robustness (1 – 99%)	SPF Resilience (1-99%)	SPC Resilience (1-99%)
1WP _CAS	1	1	20.6
1WP _TAS	1	20.6	99
1WP _D\CAS	1	99	1
1WP _TAS_DAS	1	40.2	79.4
1WP _CAS_TAS_AAO	1	59.8	40.2
1WP _CAS_TAS_IPS	1	59.8	40.2
1WP _CAS_TAS_DAS_AAO	1	79.4	59.8
1WP _CAS_TAS_DAS_IPS	1	79.4	59.8
2WP _CAS	50	1	20.6
2WP _TAS	50	20.6	99
2WP _D\CAS	50	99	1
2WP _TAS_DAS	50	40.2	79.4
2WP _CAS_TAS_AAO	50	59.8	40.2
2WP _CAS_TAS_IPS	50	59.8	40.2
2WP _CAS_TAS_DAS_AAO	50	79.4	59.8
2WP _CAS_TAS_DAS_IPS	50	79.4	59.8
3WP _CAS	99	1	20.6
3WP _TAS	99	20.6	99
3WP _D\CAS	99	99	1
3WP _TAS_DAS	99	40.2	79.4
3WP _CAS_TAS_AAO	99	59.8	40.2
3WP _CAS_TAS_IPS	99	59.8	40.2
3WP _CAS_TAS_DAS_AAO	99	79.4	59.8
3WP _CAS_TAS_DAS_IPS	99	79.4	59.8

Table 6-6: The scoring (AcSc) of security architectures based on security strength features (0% the worst → 100% the best)

(Note that the values of 0% and 100% are not taken into account because, there is no zero or perfect protection realistically).

The importance of criteria (i.e. requirements and dimensions) may vary from one domain to another. For example, security appears to be much more important in military contexts than in civil contexts because of potential hostile environments in these contexts. Therefore, the Rank Reciprocal Weighting (RRW) method (Stillwell *et al.*, 1981) is adopted for weighting the importance of different measurements and dimensions in this ranking approach (referring to the second step). Always weights, generated by this method, are between 0 and 1 and thereby this is consistent with the predefined range for scoring [0%, 100%] proposed in this approach. Also, in this method, the total of weights is always 1 regardless of the type of ranking being used (i.e. a reliable numerical weighting

method). The following formula is taken from Stillwell *et al* (1981) to reckon a weight of a particular criterion, based on its importance rank: (Note that the lower the *i* value, the higher the importance).

$$w_{criterion} = \frac{1/i}{\sum_{j=1}^{n} 1/j}\ , i = 1,2, \dots, n$$

where n is a number of criteria
i a criterion importance score
w_i a weight for i criterion

Always $\sum_{i=1}^{n} w_i = 1$

In this study, two simple levels of a criteria hierarchy can be recognised, as shown in Figure 6-10. Only performance and security strength dimensions have multiple measurements (e.g. *performance*: success ratio, failure frequency, RTT, and RTT-STDev; *security strength*: authentication protocol robustness, SPF resilience, and SPC resilience). In the second level of this hierarchy, for simplicity, it is proposed one fixed set of weights for these particular measurements. These weights are expected to be applicable to all scenarios being tackled. However, for the first level of this hierarchy, the importance of different dimensional weights will be defined according to the scenario being involved as this will be described in Section 6.6.

For the performance measurements, the success ratio and failure frequency are the most and equally important as these two measurements indicate to the desirable requirements of any system (i.e. reliability and availability). Also, other measurements (e.g. RTT, etc.) rely on them in terms of calculation. The RTT measurement is proposed to be more important than its deviation metrics. As a result, performance measurements are prioritised and the corresponding weights are calculated accordingly based on the RRW method, as shown in Table 6-7. As such, the weights of security strength measurements are estimated. However, the importance of security strength measurements are ranked in the following order, SPF and SPC resilience first and then authentication protocol robustness. This stems from that fact that securing service security on the server sides (being in operation) is much important than establishing protected links between user and server nodes.

Performance	***i = 1***	***i = 1***	***i = 2***	***i = 3***	***Total Sum***
	$w_{Success\ Ratio}$	$w_{Failure\ Frequency}$	w_{RTT}	$w_{RTT\text{-}STDev}$	
	0.3529	0.3529	0.1765	0.1176	1
Security Strength	***i = 2***	***i = 1***	***i = 1***		***Total Sum***
	$w_{AutProt.\ Robustness}$	$w_{SPF\text{-}Resilience}$	$w_{SPC\text{-}Resilience}$		
	0.2	0.4	0.4		1

Table 6-7: The performance and security strength weights based RRW method *i* an importance rank (lowest value is the most important)

The final step in this ranking approach is to generate the overall rank of all alternatives being considered. This can be worked out by using the weighted averaging which incorporates the weight of importance for a certain measurement and the scores of alternatives in calculation (as described in the formula below). An implementation of this step will be presented in next section.

$$Weighted\ AVG_{Sec\ altern.} = \frac{\sum_{i=1}^{n} r_i w_i}{\sum_{i=1}^{n} w_i}, i = 1,2, \dots, n$$

Where: n is a number of criteria
r_i a security alternative score
w_i a weight for the i criterion

In conclusion, the ranking approach used in this section is suggested to highlight the best and the worst among security architectures, depending on the experimental results of performance and communication and the implications of security strength. The aim of this approach is to enable MANET developers (i.e. decision makers) to easily perceive a whole picture (multi-dimension evaluation or multi-criteria), in order to be able to make better decisions. Table 6-4, 6-5, 6-6 and 6-7 primarily represent the preliminary evaluation (alternative scoring and importance weighting) of the key criteria for each different dimension in the proposed approach: performance, communication (i.e. MANET constraints) and security strength. In Stage 4 of the proposed approach, these tables are intended to be employed in the overall assessment of these three dimensions under different real case scenarios. This assessment will be demonstrated in the next section.

6.6 Validation and Scenarios

Generally, validation in this context means to ensure that the proposed SSAM meets the aim of this study and perform as it is expected (i.e. the system has been validated from the experiment quantitative results). In other words, validation, particularly in this study, is to prove that the SSAM model can offer the flexible approach leveraging a variety of security architectures which can be integrated in any service model deployed in large-scale MANETs in order to facilitate trust and security. This arguably motivates MANET developers to develop effective and efficient security services or applications taking into account MANETs limitations and application requirements. In this section, firstly the overall proposed approach with all elements will be summarised and highlighted. Secondly, for the SSAM approach validation, this section illustrates the usage of this approach in three different real case scenarios. Each scenario begins with a high-level description of the service/application for a specific domain and MANET characteristics, followed by the identification of requirements and preferences for the application.

6.6.1 The Real Case Scenarios

This section presents three exemplar cases, inspired by real world applications, that MANETs can be employed under different requirements. These cases are unfolded within scenarios borrowed from academia, emergency services and crisis management, and military missions. SSAM is applied on all three in order to prove the feasibility of the approach. The reason for selecting these case scenarios is that they cover a wide range of conditions and requirements, and, therefore, the flexibility and versatility of the proposed security solution can be demonstrated.

All presented case scenarios are designated to conform to all pre-conditions and assumptions of the SSAM design. For instance, each proposed scenario adheres to the authority-based infrastructure where all types of network nodes must be trusted and controlled by an authority. That is, they must have membership credentials to connect to the network and utilise its services.

Additionally, since SSAM is tested and evaluated under a low mobility model in MANET (i.e. human mobility model), as described in Section 5.3.4.1, all case scenarios must comply with this feature. Consequently, for validity purposes, other mobility models cannot be used. For example, a vehicle mobility model would show different broken links and partitioning behaviour in the network, and this would yield completely different and incomparable results.

6.6.1.1 The First Scenario – MANETs in Academia (S1)

Academic wireless networks usually cover University campuses and are accessed by a variety of users (students, staff, visitors, etc.) that may have access privileges for short or longer periods. MANETs can benefit academic environments enormously by providing autonomous networks on the fly that can be accessed by battery-powered mobile devices (e.g. smartphones, tablets, laptops, etc.). Thus, this networking technology, complementary to the traditional infrastructure-based networks, has the potential to provide the academic community with an on-demand medium for supporting and managing university services on campus (e.g. access to email accounts, intranets, e-libraries, etc.).

The MANETs in Academia scenario studies the case where the IT department of a University decides to deploy the services listed above on MANETs so that the users can access them using their own mobile devices while roaming around the University campus. As in any network, it is of utmost importance to provide a secure infrastructure. One way to increase security is to provide short-term membership certificates to users. Therefore, MANET restrictions and application domain requirements (in this case the academia particularities) must be considered when developing a security/trust service.

Generally, an academic environment is considered as a non-hostile environment characterised by benignity. Usually, the users abide with the University rules and have no malicious intentions (i.e. there is no contender). Furthermore, most of the devices used to access MANET are small and light-duty

devices, and, therefore, have limited power and storage capabilities. Therefore, performance and security are considered equally and moderately important features in this setting. Consequently, the configurations of the simulation model should comply with these characteristics. The factors considered for modelling the MANETs in Academia case scenario are described below.

Requirements and Settings:

The following factors are necessary to be taken into account for the security service that needs to be deployed in MANETs within the academic context:

- The good performance is needed:
  - Both service availability and reliability in this context appear to be more important than service quickness.
- Most devices are smartphones which have limited bandwidth, storage and power.
- The standard protection is essential.
- Academia can be characterised as a non-hostile environment.
- The case of node churning is considered as the adopted *Churn* model developed on the node lifetime in the academic domain (campus), see Section 5.3.4.6 for more details.

Results (Decision Implementation):

Based on the settings of this scenario, the results of the *Churn* and *No-Churn* cases should be taken into consideration in this decision implementation. Following the ranking approach illustrated in Section 6.5, both performance and security strength criteria should be evaluated. This is due to the fact that these two dimensions consist of more than one measurement in the second level of the criteria hierarchy. To begin with the performance measurements, relying on Table

6-5 (i.e. including ranking score.) and Table 6-7 (i.e. including importance weights for performance), a weighted average for each alternative can be estimated for the two different network scenarios (*Churn* and *No-Churn*), as shown in Table 6-8 and 6-9.

	The Churn Scenario (score between 0 and 100%) for each performance measurement SR (Success Ratio), FF (Failure Frequency), RTTD (RTT-STDev)				
Rank Reciprocal (RR) Weights	w_{SR}	w_{FF}	w_{RTT}	w_{RTTD}	**Weighted AVG (Scores)**
	35.29% ↓	35.29% ↓	17.65% ↓	11.76% ↓	
Server Architecture	**SR.**	**FF.**	**RTT**	**RTTD.**	**Total Rank [%]**
CAS	20.56	71.73	88.57	84.82	**58.2**
TAS	0.00	66.42	0.00	0.00	**23.4**
D\CAS	62.25	81.75	92.58	88.23	**77.5**
TAS_DAS	26.62	73.20	20.52	79.00	**48.1**
CAS_TAS_AAO	20.42	72.38	90.57	86.28	**58.9**
CAS_TAS_IPS	21.31	72.32	86.68	80.18	**57.8**
CAS_TAS_DAS_AAO	53.17	79.70	62.00	57.44	**64.6**
CAS_TAS_DAS_IPS	46.61	77.42	38.45	20.16	**52.9**

Table 6-8: The final ranking estimation for the performance dimension in the *Churn* network scenario

	The Churn Scenario (score between 0 and 100%) for each performance measurement SR (Success Ratio), FF (Failure Frequency), RTTD (RTT-STDev)				
Rank Reciprocal (RR) Weights	w_{SR}	w_{FF}	w_{RTT}	w_{RTTD}	**Weighted AVG (Scores)**
	35.29% ↓	35.29% ↓	17.65% ↓	11.76% ↓	
Server Architecture	**SR.**	**FF.**	**RTT**	**RTTD.**	**Total Rank [%]**
CAS	69.64	73.30	92.08	87.64	**77.0**
TAS	59.60	64.91	89.90	87.97	**70.2**
D\CAS	84.52	80.85	94.79	90.95	**85.8**
TAS_DAS	73.32	73.55	71.31	61.12	**71.6**
CAS_TAS_AAO	71.61	72.49	93.19	89.15	**77.8**
CAS_TAS_IPS	74.21	71.32	86.17	71.22	**75.0**
CAS_TAS_DAS_AAO	81.01	79.22	82.07	66.05	**78.8**
CAS_TAS_DAS_IPS	81.16	76.49	70.75	40.22	**72.9**

Table 6-9: The final ranking estimation for the performance dimension in the *No-Churn* network scenario

Eventually, regardless of the authentication protocol in use, the *D\CAS* and *CAS_TAS_DAS_AAO* architectures respectively show the best performance among the other architectures in both *Churn* and *No-Churn* scenarios in this ranking evaluation. While, the *TAS* has the worst performance in all scenarios. In both scenarios, these two lists of total ranks represent the levels of achievements for each security alternative based on the performance requirements in the academic context. The alternative with the highest percentage is considered the best choice as this indicates that this alternative is able to meet all desirable requirements in the context.

Similarly, for the security strength dimension, the scores of security alternatives for every measurement, generated in Table 6-6, and the proposed weights for this dimension, shown in Table 7-7, are incorporated for calculate the weighted averages of all alternatives, as shown in Table 6-10. This table is proposed to be applicable for both *Churn* and *No-Churn* scenarios.

Rank Reciprocal (RR) Weights	$W_{AP\ Robust.}$	W_{SPF}	W_{SPC}	**Weighted AVG (Scores)**
	20% ↓	40% ↓	40% ↓	
Security Strength / Security architecture	**Authentication Protocol Robustness (1 – 99%)**	**SPF Resilience (1-99%)**	**SPC Resilience (1-99%)**	**Total Rank [%]**
1WP _CAS	1	1	20.6	**08.84**
1WP_TAS	1	20.6	99	**48.04**
1WP_D\CAS	1	99	1	**40.20**
1WP_TAS_DAS	1	40.2	79.4	**48.04**
1WP_CAS_TAS_AAO	1	59.8	40.2	**40.2**
1WP_CAS_TAS_IPS	1	59.8	40.2	**40.2**
1WP_CAS_TAS_DAS_AAO	1	79.4	59.8	**55.88**
1WP_CAS_TAS_DAS_IPS	1	79.4	59.8	**55.88**
2WP_CAS	50	1	20.6	**18.64**
2WP_TAS	50	20.6	99	**57.84**
2WP_D\CAS	50	99	1	**50.00**
2WP_TAS_DAS	50	40.2	79.4	**57.84**
2WP_CAS_TAS_AAO	50	59.8	40.2	**50.00**
2WP_CAS_TAS_IPS	50	59.8	40.2	**50.00**
2WP_CAS_TAS_DAS_AAO	50	79.4	59.8	**65.68**
2WP_CAS_TAS_DAS_IPS	50	79.4	59.8	**65.68**
3WP_CAS	99	1	20.6	**28.44**
3WP_TAS	99	20.6	99	**67.64**
3WP_D\CAS	99	99	1	**59.80**
3WP_TAS_DAS	99	40.2	79.4	**67.64**

3WP_CAS_TAS_AAO	99	59.8	40.2	**59.8**
3WP_CAS_TAS_IPS	99	59.8	40.2	**59.8**
3WP_CAS_TAS_DAS_AAO	99	79.4	59.8	**75.48**
3WP_CAS_TAS_DAS_IPS	99	79.4	59.8	**75.48**

Table 6-10: The final ranking estimation for the security strength dimension

It appears that the *CAS_TAS_DAS_AAO, CAS_TAS_DAS_IPS, TAS_DAS* and *TAS* architectures with *3WP* or *2WP* protocols respectively present the best security strength among the other architectures. On the other side, most architectures with *1WP* protocols show lower security strength especially the *D\CAS*, *CAS_TAS_AAO*, *CAS_TAS_IPS* and *CAS* architectures, as demonstrated in Table 6-10. After completing the evaluation of the second level of criteria hierarchy, in this stage, the final rank of all top-level dimensions (i.e. performance, security strength, and MANET constraints) can be calculated for every alternative. However, the importance of these three dimensions differs based on the requirements and settings of the application being involved. Hence, according to the requirements of this real case scenario in an academic context, it can be concluded that service performance and security are equally important. On the other hand, since most devices which operate the MANET are presumed to be limited low-end, it is very important that communication and computation (i.e. communication overhead metric) are as lower as possible. Also, this MANET constraint is considered much more important than performance and security since this network relies on mobile devices for operation. As a result, the corresponding weights of importance for the dimensions can be evaluated based on the RRW method: (1) $w_{performance}$ = 25%, (2) $w_{SecurityStrength}$ = 25% and (3) $w_{communicationOverhead}$ = 50%. These weights are also applied for both *Churn* and *No-Churn* network cases. Depending on Table 6-5, 6-8 and 6-10, two lists of final ranks for the security alternatives can be created, as shown in Table 6-11 and 6-12.

Rank Reciprocal (RR) Weights	$w_{performance}$ 25% ↓	$w_{com.Overhead}$ 50% ↓	$w_{SecurityStrength}$ 25% ↓	*Churn* Scenario Weighted AVG (Scores) Final Rank [%] "Achievement Score" Highest → Lowest
Dimension / Security architecture	Performance Total Rank	Communication Overhead Total Rank	Security Strength Total Rank	
3WP_CAS_TAS_IPS	57.77	82.13	59.8	**70.46**
2WP_CAS_TAS_IPS	57.77	84.57	50.00	**69.23**
1WP_CAS_TAS_IPS	57.77	88.84	40.20	**68.91**
3WP_CAS	58.18	83.56	28.44	**63.44**
2WP_CAS	58.18	85.81	18.64	**62.11**
1WP_CAS	58.18	89.91	08.84	**61.71**
3WP_CAS_TAS_DAS_IPS	52.93	56.80	75.48	**60.50**
2WP_D/CAS	77.54	57.05	50.00	**60.41**
1WP_CAS_TAS_DAS_IPS	52.93	64.52	55.88	**59.46**
2WP_CAS_TAS_DAS_IPS	52.93	59.43	65.68	**59.37**
3WP_D/CAS	77.54	49.85	59.80	**59.26**
1WP_D/CAS	77.54	59.54	40.20	**59.20**
1WP_CAS_TAS_AAO	58.88	57.44	40.20	**53.49**
2WP_CAS_TAS_AAO	58.88	51.79	50.00	**53.12**
3WP_CAS_TAS_AAO	58.88	44.43	59.80	**51.88**
1WP_CAS_TAS_DAS_AAO	64.59	37.84	55.88	**49.04**
2WP_CAS_TAS_DAS_AAO	64.59	29.93	65.68	**47.54**
3WP_CAS_TAS_DAS_AAO	64.59	23.88	75.48	**46.96**
2WP_TAS_DAS	48.14	21.92	57.84	**37.45**
3WP_TAS_DAS	48.14	15.98	67.64	**36.94**
1WP_TAS_DAS	48.14	25.13	48.04	**36.61**
3WP_TAS	23.44	0.00	67.64	**22.77**
2WP_TAS	23.44	0.00	57.84	**20.32**
1WP_TAS	23.44	0.00	48.04	**17.87**

Table 6-11: The list of final security alternative ranks in the *Churn* setting for Scenario S1 (Academia)

Based on the percentage of alternatives' achievements (scores), both Table 6-11 and 6-12 produce different ordered lists of alternatives which enable MANET developers to choose the best nominated architectures for this specific scenario under the *Churn* or *No-Churn* settings. In the *Churn* setting, the *3/2/1WP_CAS_TAS_IPS* architectures respectively show the best achievement (approximately between 70.46 and 68.91%). This means that these architectures satisfy the most desired requirements (lower communication → better performance and security) in this scenario as presented in the green cells in Table 6-12. Also, the group of *3/2/1WP_CAS, 3/2/1WP_CAS_TAS_DAS_IPS* and

3/2/1WP_D\CAS architectures respectively comes as a second option in the list, as shown in the greenish-yellow cells within Table 6-12. Some of the other architectures demonstrate a medium level of achievements, such as *1/2/3WP_CAS_TAS_AAO*, *1/2/3WP_CAS_TAS_AAO*, etc. As expected, *3/2/1WP_TAS* architectures achieve the minimum requirements in this ordered list in the *Churn* setting. This is because these architectures are completely non-functional in this particular setting.

Rank Reciprocal (RR) Weights	$w_{performance}$ 25% ↓	$w_{com.Overhead}$ 50% ↓	$w_{SecurityStrength}$ 25% ↓	***No-Churn* Scenario** **Weighted AVG (Scores)** **Final Rank [%]** **"Achievement Score" Highest → Lowest**
Dimension / Security architecture	**Performance Total Rank**	**Communication Overhead Total Rank**	**Security Strength Total Rank**	
3WP_CAS_TAS_DAS_IPS	72.86	73.30	75.48	**73.74**
3WP_CAS_TAS_IPS	74.95	79.99	59.80	**73.68**
1WP_CAS_TAS_DAS_IPS	72.86	82.07	55.88	**73.22**
2WP_CAS_TAS_DAS_IPS	72.86	77.13	65.68	**73.20**
1WP_CAS_TAS_IPS	74.95	88.69	40.20	**73.13**
2WP_CAS_TAS_IPS	74.95	83.57	50.00	**73.02**
3WP_CAS	77.01	85.19	28.44	**68.96**
2WP_CAS	77.01	87.59	18.64	**67.71**
1WP_CAS	77.01	91.72	08.84	**67.32**
1WP_D\CAS	85.79	60.96	40.20	**61.98**
2WP_D\CAS	85.79	55.32	50.00	**61.61**
3WP_D\CAS	85.79	47.97	59.80	**60.38**
1WP_CAS_TAS_DAS_AAO	78.81	52.10	55.88	**59.72**
1WP_CAS_TAS_AAO	77.79	57.39	40.20	**58.19**
1WP_TAS	70.16	55.72	48.04	**57.41**
1WP_TAS_DAS	71.61	50.07	48.04	**54.95**
2WP_CAS_TAS_DAS_AAO	78.81	36.25	65.68	**54.25**
3WP_CAS_TAS_DAS_AAO	78.81	27.22	75.48	**52.18**
2WP_CAS_TAS_AAO	77.79	39.88	50.00	**51.89**
2WP_TAS	70.16	38.01	57.84	**51.01**
2WP_TAS_DAS	71.61	35.36	57.84	**50.05**
3WP_CAS_TAS_AAO	77.79	30.30	59.80	**49.55**
3WP_TAS	70.16	28.17	67.64	**48.53**
3WP_TAS_DAS	71.61	26.20	67.64	**47.91**

Table 6-12: The list of final security alternative ranks in the *No-Churn* setting for Scenario S1 (Academia)

Alternatively, in the *No-Churn* setting, both *3/2/1WP_CAS_TAS_DAS_IPS* and *3/2/1WP_CAS_TAS _IPS* architectures become in one group which is considered the best architectures group accomplishing almost all demands of this context (approximately 73%), as shown in the green cells in Table 6-12.Thereafter, the *3/2/1WP_CAS* architectures take the next place with approximately 67% achievement score among the other architectures under this setting, as displayed in the greenish-yellow cells. In the list, different security architectures using the *1WP* protocol show the in-between order (approximately between 62 and 57%), for example *1/2/3WP_D/CAS*, *1WP_CAS_TAS_DAS_AAO*, *1WP_CAS_TAS _AAO*, and *1WP_TAS* architectures respectively. The architectures which are characterised with the least success in achieving the requirements of this context are *3WP_CAS_TAS_AAO*, *3WP _TAS* and *3WP_TAS_DAS* architectures, as shown in Table 6-12.

Eventually, two- and three-level hierarchical security architectures, especially *3/2/1WP_CAS_TAS_IPS* and *3/2/1WP_CAS_TAS_DAS_IPS,* offer the best solutions for this scenario as the settings of this scenario are constrained with a low communication overhead and a standard level of security and performance.

6.6.1.2 The Second Scenario – MANETs in Emergency and Crisis Domains (S2)

In a crisis (i.e. physical disaster such as earthquake, flood, hurricane, etc. or man-made disaster such as terrorist attack, explosion, etc.), the three main emergency services (i.e. ambulance, fire, and police) cooperate in order to provide timely and efficient response. Communication among the various rescue teams is essential in emergency situations. In the same time, communication means that depend on network infrastructures are considered insufficient, especially in semi-urban or rural areas that are not well facilitated by network infrastructures. Therefore, MANETs have the potential to offer an infrastructure-less solution since they do not depend on any infrastructure in order to operate. In the case of emergency

response in disasters, this is an important advantage over traditional pre-established networks, such as cellular networks, that might not exist or are damaged.

The second case scenario assumes that the rescue team members are equipped with handheld devices, such as smartphones, tablets, laptops, etc., that can utilise communication technologies for building a MANET (e.g. Wi-Fi). The deployed services can facilitate information sharing and communication services, such as up-to-date news feed, real-time multimedia streaming (e.g. voice or video calling services such as Walkie-Talkie, Push-to-Talk, VoIP, etc.), and instant messaging. To accomplish secure utilisation of those resources and services, a security/trust system should be adopted. The security system will ensure the distribution of security credentials in order to allow users to access the MANET.

The life-critical nature of crisis management designates the importance of network performance for efficient communication while the security, although important, comes as a second priority. This is supported by the literature where, for example, Lien *et al.*(2009) and Jang *et al.* (2009) point out that performance, especially efficiency, is more important than security for the emergency response communications. However, some security aspects are considered essential for this case (e.g. integrity and authentication of users). Furthermore, the communication environment in the case of crisis management is considered a non-hostile environment that rarely has intended adversaries. Therefore, performance is considered a critical requirement. In the same time, integrity and authentication are considered more important security requirements than confidentiality. Thus, the configurations of the SSAM (i.e. simulation model) should consider these requirements. The factors considered for modelling the MANETs in the Emergency Services case scenario are described below.

Requirements and Settings:

The following factors have to be taken into consideration when the security service is required to be deployed in MANETs within an emergency and crisis contexts:

- There is a demand for high performance service as time is very critical in this context.
- The security strength should be acceptable.
- In emergency and crisis situations, it is expected to have a non-hostile environment.
- Many devices, exploited by the rescue team members, may suffer from problems of limited bandwidth, storage and power.

Results (Decision Implementation):

Similar to the decision application of the first scenario described in Section 6.6.1.1, Tables 6-8 and 6-10 are used in this scenario because the weights of performance and security strength metrics are presumed to be fixed for all scenarios in this study. On the other side, only the results under the *No-Churn* setting can be applicable to this scenario as the *Churn* setting (a lifetime node) is established on an academic context. Considering the prescribed requirements for this scenario, performance is much more important than security and communication because applications in this context are time-sensitive. Also, since MANETs in this scenario appear to be limited, the communication overhead is much important than security. Therefore, the corresponding weights of importance for the dimensions are reckoned via applying the RRW method: (1) $w_{performance}$ = 54.55%, (2) $w_{SecurityStrength}$ = 18.18% and (3) $w_{communicationOverhead}$ = 27.27%. Thereafter, by using the resultant values of Tables 6-5, 6-8 and 6-10, the list of final security alternative ranks is calculated, as shown in Table 6-13.

Rank Reciprocal (RR) Weights	$w_{performance}$ 54.55% ↓	$w_{com.Overhead}$ 27.27% ↓	$w_{SecurityStrength}$ 18.18% ↓	*No-Churn* Scenario Weighted AVG (Scores) Final Rank [%] "Achievement Score" Highest → Lowest
Dimension / Security architecture	Performance Total Rank	Communication Overhead Total Rank	Security Strength Total Rank	
3WP_CAS_TAS_IPS	74.95	79.99	59.80	**73.57**
3WP_CAS_TAS_DAS_IPS	72.86	73.30	75.48	**73.46**
2WP_CAS_TAS_IPS	74.95	83.57	50.00	**72.77**
2WP_CAS_TAS_DAS_IPS	72.86	77.13	65.68	**72.72**
1WP_CAS_TAS_IPS	74.95	88.69	40.20	**72.38**
1WP_CAS_TAS_DAS_IPS	72.86	82.07	55.88	**72.28**
2WP_D/CAS	85.79	55.32	50.00	**70.98**
3WP_D/CAS	85.79	47.97	59.80	**70.75**
1WP_D/CAS	85.79	60.96	40.20	**70.73**
3WP_CAS	77.01	85.19	28.44	**70.41**
2WP_CAS	77.01	87.59	18.64	**69.28**
1WP_CAS	77.01	91.72	8.84	**68.63**
1WP _CAS_TAS_DAS_AAO	78.81	52.10	55.88	**67.35**
1WP_CAS_TAS_AAO	77.79	57.39	40.20	**65.39**
2WP_CAS_TAS_DAS_AAO	78.81	36.25	65.68	**64.81**
3WP_CAS_TAS_DAS_AAO	78.81	27.22	75.48	**64.13**
2WP_CAS_TAS_AAO	77.79	39.88	50.00	**62.40**
1WP_TAS	70.16	55.72	48.04	**62.20**
3WP_CAS_TAS_AAO	77.79	30.30	59.80	**61.57**
1WP_TAS_DAS	71.61	50.07	48.04	**61.45**
2WP _TAS_DAS	71.61	35.36	57.84	**59.22**
2WP_TAS	70.16	38.01	57.84	**59.15**
3WP_TAS_DAS	71.61	26.20	67.64	**58.50**
3WP_TAS	70.16	28.17	67.64	**58.25**

Table 6-13: The final list of security alternative ranks in the *No-Churn* setting for Scenario S2 (Crisis Management)

The *3/2/1WP_CAS_TAS_DAS_IPS* and *3/2/1WP_CAS_TAS_IPS* architectures are considered the best alternatives in meeting all requirements for this scenario (approximately between 72% and 73%), as displayed in the green cells in Table 6-13. In spite of the high communication and fair security strength, the *2/3/1WP_D\CAS* architectures show fairly proper choice as expected

(approximately 70%). This is because these architectures are already characterised with the highest performance among all architectures. The *2/1WP_TAS_DAS, 2/3WP_TAS* and *3WP_TAS_DAS* architectures reveal the minimum achievement scores in this scenario. The different achievements of the rest of security architectures can be recognised by looking at Table 6-13.

6.6.1.3 The Third Scenario – MANETs in Military Domain (S3)

The third case scenario aims to study MANETs under the requirements of military missions. For the purpose of this experiment, the case of infantry troops using MANET for tactical-level communications is explored. MANET in this case can be used to provide services, such as situational awareness, positional awareness, command-and-control information systems (C2IS), SMS, and real-time multimedia streaming for voice push-to-talk (PTT), voice push-to-group (PTG), and video from surveillance monitors.

Battlefields are generally hostile environments where the enemy will attempt to intercept communications. Therefore, the soldiers need secure and resilient communication systems. In the same time, speedy communication and accurate access to information can prove to be life-saving to ground soldiers. Consequently, in this setting, both security and performance are considered highly important features for the MANET configurations. Furthermore, military equipment is usually high advanced technologically, therefore, it is assumed that the soldiers' gear includes handheld devices with no power and capacity limitations. The factors considered for modelling the MANETs in Military Missions case scenario are described below.

Requirements and Settings:

The following factors have to be taken into account once the security service needs to be designed and implemented in MANETs within the military context:

- Security is more important than performance in this context.

- The environment of this scenario is normally characterised with hostility (i.e. enemies).
- The good performance is necessary.
- In the military domain, most devices that are used, have no power and capacity limitations.

Results (Decision Implementation):

Similar to the decision application of the first and second scenarios in Sections 6.6.1.1 and 6.6.1.2, Tables 6-8 and 6-10 are involved in this scenario due to having fixed weights of performance and security strength metrics. However, only the results of the *No-Churn* setting are appropriate to this scenario as the *Churn* setting (a lifetime node) is defined only within an academic context (see Section 5.3.4.6). Based on the identified requirements of this scenario, security is much more important than performance and communication because of critical and hostile environments. On the other hand, MANETs in this context appear not to have resource restrictions in terms of power and computation. In this case, performance becomes more important than the communication overhead (i.e. the context of MANET constraints). Accordingly, the corresponding weights of importance for the dimensions in this scenario are estimated according to the RRW method: (1) $w_{performance}$ = 27.27% (2) $w_{SecurityStrength}$ = 54.55% and (3) $w_{communicationOverhead}$ = 18.18%. Then, by taking advantage of Tables 6-5, 6-8 and 6-10, the list of final security alternative ranks is generated as shown in Table 6-14.

In this scenario, the best architecture in terms of satisfying the desired requirements is the *3WP_CAS_TAS_DAS_IPS* architecture, as shown Table 6-14. As shown from its performance, communication and security analyses, this architecture assures the fulfilment of the high protection, high performance and fairly low communication. The *2WP_CAS_TAS_DAS_IPS, 3WP_CAS_ TAS_DAS_AAO* and *3WP_CAS_TAS_IPS* architectures respectively present great potential in this scenario apart from the top-rank architecture, as shown in Table

6-14. However, the *2/1WP_CAS* architectures have the lowest achievement scores, as anticipated. This is due to the fact that these architectures are the weakest architectures in the context of security and performance.

Rank Reciprocal (RR) Weights	$w_{performance}$ 27.27% ↓	$w_{com.Overhead}$ 18.18% ↓	$w_{SecurityStrength}$ 54.55% ↓	***No-Churn* Scenario** **Weighted AVG (Scores)** **Final Rank [%]** **"Achievement Score"** **Highest → Lowest**
Dimension / **Security architecture**	**Performance Total Rank**	**Communication Overhead Total Rank**	**Security Strength Total Rank**	
3WP_CAS_TAS_DAS_IPS	72.86	73.30	75.48	**74.37**
2WP_CAS_TAS_DAS_IPS	72.86	77.13	65.68	**69.72**
3WP_CAS_TAS_DAS_AAO	78.81	27.22	75.48	**67.61**
3WP_CAS_TAS_IPS	74.95	79.99	59.80	**67.60**
1WP_CAS_TAS_DAS_IPS	72.86	82.07	55.88	**65.27**
3WP_D\CAS	85.79	47.97	59.80	**64.74**
2WP_CAS_TAS_DAS_AAO	78.81	36.25	65.68	**63.91**
2WP_CAS_TAS_IPS	74.95	83.57	50.00	**62.91**
1WP_CAS_TAS_DAS_AAO	78.81	52.10	55.88	**61.45**
3WP_TAS_DAS	71.61	26.20	67.64	**61.19**
3WP_TAS	70.16	28.17	67.64	**61.15**
2WP_D\CAS	85.79	55.32	50.00	**60.73**
3WP_CAS_TAS_AAO	77.79	30.30	59.80	**59.34**
1WP_CAS_TAS_IPS	74.95	88.69	40.20	**58.49**
2WP_TAS	70.16	38.01	57.84	**57.59**
2WP_TAS_DAS	71.61	35.36	57.84	**57.51**
1WP_D\CAS	85.79	60.96	40.20	**56.41**
2WP_CAS_TAS_AAO	77.79	39.88	50.00	**55.74**
1WP_TAS	70.16	55.72	48.04	**55.47**
1WP_TAS_DAS	71.61	50.07	48.04	**54.84**
1WP_CAS_TAS_AAO	77.79	57.39	40.20	**53.58**
3WP_CAS	77.01	85.19	28.44	**52.00**
2WP_CAS	77.01	87.59	18.64	**47.10**
1WP_CAS	77.01	91.72	8.84	**42.50**

Table 6-14: The final list of security alternative ranks in the *No-Churn* setting for Scenario S3 (Military)

In conclusion, along with the criteria and application requirements, the results generated from applying the proposed ranking approach in different scenarios can be used as guidelines for MANET developers to find appropriate security

architectures which meet not only the security demands but also the performance and communication expectations for any specific application. Also, it is important to indicate that the security architectures with two- or three-level hybrid and hierarchical servers offer best alternatives in all scenarios being tacked, such as *1/2/3WP_CAS_TAS_DAS_IPS, 1/2/3WP_CAS_TAS_DAS_AAO*, and *1/2/3WP_ CAS_ TAS_IPS* architectures.

6.7 Conclusion

This chapter initially presented the evaluation of the model results obtained from conducting the performance and communication testing, for the proposed security architectures in SSAM under different network scenarios (i.e. the *Churn* and No-*Churn* settings). These performance and communication results were intended to be integrated in the dimensions of performance and MANETs constraints respectively in the proposed methodological approach. In addition, the security architectures were examined in a qualitative way based on three main security concerns (i.e. single-point-of-failure (SPF) and single-point-compromise (SPC) and authentication protocol weaknesses) for ranking purposes. The results of this particular security analysis were involved in the dimension of the security strength in the proposed approach.

Hence, the methodological approach, which was developed to evaluate security services in MANETs under different considerations, was applied incorporating the SSAM model for validation purposes. Also, in order to demonstrate the feasibility of the proposed approach in practical settings, three unique real case scenarios (i.e. academic, emergency, and military contexts) for MANETs were exploited. On the other side, in the proposed multi-dimensional methodology, a simple ranking approach is suggested and employed as a decision support framework. This ranking approach was established on achievement scoring (i.e. rating), reciprocal ranking weighting (RRW) and weighted averaging methods. The aim of this approach was to enable MANET developers to prioritise their security strength, communication and performance requirements so as to rank the available

security architecture alternatives in SSAM based on their appropriateness for a specified application. As a result of implementing this approach, three key lists which included the final ranks of security alternatives were generated based on the requirements and settings of three representative scenarios. These lists could allow security developers to define a proper security service policy matching the demands of the context being tackled. Last but not least, it is worth pointing out that the hybrid and hierarchical security architectures in SSAM provide the best alternatives among three focused scenarios, for example *1/2/3WP_CAS_TAS_DAS_IPS, 1/2/3WP_CAS_TAS_DAS_AAO,* and *1/2/3WP _CAS_TAS_IPS*.

Chapter 7: Conclusion

7.1 Thesis Overview

This thesis has seven chapters. This chapter presents the thesis contributions, and the research aim with its objectives revisited. It also highlights the limitations and future research directions for this research. An overview of the previous six chapters is outlined below:

Chapter 1 presented the introduction for this thesis, in which the motivation of this research was addressed. Several dimensions (i.e. strength, MANET constraints, application requirements, performance and operational levels), which have a significant impact on the security design of MANETs, were underlined. Therefore, the research aim in this study defined in this chapter was to develop a systematic approach which could support MANET developers to select the best appropriate server-based security infrastructures that satisfy the security, performance and context requirements of a given application in MANETs. Then, the six research objectives were identified to reach the aim of this research and the simulation approach was adopted to carry out this research. Also, a number of relevant definitions used in this thesis were well explained to avoid misinterpretation.

Chapter 2 highlighted the relevant subjects related to MANETs and their security issues. Initially, the discussion mainly indicated the unique MANET characteristics, the domains of MANET applications, the encountered MANET challenges for leveraging a better understanding about MANETs in practice. Then, most MANET-related security fundamentals and current trust models, which were involved in formulating a part of the proposed approach in this study (e.g. security mechanisms, MANET threats and attacks, etc.), were described.

This chapter set up a foundation for developing this study's approach by means of identifying a number of elements such as distinct MANET characteristics, MANET security aspects and available trust infrastructures.

Chapter 3 clarified the research methodology adopted in this thesis. The overall research approach for this research was presented initially. Then, this chapter established the background on a simulation technique and its methodology, which were used for testing the performance and communication of the SSAM model. Typically, the simulation methodology which was composed of three different stages (the problem definition, the model development and the decision support) was detailed and linked to this study. Furthermore, the verification validation and testing techniques (VV&T), were defined and exploited in this research approach for leveraging its credibility.

Chapter 4 had two parts (Part A and B). Part A demonstrated the proposed conceptual security framework for MANETs. This framework highlighted the important security-design-related elements and dimensions (i.e. an operational level, security/trust components, security strength, performance, the context of MANET constraints and application settings) associated with design and development of a trust/security infrastructure in MANETs. Also, based on this framework, a security-based methodology was proposed to enable MANET security developers to develop effective security solutions suiting different MANET contexts. Part B presented the proposal and design of the security model (SSAM). In accordance with the building blocks of a security/trust infrastructure in MANET, the SSAM components: a server architecture, an authentication protocol, a cryptosystem, a security credential, a strategy of calling, and MANET settings, were identified. Then, the model design of SSAM activities, communications and processes was illustrated. The aim of the SSAM model came with two key purposes. The first purpose was to offer a new security design for MANETs in the service operational level while the second purpose was to validate the suggested multi-dimension approach established from the framework proposed in Part A.

Chapter 5 discussed the implementation and experimentation of the SSAM model being conceptualised in Chapter 4 – Part B, considering the SSAM model assumptions. The OMNeT++ simulator was used to build the SSAM prototype which was composed of the definitions of necessary network, node and messages structures (i.e. *.ned and *.msg files) and the creation of C++ classes related to the SSAM activity, communication and process models mentioned in Chapter 4 – Part B. In addition, the key configurations and initialisations for this prototype were determined in order to properly conduct the performance and communication testing. Two different sets of configurations were described and justified in this chapter, the SSAM-related security and MANET-related network configurations. Eventually, the experimental design for performing the SSAM simulation was explained through identifying proper performance and communication metrics, specifying required test cases and experimental parameters, and deciding the requisite replications of each simulation experiment.

Chapter 6 provided an evaluation of the proposed approach. This was achieved primarily by interpreting the results of applying the SSAM model over MANETs. This chapter started with analysing the performance and communication results obtained from the simulation experiments of the SSAM under specific network scenarios (i.e. different node populations and node *No-Churn* and *Churn* settings). On the other side, the security strength for every security architecture in SSAM was evaluated based on two security criteria. The first one was associated with server architecture problems (i.e. Single-Point-of-Failure (SPF) and Single-Point-of-Compromise (SPC)). The second one represented the level of protocol robustness which was be examined by checking whether an authentication protocol satisfied certain protection requirements. The simple ranking and selecting method (i.e. includes achievement scoring, reciprocal ranking weighting (RRW) and weighted averaging) was utilised for facilitating the ranking of different security architectures in the SSAM model, based on certain criteria (i.e. different requirements). The applicability of the proposed methodology in this study was substantiated by using three distinct real case scenarios. The results generated from implementing this methodology would help MANET developers

to choose the best security alternatives matching the requirements of the context being considered.

7.2 Research Contribution

The study of this thesis produces three significant contributions. These contributions are distinguished with different themes (a methodology, a framework and SSAM) and analysed next.

7.2.1 Methodology Contribution

The main contribution of this study is a methodological approach whose mission is to describe how to evaluate and find the best suited security architecture for MANETs in a particular operational level that comply with *security strength*, *performance* and *application context* requirements and also handles *MANET constraints* effectively. The structure of this approach is established on the conceptual multi-dimensional security framework which is considered the second contribution in this thesis. This approach consists of four well-organised steps (see Section 4.5, 6.4 and 6.5) which can be used as a roadmap for MANET developers to produce a set of guidelines that can support decision making on selecting an appropriate security solution fitting their applications in MANETs. In this approach, simulation and certain ranking techniques are incorporated for implementation and evaluation purposes. This proposed methodology can contribute to the design evaluation knowledge for MANETs.

7.2.2 Security Framework Contribution

The conceptual multi-dimensional security framework for MANETs proposed in this research is the second contribution. This well-structured framework is intended to help to recognise the roles of different dimensions (i.e. *operational*

level, *security components*, *security strength*, *performance*, and the *context* of *MANET constraints and application settings*) in the development of a security/trust infrastructure at the *service level* (or any *operational level*) of MANETs. Also, this framework can be used to establish a better understanding about relationships and dependencies among those dimensions. Finally, the framework leverages a design approach (i.e. building blocks of a security/trust infrastructure) which enables MANET security developers to design and standardise an effective security solution for MANETs.

7.2.3 SSAM Contribution (Model and Tool)

Both SSAM model and its implementation (i.e. a simulation tool) are considered the third contribution in this research. Firstly, the design of SSAM, which is based on Server-based Security Architectures for MANETs (SSAM), is created to offer a suite of different security architectures. These particular security architectures can typically be applied in the service level of MANETs for deploying a certain security/trust service. In addition, the proposed security architectures are established upon four different types of authority servers (i.e. *CAS*, *TAS*s, *D\CAS* and *DASs*) and the three different standard authentication protocols (i.e. *X509 1WP*, *2WP* and *3WP*) for providing membership certificates to user nodes in MANETs. The novelty of this model is to present versatile security architectures, especially the hybrid and hierarchical ones (e.g. *CAS_TAS, CAS_TAS_DAS*, and *TAS_DAS*) catering for an enhancement in security service availability and utilisation. Besides, two new strategies of calling (i.e. All At Once – *AAO* and In Priority Sequence – *IPS*) are suggested for the hybrid and hierarchical security architectures for flexibility purposes.

Secondly, the SSAM tool offers an open simulation model which is implemented in C++ using the OMNeT++ simulator for performance and communication testing. This tool can be reused by MANET practitioners in other scenarios or be furthered developed. However, this tool also promotes a new

feature to the area of MANET simulation through considering the "*No-Churn*" and "*Churn*" models which represent the cases of growing and shrinking MANETs, stemming from node joining (i.e. arrivals) and node churning (i.e. a node lifetime and leaving). In addition, incorporating two different fixed and base-2 exponential interval models for the re-authentication process in the SSAM tool (as shown in Section 5.3.3.2) is considered advantageous as this can improve the success rate in calling server.

7.3 The Research Aim and Objectives Revisited

There are a number of different scopes affecting the design of SSAM (i.e. server architectures and authentication protocols, etc.) in the service level of MANETs. Developing a security solution for a given MANET often requires an insight into different issues which are an operational level, security, performance and the contexts of MANET constraints and application settings. Therefore, the main aim of this research is to develop a methodological approach that enables MANET developers to choose the best appropriate server-based security architecture that satisfies the security, performance and context requirements of a given application. Table 7-1 present how the core chapter's achievements of this thesis have successfully met the claimed research objectives specified in Section 1.2.

Objective	Chapter Achievements
Objective 1: Conduct a literature review to evaluate the current and most common security fundamentals and authoritarian trust models in the MANET domain to highlight their capabilities, requirements and limitations which will be a cornerstone for developing the proposed approach in the aim of this study	The first objective was achieved in Chapter 2: Several issues, related to MANETs (characteristics, challenges, and applications) and security (security requirements, threats and techniques) were highlighted. Also, various exiting credential-based trust models were analysed, shown in Figure 2-6.

Objective 2: Design a conceptual security framework for identifying the building blocks of the proposed approach of this study: the security operational level design; the security/trust infrastructure; security strength; performance; context and addressing the relationship between them as well as establishing this approach.	The second objective was achieved in Chapter 4 –part A: The conceptual security framework was developed, based on considering operational security level design (shown in Figure 4-2), the building blocks of the security/trust infrastructure (shown in Figure 4-3), the influential dimensions of security strength, performance and MANET context (shown in Figure 4-4). This framework constructed the proposed methodological approach (shown in Figure 4-5).
Objective3: Design a new security model of server-based security architectures for MANETs (SSAM), based on the components of the security/trust infrastructure for the service level of MANETs and also for checking the applicability of the proposed approach designed in pervious objectives.	The third objective was achieved in Chapter 4 –part B: The SSAM model was designed (shown in Figure 4-6) and its components were formulated (i.e. server architecture, authentication protocol, cryptosystem, credential, strategy of calling and MANET settings). Also, the activity, communication and process models for SSAM were presented.
Objective 4: Develop a simulation model for the proposed server-based security architectures for MANETs (SSAM) and experimentally test them under different scenarios using a	The forth objective was achieved in Chapter 5: The SSAM was implemented using the C++-based OMENT++ simulator.

network simulator in order to understand the cost of applying them on the performance and communication of MANETs, i.e. to create a performance evaluation model.	Important configurations were defined to prepare the model for experimentation. Several experiments were performed to compare the performance and communication of different security architectures in SSAM under different MANET scenarios.
Objective 5: Develop the proposed approach based on the results collected from the simulation experiments from the previous objective and the consideration of the other security strength and context requirements.	These objectives were achieved in Chapter 6: The results from the conducted experimentation in pervious chapter were analysed and presented. The Security strength in SSAM was examined. The methodological approach was validated by applying SSAM in three different real case scenarios.
Objective 6: Evaluate the proposed approach through the use of real case scenarios so as to demonstrate and validate the benefit and value of the approach.	

Table 7-1: Research Objectives Vs Chapter Achievements

7.4 Research Limitations and Future Directions

This research was conducted in the view of some assumptions and considerations that entail some research limitations listed as follows:

1- Nevertheless, specific security architectures in SSAM primarily comprising of different server architectures and authentication protocols for MANETs was sensibly nominated to be evaluated for this study; there is a study limitation of the number of those evaluated architectures. It is clear that there are other

security architectures (e.g. clustered ones) and protocols (e.g. SSL\TLS, etc.) that can be covered and expand the scope of security infrastructures in this research.

2- In this study, only one type of a cryptosystem (i.e. asymmetric RSA) and one type of a credential (i.e. *X.509-v3* certificate) were exploited with a generalised processing cost for performance testing although there are several types of those cryptographic functions with different computation and configurations costs.

3- Using limited case scenarios and MANET settings that incorporated in the simulation model of the SSAM model, for examples:

   - Covering only one type of a mobility pattern – i.e. the steady-state random waypoint mobility.
   - Authority servers were not mobile. Also, the distributed and hybrid server architectures used a fixed and small number of distributed servers as their servers are presumed to exist in a particular region within the playground.
   - Nodes in use of security architecture in SSAM have followed a general model of arrivals and churn (e.g. Poisson Process) because of the lack of real dataset about node interest that represents the node behaviour. There is a demand of obtaining an empirical dataset which may be translated into the statistical model to be used in simulation of MANET node lifecycle.

Apart from limitation discussed above which may require a future work, there are also other future research opportunities that would extend this study as follows:

- Using other new decision making methods for ranking and selection in the evaluation stage of the proposed methodological approach.
- Investigating the effect of applying the behaviour-monitored trust management along with the credential-based trust management that has been covered in this study on network performance and communication.

- In this study, investigating SSAM model was only on a side of obtaining a membership certificates from particular security servers. However, it is vital to take into account the use of these certificates (between user nodes and service provider nodes) to complete the picture of the usage side.
- As this study focuses on a usage side of the SSAM model, it is important to consider the management side of the SSAM model, such as handling key server share update and server aliveness.
- SSAM Future research can be directed towards enhancing the capabilities of the SSAM simulation prototype by upgrading it with new features and settings, for example:
  - Investigate power consumption in MANET nodes (i.e. battery models tailored with cryptographic processing and communications costs).
  - Test SSAM under different traffic models (multimedia streaming, web browsing etc.).
  - Apply other authentication communication protocols, such as SSL\TLS with TCP, etc.
  - Tackle different numbers of servers in the distributed security architectures in SSAM as this study considered only maximum seven servers.
  - Make use of other mobility patterns especially traced ones such as Gauss Markov, Manhattan Grid, and Reference Point Group Mobility models.
  - Take advantage of other different transport protocols - e.g. TCP, etc. and routing protocols - e.g. DSR, OLSR, DSDV, etc.

  However, extending SSAM may entail some complexities and problems. Even though the SSAM tool is modular-based, adapting a new module to this tool, especially for handling more functions of routing, mobility, churn and power may require some considerations, and few new code amendments and configurations. For example, using proactive routing protocols (e.g. OLSR and DSDV) in a MANET needs to consider the time delay for updating a routing table of a new joining node, before this node becomes a part of the network

and can call a certain set of security servers within a pre-deployed security architecture. Involving other different protocols and security architectures in the SSAM tool might introduce new components (e.g. additional messages or servers) which need to be integrated and consistently linked up with the existed components of SSAM tool. Therefore, it is important to make sure that any new amendment to a certain module in the SSAM tool would not conflict with the functionality of other modules.

7.5 Concluding Remarks

Security for MANETs is still a dilemma that obstructs development and acceptance. This research has basically investigated the variety of existing security architectures in the SSAM model to create more in-depth understanding about their suitability into different MANET contexts at the service operational level. The approach of this study is intended to provide a strategy along with guidelines for how to consider the better security architecture that satisfies the settings and requirements of given MANET context. This arguably would aid MANET developers in their security system development and deployment.

References

Abdulrahman, A., (1997). The PGP Trust Model. *The Journal of Electronic Commerce*.

Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J. & Levkowetz, H., (2004):*Extensible authentication protocol (EAP).* RFC 3748, June. Available from:< http://tools.ietf.org/html/rfc3748 > [Accessed 1/12/2013].

Abusalah, L., Khokhar, A. & Guizani, M., (2008). A survey of secure mobile Ad Hoc routing protocols. *Communications Surveys & Tutorials, IEEE,* 10(4)**,**pp. 78-93.

Aivaloglou, E., Gritzalis, S. & Skianis, C., (2006). Trust Establishment in Ad Hoc and Sensor Networks. *In* Lopez, J. (ed.) *Critical Information Infrastructures Security.* Springer Berlin Heidelberg,4347,pp.179-194.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. & Cayirci, E., (2002). Wireless sensor networks: a survey. *Computer Networks,* 38(4)**,**pp. 393-422.

Al-Bayatti, A. H., Zedan, H. & Cau, A., (2009). Security solution for mobile ad hoc network of networks (manon). *In*: *Networking and Services, 2009. ICNS'09. Fifth International Conference on*IEEE,pp. 255-262.

Aljnidi, M. & Leneutre, J., (2007). Towards an Autonomic Security System for Mobile Ad Hoc Networks. *In*: *Information Assurance and Security, 2007. IAS 2007. Third International Symposium on*,pp. 29-32.

Aschenbruck, N., Ernst, R., Gerhards-Padilla, E. & Schwamborn, M., 2010. BonnMotion: a mobility scenario generation and analysis tool. *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques.* Torremolinos, Malaga, Spain: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering),pp.1-10.

Asokan, N. & Ginzboorg, P., (2000). Key agreement in ad hoc networks. *Computer Communications,* 23(17)**,**pp. 1627-1637.

Balakrishnan, V. & Varadharajan, V., (2005). Designing secure wireless mobile ad hoc networks. *In*: *Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on*,pp. 5-8 vol.2.

Balci, O., (1990). Guidelines for successful simulation studies. *In*: *Simulation Conference, 1990. Proceedings., Winter*,pp. 25-32.

Balci, O., (1994). Validation, verification, and testing techniques throughout the life cycle of a simulation study. *Annals of Operations Research,* 53(1)**,**pp. 121-173.

Balfanz, D., Smetters, D., Stewart, P. & Wong, H., (2002). Talking to strangers: Authentication in ad-hoc wireless networks. *In*: *Symposium on Network and Distributed Systems Security (NDSS'02), San Diego, California*.

Bechler, M., Hof, H. J., Kraft, D., Pahlke, F. & Wolf, L., (2004). A cluster-based security architecture for ad hoc networks. *In*: *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*,pp. 2393-2403.

Boukerche, A., Zarrad, A. & Araujo, R. B., (2010). A Cross-Layer Approach-Based Gnutella for Collaborative Virtual Environments over Mobile Ad Hoc Networks. *Parallel and Distributed Systems, IEEE Transactions on,* 21(7)**,**pp. 911-924.

Boyd, C. & Mathuria, A., (1998). Key establishment protocols for secure mobile communications: A selective survey. *In* Boyd, C. & Dawson, E. (eds.) *Information Security and Privacy.* Springer Berlin Heidelberg,1438,pp.344-355.

Boyd, C. & Mathuria, A., (2003). *Protocols for Authentication and Key Establishment*: Springer-Verlag Berlin Heidelberg.

Bredel, M. & Bergner, M., (2009). On the accuracy of ieee 802.11 g wireless lan simulations using omnet++. *In*: *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering),pp. 81.

Broch, J., Maltz, D. A., Johnson, D. B., Hu, Y.-C. & Jetcheva, J., (1998). A performance comparison of multi-hop wireless ad hoc network routing protocols. *In*: *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*ACM,pp. 85-97.

Buchegger, S. & Le Boudec, J.-Y., (2002). Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. *In*: *Parallel, Distributed and Network-based Processing, 2002. Proceedings. 10th Euromicro Workshop on*IEEE,pp. 403-410.

Burrows, M., Abadi, M. & Needham, R., (1990). A logic of authentication. *ACM Transcations on Computer Systems,* 8(1)**,**pp. 18-36.

Camp, T., Boleng, J. & Davies, V., (2002). A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing,* 2(5)**,**pp. 483-502.

Capkun, S., Buttyan, L. & Hubaux, J.-P., (2002). Small worlds in security systems: an analysis of the PGP certificate graph. *In*: *Proceedings of the 2002 workshop on New security paradigms*, Virginia Beach, Virginia: ACM,pp. 28-35.

Capkun, S., Buttyan, L. & Hubaux, J. P., (2003). Self-organized public-key management for mobile ad hoc networks. *Mobile Computing, IEEE Transactions on,* 2(1)**,**pp. 52-64.

Carey, N., (2005). Establishing pedestrian walking speeds. *Karen Aspelin, Portland State University*.

Carvalho, M., (2008). Security in Mobile Ad Hoc Networks. *Security & Privacy, IEEE,* 6(2)**,**pp. 72-75.

Cayirci, E. & Rong, C., (2008). *Security in wireless ad hoc and sensor networks*: John Wiley & Sons.

Chadha, R. & Kant, L., (2008). *Policy-Driven Mobile Ad hoc Network Management*: John Wiley & Sons.

Chakeres, I. D. & Belding-Royer, E. M., (2004). AODV routing protocol implementation design. *In*: *Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on*,pp. 698-703.

Chlamtac, I., Conti, M. & Liu, J. J. N., (2003). Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks,* 1(1)**,**pp. 13-64.

Cho, J.-H., Swami, A. & Chen, I.-R., (2011). A Survey on Trust Management for Mobile Ad Hoc Networks. *Communications Surveys & Tutorials, IEEE,* 13(4)**,**pp. 562-583.

Choi, H.-D., Park, H.-H. & Woo, M., (2006). An Enhanced Gnutella for Ad-Hoc Networks. *In*: *Systems and Networks Communications, 2006. ICSNC '06. International Conference on*,pp. 3-3.

Chokhani, S., Ford, W., Sabett, R., Merrill, C. & Wu, S., (2003). Internet X509 PKI Certificate Policy and Certification Practices Framework. *RFC−3647*.

Clausen, T., Jacquet, P., Adjih, C., Laouiti, A., Minet, P., Muhlethaler, P., Qayyum, A. & Viennot, L., (2003). Optimized link state routing protocol (OLSR). *RFC-3626, IETF*.

Cocorada, S., (2008). An IEEE 802.11 g simulation model with extended debug capabilities. *In*: *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering),pp. 81.

Conti, M. & Giordano, S., (2007). Multihop Ad Hoc Networking: The Theory. *Communications Magazine, IEEE,* 45(4)**,**pp. 78-86.

Conti, M., Maselli, G., Turi, G. & Giordano, S., (2004). Cross-layering in mobile ad hoc network design. *Computer,* 37(2)**,**pp. 48-51.

Cook, K., (2001). *Trust in society*: Russell Sage Foundation.

Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R. & Polk, W., (2008):*Internet X. 509 Public Key Infrastructure–Certificate and CRL Profile*. IETF RFC 5280 Available from:< https://tools.ietf.org/html/rfc5280 > [Accessed 30/11/2014].

Cordeiro, C. D. M. & Agrawal, D. P., (2011). *Ad hoc and sensor networks: theory and applications*: World Scientific.

Corson, M. S., Macker, J. P. & Cirincione, G. H., (1999). Internet-based mobile ad hoc networking. *Internet Computing, IEEE,* 3(4)**,**pp. 63-70.

Corson, S. & Macker, J., (1999). *Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations*: RFC Editor.

Cox, M., Engelschall, R., Henson, S. & Laurie, B., (2002):*The openssl project*. Available from:< https://www.openssl.org/, https://www.openssl.org/docs/ssl/SSL_CTX_set_max_cert_list.html > [Accessed 29/05/2014].

Dawoud, D., Gordon, R. L. & Suliman, A., (2011). Trust Establishment in Mobile Ad Hoc Networks: Key Management. *Mobile Ad hoc Networks: Application: InTech.*

Desmedt, Y. & Frankel, Y., (1990). Threshold Cryptosystems. *In*: *Proceedings of the 9th Annual International Cryptology Conference on Advances in Cryptology*Springer-Verlag,pp. 307-315.

Desmedt, Y. & Frankel, Y., (1992). Shared generation of authenticators and signatures. *In* Feigenbaum, J. (ed.) *Advances in Cryptology — CRYPTO '91*. Springer Berlin Heidelberg,576,pp.457-469.

Dey, A. K., (2001). Understanding and Using Context. *Personal Ubiquitous Comput.,* 5(1)**,**pp. 4-7.

Dierks, T., (2008). The transport layer security (TLS) protocol version 1.2.

Djenouri, D., Khelladi, L. & Badache, A. N., (2005). A survey of security issues in mobile ad hoc and sensor networks. *Communications Surveys & Tutorials, IEEE,* 7(4)**,**pp. 2-28.

Dong, Y., Sui, A.-F., Yiu, S. M., Li, V. O. K. & Hui, L. C. K., (2007). Providing distributed certificate authority service in cluster-based mobile ad hoc networks. *Computer Communications,* 30(11–12)**,**pp. 2442-2452.

Dossogne, J., Lafitte, F. & Heule, D. V., 2013. Secure and practical threshold RSA. *Proceedings of the 6th International Conference on Security of Information and Networks.* Aksaray, Turkey: ACM,pp.79-85.

Eriksson, J., Faloutsos, M. & Krishnamurthy, S., (2005). Routing Scalability in MANETs. *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks, Wu, J.(Ed.). Taylor and Francis, New York***,**pp. 17-34.

Freebersyser, J. A. & Leiner, B., (2001). A DoD perspective on mobile Ad hoc networks. *In Ad hoc networking.* Addison-Wesley Longman Publishing Co., Inc.,pp.29-51.

Gennaro, R., Halevi, S., Krawczyk, H. & Rabin, T., (2008). Threshold RSA for Dynamic and Ad-Hoc Groups. *In* Smart, N. (ed.) *Advances in Cryptology – EUROCRYPT 2008.* Springer Berlin Heidelberg,4965,pp.88-107.

Gennaro, R., Rabin, T., Jarecki, S. & Krawczyk, H., (2000). Robust and Efficient Sharing of RSA Functions. *Journal of Cryptology,* 13(2)**,**pp. 273-300.

Giannoulis, S., Antonopoulos, C., Topalis, E., Athanasopoulos, A., Prayati, A. & Koubias, S., (2009). TCP vs. UDP Performance Evaluation for CBR Traffic On Wireless Multihop Networks. *Simulation,* 14**,**pp. 43.

Goyal, P., Parmar, V. & Rishi, R., (2011). Manet: vulnerabilities, challenges, attacks, application. *IJCEM International Journal of Computational Engineering & Management,* 11**,**pp. 32-37.

Gruber, I., Schollmeier, R. & Kellerer, W., (2004). Performance evaluation of the mobile peer-to-peer service. *In*: *Cluster Computing and the Grid, 2004. CCGrid 2004. IEEE International Symposium on*,pp. 363-371.

Guarnera, M., Villari, M., Zaia, A. & Puliafito, A., (2002). MANET: possible applications with PDA in wireless imaging environment. *In*: *Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on*,pp. 2394-2398 vol.5.

Gupta, A. K., Sadawarti, H. & Verma, A. K., (2010). Performance analysis of AODV, DSR & TORA routing protocols. *IACSIT international journal of Engineering and Technology,* 2(2)**,**pp. 226-231.

Hadjichristofi, G. C., Adams, W. J. & Davis, N. J., (2005a). A Framework for Key Management in Mobile Ad Hoc Networks. *International Journal of Information Technology,* 11(2).

Hadjichristofi, G. C., Adams, W. J. & Davis, N. J. I., (2005b). A framework for key management in mobile ad hoc networks. *In*: *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*,pp. 568-573 Vol. 2.

Halls, G. A., (1994). HIPERLAN: the high performance radio local area network standard. *Electronics & Communication Engineering Journal,* 6(6)**,**pp. 289-296.

Herrera, O. & Znati, T., (2007). Modeling Churn in P2P Networks. *In*: *Simulation Symposium, 2007. ANSS '07. 40th Annual*,pp. 33-40.

Hoeper, K. & Gong, G., (2004). *Models of Authentication in Ad Hoc Networks and Their Related Network Properties.* Technical Report CACR 2004-03, Centre for Applied Cryptographic Research, Waterloo, Canada. Available from:< http://www.comsec.uwaterloo.ca/~khoeper/cacr2004-03.pdf > [Accessed 02/11/2013].

Hogie, L., Bouvry, P. & Guinand, F., (2006). An Overview of MANETs Simulation. *Electronic Notes in Theoretical Computer Science,* 150(1)**,**pp. 81-101.

Hutchins, R. & Zegura, E. W., (2002). Measurements from a campus wireless network. *In*: *Communications, 2002. ICC 2002. IEEE International Conference on*,pp. 3161-3167 vol.5.

I'anson, C. & Mitchell, C., (1990). Security defects in CCITT Recommendation X. 509–The directory authentication framework. *Computer Communications Review,* 20(2)**,**pp. 30-34.

IEEE802, (2012). *IEEE 802.15 WPAN Task Group 1* [online]. Available from:< http://www.ieee802.org/15/pub/TG1.html > [Accessed 13/10/2013].

IEEE802.11, (1997):*Wireless LAN medium access control (MAC) and physical layer (PHY) specifications*.

IEEE802.11g, (2003):*Further Higher Data Rate Extension in the 2.4GHz Band, June*.

IEEE, *IEEE Standards Association* [online]. Available from:< http://standards.ieee.org/reading/ieee/interp/ > [Accessed 01/09/2013].

ISO, (1996):*ISO/IEC 7498-1: 1994 Information Technology-Open Systems Interconnection-Basic Reference Model: The Basic Model*. Available from:< http://www.ecma-international.org/activities/Communications/TG11/s020269e.pdf > [Accessed 21/01/2014].

ISO, I., (1989):*ISO 7498-2. Information processing systems—Open Systems Interconnection—Basic Reference Model. Part 2: Security Architecture*. Available from:< https://www.iso.org/obp/ui/#iso:std:iso:7498:-2:ed-1:v1:en > [Accessed 11/10/2013].

ISO/IEC 9798-2, I. O. F. S., (2008):*ISO/IEC 9798-2:, Information technology - Security techniques - Entity Authentication - Part 2: Mechanisms using symmetric encipherment algorithms, 2008. Third edition.*

ISO/IEC 9798-3, I. O. F. S., (1998/Cor.1:2009):*ISO/IEC 9798-3: Information technology - Security techniques - Entity Authentication - Part 3: Mechanisms using digital signature techniques. Technical Corrigendum 1, 2009.* Available from:< http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=31488 > [Accessed

ITU-T, (1989):*X. 509: The Directory Authentication Framework. International Telephone and Telegraph*. Available from:< http://www.itu.int/rec/T-REC-X.509-198811-S/en > [Accessed 04/11/2014].

ITU-T, (2008):*ITU-T Recommendation X.509 (ISO/IEC 9594-8) – The Directory: Public-key and attribute certificate frameworks*. Available from:< http://www.itu.int/rec/T-REC-X.509-200811-S > [Accessed 15/02/2013].

ITU, (1991):*ITU-T Rec. X.800 Security Architecture for Open Systems Interconnection for CCITT applications. ITU-T (CCITT) Recommendation*. Available from:< https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-X.800-199103-I!!PDF-E&type=items > [Accessed 10/11/2013].

ITU, (2003):*ITU-T Recommendation X.805 - Security Architecture for Systems Providing End-to-End Communications*. ITU. Available from:< https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-X.805-200310-I!!PDF-E&type=items > [Accessed 11/10/2013].

Jain, R., (1991). *The art of computer system performance analysis: techniques for experimental design, measurement, simulation and modeling* New York: John Willey.

Jang, H.-C., Lien, Y.-N. & Tsai, T.-C., 2009. Rescue information system for earthquake disasters based on MANET emergency communication platform. *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly.* Leipzig, Germany: ACM,pp.623-627.

Jansen, W., Karygiannis, T., Korolev, V., Gavrila, S. & Iorga, M., (2003). Policy Expression and Enforcement for Handheld Devices. *Technical report*.

Johnson, D. B. & Maltz, D. A., (1996). Dynamic source routing in ad hoc wireless networks. *In Mobile computing.* Springer,pp.153-181.

Jun-Zhao, S., (2001). Mobile ad hoc networking: an essential technology for pervasive computing. *In*: *Info-tech and Info-net, 2001. Proceedings. ICII 2001 - Beijing. 2001 International Conferences on*,pp. 316-321.

Keoh, S. L., Lupu, E. & Sloman, M., (2004). PEACE : a policy-based establishment of ad-hoc communities. *In*: *20th Annual Computer Security Applications Conference, 2004*,pp. 386-395.

Khakpour, A. R., Laurent-Maknavicius, M. & Chaouchi, H., (2008). WATCHMAN: An overlay distributed AAA architecture for mobile ad hoc networks. *In*: *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*IEEE,pp. 144-152.

Khun-Jush, J., Malmgren, G., Schramm, P. & Torsner, J., (2000). HIPERLAN type 2 for broadband wireless communication. *Ericsson Review* 77(2)**,**pp. 108-119.

Kim, S., Kim, J., Cheon, J. H. & Ju, S.-H., (2011). Threshold signature schemes for ElGamal variants. *Computer Standards & Interfaces,* 33(4)**,**pp. 432-437.

Klein-Berndt, L., (2010). Kernel AODV from National Institute of Standards and Technology (NIST). *Routing Algorithm for Wireless Sensor Networks,* 10(10)**,**pp. 9493-9511.

Klemm, A., Lindemann, C. & Waldhorst, O. P., (2003). A special-purpose peer-to-peer file sharing system for mobile ad hoc network. *In*: *IEEE Vehicular Technology Conference*,pp. 2758-2763.

Komninos, N., Vergados, D. & Douligeris, C., (2006). Layered security design for mobile ad hoc networks. *Computers & Security,* 25(2)**,**pp. 121-130.

Komninos, N., Vergados, D. & Douligeris, C., (2007). Authentication in a layered security approach for mobile ad hoc networks. *Computers & Security,* 26(5)**,**pp. 373-380.

Kong, J., Zerfos, P., Luo, H., Lu, S. & Zhang , L., (2001). Providing robust and ubiquitous security support for mobile ad-hoc networks. *In*: *Network Protocols, 2001. Ninth International Conference on*,pp. 251-260.

Kurkowski, S., Camp, T. & Colagrosso, M., (2005). MANET simulation studies: the incredibles. *SIGMOBILE Mob. Comput. Commun. Rev.,* 9(4)**,**pp. 50-61.

Larafa, S. & Laurent, M., (2009). Protocols for distributed AAA framework in mobile ad-hoc networks. *In*: *Proc. Workshop on Mobile and Wireless Networks Security (MWNS 2009)*Citeseer,pp. 75-86.

Larafa, S. & Laurent, M., (2011). Towards multiple-exchange protocol use in distributed AAA frameworks for more autonomy in MANETs. *In*: *Computers and Communications (ISCC), 2011 IEEE Symposium on*IEEE,pp. 856-863.

Law, A. M. & Mccomas, M. G., (1991). Secrets of successful simulation studies. *In*: *Simulation Conference, 1991. Proceedings., Winter*,pp. 21-27.

Lessmann, J., Janacik, P., Lachev, L. & Orfanus, D., (2008). Comparative Study of Wireless Network Simulators. *In*: *Networking, 2008. ICN 2008. Seventh International Conference on*,pp. 517-523.

Li, H. & Singhal, M., (2007). Trust management in distributed systems. *IEEE Computer,* 40(2)**,**pp. 45-53.

Lien, Y.-N., Jang, H.-C. & Tsai, T.-C., (2009). A MANET based emergency communication and information system for catastrophic natural disasters. *In*: *Distributed Computing Systems Workshops, 2009. ICDCS Workshops' 09. 29th IEEE International Conference on*IEEE,pp. 412-417.

Louta, M., Kraounakis, S. & Michalas, A., (2010). A survey on reputation-based cooperation enforcement schemes in wireless ad hoc networks. *In*: *Wireless Information Networks and Systems (WINSYS), Proceedings of the 2010 International Conference on*,pp. 1-4.

Luo, H., Kong, J., Zerfos, P., Lu, S. & Zhang, L., (2004). URSA: ubiquitous and robust access control for mobile ad hoc networks. *Networking, IEEE/ACM Transactions on,* 12(6)**,**pp. 1049-1063.

Luo, J., Hubaux, J. P. & Eugster, P. T., (2005). DICTATE: DIstributed CerTification Authority with probabilisTic frEshness for ad hoc networks. *Dependable and Secure Computing, IEEE Transactions on,* 2(4)**,**pp. 311-323.

Mallapur, S. V. & Patil, S. R., (2012). Survey on Simulation Tools for Mobile Ad-Hoc Networks. *International Journal of Computer Networks and Wireless Communications (IJCNWC).*

Manet, (2012). *IETF Group* [online]. Available from:< http://datatracker.ietf.org/wg/manet/charter/ > [Accessed 13/10/2012].

Marias, G. F., Georgiadis, P., Flitzanis, D. & Mandalas, K., (2006). Cooperation enforcement schemes for MANETs: a survey. *Wireless Communications and Mobile Computing,* 6(3)**,**pp. 319-332.

Martucci, L. A., Schweitzer, C. M., Venturini, Y. R., Carvalho, T. C. M. B. & Ruggiero, W. V., (2004). A Trust-Based Security Architecture for Small and Medium-Sized Mobile Ad Hoc Networks. *In*: *the 3rd Annual Mediterranean Ad Hoc Networking Workshop*Med-Hoc-Net,pp. 278–290.

Menezes, A. J., Van Oorschot, P. C. & Vanstone, S. A., (1996). *Handbook of applied cryptography*: CRC press.

Merwe, J. V. D., Dawoud, D. & Mcdonald, S., (2007). A survey on peer-to-peer key management for mobile ad hoc networks. *ACM Comput. Surv.,* 39(1)**,**pp. 1.

Messerges, T. S., Cukier, J., Kevenaar, T. a. M., Puhl, L., Struik, R. & Callaway, E., (2003). A security design for a general purpose, self-organizing, multihop ad hoc wireless network. *In*: *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, Fairfax, Virginia: ACM,pp. 1-11.

Moore, D. & Hebeler, J., (2002). *Peer-to-peer : building secure, scalable, and manageable networks* Berkeley, Calif: McGraw-Hill/Osborne.

Murthy, C. S. R. & Manoj, B. S., (2004). *Ad Hoc Wireless Networks: Architectures and Protocols*: Prentice Hall PTR, New Jersey

Nance, R., (1994). The Conical Methodology and the evolution of simulation model development. *Annals of Operations Research,* 53(1)**,**pp. 1-45.

Navidi, W. & Camp, T., (2004). Stationary distributions for the random waypoint mobility model. *Mobile Computing, IEEE Transactions on,* 3(1)**,**pp. 99-108.

Navidi, W., Camp, T. & Bauer, N., (2004). Improving the accuracy of random waypoint simulations through steady-state initialization. *In*: *Proceedings of the 15th International Conference on Modeling and Simulation*,pp. 319-326.

Nelson, B. L., Carson, J. S. & Banks, J., (2001). *Discrete event system simulation*: Prentice hall.

Neuman, B. C. & Ts'o, T., (1994). Kerberos: an authentication service for computer networks. *Communications Magazine, IEEE,* 32(9)**,**pp. 33-38.

Neuman, C., Yu, T., Hartman, S. & Raeburn, K., (2005):*RFC4120: The Kerberos Network Authentication Service (V5).* IETF. Available from:< https://www.ietf.org/rfc/rfc4120.txt >

Ngai, E. C. H. & Lyu, M. R., (2004). Trust- and clustering-based authentication services in mobile ad hoc networks. *In*: *Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on*,pp. 582-587.

Omar, M., Challal, Y. & Bouabdallah, A., (2007). NetTRUST: mixed NETworks Trust infrastRUcture baSed on Threshold cryptography. *In*: *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*,pp. 2-10.

OMNeT++, [online]. Available from:< http://www.omnetpp.org/ > [Accessed 1 Apr 2012].

Pal, A., Singh, J. & Dutta, P., (2011). A Study on the Effect of Traffic Patterns in Mobile Ad Hoc Network. *In* Abraham, A., Lloret Mauri, J., Buford, J., Suzuki, J. & Thampi, S. (eds.) *Advances in Computing and Communications.* Springer Berlin Heidelberg,190,pp.83-90.

Papadopouli, M., Shen, H. & Spanakis, M., (2005). Modeling client arrivals at access points in wireless campus-wide networks. *In*: *Local and Metropolitan Area Networks, 2005. LANMAN 2005. The 14th IEEE Workshop on*,pp. 6 pp.-6.

Perkins, C., Belding-Royer, E. & Das, S., (2003):*Ad hoc On-Demand Distance Vector (AODV) Routing RFC 3561*. IETF. Available from:< > [Accessed

Perkins, C. E. & Bhagwat, P., (1994). Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *In*: *ACM SIGCOMM Computer Communication Review*ACM,pp. 234-244.

Perlman, R., (1999). An overview of PKI trust models. *Network, IEEE,* 13(6)**,**pp. 38-43.

Pirzada, A. A. & Mcdonald, C., 2004. Kerberos assisted Authentication in Mobile Ad-hoc Networks. *Proceedings of the 27th Australasian conference on Computer science - Volume 26.* Dunedin, New Zealand: Australian Computer Society, Inc.,pp.41-46.

Postel, J., (1980). RFC 768: User datagram protocol. *URL http://www.ietf.org/rfc/rfc768.txt*.

Postel, J., (1981). RFC–793 Transmission Datagram Protocol. *Information Sciences Institute, USC, CA*.

Rachedi, A. & Benslimane, A., (2006). Trust and Mobility-based Clustering Algorithm for Secure Mobile Ad Hoc Networks. *In*: *Systems and Networks Communications, 2006. ICSNC '06. International Conference on*,pp. 72-72.

Rachedi, A., Benslimane, A., Lei, G. & Assi, C., (2007). A Confident Community to Secure Mobile Ad Hoc Networks. *In*: *Communications, 2007. ICC '07. IEEE International Conference on*,pp. 1254-1259.

Raghani, S., Toshniwal, D. & Joshi, R., (2006). Dynamic Support for Distributed Certification Authority in Mobile Ad Hoc Networks. *In*: *Hybrid Information Technology, 2006. ICHIT '06. International Conference on*,pp. 424-432.

Rahman, A. H. A. & Zukarnain, Z. A., (2009). Performance comparison of AODV, DSDV and I-DSDV routing protocols in mobile ad hoc networks. *European Journal of Scientific Research,* 31(4)**,**pp. 566-576.

Ramanathan, R. & Redi, J., (2002). A brief overview of ad hoc networks: challenges and directions. *Communications Magazine, IEEE,* 40(5)**,**pp. 20-22.

Rivest, R. L., Shamir, A. & Adleman, L., (1978). A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM,* 21(2)**,**pp. 120-126.

Robinson, S., (2004). *Simulation: the practice of model development and use*: John Wiley & Sons Chichester.

Royer, E. M. & Perkins, C. E., (2000). An implementation study of the AODV routing protocol. *In*: *Wireless Communications and Networking Confernce, 2000. WCNC. 2000 IEEE*,pp. 1003-1008 vol.3.

Salmanian, M., Hu, J., Pan, L., Mason, P. C. & Li, M., (2010). Supporting periodic, strong re-authentication in MANET scenarios. *In*: *MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010*,pp. 19-25.

Saremi, F., Mashayekhi, H., Movaghar, A. & Jalili, R., (2009). CEBAC: A Decentralized Cooperation Enforcement Based Access Control Framework in MANETs. *In* Sarbazi-Azad, H., Parhami, B., Miremadi, S.-G. & Hessabi, S. (eds.) *Advances in Computer Science and Engineering.* Springer Berlin Heidelberg,6,pp.427-434.

Sarkar, S. K., Basavaraju, T. & Puttamadappa, C., (2007). *Ad hoc mobile wireless networks: principles, protocols and applications*: CRC Press.

Savola, R. & Uusitalo, I., (2006). Towards Node-Level Security Management in Self-Organizing Mobile Ad Hoc Networks. *In*: *Telecommunications, 2006. AICT-ICIW '06. International Conference on Internet and Web Applications and Services/Advanced International Conference on*,pp. 36-36.

Saxena, N., Tsudik, G. & Yi, J. H., (2003). Admission control in Peer-to-Peer: design and performance evaluation. *In*: *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*ACM,pp. 104-113.

Saxena, N., Tsudik, G. & Yi, J. H., (2007). Threshold cryptography in P2P and MANETs: The case of access control. *Computer Networks,* 51(12)**,**pp. 3632-3649.

Scheneier, B., (1996). *Applied Cryptography Second Edition: protocols, algorithms, and source code in C*: John Wiley and Sons.

Schollmeier, R., (2001). A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications. *In*: *Peer-to-Peer Computing, IEEE International Conference on*IEEE Computer Society,pp. 0101-0101.

Scott, W., Houle, A. & Martin, A., 2006. Information Assurance Issues for an SDR Operating in a MANET Network. *SDR Forum.*

Sehgal, A., Ahuja, R. & Kumari, S., (2011). A security architecture for Mobile Ad Hoc Networks. *In*: *International Conference on Science and Engineering (ICSE 2011)*, India: RG Education Society,pp. 255-260.

Shamir, A., (1979). How to share a secret. *Commun. ACM,* 22(11)**,**pp. 612-613.

Shoup, V., (2000). Practical threshold signatures. *In*: *Proceedings of the 19th international conference on Theory and application of cryptographic techniques*, Bruges, Belgium: Springer-Verlag,pp. 207-220.

Smart, N. P., (2003). *Cryptography: an introduction*: McGraw-Hill New York.

Sokolowski, J. A. & Banks, C. M., (2009). *Principles of modeling and simulation: a multidisciplinary approach*: John Wiley & Sons.

Stallings, W., (2010). *Cryptography and Network Security: Principles and Practice*: Prentice Hall Press.

Steinmetz, R. & Wehrle, K., (2005). 2. What Is This "Peer-to-Peer" About? *In* Steinmetz, R. & Wehrle, K. (eds.) *Peer-to-Peer Systems and Applications.* Springer Berlin Heidelberg,3485,pp.9-16.

Stillwell, W. G., Seaver, D. A. & Edwards, W., (1981). A comparison of weight approximation techniques in multiattribute utility decision making. *Organizational Behavior and Human Performance,* 28(1)**,**pp. 62-77.

Stutzbach, D. & Rejaie, R., 2006. Understanding churn in peer-to-peer networks. *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement.* Rio de Janeriro, Brazil: ACM,pp.189-202.

Tanenbaum, A. S. & Van Steen, M., (2007). *Distributed Systems: Principles and Paradigms,* 2 ed.: Prentice Hall.

Tang, B., Zhou, Z., Kashyap, A. & Chiueh, T. C., (2005). An integrated approach for P2P file sharing on multi-hop wireless networks. *In*: *2005 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob'2005*,pp. 268-274.

Taylor, I. J. & Harrison, A., (2006). *From P2P to Web services and grids: peers in a client/server world*: Springer.

Toh, C. K. K., (2001). *Ad Hoc Wireless Networks: Protocols and Systems*: Prentice Hall PTR, NJ, USA.

Toubiana, V. & Labiod, H., (2008). Towards a flexible security management solution for dynamic MANETs. *In*: *Network Operations and Management Symposium, 2008. NOMS 2008. IEEE*,pp. 963-966.

Transafety, I., (1997):*Study Compares Older and Younger Pedestrian Walking Speeds*. October. Available from:< http://www.usroads.com/journals/p/rej/9710/re971001.htm > [Accessed 29/5/2014].

Umar, A., (1997). *Object-oriented client/server Internet environments*: Prentice Hall Press.

Umar, A. & Fraser, C. W., (1993). *Distributed computing: a practical synthesis of networks, client-server systems, distributed applications, and open systems*: Prentice-Hall, Inc.

Venkatraman, L. & Agrawal, D. P., (2000). A novel authentication scheme for ad hoc networks. *In*: *Wireless Communications and Networking Confernce, 2000. WCNC. 2000 IEEE*,pp. 1268-1273 vol.3.

Verma, R., O'mahony, D. & Tewari, H., .. 2004. Progressive authentication in ad hoc networks. *5th European Wireless Conference.*

Wang, C.-T., Lin, C.-H. & Chang, C.-C., (1998). Threshold signature schemes with traceable signers in group communications. *Computer Communications,* 21(8)**,**pp. 771-776.

Wu, B., Chen, J., Wu, J. & Cardei, M., (2007a). A survey of attacks and countermeasures in mobile ad hoc networks. *In Wireless Network Security.* Springer,pp.103-135.

Wu, B., Wu, J., Fernandez, E. B., Ilyas, M. & Magliveras, S., (2007b). Secure and efficient key management in mobile ad hoc networks. *J. Netw. Comput. Appl.,* 30(3)**,**pp. 937-954.

Yang, C., Wu, H., Itoh, M. & Zhou, T., (2010). On User Arrival Patterns in the Wireless Network. *Journal of Computational Information Systems,* 6(14)**,**pp. 4827-4834.

Yang, H., Luo, H., Ye, F., Lu, S. & Zhang, L., (2004). Security in mobile ad hoc networks: challenges and solutions. *Wireless Communications, IEEE,* 11(1)**,**pp. 38-47.

Yang, H., Shu, J., Meng, X. & Lu, S., (2006). SCAN: self-organized network-layer security in mobile ad hoc networks. *Selected Areas in Communications, IEEE Journal on,* 24(2)**,**pp. 261-273.

Yi, S. & Kravets, R., (2003). MOCA: mobile certificate authority for wireless ad hoc networks. *In*: *2nd annual PKI research workshop*,pp. 65-79.

Yi, S. & Kravets, R., (2004). Composite key management for ad hoc networks. *In*: *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*,pp. 52-61.

Yoon, J., Mingyan, L. & Noble, B., (2003). Random waypoint considered harmful. *In*: *INFOCOM 2003. Twenty-Second Annual Joint Conference of*

the IEEE Computer and Communications. IEEE Societies,pp. 1312-1321 vol.2.

Yu, S., Zhang, Y., Song, C. & Chen, K., (2003). A security architecture for mobile ad hoc networks. *In*: *Proceedings of APAN Network Research Workshop. Cairns, Australia*.

Yunfang, F., (2007). Adaptive Trust Management in MANET. *In*: *Computational Intelligence and Security, 2007 International Conference on*,pp. 804-808.

Yung-Chih, C., Kurose, J. & Towsley, D., (2012). A mixed queueing network model of mobility in a campus wireless network. *In*: *INFOCOM, 2012 Proceedings IEEE*,pp. 2656-2660.

Zhou, L. & Haas, Z. J., (1999). Securing ad hoc networks. *Network, IEEE,* 13(6)**,**pp. 24-30.

Appendix A

A.1 The BonnMotion Tool

1- How to install the BonnMotion application?

Please find its manual available online in the following links:

http://sys.cs.uos.de/bonnmotion/doc/BonnMotion_Docu.pdf

2- How to generate a trace file for a particular mobility model using the BonnMotion tool?

We should create and initialise a parameter file (*.params) which includes all configuration parameters about a specific mobility scenario. Since this study depends on the steady-state Random Waypoint model, the parameters need to be adjusted: the type of mobility model, a number of nodes, simulation time, space, speed and pause time. In addition, there are different scenarios having different a number of nodes (100, 250, and 500 nodes) in this study, so it is required to prepare three parameter files as following:

The *.params File1 Content (100 Nodes)	The *.params File2 Content (250 Nodes)	The *.params File3 Content (500 Nodes)
model=SteadyStateRandomWaypoint	model=SteadyStateRandomWaypoint	model=SteadyStateRandomWaypoint
x=1500	x=1500	x=1500
y=1500	y=1500	y=1500
duration= 7300	duration= 16800	duration= 35000
nn=100	nn=250	nn=500
speedMean=1	speedMean=1	speedMean=1
speedDelta=0.5	speedDelta=0.5	speedDelta=0.5
pauseMean=60.0	pauseMean=60.0	pauseMean=60.0
pauseDelta=30	pauseDelta=30	pauseDelta=30

Figure A.1: Different BonnMotion configurations (params files)

After creating these *.params files, we will produce the actual trace or movements files for each scenario and also each one must be replicated 30 times according to the predefined a number of replications required for each runs in simulation. Since this tool relies on a certain random generator along with a seed for generating random positions and movements, every repeat for the same *.params file without defining the seed parameter would produce a different trace file scenario as it makes use of different seeds in every replication. We have coded a bat file to run BonnMotion application and repeats n time the same *.params file for each of three scenario discussed above.

```
FOR /L %%A IN (0,1,29) DO

bm -f MySteadyStateWPMobSenario%%A -I SteadyStateWPMobility.params
SteadyStateRandomWaypoint
```

Figure A.6: The *.Bat file for creating trace files

After running the bat file indicating to the *.params file, thirty (*.movements) trace files will be created to represent thirty different replications for one run. Also, these steps will be repeated to the other scenarios 250 and 500 nodes. Finally, these files will be fed into OMNeT++ simulator to simulate node mobility.

A.2 Node NED File (The Node-Level Structure)

The following script is the NED file for defining the node structure:

```
module MobileManetRoutingHost_AAM_F
{
    parameters:
        @node();
        int numTcpApps = default(0);
        int numUdpApps = default(0);
@display("i=device/pocketpc_s;bgb=395,372");
    gates:
        input radioIn @directIn;
    submodules:

        notificationBoard: NotificationBoard {
            parameters:
                @display("p=60,70");
        }

        interfaceTable: InterfaceTable {
            parameters:
                @display("p=60,154");
        }
        routingTable: RoutingTable {
            parameters:
                IPForward = true;
                routerId = "";
                routingFile = routingFile;
                @display("p=60,230");
        }
        tcpApp[numTcpApps]: <tcpAppType> like TCPApp {
            parameters:
                @display("p=170,47");
        }
        tcp: TCP {
            parameters:
                @display("p=179,161");
        }
        udpApp[numUdpApps]: <udpAppType> like UDPApp {
            parameters:
                @display("p=295,47");
        }
        udp: UDP {
            parameters:
                @display("p=279,143");
           }
        networkLayer: NetworkLayerGlobalArp {
            parameters:
                proxyARP = false;
                globalARP = true;
                @display("p=256,230;q=queue");
            gates:
                ifIn[1];
                ifOut[1];
        }
        manetrouting: ManetRouting {
```

```
            @display("p=153,230;i=block/network2");
        }
        wlan: Ieee80211gNicAdhoc {
            parameters:
                @display("p=256,310;q=queue");
        }
        mobility: <mobilityType> like BasicMobility {
            parameters:
                @display("p=153,301;i=block/cogwheel");
        }

    connections allowunconnected:
        for i=0..numTcpApps-1 {
            tcpApp[i].tcpOut --> tcp.appIn++;
            tcpApp[i].tcpIn <-- tcp.appOut++;
        }

        tcp.ipOut --> networkLayer.tcpIn;
        tcp.ipIn <-- networkLayer.TCPOut;

        for i=0..numUdpApps-1 {
            udpApp[i].udpOut --> udp.appIn++;
            udpApp[i].udpIn <-- udp.appOut++;
        }

        udp.ipOut --> networkLayer.udpIn;
        udp.ipIn <-- networkLayer.udpOut;

        networkLayer.MANETOut --> manetrouting.from_ip;
        networkLayer.MANETIn <-- manetrouting.to_ip;

        // connections to network outside
        radioIn --> wlan.radioIn;
        wlan.uppergateOut --> networkLayer.ifIn[0];
        wlan.uppergateIn <-- networkLayer.ifOut[0];
}
```

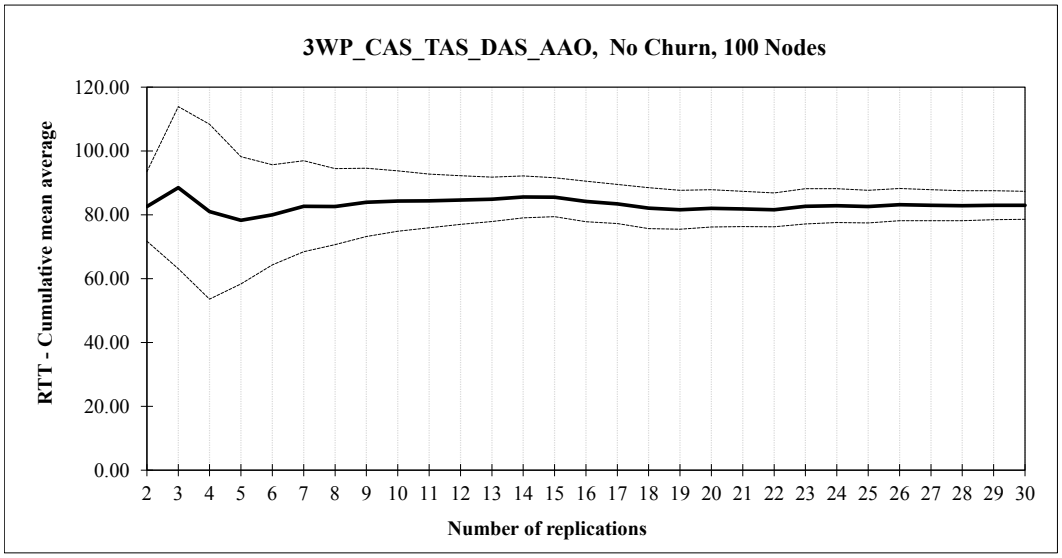
Appendix B

B.1. The Number of Replication Tables and Charts

- **Round Tip Time**

Replication	Result *RTT*	Cum. mean average	Standard deviation	Confidence Interval (95%) Lower Interval CI_{lower}	Confidence Interval (95%) Upper Interval CI_{Upper}	% Deviation STDevPRC
1	83.49881476	83.50	n/a	n/a	n/a	n/a
2	81.78117972	82.64	1.215	71.73	93.55	13.20%
3	100.2866359	88.52	10.224	63.12	113.92	28.69%
4	58.42857004	81.00	17.208	53.62	108.38	33.80%
5	67.53103665	78.31	16.073	58.35	98.26	25.49%
6	88.49828636	80.00	14.967	64.30	95.71	19.63%
7	98.84478482	82.70	15.407	68.45	96.94	17.23%
8	81.82966404	82.59	14.267	70.66	94.52	14.44%
9	94.62292974	83.92	13.936	73.21	94.64	12.76%
10	88.04561784	84.34	13.203	74.89	93.78	11.20%
11	84.76434445	84.38	12.526	75.96	92.79	9.97%
12	87.79690215	84.66	11.984	77.05	92.28	8.99%
13	87.33322957	84.87	11.498	77.92	91.81	8.19%
14	95.13210013	85.60	11.382	79.03	92.17	7.68%
15	84.64344801	85.54	10.971	79.46	91.61	7.10%
16	64.13835689	84.20	11.873	77.87	90.52	7.51%
17	70.83705828	83.41	11.944	77.27	89.55	7.36%
18	59.79512301	82.10	12.855	75.71	88.49	7.79%
19	72.77746642	81.61	12.674	75.50	87.72	7.49%
20	89.71019502	82.01	12.469	76.18	87.85	7.12%
21	78.39385745	81.84	12.179	76.30	87.39	6.77%
22	76.02534807	81.58	11.950	76.28	86.88	6.49%
23	106.62831	82.67	12.790	77.14	88.20	6.69%
24	87.88964384	82.88	12.554	77.58	88.19	6.40%
25	75.77491896	82.60	12.372	77.49	87.71	6.18%
26	98.02632687	83.19	12.494	78.15	88.24	6.07%
27	78.18876649	83.01	12.289	78.15	87.87	5.86%
28	79.20968365	82.87	12.081	78.19	87.56	5.65%
29	87.19402313	83.02	11.890	78.50	87.54	5.45%
30	81.84488823	82.98	11.685	78.62	87.35	5.26%

The Confidence Interval Method: Results of RTT in the case of *3WP_CAS_TAS_DAS_AAO* with 100-Node and *No-Churn* for 30 replications.

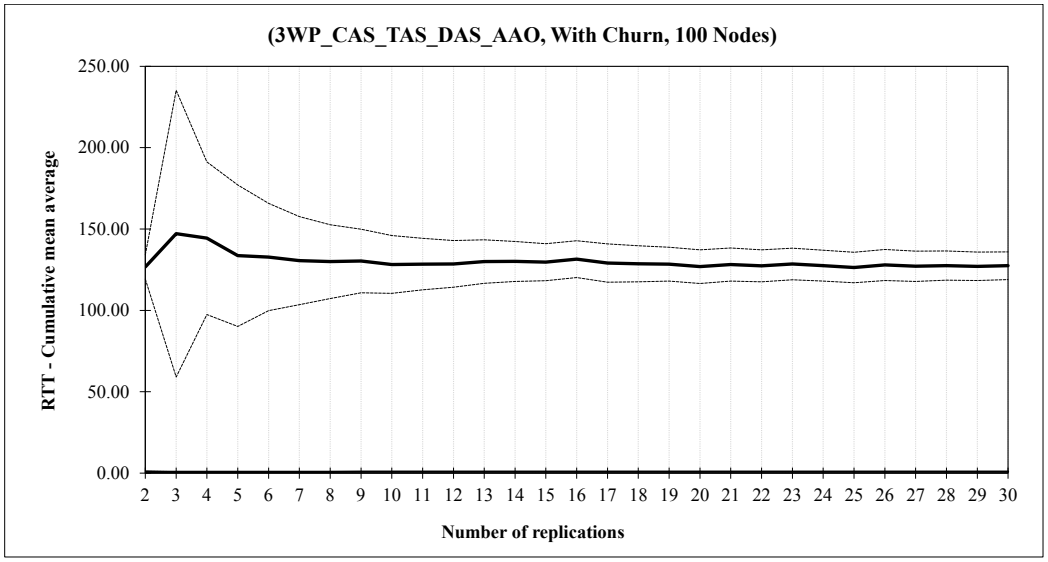


The plot represents the 95% confidence intervals and cumulative mean of RTT in the case of no node churning

Replication	Result: *RTT*	Cum. mean average	Standard deviation	Confidence Interval (95%) Lower Interval CI_{lower}	Confidence Interval (95%) Upper Interval CI_{Upper}	% Deviation STDevPRC
1	126.082017	126.08	n/a	n/a	n/a	n/a
2	127.256032	126.67	0.830	119.21	134.13	5.89%
3	188.046861	147.13	35.441	59.09	235.17	59.84%
4	135.960525	144.34	29.472	97.44	191.23	32.49%
5	90.6785896	133.60	35.032	90.11	177.10	32.56%
6	128.455586	132.75	31.404	99.79	165.70	24.83%
7	117.163723	130.52	29.267	103.45	157.59	20.74%
8	126.052563	129.96	27.142	107.27	152.65	17.46%
9	133.409343	130.35	25.415	110.81	149.88	14.99%
10	109.145182	128.23	24.881	110.43	146.02	13.88%
11	130.713262	128.45	23.617	112.59	144.32	12.35%
12	129.841657	128.57	22.521	114.26	142.88	11.13%
13	146.531122	129.95	22.130	116.58	143.32	10.29%
14	131.860576	130.09	21.268	117.81	142.37	9.44%
15	123.012032	129.61	20.576	118.22	141.01	8.79%
16	159.869585	131.50	21.269	120.17	142.84	8.62%
17	90.5187756	129.09	22.867	117.34	140.85	9.11%
18	121.235243	128.66	22.261	117.59	139.73	8.60%
19	123.824514	128.40	21.663	117.96	138.84	8.13%
20	97.8777668	126.88	22.162	116.50	137.25	8.18%
21	154.205019	128.18	22.409	117.98	138.38	7.96%
22	109.874724	127.35	22.214	117.50	137.20	7.73%
23	153.832231	128.50	22.395	118.81	138.18	7.54%
24	104.520787	127.50	22.443	118.02	136.98	7.43%
25	98.4398804	126.34	22.726	116.96	135.72	7.43%

26	167.157884	127.91	23.663	118.35	137.46	7.47%
27	106.289055	127.11	23.573	117.78	136.43	7.34%
28	139.227966	127.54	23.246	118.52	136.55	7.07%
29	112.907671	127.03	22.988	118.29	135.78	6.88%
30	138.860912	127.43	22.691	118.96	135.90	6.65%

The Confidence Interval Method: Results of RTT in the case of *3WP_CAS_TAS_DAS_AAO* with 100-Node and *Churn* for 30 replications.



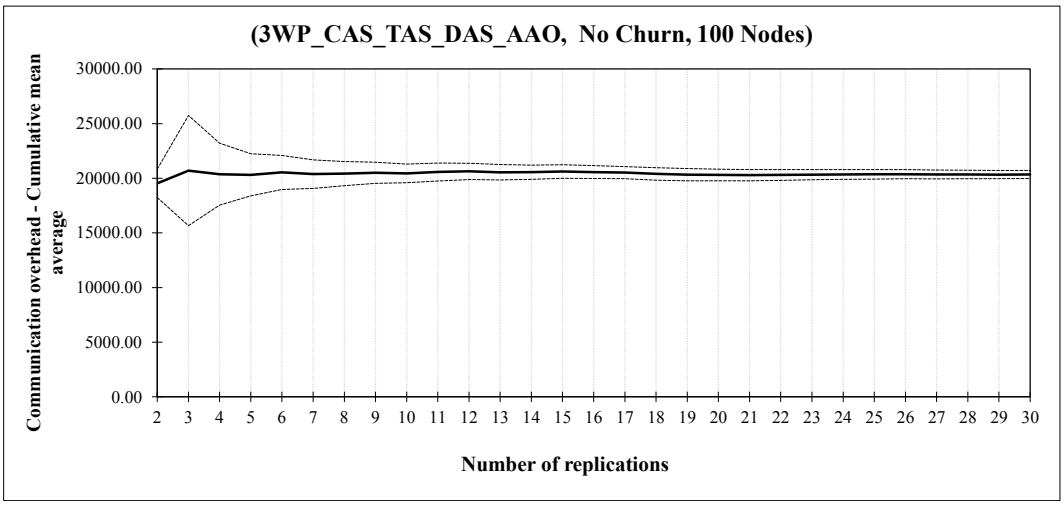
The plot represents the 95% confidence intervals and cumulative mean of RTT in the case of node churning

- **Communication Overhead:**

Replication	Result: *Communi Overhead*	Cum. mean average	Standard deviation	Confidence Interval (95%) Lower Interval CI_{lower}	Upper Interval CI_{Upper}	% Deviation STDevPRC
1	19636	19635.77	n/a	n/a	n/a	n/a
2	19433	19534.33	143.453	18245.46	20823.20	6.60%
3	23044	20704.34	2029.046	15663.91	25744.77	24.34%
4	19384	20374.19	1783.444	17536.33	23212.05	13.93%
5	20073	20313.93	1550.374	18388.89	22238.98	9.48%
6	21632	20533.62	1487.450	18972.64	22094.61	7.60%
7	19490	20384.53	1413.986	19076.81	21692.25	6.42%

8	20751	20430.37	1315.502	19330.58	21530.16	5.38%
9	21105	20505.36	1250.937	19543.81	21466.92	4.69%
10	19861	20440.89	1196.890	19584.68	21297.09	4.19%
11	21872	20571.02	1214.732	19754.95	21387.09	3.97%
12	21294	20631.28	1176.861	19883.54	21379.02	3.62%
13	19516	20545.48	1168.456	19839.39	21251.57	3.44%
14	20721	20558.03	1123.598	19909.29	21206.78	3.16%
15	21536	20623.26	1111.809	20007.56	21238.96	2.99%
16	19759	20569.21	1095.649	19985.38	21153.04	2.84%
17	19633	20514.17	1084.865	19956.38	21071.95	2.72%
18	18443	20399.12	1160.145	19822.20	20976.05	2.83%
19	19011	20326.08	1171.555	19761.40	20890.75	2.78%
20	19826	20301.06	1145.784	19764.81	20837.30	2.64%
21	19920	20282.91	1119.866	19773.15	20792.66	2.51%
22	20672	20300.59	1096.021	19814.64	20786.54	2.39%
23	21011	20331.48	1081.021	19864.01	20798.95	2.30%
24	20763	20349.46	1060.921	19901.47	20797.45	2.20%
25	20615	20360.07	1039.938	19930.80	20789.33	2.11%
26	20708	20373.44	1021.205	19960.97	20785.91	2.02%
27	19586	20344.27	1012.781	19943.63	20744.91	1.97%
28	20675	20356.08	995.810	19969.94	20742.21	1.90%
29	19771	20335.89	983.889	19961.64	20710.14	1.84%
30	20630	20345.70	968.268	19984.14	20707.26	1.78%

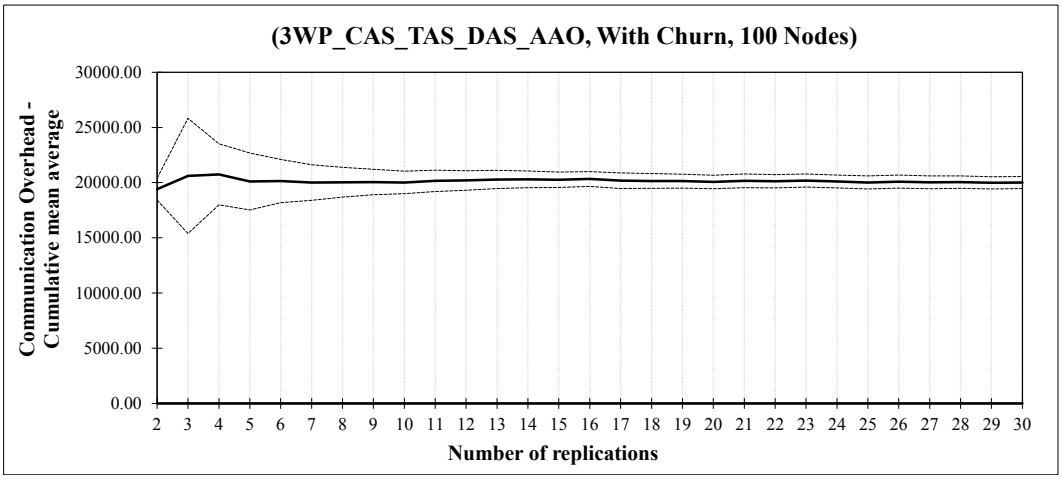
The Confidence Interval Method: Results of Communication Overheads in the case of *3WP_CAS_TAS_DAS_AAO* with 100-Node and *No-Churn* for 30 replications.



The plot represents the 95% confidence intervals and cumulative mean of Communication Overheads in the case of no churning

Replication	Result: *Communi Overhead*	Cum. mean average	Standard deviation	**Confidence Interval (95%)** Lower Interval CI_{lower}	Upper Interval CI_{Upper}	% Deviation STDevPRC
1	19325	19324.86	n/a	n/a	n/a	n/a
2	19481	19402.91	110.371	18411.27	20394.55	5.11%
3	23036	20613.90	2098.950	15399.82	25827.98	25.29%
4	21157	20749.75	1735.189	17988.68	23510.82	13.31%
5	17536	20107.00	2079.374	17525.12	22688.88	12.84%
6	20368	20150.45	1862.892	18195.47	22105.44	9.70%
7	19172	20010.67	1740.327	18401.14	21620.21	8.04%
8	20204	20034.82	1612.677	18686.59	21383.05	6.73%
9	20226	20056.04	1509.863	18895.45	21216.62	5.79%
10	19679	20018.34	1428.496	18996.45	21040.22	5.10%
11	21627	20164.61	1439.412	19197.60	21131.62	4.80%
12	20520	20194.26	1376.263	19319.83	21068.70	4.33%
13	21387	20286.04	1358.591	19465.06	21107.03	4.05%
14	20520	20302.78	1306.793	19548.26	21057.29	3.72%
15	19717	20263.76	1268.292	19561.40	20966.11	3.47%
16	21407	20335.23	1258.198	19664.78	21005.68	3.30%
17	17700	20180.22	1375.724	19472.88	20887.55	3.51%
18	19715	20154.38	1339.142	19488.44	20820.32	3.30%
19	19910	20141.54	1302.616	19513.70	20769.38	3.12%
20	18560	20062.48	1316.241	19446.46	20678.50	3.07%
21	22207	20164.62	1365.629	19542.99	20786.25	3.08%
22	19348	20127.50	1344.041	19531.59	20723.42	2.96%
23	21633	20192.97	1350.155	19609.12	20776.82	2.89%
24	18128	20106.93	1386.122	19521.62	20692.24	2.91%
25	17908	20018.98	1426.410	19430.19	20607.78	2.94%
26	22003	20095.30	1450.758	19509.33	20681.28	2.92%
27	18282	20028.16	1464.748	19448.72	20607.59	2.89%
28	20805	20055.90	1444.843	19495.65	20616.15	2.79%
29	17893	19981.33	1474.539	19420.44	20542.21	2.81%
30	20936	20013.16	1459.347	19468.24	20558.09	2.72%

The Confidence Interval Method: Results of Communication Overheads in the case of *3WP_CAS_TAS_DAS_AAO* with 100-Node and *Churn* for 30 replications.



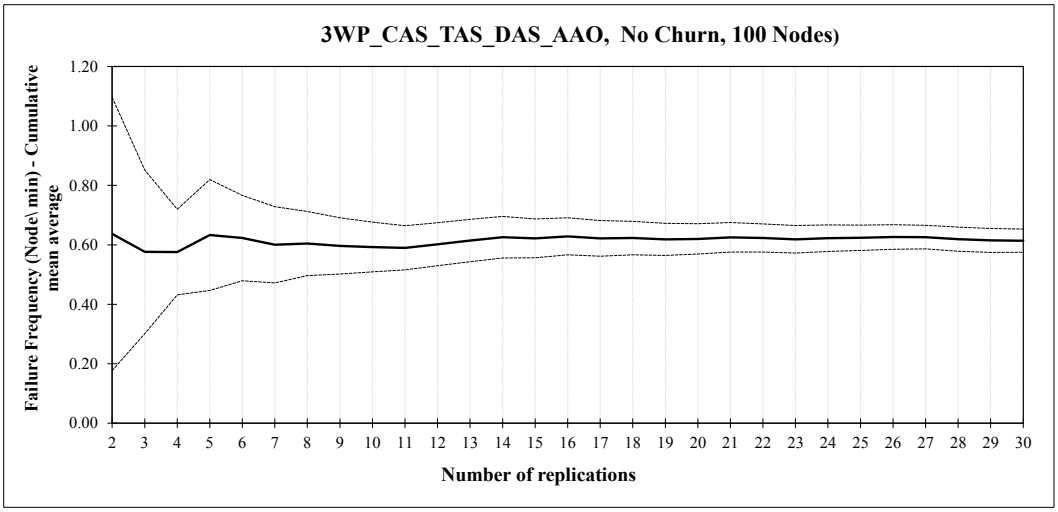
The plot represents the 95% confidence intervals and cumulative mean of Communication Overheads in the case of node churning

- **Failure Frequency:**

Replication	Result: *Failure Frequency*	Cum. mean average	Standard deviation	Confidence Interval (95%) Lower Interval CI_{lower}	Confidence Interval (95%) Upper Interval CI_{Upper}	% Deviation STDevPRC
1	0.60073938	0.60	n/a	n/a	n/a	n/a
2	0.6731342	0.64	0.051	0.18	1.10	72.21%
3	0.45541597	0.58	0.111	0.30	0.85	47.78%
4	0.57348419	0.58	0.091	0.43	0.72	25.03%
5	0.86246911	0.63	0.150	0.45	0.82	29.48%
6	0.57140248	0.62	0.137	0.48	0.77	23.05%
7	0.46423855	0.60	0.139	0.47	0.73	21.34%
8	0.63529531	0.60	0.129	0.50	0.71	17.82%
9	0.53147471	0.60	0.123	0.50	0.69	15.85%
10	0.55944475	0.59	0.117	0.51	0.68	14.06%
11	0.5629248	0.59	0.111	0.52	0.66	12.63%
12	0.73387799	0.60	0.114	0.53	0.67	11.99%
13	0.76509566	0.61	0.118	0.54	0.69	11.58%
14	0.77337041	0.63	0.121	0.56	0.70	11.15%
15	0.56281647	0.62	0.118	0.56	0.69	10.48%
16	0.73309827	0.63	0.117	0.57	0.69	9.92%
17	0.50978506	0.62	0.117	0.56	0.68	9.67%
18	0.64436495	0.62	0.114	0.57	0.68	9.06%
19	0.53853186	0.62	0.112	0.56	0.67	8.73%
20	0.65015359	0.62	0.109	0.57	0.67	8.25%
21	0.72604383	0.63	0.109	0.58	0.67	7.93%
22	0.57864836	0.62	0.107	0.58	0.67	7.60%
23	0.52505412	0.62	0.106	0.57	0.66	7.43%
24	0.70278216	0.62	0.105	0.58	0.67	7.15%

25	0.65918871	0.62	0.103	0.58	0.67	6.85%
26	0.69972266	0.63	0.102	0.59	0.67	6.60%
27	0.60637323	0.63	0.101	0.59	0.67	6.35%
28	0.43933969	0.62	0.105	0.58	0.66	6.56%
29	0.48974322	0.61	0.106	0.57	0.65	6.54%
30	0.59089637	0.61	0.104	0.58	0.65	6.32%

The Confidence Interval Method: Results of Failure Frequency the case of *3WP_CAS_TAS_DAS_AAO* with 100-Node and *No-Churn* for 30 replications.

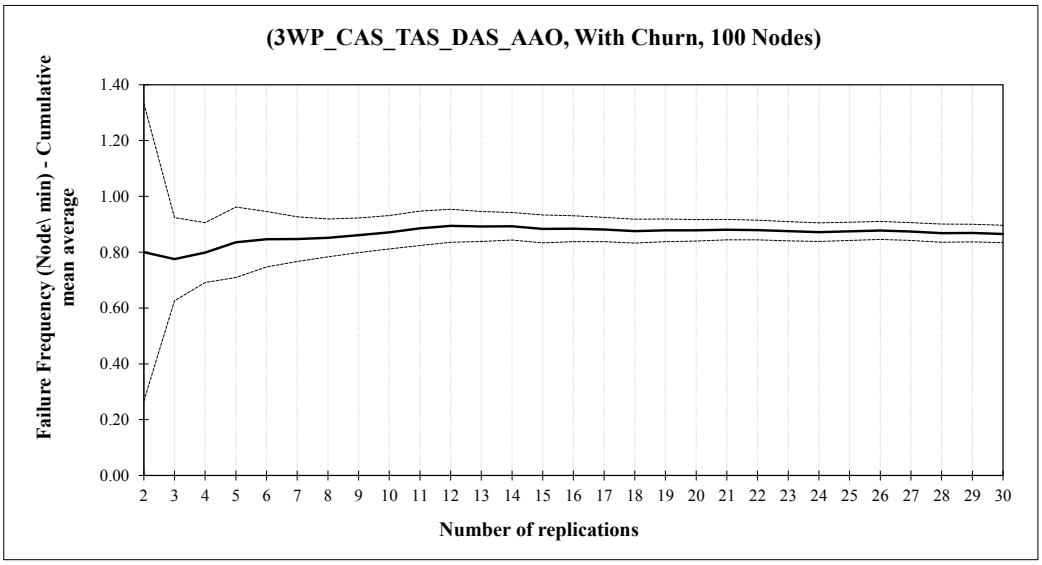


The plot represents the 95% confidence intervals and cumulative mean of Failure Frequency in the case of no node churning

Replication	Result: *Failure Frequency*	Cum. mean average	Standard deviation	Confidence Interval (95%) Lower Interval CI_{lower}	Upper Interval CI_{Upper}	% Deviation STDevPRC
1	0.75808909	0.76	n/a	n/a	n/a	n/a
2	0.84195966	0.80	0.059	0.27	1.33	66.60%
3	0.72567907	0.78	0.060	0.63	0.92	19.23%
4	0.86831985	0.80	0.068	0.69	0.91	13.47%
5	0.98397041	0.84	0.102	0.71	0.96	15.08%
6	0.90044284	0.85	0.095	0.75	0.95	11.73%
7	0.84911737	0.85	0.086	0.77	0.93	9.43%
8	0.88266914	0.85	0.081	0.78	0.92	7.95%
9	0.93565921	0.86	0.081	0.80	0.92	7.21%
10	0.96761638	0.87	0.083	0.81	0.93	6.84%
11	1.02750841	0.89	0.092	0.82	0.95	6.98%
12	0.99177006	0.89	0.093	0.84	0.95	6.60%
13	0.86441505	0.89	0.089	0.84	0.95	6.05%
14	0.90204036	0.89	0.086	0.84	0.94	5.55%

15	0.75446905	0.88	0.090	0.83	0.93	5.65%
16	0.88787138	0.88	0.087	0.84	0.93	5.25%
17	0.84087753	0.88	0.085	0.84	0.93	4.96%
18	0.77926423	0.88	0.086	0.83	0.92	4.88%
19	0.92307735	0.88	0.084	0.84	0.92	4.62%
20	0.87770454	0.88	0.082	0.84	0.92	4.37%
21	0.92988431	0.88	0.081	0.84	0.92	4.17%
22	0.8535507	0.88	0.079	0.84	0.91	3.98%
23	0.78380083	0.88	0.080	0.84	0.91	3.93%
24	0.79288786	0.87	0.080	0.84	0.91	3.86%
25	0.948151	0.87	0.079	0.84	0.91	3.75%
26	0.95361802	0.88	0.079	0.85	0.91	3.65%
27	0.77250133	0.87	0.080	0.84	0.91	3.64%
28	0.71477908	0.87	0.084	0.84	0.90	3.77%
29	0.88147031	0.87	0.083	0.84	0.90	3.63%
30	0.76745807	0.87	0.084	0.83	0.90	3.61%

The Confidence Interval Method: Results of Failure Frequency the case of *3WP_CAS_TAS_DAS_AAO* with 100-Node and *Churn* for 30 replications.

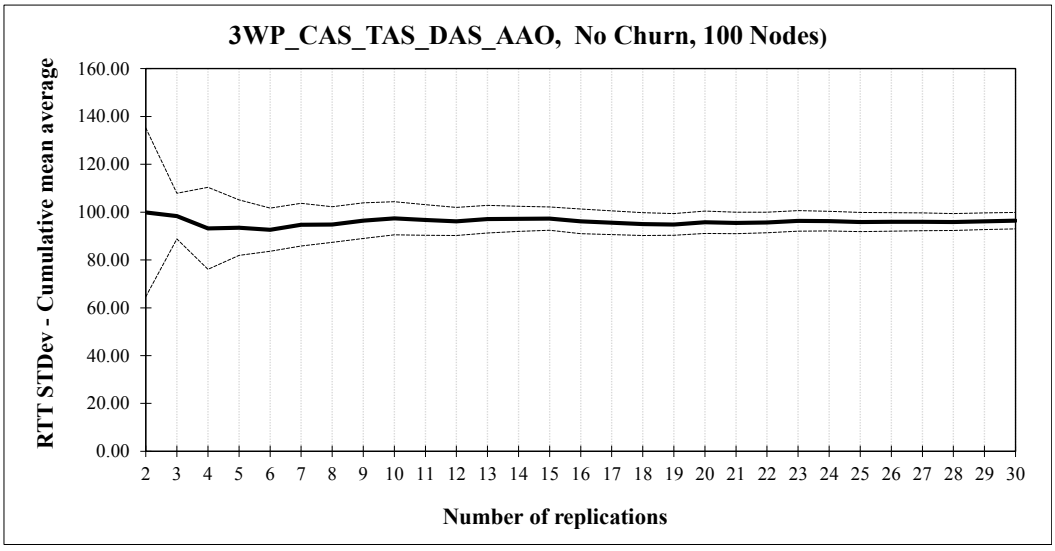


The plot represents the 95% confidence intervals and cumulative mean of Failure Frequency in the case of node churning

- **RTT-STDev:**

Replication	Result: *RTT STDev*	Cum. mean average	Standard deviation	Confidence Interval (95%) Lower Interval CI_{lower}	Confidence Interval (95%) Upper Interval CI_{Upper}	% Deviation STDevPRC
1	102.681256	102.68	n/a	n/a	n/a	n/a
2	97.1252856	99.90	3.929	64.61	135.20	35.33%
3	95.3094745	98.37	3.841	88.83	107.91	9.70%
4	77.77915	93.22	10.763	76.10	110.35	18.37%
5	94.6413099	93.51	9.343	81.91	105.11	12.41%
6	88.3666459	92.65	8.616	83.61	101.69	9.76%
7	107.380728	94.75	9.636	85.84	103.67	9.41%
8	94.986537	94.78	8.922	87.32	102.24	7.87%
9	109.433696	96.41	9.669	88.98	103.84	7.71%
10	106.61279	97.43	9.670	90.51	104.35	7.10%
11	89.6175139	96.72	9.472	90.36	103.08	6.58%
12	89.4899075	96.12	9.269	90.23	102.01	6.13%
13	108.285063	97.05	9.494	91.32	102.79	5.91%
14	99.2592639	97.21	9.141	91.93	102.49	5.43%
15	97.8462883	97.25	8.810	92.38	102.13	5.02%
16	79.3747095	96.14	9.614	91.01	101.26	5.33%
17	86.3251549	95.56	9.608	90.62	100.50	5.17%
18	86.1495236	95.04	9.581	90.27	99.80	5.01%
19	91.4766939	94.85	9.347	90.34	99.35	4.75%
20	113.493686	95.78	10.007	91.10	100.47	4.89%
21	90.0730483	95.51	9.833	91.03	99.99	4.69%
22	99.4121463	95.69	9.632	91.42	99.96	4.46%
23	111.423429	96.37	9.966	92.06	100.68	4.47%
24	92.8019945	96.22	9.774	92.10	100.35	4.29%
25	86.7215928	95.84	9.756	91.82	99.87	4.20%
26	98.1360501	95.93	9.569	92.07	99.80	4.03%
27	96.7186584	95.96	9.384	92.25	99.67	3.87%
28	93.1835191	95.86	9.224	92.28	99.44	3.73%
29	104.76482	96.17	9.207	92.67	99.67	3.64%
30	103.395857	96.41	9.143	92.99	99.82	3.54%

The Confidence Interval Method: Results of RTT-STDev the case of *3WP_CAS_TAS_DAS_AAO* with 100-Node and *No-Churn* for 30 replications.

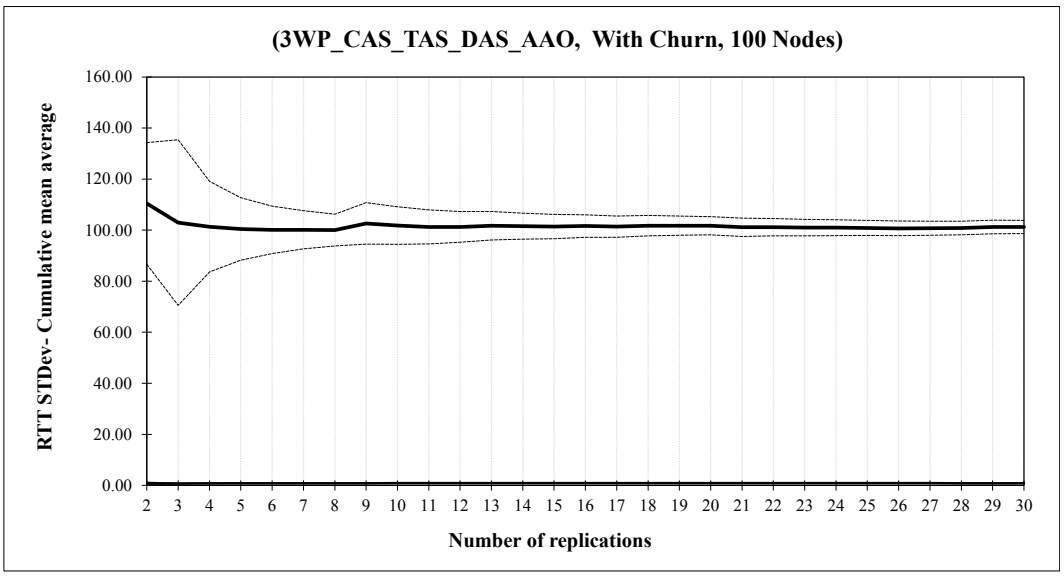


The plot represents the 95% confidence intervals and cumulative mean of RTT-STDev in the case of no node churning

Replication	Result: *RTT STDev*	Cum. mean average	Standard deviation	Confidence Interval (95%) Lower Interval CI_{lower}	Upper Interval CI_{Upper}	% Deviation STDevPRC
1	108.549528	108.55	n/a	n/a	n/a	n/a
2	112.302039	110.43	2.653	86.59	134.27	21.59%
3	88.0597927	102.97	13.049	70.56	135.38	31.48%
4	96.5018923	101.35	11.134	83.64	119.07	17.48%
5	96.8323523	100.45	9.852	88.22	112.68	12.18%
6	98.4404086	100.11	8.850	90.83	109.40	9.28%
7	100.066229	100.11	8.079	92.64	107.58	7.46%
8	99.4203286	100.02	7.484	93.77	106.28	6.26%
9	123.597927	102.64	10.525	94.55	110.73	7.88%
10	94.2117298	101.80	10.274	94.45	109.15	7.22%
11	95.8901422	101.26	9.909	94.60	107.92	6.57%
12	101.146486	101.25	9.448	95.25	107.25	5.93%
13	107.191111	101.71	9.194	96.15	107.26	5.46%
14	99.7369504	101.57	8.849	96.46	106.68	5.03%
15	99.0006672	101.40	8.553	96.66	106.13	4.67%
16	104.731066	101.60	8.305	97.18	106.03	4.36%
17	97.4155113	101.36	8.105	97.19	105.53	4.11%
18	108.279673	101.74	8.031	97.75	105.74	3.93%
19	101.497978	101.73	7.805	97.97	105.49	3.70%
20	100.928873	101.69	7.598	98.13	105.25	3.50%
21	89.7031045	101.12	7.854	97.54	104.69	3.54%
22	101.192756	101.12	7.665	97.72	104.52	3.36%
23	98.539274	101.01	7.508	97.76	104.26	3.21%

24	100.315393	100.98	7.345	97.88	104.08	3.07%
25	98.1706019	100.87	7.212	97.89	103.85	2.95%
26	96.2420345	100.69	7.124	97.81	103.57	2.86%
27	102.613597	100.76	6.996	97.99	103.53	2.75%
28	102.692951	100.83	6.875	98.17	103.50	2.64%
29	112.14903	101.22	7.070	98.53	103.91	2.66%
30	100.739418	101.21	6.948	98.61	103.80	2.56%

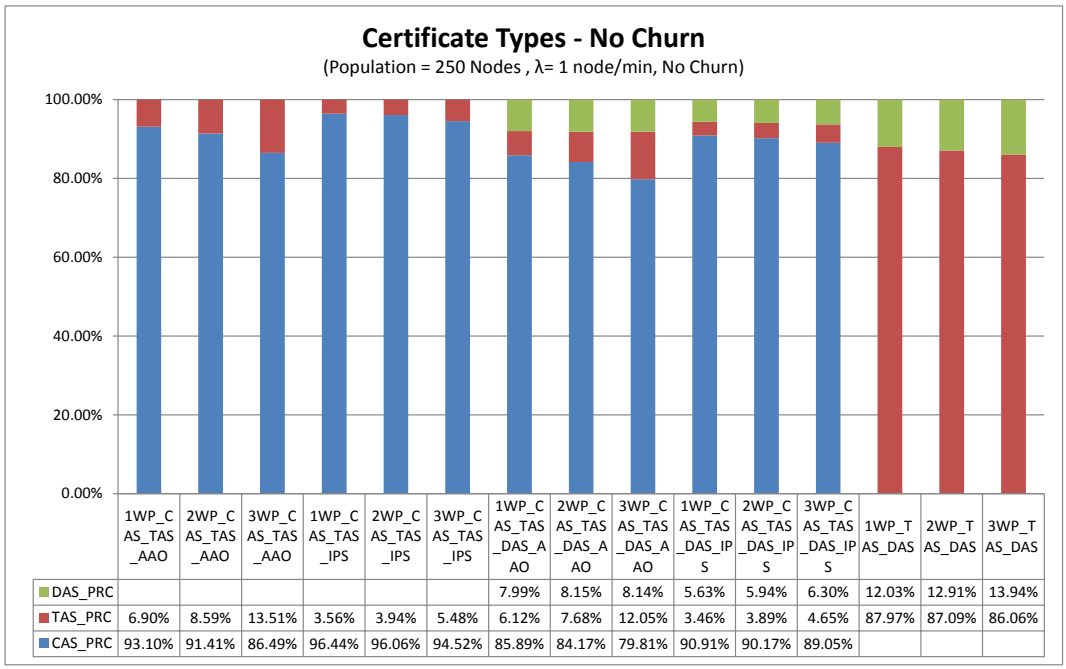
The Confidence Interval Method: Results of RTT-STDev the case of *3WP_CAS_TAS_DAS_AAO* with 100-Node and *Churn* for 30 replications.



The plot represents the 95% confidence intervals and cumulative mean of RTT-STDev in the case of node churning

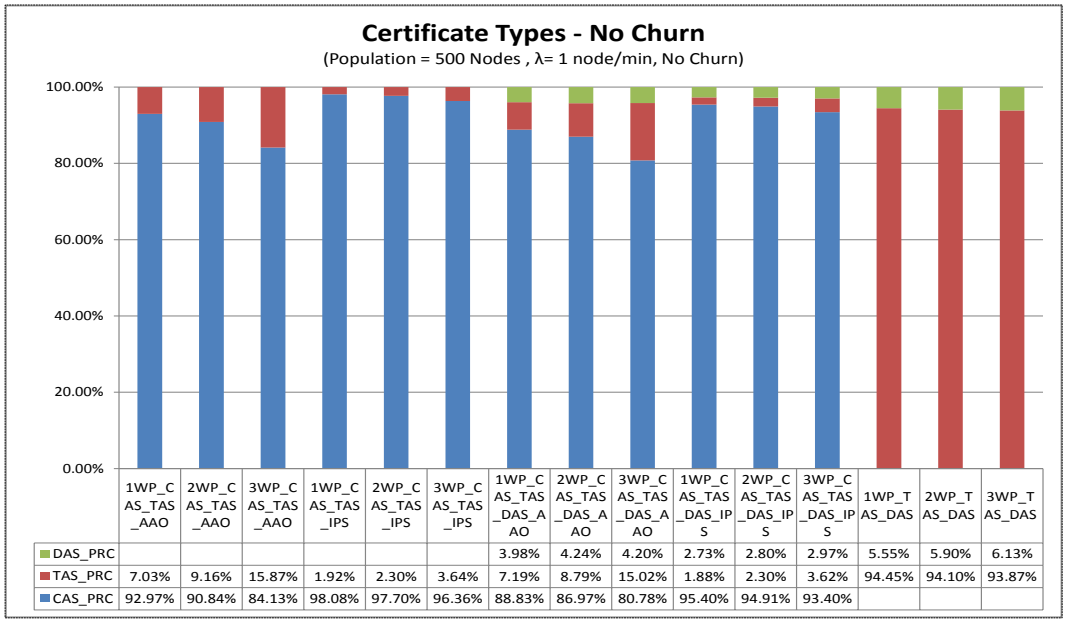
B.2. Certificate Acquisition

B.2.1. The Certificate Type



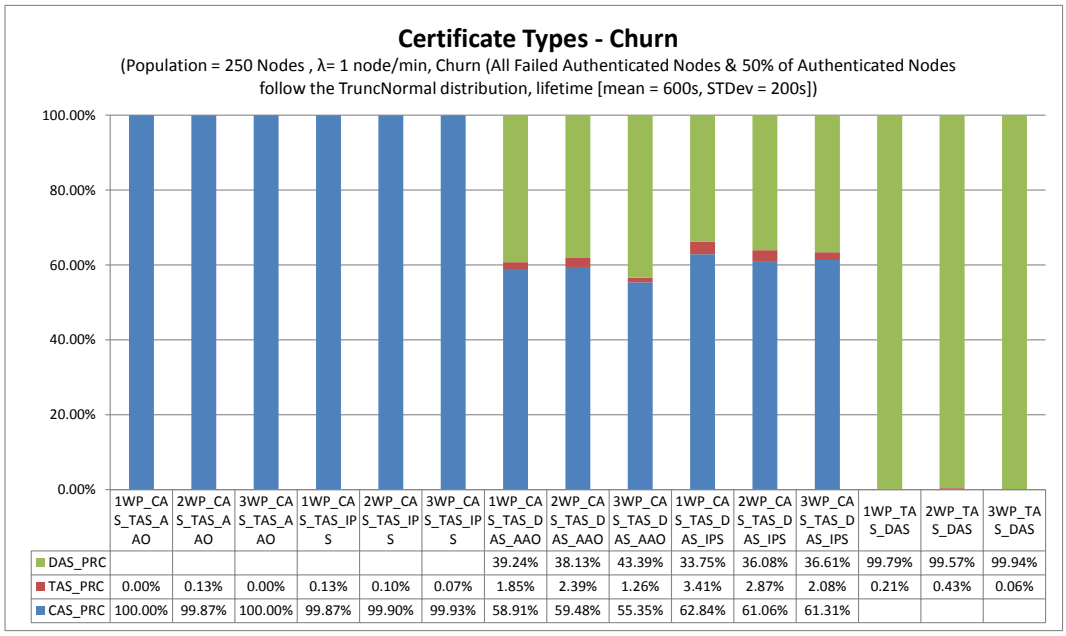
	1WP_CAS_TAS_AAO	2WP_CAS_TAS_AAO	3WP_CAS_TAS_AAO	1WP_CAS_TAS_IPS	2WP_CAS_TAS_IPS	3WP_CAS_TAS_IPS	1WP_CAS_TAS_DAS_AAO	2WP_CAS_TAS_DAS_AAO	3WP_CAS_TAS_DAS_AAO	1WP_CAS_TAS_DAS_IPS	2WP_CAS_TAS_DAS_IPS	3WP_CAS_TAS_DAS_IPS	1WP_TAS_DAS	2WP_TAS_DAS	3WP_TAS_DAS
DAS_PRC							7.99%	8.15%	8.14%	5.63%	5.94%	6.30%	12.03%	12.91%	13.94%
TAS_PRC	6.90%	8.59%	13.51%	3.56%	3.94%	5.48%	6.12%	7.68%	12.05%	3.46%	3.89%	4.65%	87.97%	87.09%	86.06%
CAS_PRC	93.10%	91.41%	86.49%	96.44%	96.06%	94.52%	85.89%	84.17%	79.81%	90.91%	90.17%	89.05%			

The percentage of different certificate type obtainability in the case of the *"250 Nodes No-Churn"* scenario.



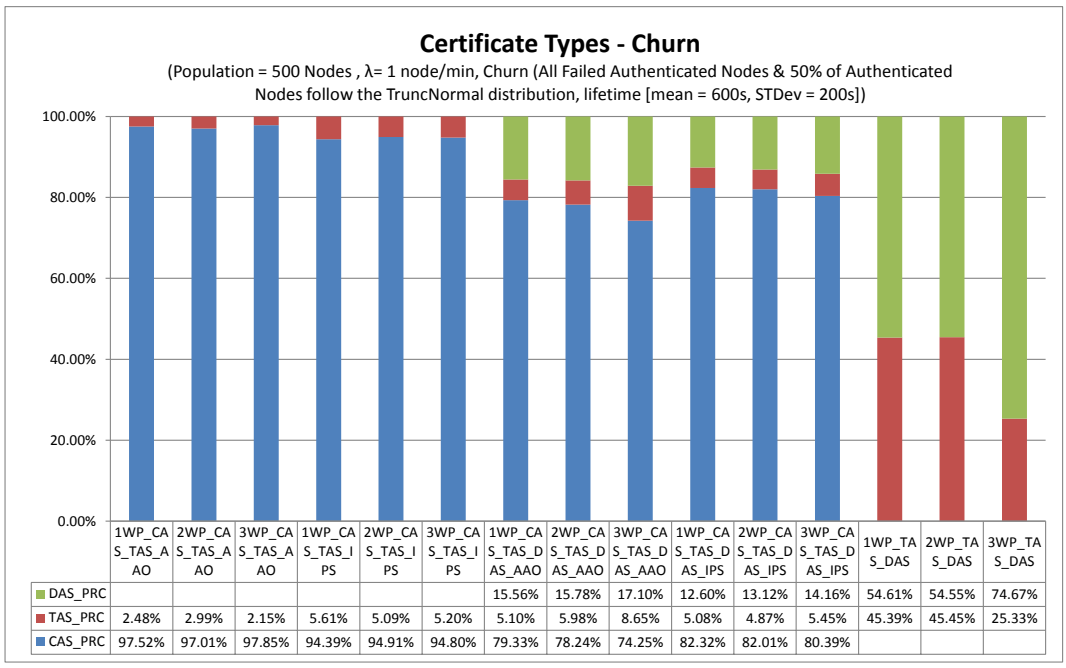
	1WP_CAS_TAS_AAO	2WP_CAS_TAS_AAO	3WP_CAS_TAS_AAO	1WP_CAS_TAS_IPS	2WP_CAS_TAS_IPS	3WP_CAS_TAS_IPS	1WP_CAS_TAS_DAS_AAO	2WP_CAS_TAS_DAS_AAO	3WP_CAS_TAS_DAS_AAO	1WP_CAS_TAS_DAS_IPS	2WP_CAS_TAS_DAS_IPS	3WP_CAS_TAS_DAS_IPS	1WP_TAS_DAS	2WP_TAS_DAS	3WP_TAS_DAS
DAS_PRC							3.98%	4.24%	4.20%	2.73%	2.80%	2.97%	5.55%	5.90%	6.13%
TAS_PRC	7.03%	9.16%	15.87%	1.92%	2.30%	3.64%	7.19%	8.79%	15.02%	1.88%	2.30%	3.62%	94.45%	94.10%	93.87%
CAS_PRC	92.97%	90.84%	84.13%	98.08%	97.70%	96.36%	88.83%	86.97%	80.78%	95.40%	94.91%	93.40%			

The percentage of different certificate type obtainability in the case of the *"500 Nodes No-Churn"* scenario.



	1WP_CAS_TAS_AAO	2WP_CAS_TAS_AAO	3WP_CAS_TAS_AAO	1WP_CAS_TAS_IPS	2WP_CAS_TAS_IPS	3WP_CAS_TAS_IPS	1WP_CAS_TAS_DAS_AAO	2WP_CAS_TAS_DAS_AAO	3WP_CAS_TAS_DAS_AAO	1WP_CAS_TAS_DAS_IPS	2WP_CAS_TAS_DAS_IPS	3WP_CAS_TAS_DAS_IPS	1WP_TAS_DAS	2WP_TAS_DAS	3WP_TAS_DAS
DAS_PRC							39.24%	38.13%	43.39%	33.75%	36.08%	36.61%	99.79%	99.57%	99.94%
TAS_PRC	0.00%	0.13%	0.00%	0.13%	0.10%	0.07%	1.85%	2.39%	1.26%	3.41%	2.87%	2.08%	0.21%	0.43%	0.06%
CAS_PRC	100.00%	99.87%	100.00%	99.87%	99.90%	99.93%	58.91%	59.48%	55.35%	62.84%	61.06%	61.31%			

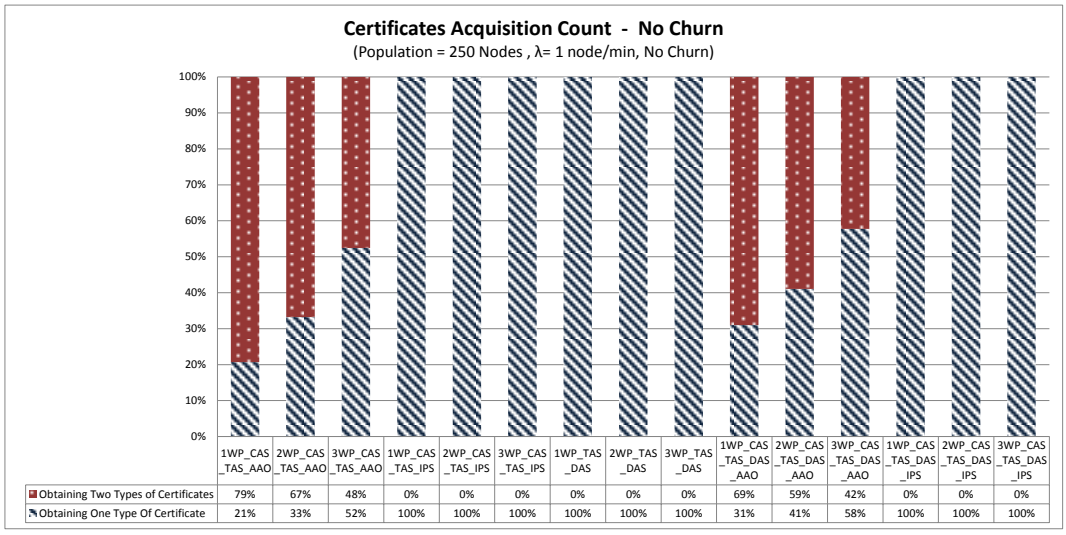
The percentage of different certificate type obtainability in the case of the *"250 Nodes Churn"* scenario.



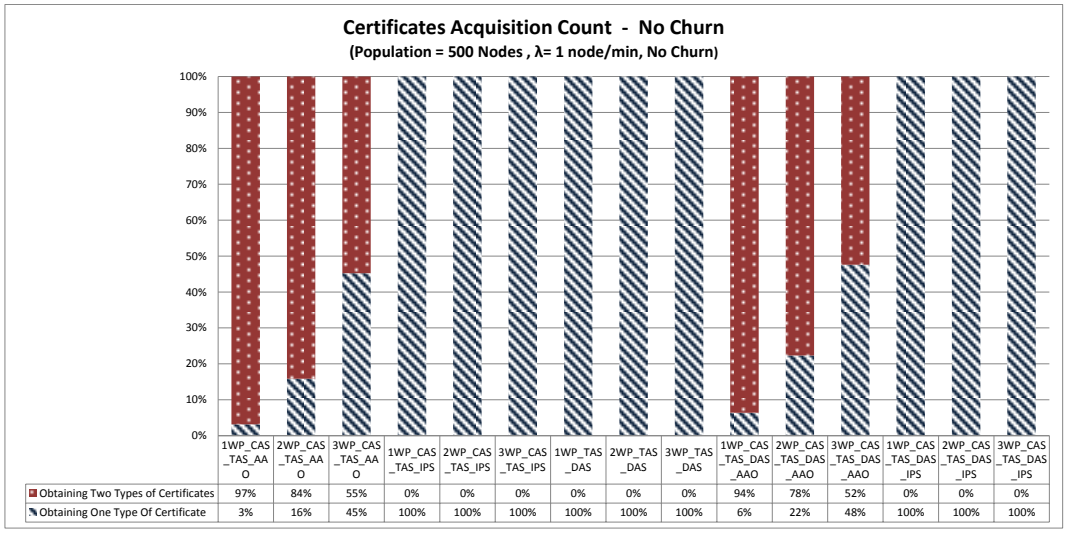
	1WP_CAS_TAS_AAO	2WP_CAS_TAS_AAO	3WP_CAS_TAS_AAO	1WP_CAS_TAS_IPS	2WP_CAS_TAS_IPS	3WP_CAS_TAS_IPS	1WP_CAS_TAS_DAS_AAO	2WP_CAS_TAS_DAS_AAO	3WP_CAS_TAS_DAS_AAO	1WP_CAS_TAS_DAS_IPS	2WP_CAS_TAS_DAS_IPS	3WP_CAS_TAS_DAS_IPS	1WP_TAS_DAS	2WP_TAS_DAS	3WP_TAS_DAS
DAS_PRC							15.56%	15.78%	17.10%	12.60%	13.12%	14.16%	54.61%	54.55%	74.67%
TAS_PRC	2.48%	2.99%	2.15%	5.61%	5.09%	5.20%	5.10%	5.98%	8.65%	5.08%	4.87%	5.45%	45.39%	45.45%	25.33%
CAS_PRC	97.52%	97.01%	97.85%	94.39%	94.91%	94.80%	79.33%	78.24%	74.25%	82.32%	82.01%	80.39%			

The percentage of different certificate type obtainability in the case of the *"500 Nodes Churn"* scenario.

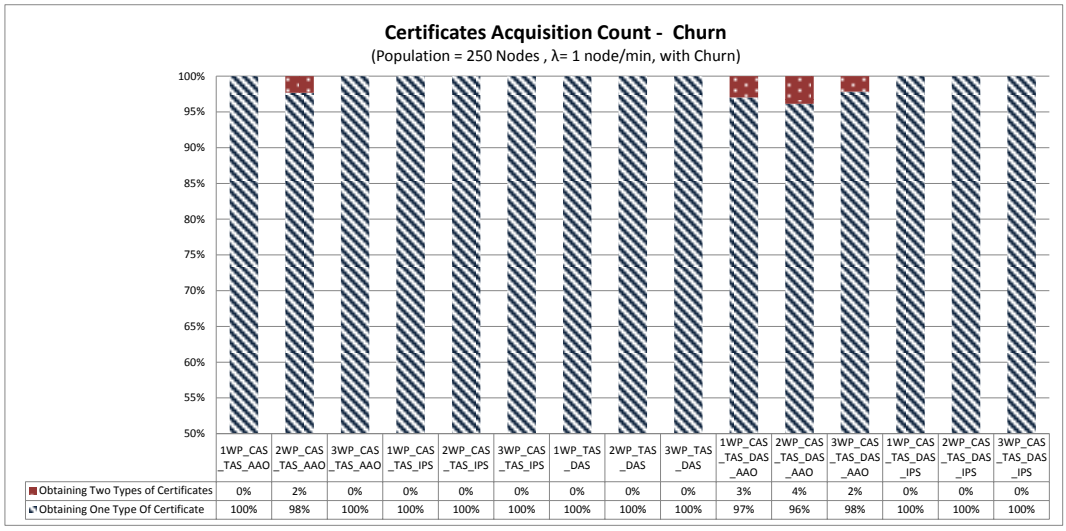
B.2.2. The Certificate Count



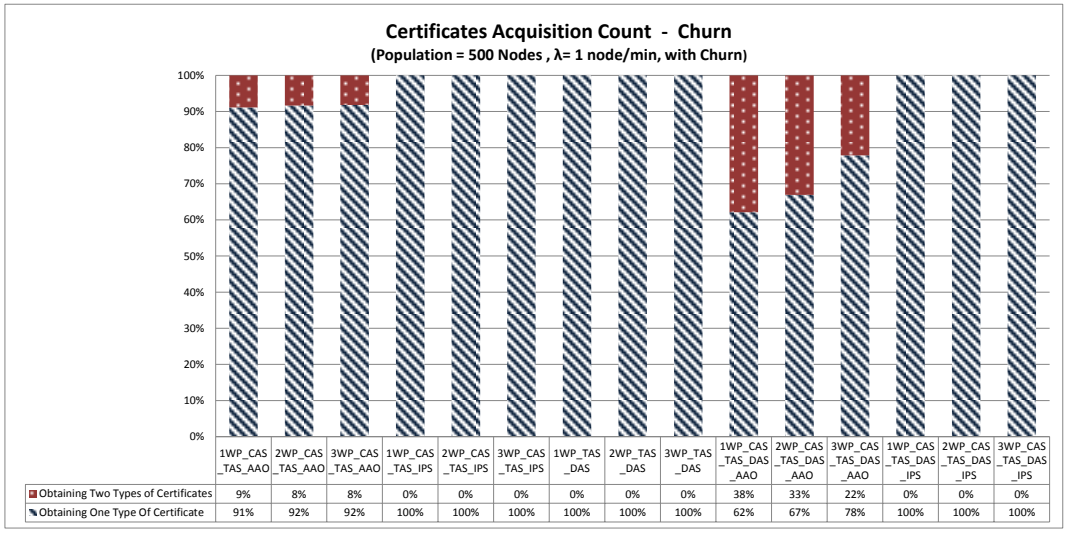
The percentage of certificate count obtainability in the case of the "*250-Nodes No-Churn*" scenario.



The percentage of certificate count obtainability in the case of the "*500-Nodes No-Churn*" scenario.



The percentage of certificate count obtainability in the case of the "*250-Nodes Churn*" scenario.

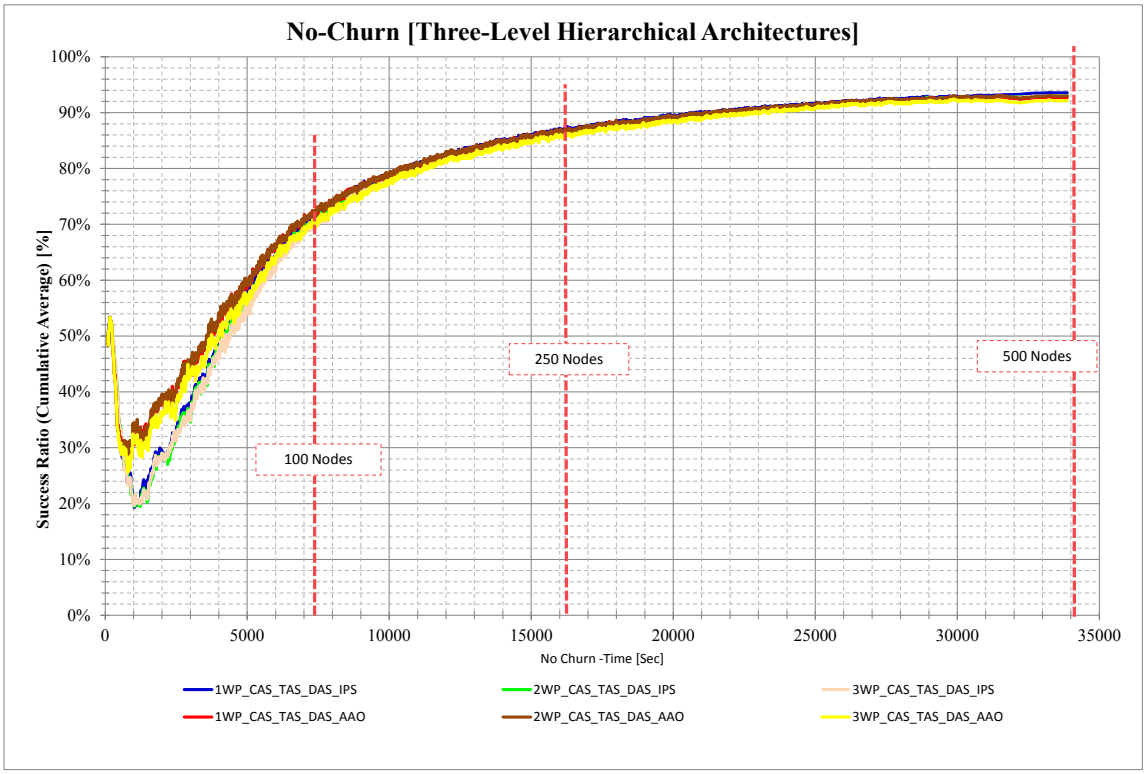


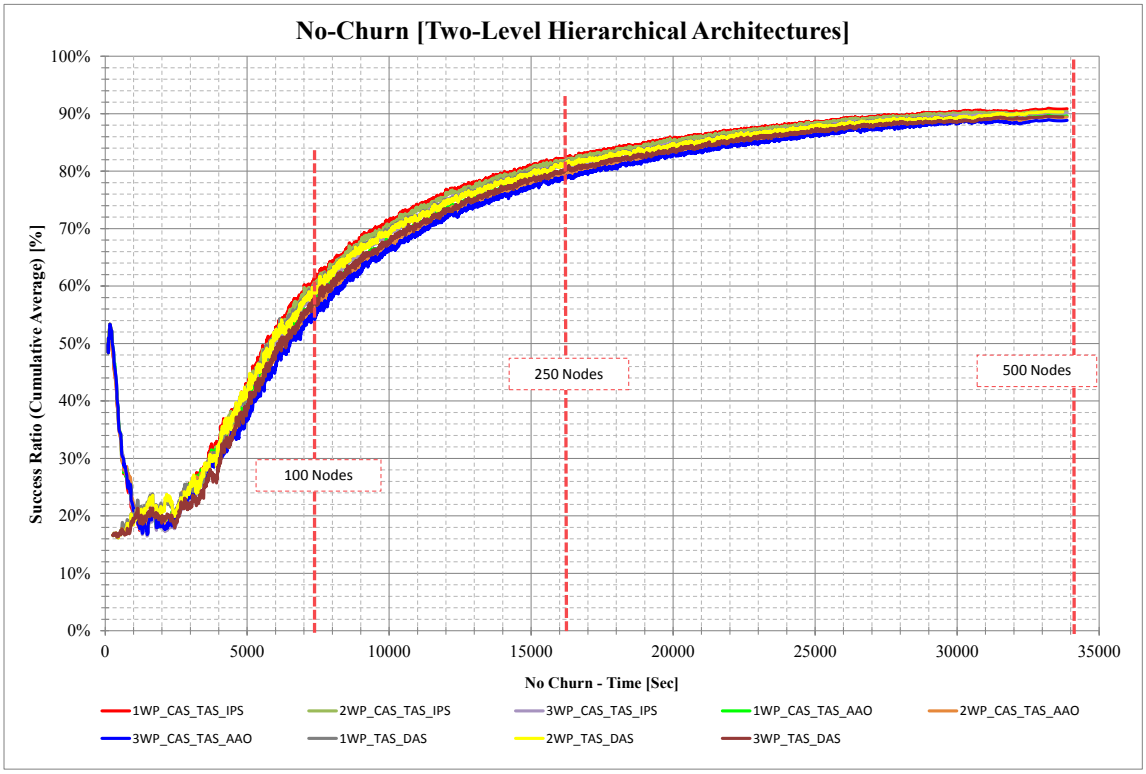
The percentage of certificate count obtainability in the case of the "*500-Nodes Churn*" scenario.

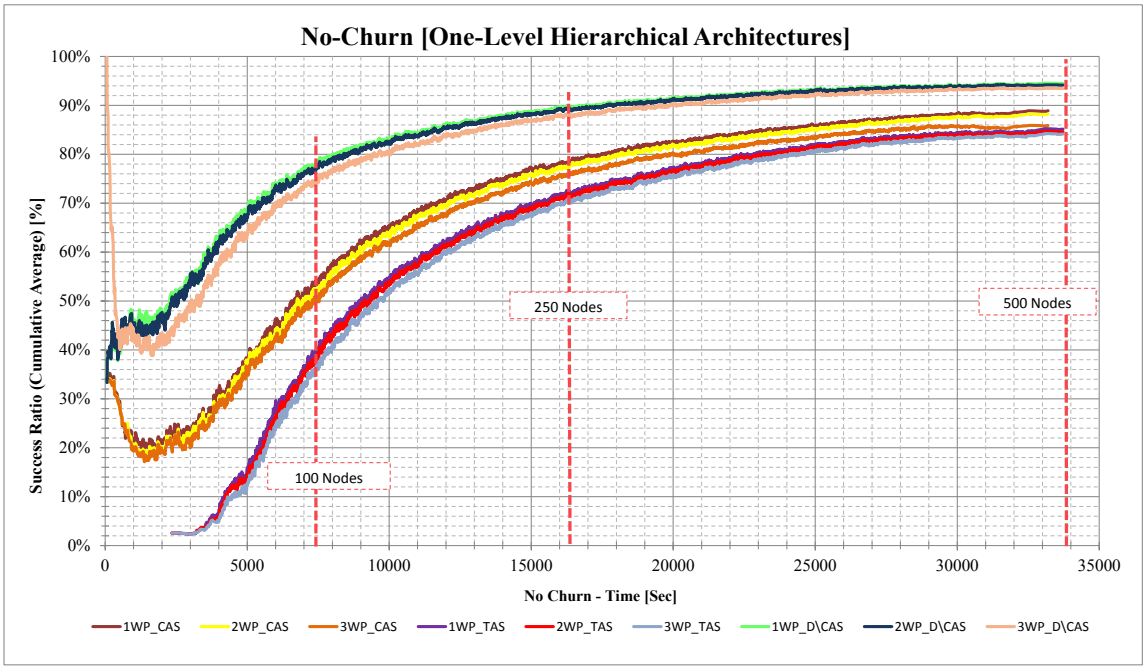
B.3. Time Series for the Common Main Metrics

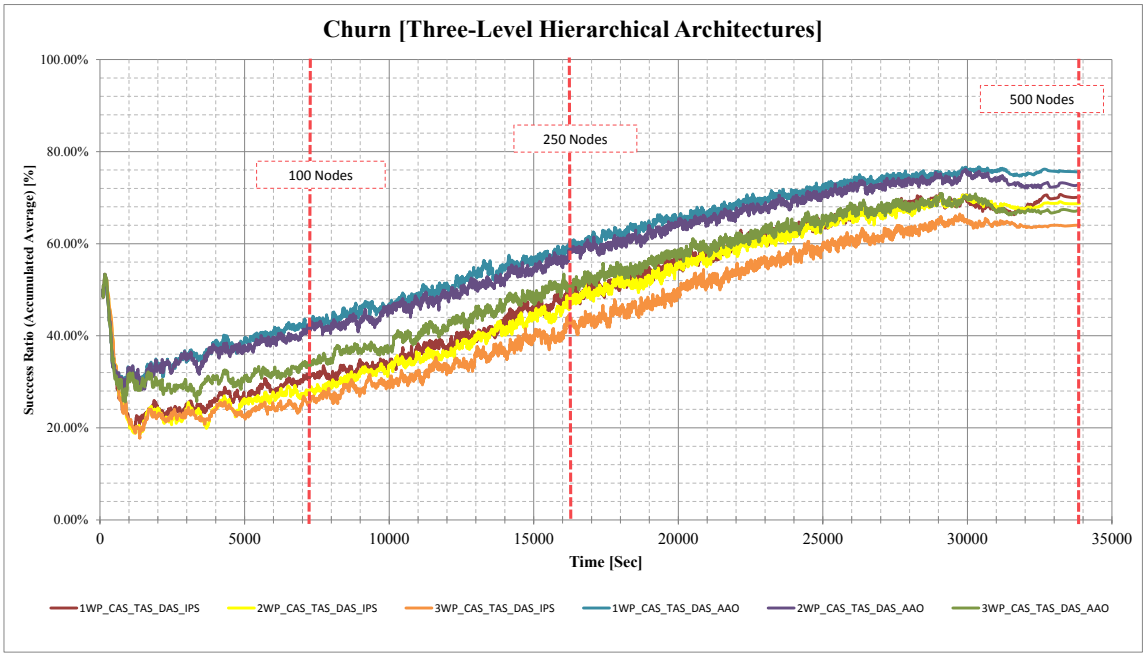
As the actual time-based scattered results are aggregated from all replications of each experiment, the results of each experiment that are presented in this section are estimated according to the so-called *Moving Average with a predefined smooth factor (α = 0.05)* in order to notice the output data variances across time and also for showing the result trend of each experiment.

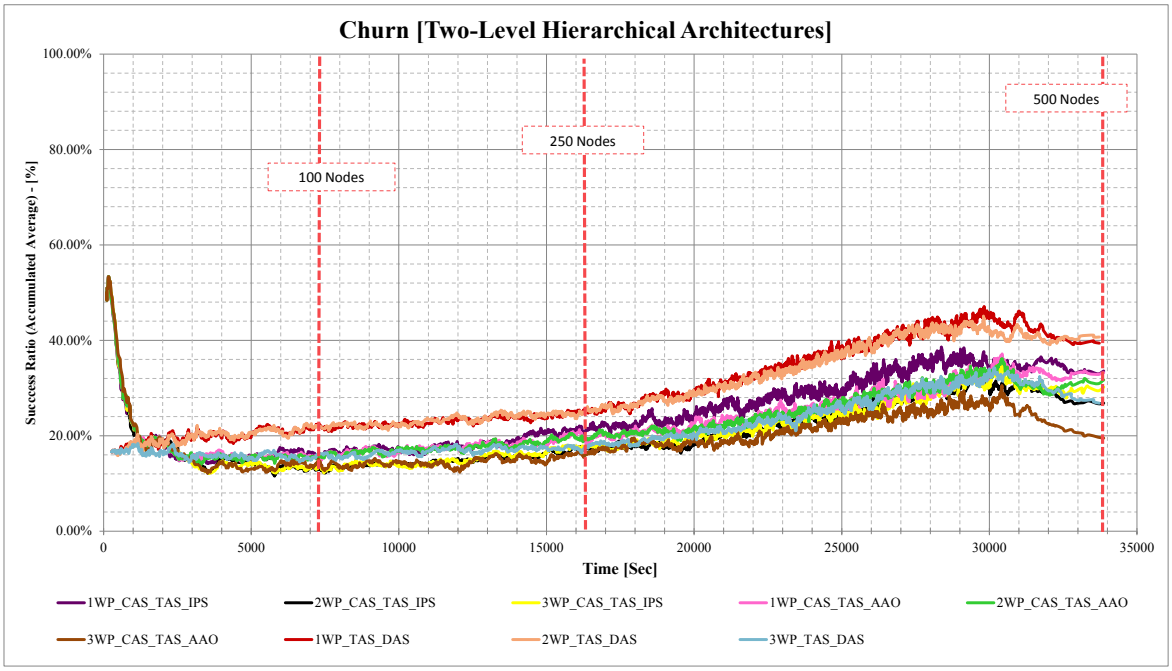
B.3.1. Success Ratio Vs Sim-Time

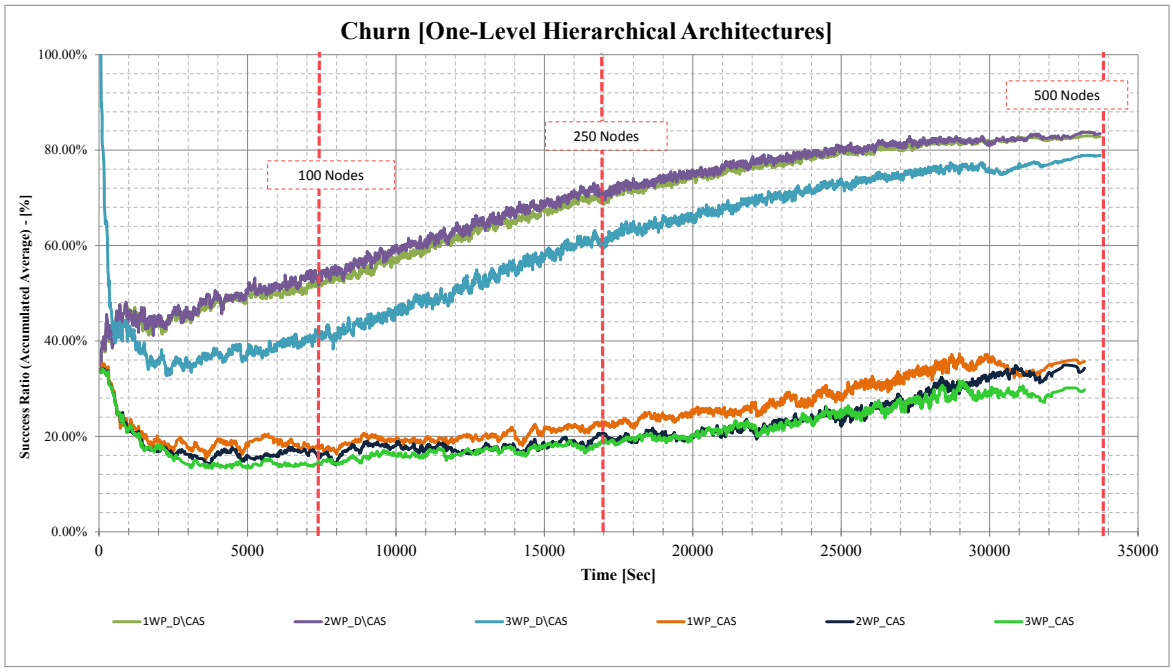



No-Churn [Two-Level Hierarchical Architectures]
Success Ratio (Cumulative Average) [%]
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
0
5000
10000
15000
20000
25000
30000
35000
No Churn - Time [Sec]
100 Nodes
250 Nodes
500 Nodes
1WP_CAS_TAS_IPS
2WP_CAS_TAS_IPS
3WP_CAS_TAS_IPS
1WP_CAS_TAS_AAO
2WP_CAS_TAS_AAO
3WP_CAS_TAS_AAO
1WP_TAS_DAS
2WP_TAS_DAS
3WP_TAS_DAS

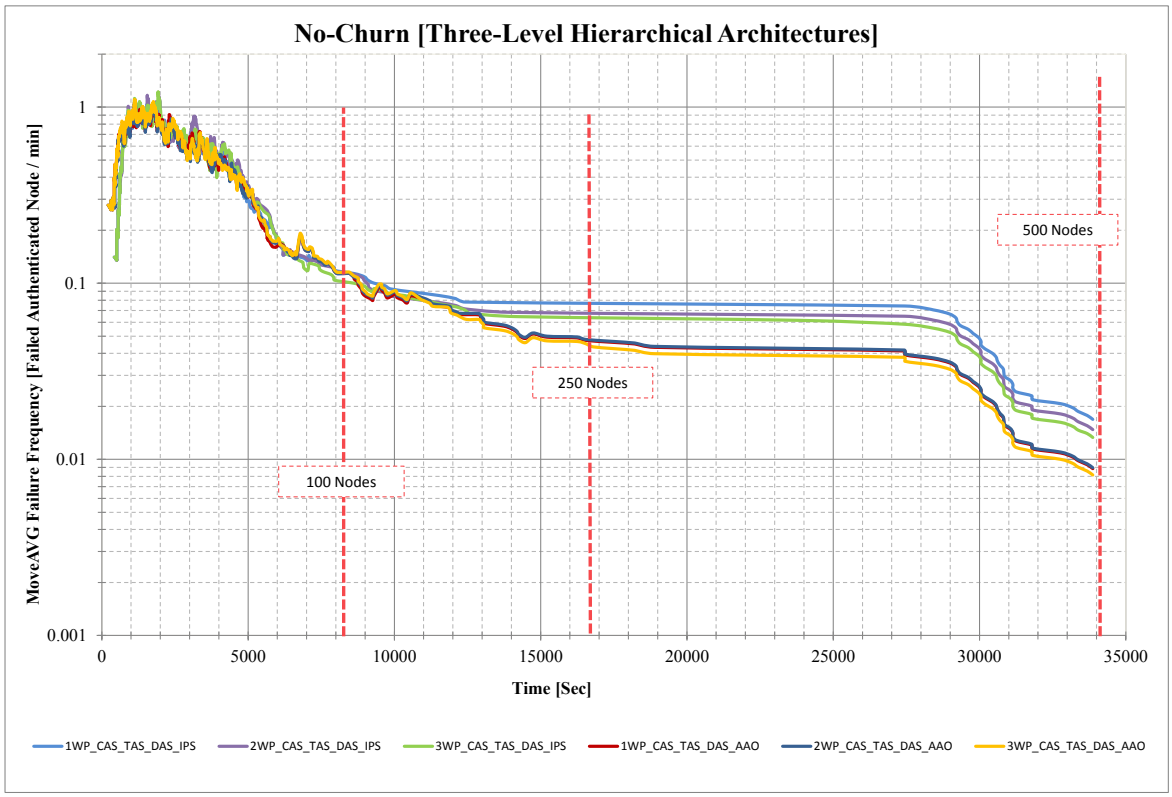

No-Churn [One-Level Hierarchical Architectures]
Success Ratio (Cumulative Average) [%]
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
0
5000
10000
15000
20000
25000
30000
35000
No Churn - Time [Sec]
100 Nodes
250 Nodes
500 Nodes
1WP_CAS
2WP_CAS
3WP_CAS
1WP_TAS
2WP_TAS
3WP_TAS
1WP_D\CAS
2WP_D\CAS
3WP_D\CAS

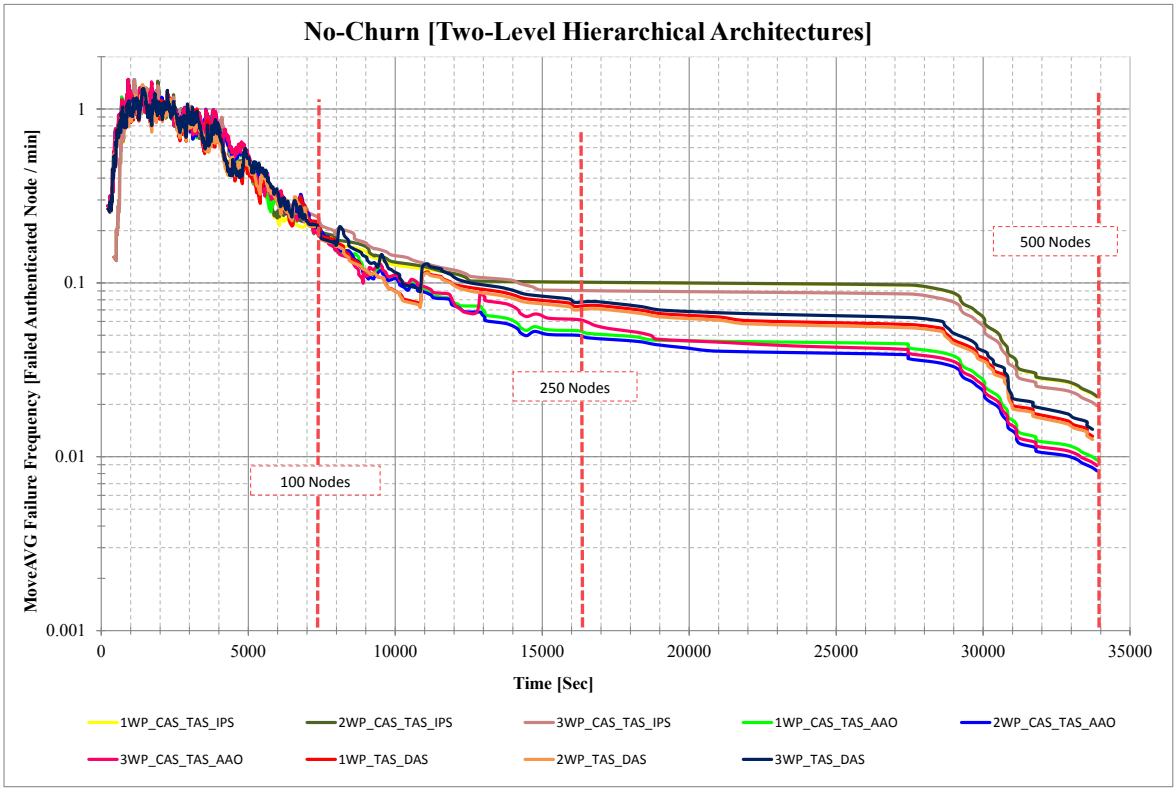

Churn [Three-Level Hierarchical Architectures]
100.00%
80.00%
60.00%
40.00%
20.00%
0.00%
Success Ratio (Accumulated Average) [%]
100 Nodes
250 Nodes
500 Nodes
0
5000
10000
15000
20000
25000
30000
35000
Time [Sec]
1WP_CAS_TAS_DAS_IPS
2WP_CAS_TAS_DAS_IPS
3WP_CAS_TAS_DAS_IPS
1WP_CAS_TAS_DAS_AAO
2WP_CAS_TAS_DAS_AAO
3WP_CAS_TAS_DAS_AAO

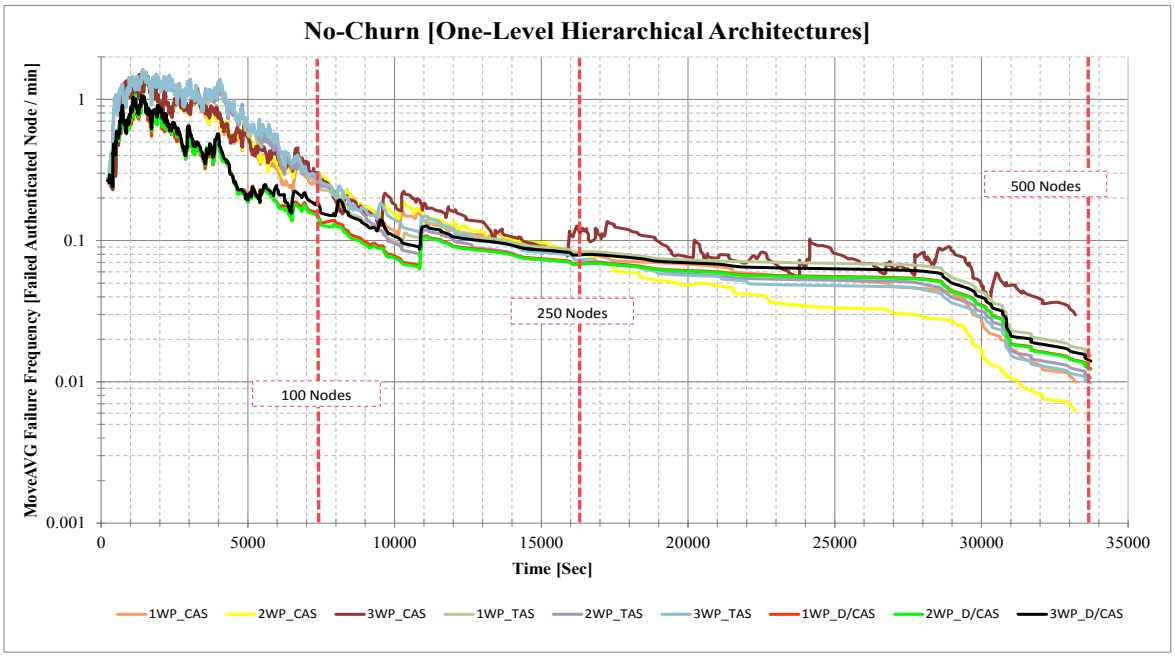

Churn [Two-Level Hierarchical Architectures]
100.00%
80.00%
60.00%
40.00%
20.00%
0.00%
Success Ratio (Accumulated Average) - [%]
100 Nodes
250 Nodes
500 Nodes
0
5000
10000
15000
20000
25000
30000
35000
Time [Sec]
1WP_CAS_TAS_IPS
2WP_CAS_TAS_IPS
3WP_CAS_TAS_IPS
1WP_CAS_TAS_AAO
2WP_CAS_TAS_AAO
3WP_CAS_TAS_AAO
1WP_TAS_DAS
2WP_TAS_DAS
3WP_TAS_DAS

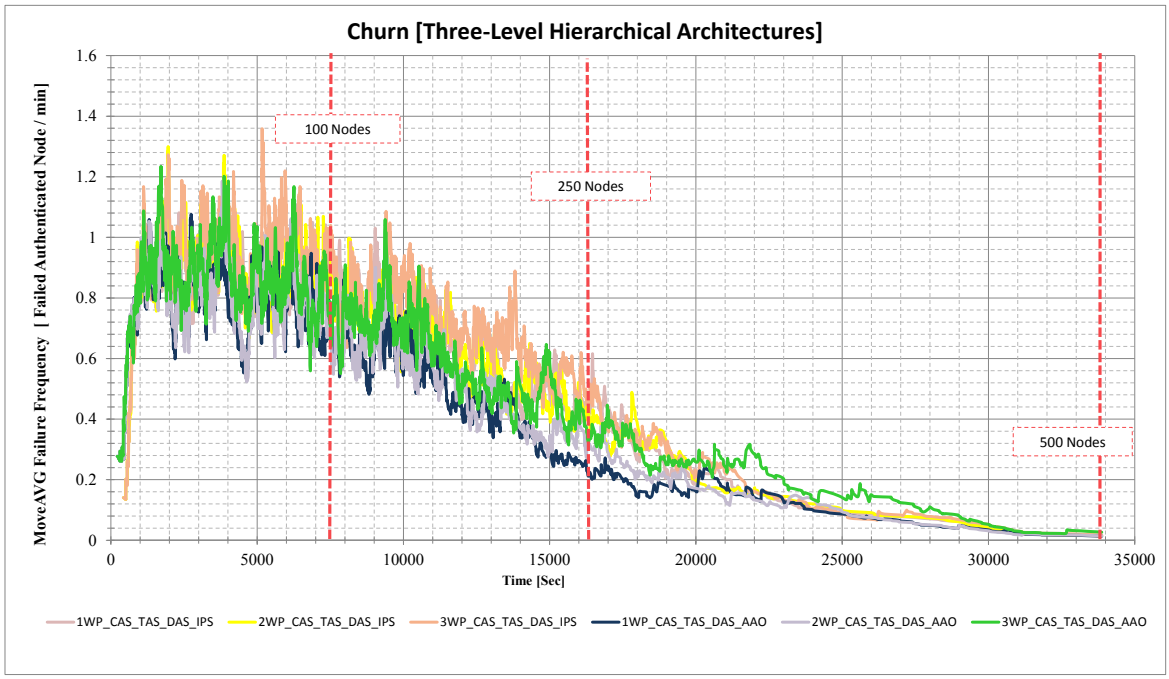


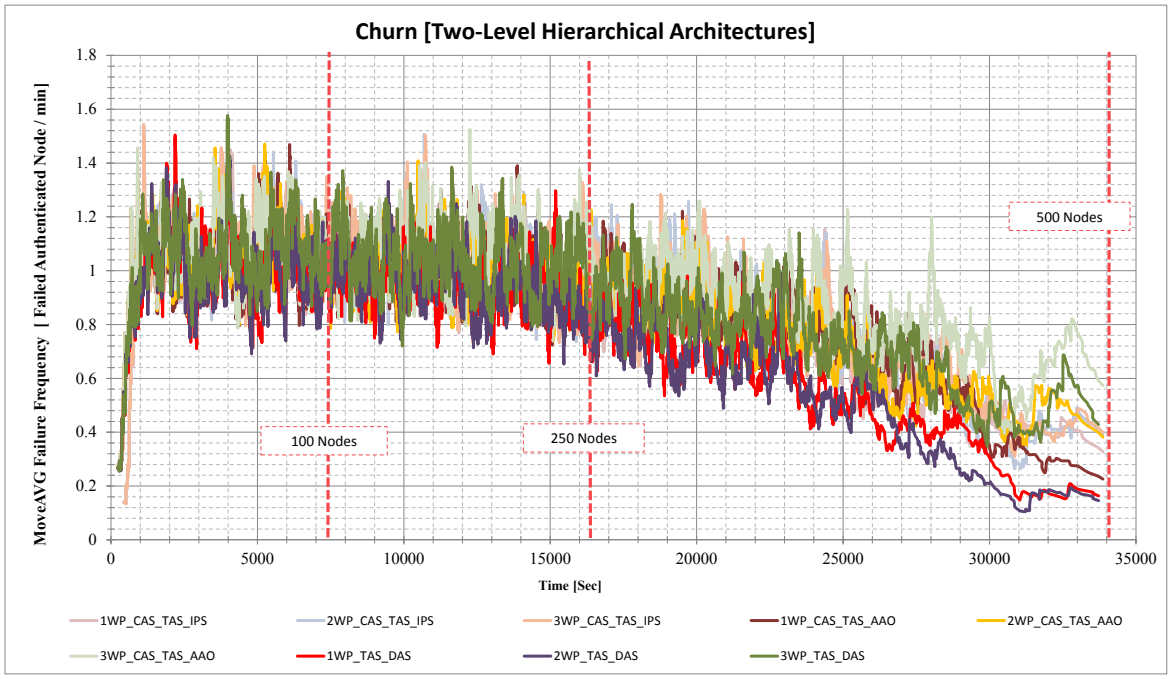
B.3.2. Failure Frequency Vs Sim-Time

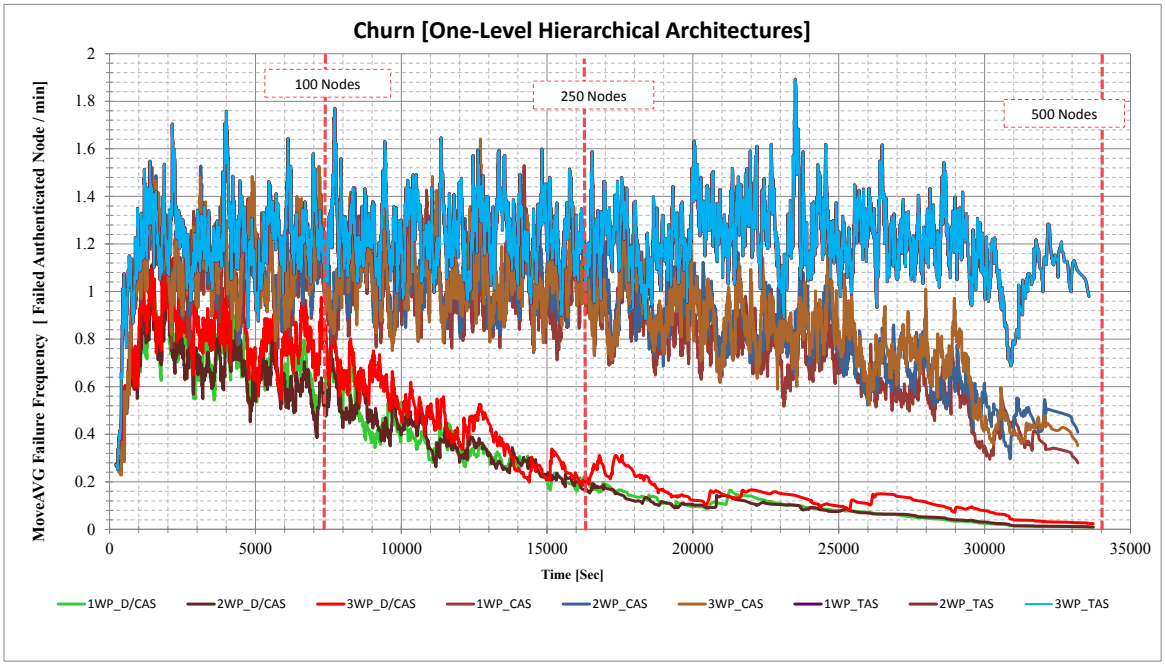



No-Churn [Two-Level Hierarchical Architectures]
MoveAVG Failure Frequency [Failed Authenticated Node / min]
1
0.1
0.01
0.001
0
5000
10000
15000
20000
25000
30000
35000
Time [Sec]
100 Nodes
250 Nodes
500 Nodes
1WP_CAS_TAS_IPS
2WP_CAS_TAS_IPS
3WP_CAS_TAS_IPS
1WP_CAS_TAS_AAO
2WP_CAS_TAS_AAO
3WP_CAS_TAS_AAO
1WP_TAS_DAS
2WP_TAS_DAS
3WP_TAS_DAS

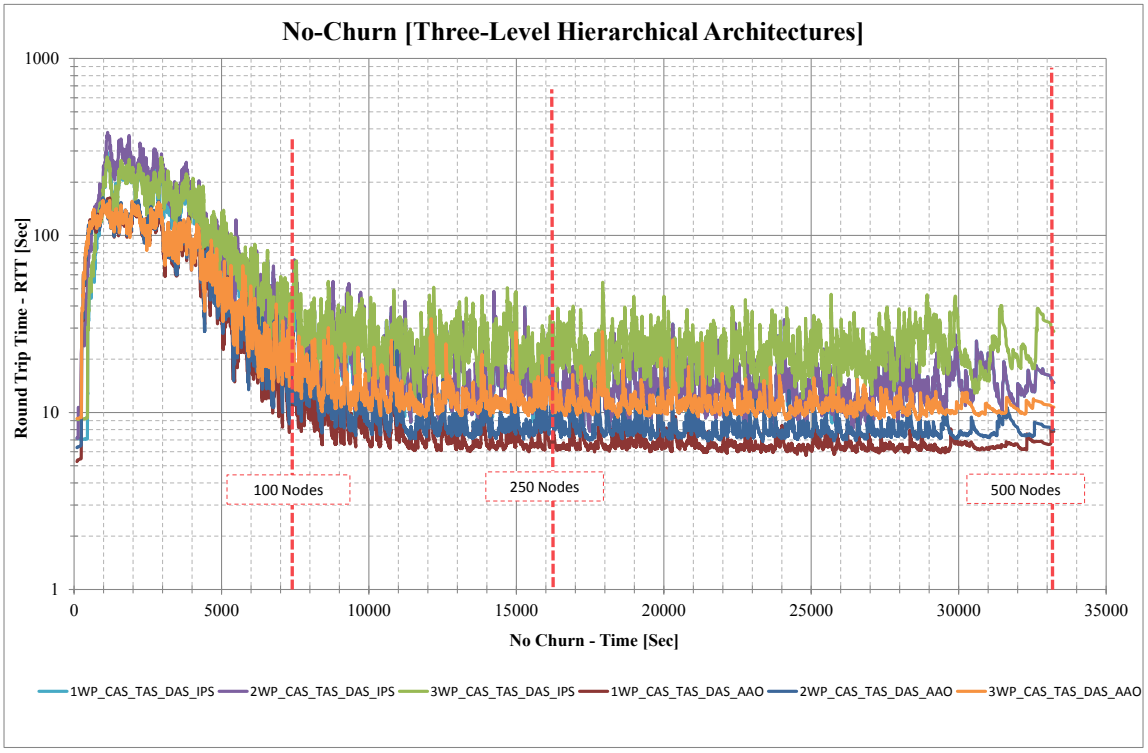

No-Churn [One-Level Hierarchical Architectures]
MoveAVG Failure Frequency [Failed Authenticated Node / min]
1
0.1
0.01
0.001
0
5000
10000
15000
20000
25000
30000
35000
Time [Sec]
100 Nodes
250 Nodes
500 Nodes
1WP_CAS
2WP_CAS
3WP_CAS
1WP_TAS
2WP_TAS
3WP_TAS
1WP_D/CAS
2WP_D/CAS
3WP_D/CAS

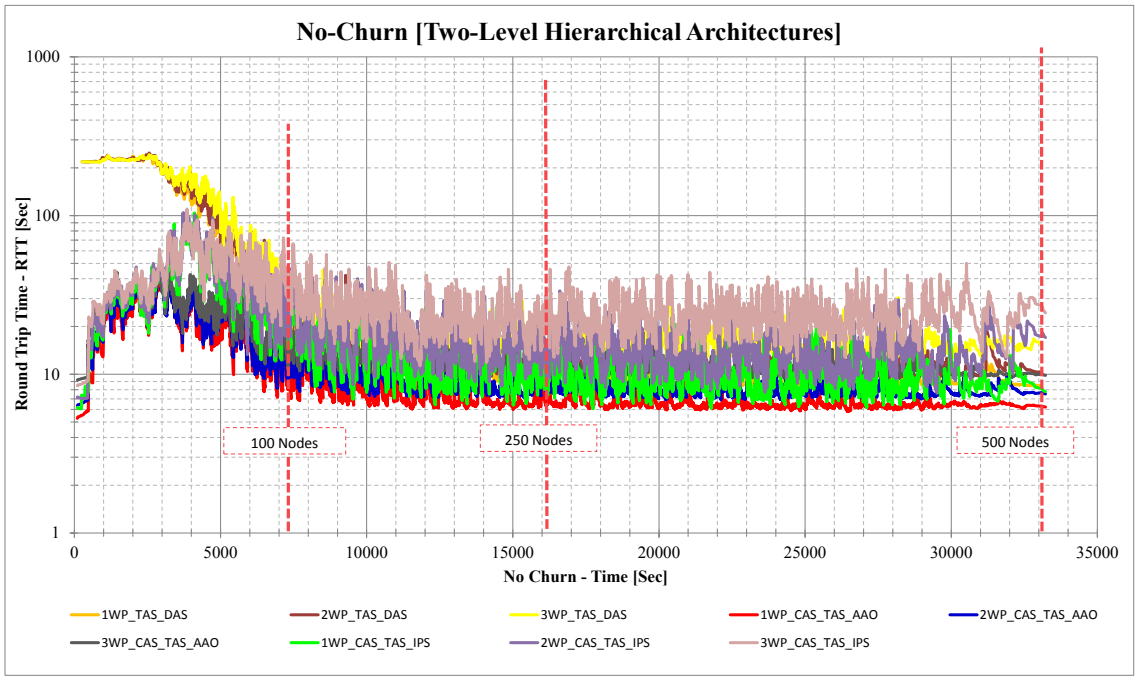

Churn [Three-Level Hierarchical Architectures]
MoveAVG Failure Frequency [Failed Authenticated Node / min]
Time [Sec]
100 Nodes
250 Nodes
500 Nodes
1WP_CAS_TAS_DAS_IPS
2WP_CAS_TAS_DAS_IPS
3WP_CAS_TAS_DAS_IPS
1WP_CAS_TAS_DAS_AAO
2WP_CAS_TAS_DAS_AAO
3WP_CAS_TAS_DAS_AAO

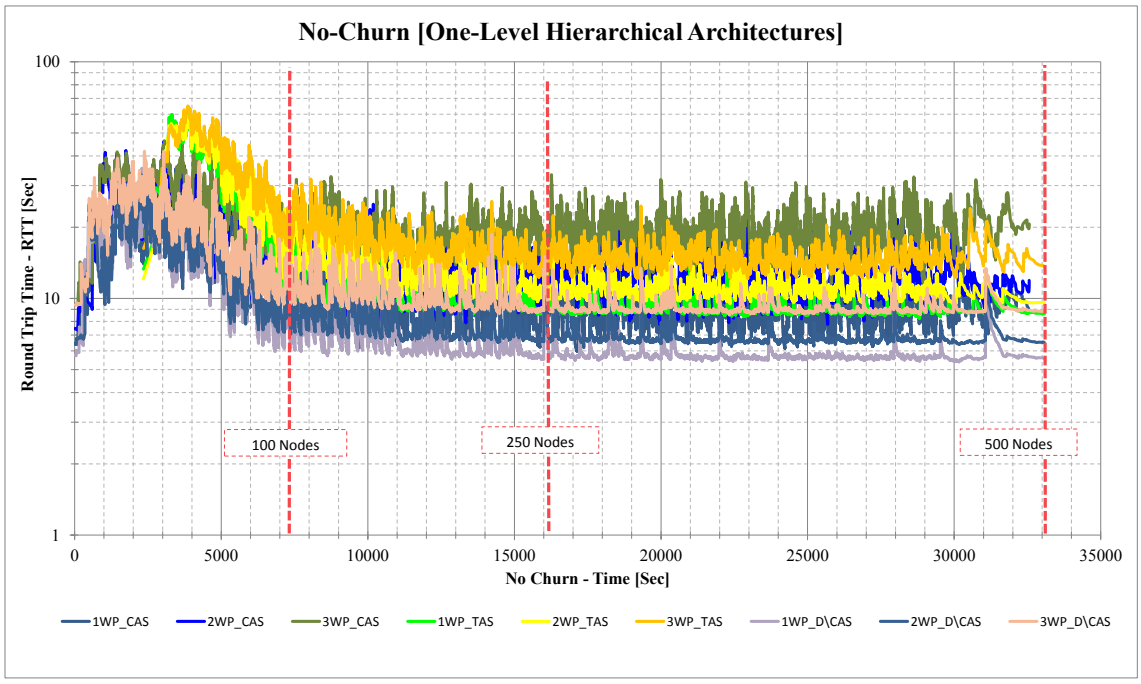

Churn [Two-Level Hierarchical Architectures]
MoveAVG Failure Frequency [Failed Authenticated Node / min]
Time [Sec]
100 Nodes
250 Nodes
500 Nodes
1WP_CAS_TAS_IPS
2WP_CAS_TAS_IPS
3WP_CAS_TAS_IPS
1WP_CAS_TAS_AAO
2WP_CAS_TAS_AAO
3WP_CAS_TAS_AAO
1WP_TAS_DAS
2WP_TAS_DAS
3WP_TAS_DAS

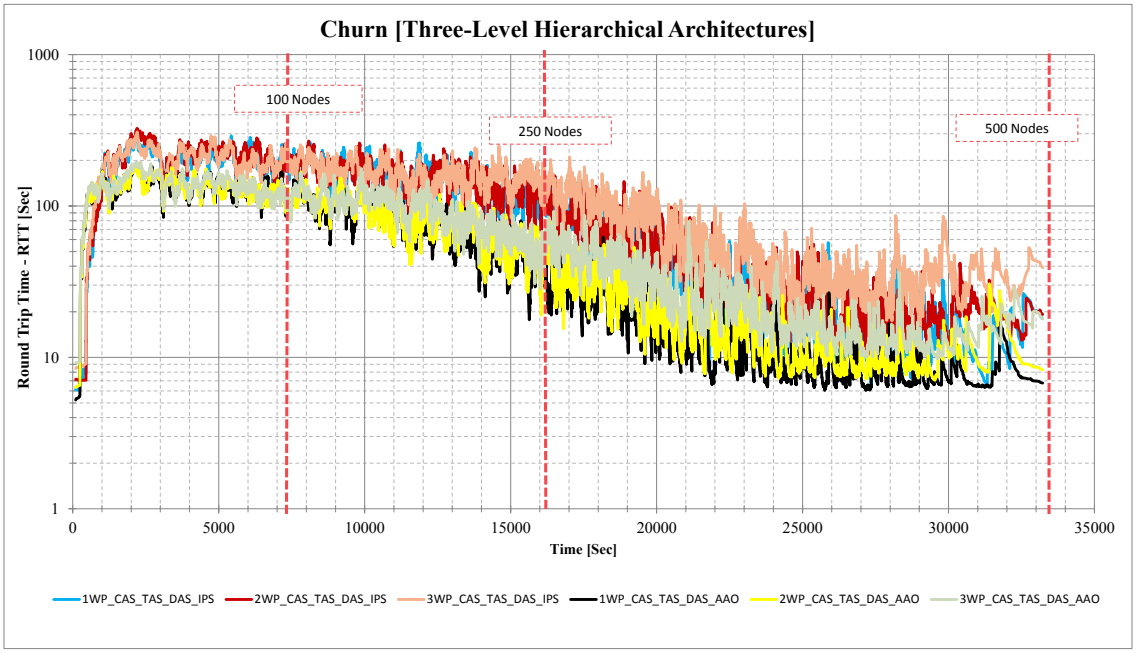


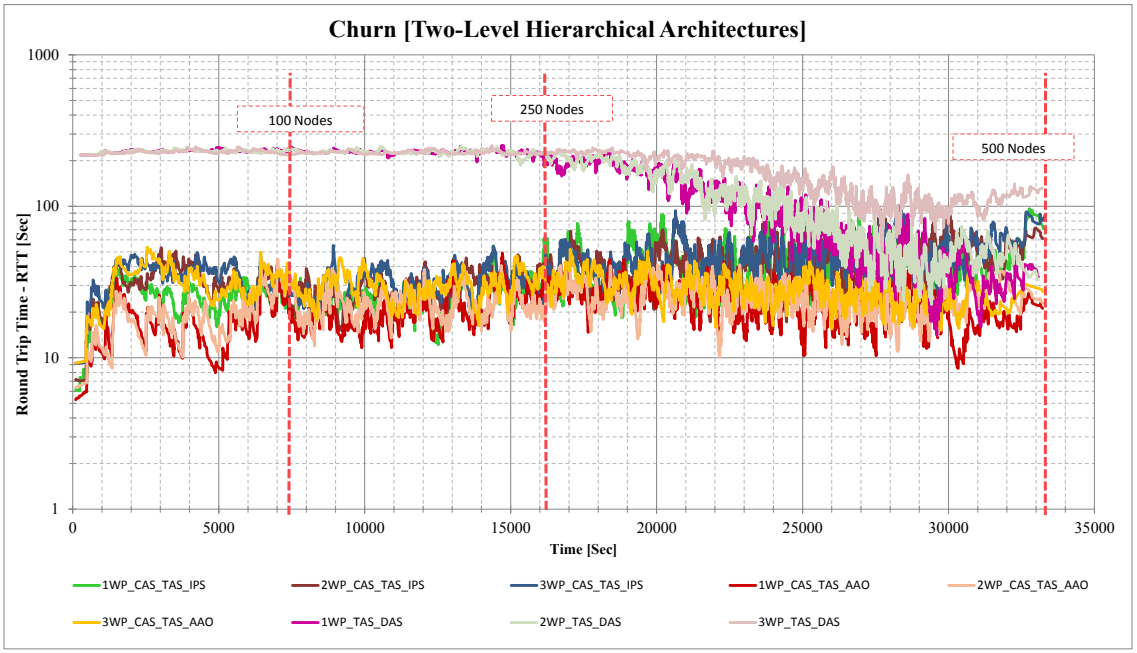
B.3.3. RTT Vs Sim-Time

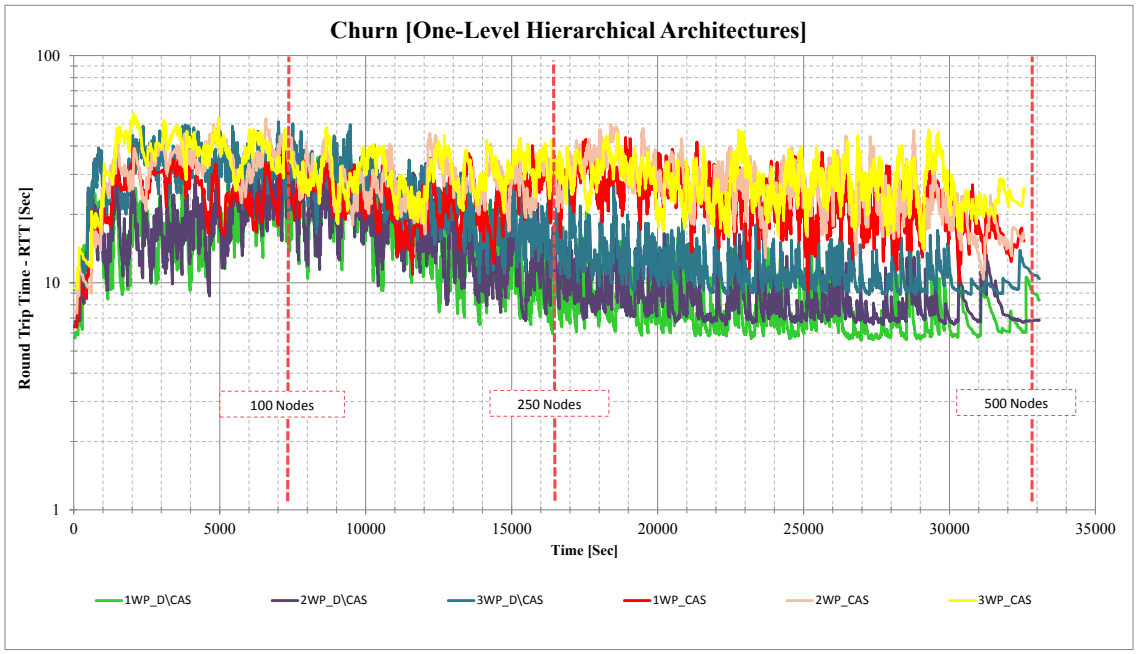



No-Churn [Two-Level Hierarchical Architectures]
Round Trip Time - RTT [Sec]
1000
100
10
1
0
5000
10000
15000
20000
25000
30000
35000
No Churn - Time [Sec]
100 Nodes
250 Nodes
500 Nodes
1WP_TAS_DAS
2WP_TAS_DAS
3WP_TAS_DAS
1WP_CAS_TAS_AAO
2WP_CAS_TAS_AAO
3WP_CAS_TAS_AAO
1WP_CAS_TAS_IPS
2WP_CAS_TAS_IPS
3WP_CAS_TAS_IPS

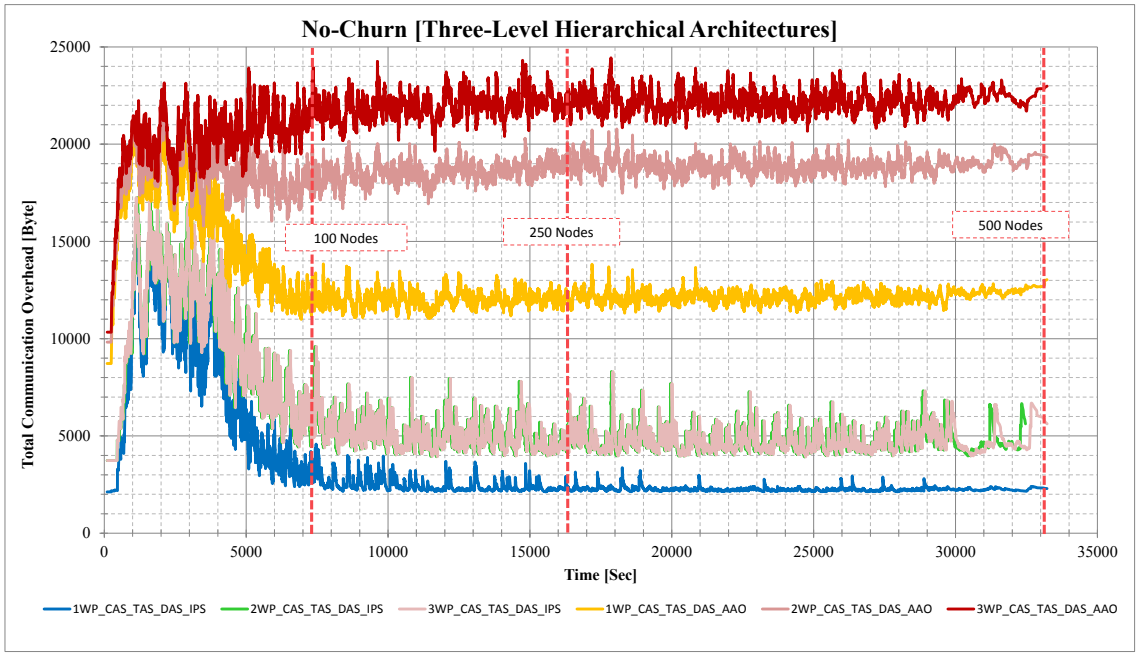

No-Churn [One-Level Hierarchical Architectures]
Round Trip Time - RTT [Sec]
100
10
1
0
5000
10000
15000
20000
25000
30000
35000
No Churn - Time [Sec]
100 Nodes
250 Nodes
500 Nodes
1WP_CAS
2WP_CAS
3WP_CAS
1WP_TAS
2WP_TAS
3WP_TAS
1WP_D\CAS
2WP_D\CAS
3WP_D\CAS

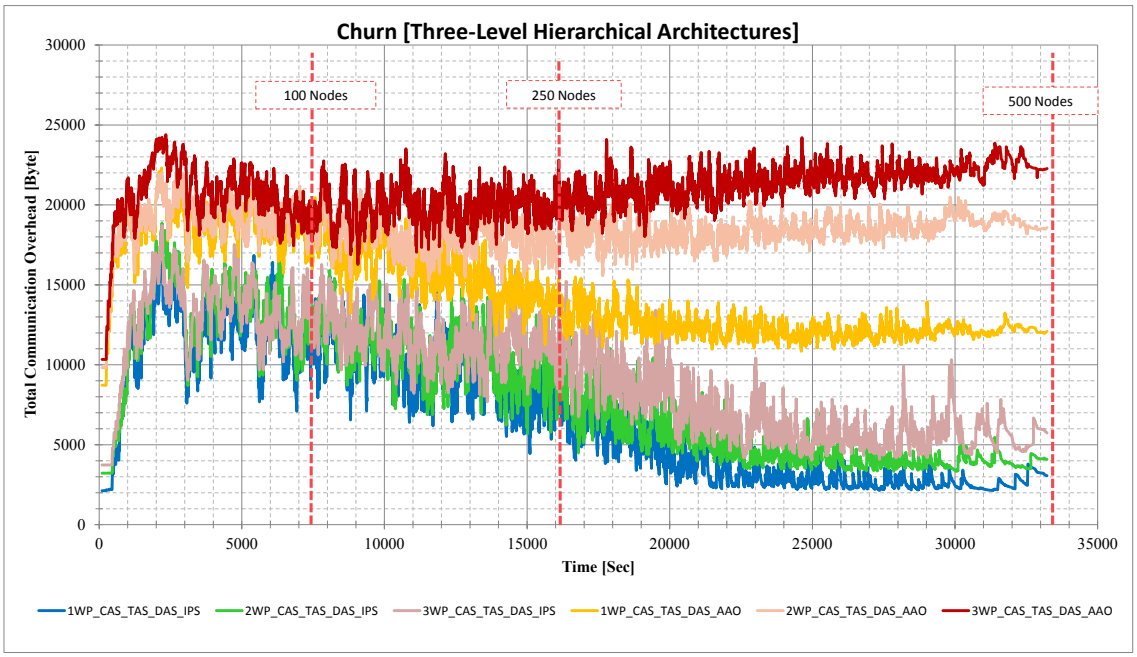

Churn [Three-Level Hierarchical Architectures]
Round Trip Time - RTT [Sec]
1000
100
10
1
0
5000
10000
15000
20000
25000
30000
35000
Time [Sec]
100 Nodes
250 Nodes
500 Nodes
1WP_CAS_TAS_DAS_IPS
2WP_CAS_TAS_DAS_IPS
3WP_CAS_TAS_DAS_IPS
1WP_CAS_TAS_DAS_AAO
2WP_CAS_TAS_DAS_AAO
3WP_CAS_TAS_DAS_AAO

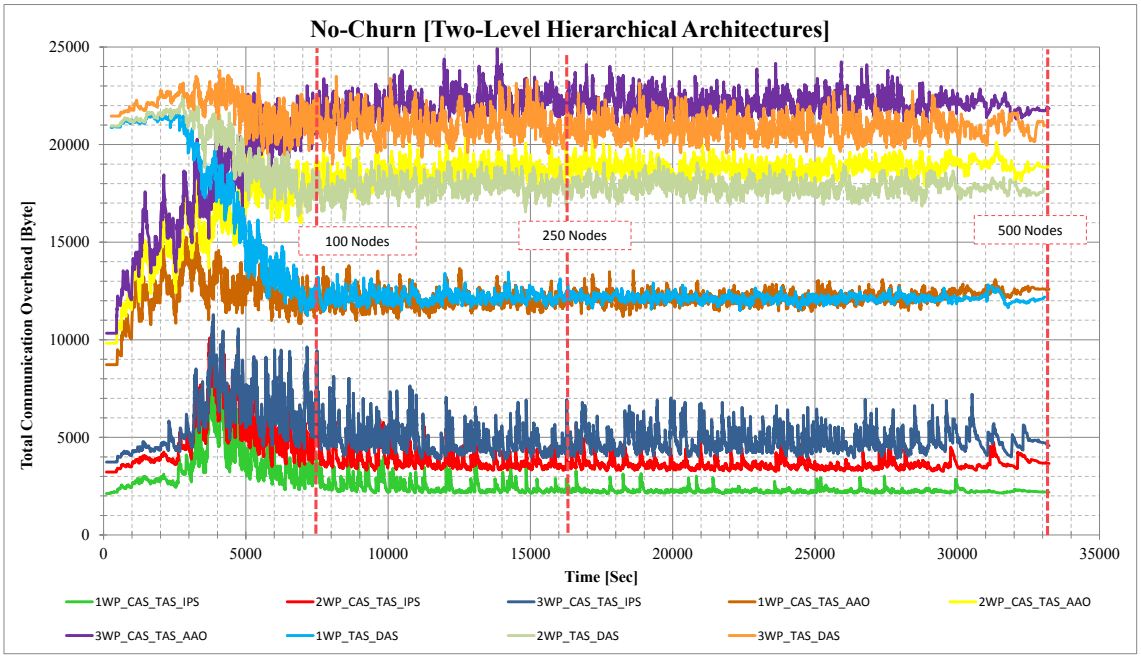

Churn [Two-Level Hierarchical Architectures]
Round Trip Time - RTT [Sec]
1000
100
10
1
0
5000
10000
15000
20000
25000
30000
35000
Time [Sec]
100 Nodes
250 Nodes
500 Nodes
1WP_CAS_TAS_IPS
2WP_CAS_TAS_IPS
3WP_CAS_TAS_IPS
1WP_CAS_TAS_AAO
2WP_CAS_TAS_AAO
3WP_CAS_TAS_AAO
1WP_TAS_DAS
2WP_TAS_DAS
3WP_TAS_DAS

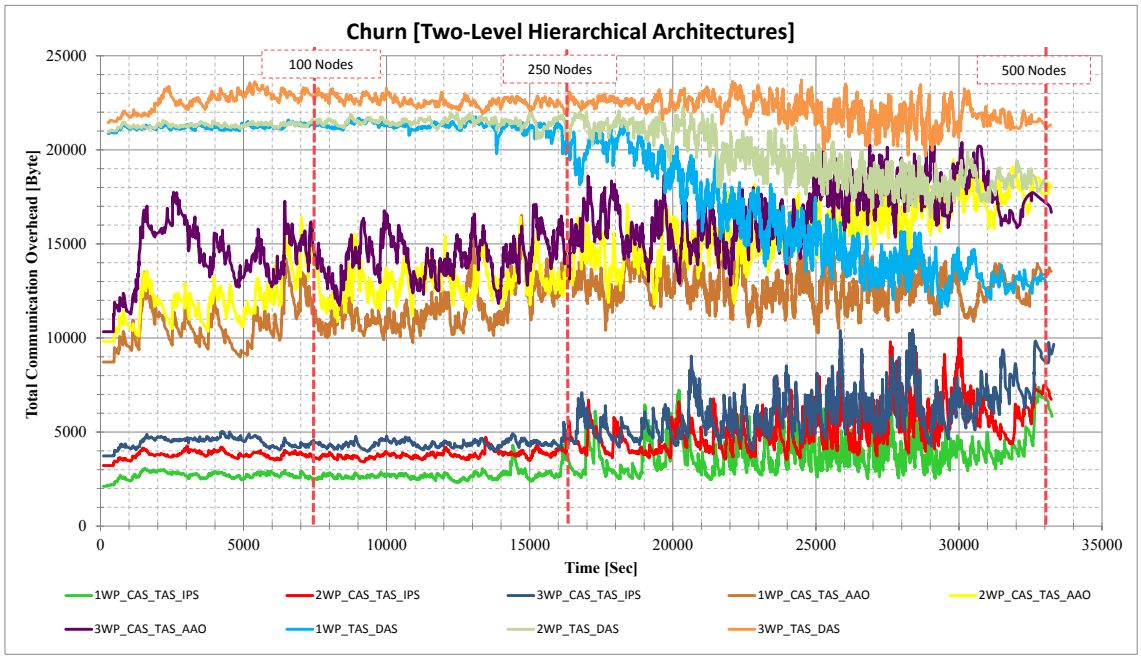


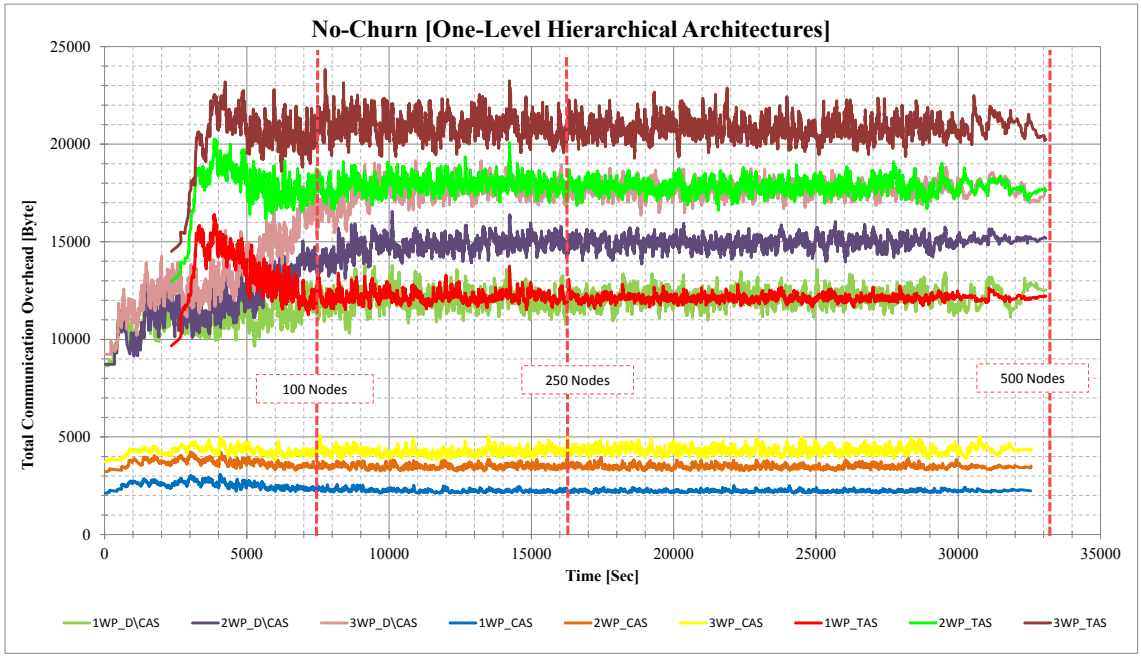
B.3.4. Communication Overhead Vs Sim-Time

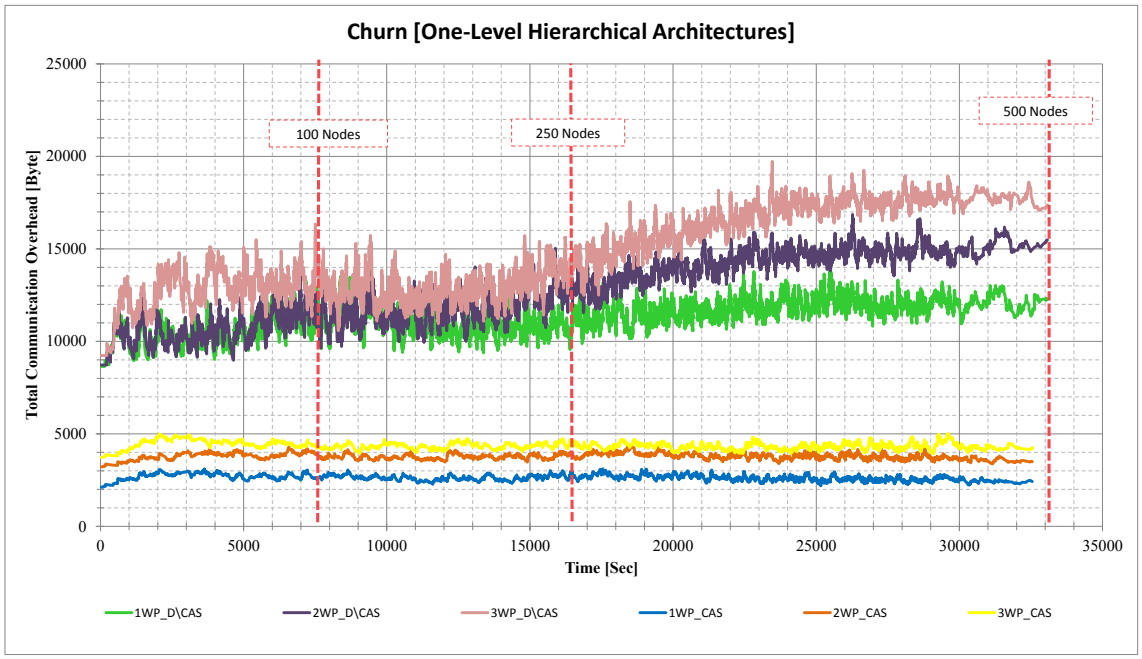



Churn [Three-Level Hierarchical Architectures]
Total Communication Overhead [Byte]
Time [Sec]
100 Nodes
250 Nodes
500 Nodes
1WP_CAS_TAS_DAS_IPS
2WP_CAS_TAS_DAS_IPS
3WP_CAS_TAS_DAS_IPS
1WP_CAS_TAS_DAS_AAO
2WP_CAS_TAS_DAS_AAO
3WP_CAS_TAS_DAS_AAO

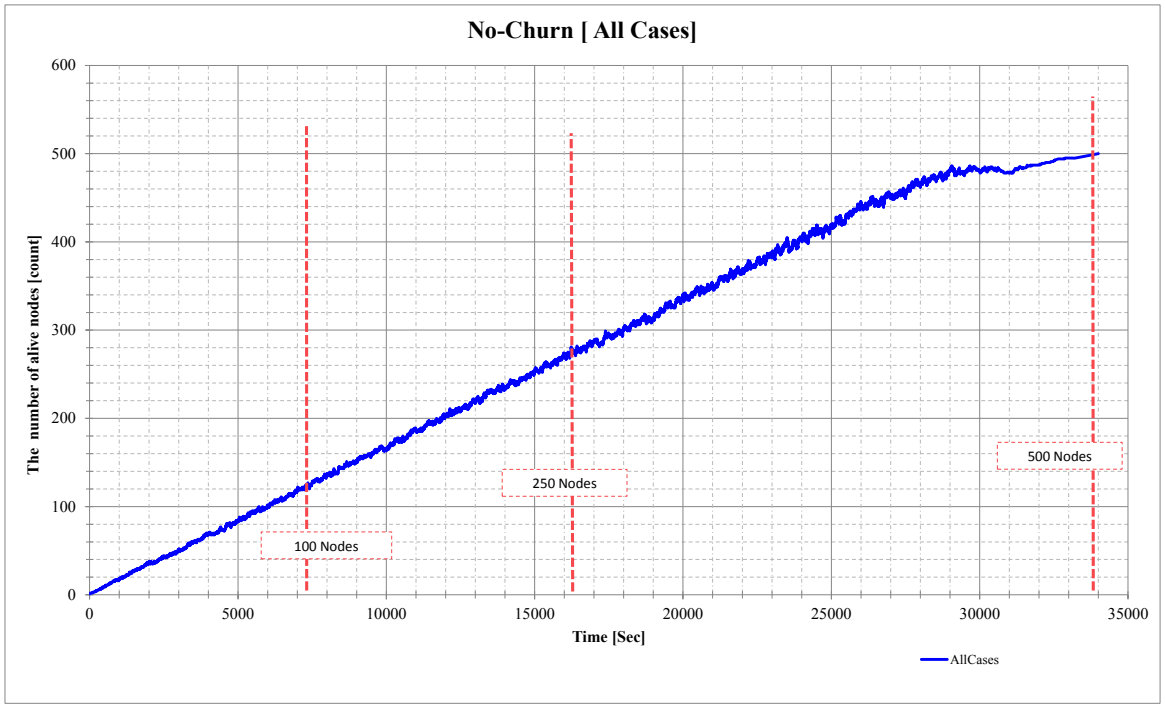

No-Churn [Two-Level Hierarchical Architectures]
Total Communication Overhead [Byte]
Time [Sec]
100 Nodes
250 Nodes
500 Nodes
1WP_CAS_TAS_IPS
2WP_CAS_TAS_IPS
3WP_CAS_TAS_IPS
1WP_CAS_TAS_AAO
2WP_CAS_TAS_AAO
3WP_CAS_TAS_AAO
1WP_TAS_DAS
2WP_TAS_DAS
3WP_TAS_DAS

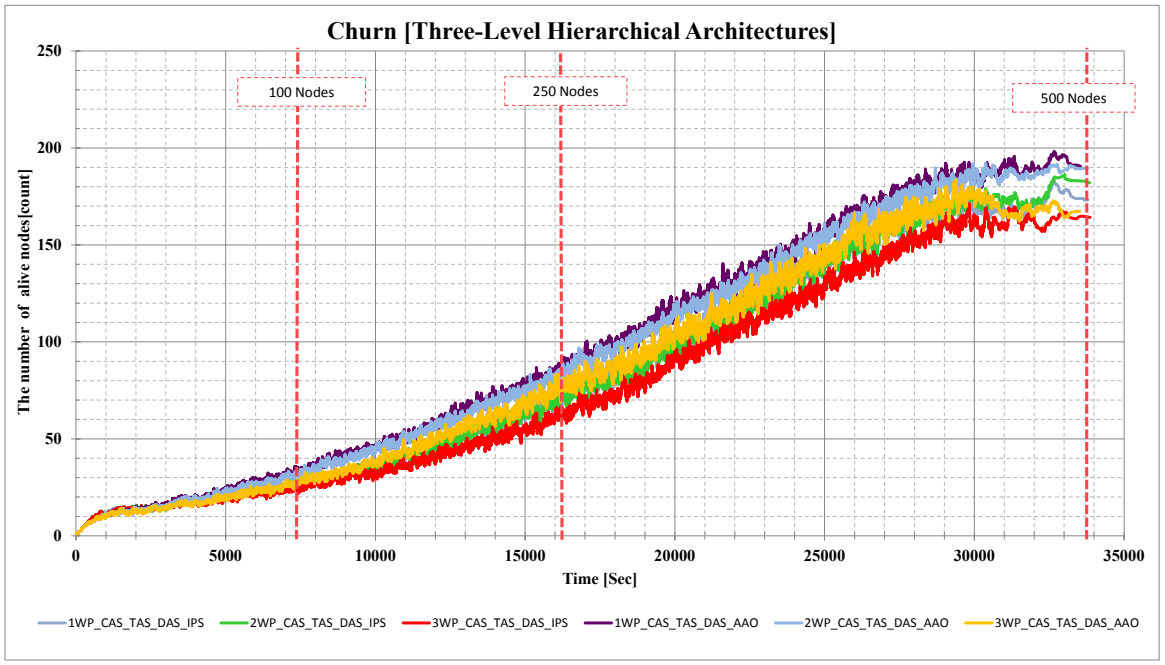

Churn [Two-Level Hierarchical Architectures]
100 Nodes
250 Nodes
500 Nodes
Total Communication Overhead [Byte]
Time [Sec]
1WP_CAS_TAS_IPS
2WP_CAS_TAS_IPS
3WP_CAS_TAS_IPS
1WP_CAS_TAS_AAO
2WP_CAS_TAS_AAO
3WP_CAS_TAS_AAO
1WP_TAS_DAS
2WP_TAS_DAS
3WP_TAS_DAS

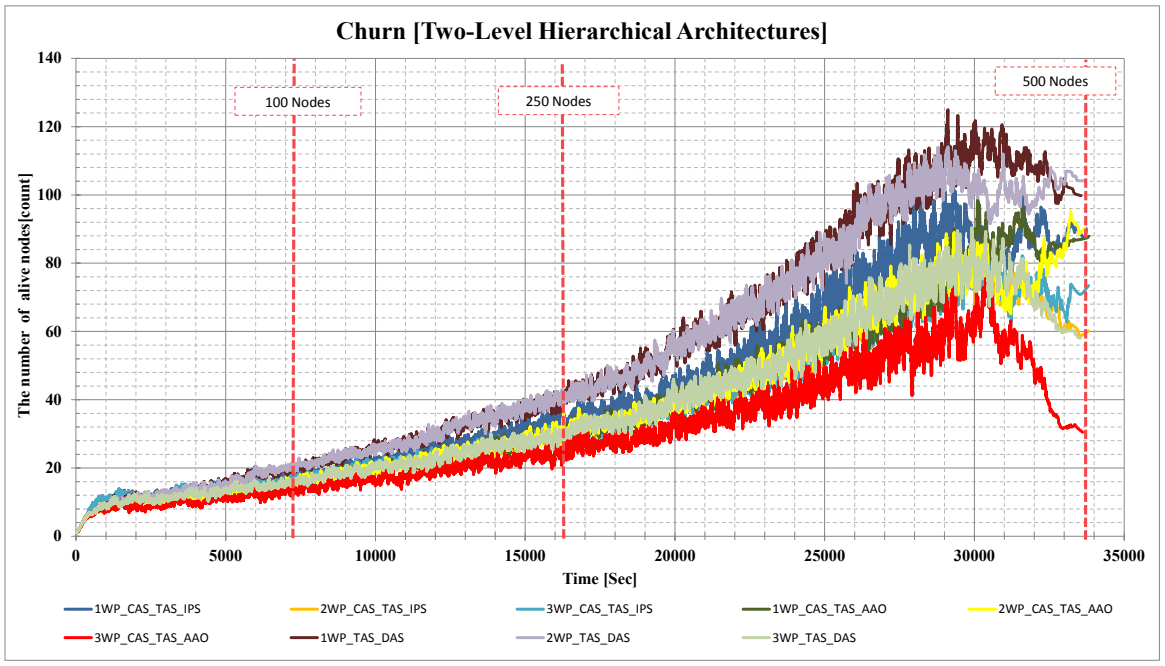

No-Churn [One-Level Hierarchical Architectures]
100 Nodes
250 Nodes
500 Nodes
Total Communication Overhead [Byte]
Time [Sec]
1WP_D\CAS
2WP_D\CAS
3WP_D\CAS
1WP_CAS
2WP_CAS
3WP_CAS
1WP_TAS
2WP_TAS
3WP_TAS

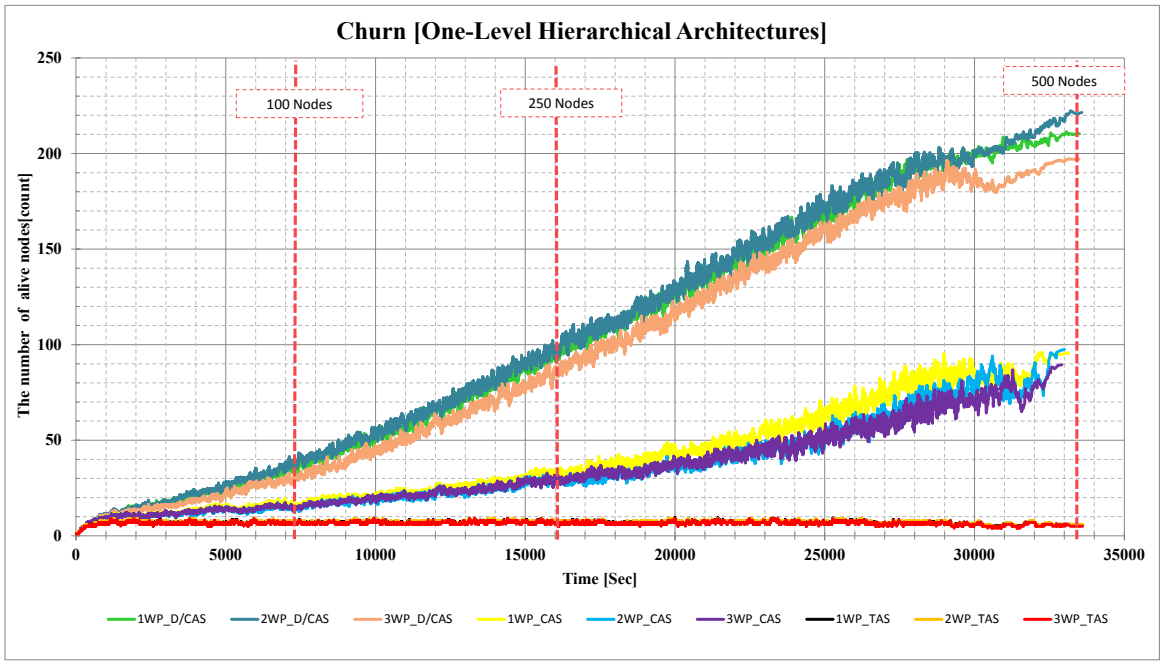


B.3.5. Alive-Node Vs Sim-Time




Churn [Three-Level Hierarchical Architectures]
100 Nodes
250 Nodes
500 Nodes
The number of alive nodes[count]
Time [Sec]
1WP_CAS_TAS_DAS_IPS
2WP_CAS_TAS_DAS_IPS
3WP_CAS_TAS_DAS_IPS
1WP_CAS_TAS_DAS_AAO
2WP_CAS_TAS_DAS_AAO
3WP_CAS_TAS_DAS_AAO


Churn [Two-Level Hierarchical Architectures]
100 Nodes
250 Nodes
500 Nodes
The number of alive nodes[count]
Time [Sec]
1WP_CAS_TAS_IPS
2WP_CAS_TAS_IPS
3WP_CAS_TAS_IPS
1WP_CAS_TAS_AAO
2WP_CAS_TAS_AAO
3WP_CAS_TAS_AAO
1WP_TAS_DAS
2WP_TAS_DAS
3WP_TAS_DAS


Churn [One-Level Hierarchical Architectures]
100 Nodes
250 Nodes
500 Nodes
The number of alive nodes[count]
Time [Sec]
0
50
100
150
200
250
0
5000
10000
15000
20000
25000
30000
35000
1WP_D/CAS
2WP_D/CAS
3WP_D/CAS
1WP_CAS
2WP_CAS
3WP_CAS
1WP_TAS
2WP_TAS
3WP_TAS

Appendix C

Success Ratio

Experiment ID = 100 Nodes (The "*Churn*" Scenario)
Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Success Ratio [%]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	16.35%	5.0%	1.0%	14.30%	18.40%
TAS	0%	NA	NA	NA	NA
TAS_DAS	20.60%	5.0%	1.0%	18.60%	22.60%
CAS_TAS_AAO	15.73%	4.6%	1.0%	13.70%	17.80%
CAS_TAS_IPS	15.63%	4.9%	1.0%	13.60%	17.70%
CAS_TAS_DAS_AAO	38.83%	5.7%	1.0%	36.80%	40.90%
CAS_TAS_DAS_IPS	30.07%	6.9%	1.0%	28.00%	32.10%
D\CAS	50.10%	7.0%	1.0%	48.10%	52.10%

a. Experiment ID = 100 (Ch), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Success Ratio [%]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	13.08%	5.7%	1.1%	10.90%	15.20%
TAS	0%	NA	NA	NA	NA
TAS_DAS	21.07%	5.1%	1.1%	18.90%	23.20%
CAS_TAS_AAO	16.13%	4.5%	1.1%	14.00%	18.30%
CAS_TAS_IPS	14.10%	5.2%	1.1%	12.00%	16.20%
CAS_TAS_DAS_AAO	38.70%	6.1%	1.1%	36.60%	40.80%
CAS_TAS_DAS_IPS	28.13%	5.9%	1.1%	26.00%	30.30%
D\CAS	49.43%	8.4%	1.1%	47.30%	51.60%

a. Experiment ID = 100 (Ch), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable:Success Ratio [%]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	13.18%	6.3%	1.1%	11.10%	15.30%
TAS	0%	NA	NA	NA	NA
TAS_DAS	14.67%	4.5%	1.1%	12.60%	16.80%
CAS_TAS_AAO	13.73%	5.2%	1.1%	11.60%	15.80%
CAS_TAS_IPS	13.07%	4.7%	1.1%	11.00%	15.20%
CAS_TAS_DAS_AAO	30.23%	6.0%	1.1%	28.10%	32.30%
CAS_TAS_DAS_IPS	26.30%	4.9%	1.1%	24.20%	28.40%
D\CAS	36.73%	8.5%	1.1%	34.60%	38.80%

a. Experiment ID = 100 (Ch), Authentication Protocol = 3WP

Experiment ID = 250 Nodes (The "*Churn*" Scenario)
Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Success Ratio [%]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	18.70%	3.6%	1.1%	16.50%	20.90%
TAS	0%	NA	NA	NA	NA
TAS_DAS	21.74%	4.3%	1.1%	19.50%	24.00%
CAS_TAS_AAO	17.87%	4.0%	1.1%	15.70%	20.10%
CAS_TAS_IPS	18.42%	3.6%	1.1%	16.20%	20.60%
CAS_TAS_DAS_AAO	51.77%	8.4%	1.1%	49.60%	54.00%
CAS_TAS_DAS_IPS	45.69%	9.7%	1.1%	43.50%	47.90%
D\CAS	64.17%	6.2%	1.1%	62.00%	66.40%

a. Experiment ID = 250 (Ch), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Success Ratio [%]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	16.09%	4.2%	1.1%	14.00%	18.20%
TAS	0%	NA	NA	NA	NA
TAS_DAS	21.91%	3.7%	1.1%	19.80%	24.00%
CAS_TAS_AAO	17.25%	4.3%	1.1%	15.20%	19.30%
CAS_TAS_IPS	16.79%	4.3%	1.1%	14.70%	18.90%
CAS_TAS_DAS_AAO	53.34%	7.4%	1.1%	51.30%	55.40%
CAS_TAS_DAS_IPS	43.24%	7.9%	1.1%	41.20%	45.30%
D\CAS	65.48%	7.0%	1.1%	63.40%	67.60%

a. Experiment ID = 250 (Ch), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Success Ratio [%]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	15.02%	3.7%	1.1%	12.80%	17.20%
TAS	0%	NA	NA	NA	NA
TAS_DAS	16.30%	4.0%	1.1%	14.10%	18.50%
CAS_TAS_AAO	14.18%	3.9%	1.1%	12.00%	16.40%
CAS_TAS_IPS	15.53%	4.5%	1.1%	13.30%	17.70%
CAS_TAS_DAS_AAO	42.97%	8.2%	1.1%	40.80%	45.20%
CAS_TAS_DAS_IPS	39.73%	8.2%	1.1%	37.50%	41.90%
D\CAS	51.58%	8.0%	1.1%	49.40%	53.80%

a. Experiment ID = 250 (Ch), Authentication Protocol = 3WP

Experiment ID = 500 Nodes (The "*Churn*" Scenario)
Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Success Ratio [%]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	34.92%	9.1%	1.4%	32.10%	37.70%
TAS	0%	NA	NA	NA	NA
TAS_DAS	45.61%	8.2%	1.4%	42.80%	48.40%
CAS_TAS_AAO	32.05%	9.6%	1.4%	29.20%	34.90%
CAS_TAS_IPS	36.90%	10%	1.4%	34.10%	39.70%
CAS_TAS_DAS_AAO	76.59%	4.0%	1.4%	73.80%	79.40%
CAS_TAS_DAS_IPS	70.45%	5.7%	1.4%	67.60%	73.30%
D\CAS	82.44%	3.3%	1.4%	79.60%	85.30%

a. Experiment ID = 500 (Ch), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Success Ratio [%]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	29.47%	9.7%	1.3%	27.00%	32.00%
TAS	0%	NA	NA	NA	NA
TAS_DAS	45.47%	6.1%	1.3%	43.00%	47.90%
CAS_TAS_AAO	32.32%	8.7%	1.3%	29.80%	34.80%
CAS_TAS_IPS	31.22%	8.1%	1.3%	28.70%	33.70%
CAS_TAS_DAS_AAO	75.25%	4.6%	1.3%	72.80%	77.70%
CAS_TAS_DAS_IPS	69.94%	5.2%	1.3%	67.50%	72.40%
D\CAS	83.15%	3.5%	1.3%	80.70%	85.60%

a. Experiment ID = 500 (Ch), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Success Ratio [%]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	28.23%	8.7%	1.4%	25.40%	31.00%
TAS_DAS	32.17%	8.2%	1.4%	29.40%	35.00%
CAS_TAS_AAO	24.53%	10.3%	1.4%	21.70%	27.30%
CAS_TAS_IPS	30.11%	9.5%	1.4%	27.30%	32.90%
CAS_TAS_DAS_AAO	70.87%	5.5%	1.4%	68.10%	73.70%
CAS_TAS_DAS_IPS	65.93%	5.9%	1.4%	63.10%	68.70%
D\CAS	77.21%	4.3%	1.4%	74.40%	80.00%

a. Experiment ID = 500 (Ch), Authentication Protocol = 3WP

Experiment ID = 100 Nodes (The "*No-Churn*" Scenario)
Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Success Ratio [%]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	46.70%	6.9%	1%	44.70%	48.70%
TAS	27.50%	3.7%	1%	25.50%	29.50%
TAS_DAS	52.40%	6.2%	1%	50.40%	54.40%
CAS_TAS_AAO	48.67%	5.3%	1%	46.70%	50.60%
CAS_TAS_IPS	53.13%	5.0%	1%	51.20%	55.10%
CAS_TAS_DAS_AAO	66.33%	5.9%	1%	64.40%	68.30%
CAS_TAS_DAS_IPS	66.07%	4.8%	1%	64.10%	68.00%
D\CAS	73.97%	5.3%	1%	72.00%	75.90%

a. Experiment ID = 100 (NoCh), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Success Ratio [%]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	45.87%	7.0%	1%	43.90%	47.80%
TAS	25.77%	3.7%	1%	23.80%	27.70%
TAS_DAS	51.77%	6.0%	1%	49.80%	53.70%
CAS_TAS_AAO	48.27%	5.0%	1%	46.30%	50.20%
CAS_TAS_IPS	52.33%	5.3%	1%	50.40%	54.30%
CAS_TAS_DAS_AAO	66.07%	5.7%	1%	64.10%	68.00%
CAS_TAS_DAS_IPS	65.20%	5.2%	1%	63.20%	67.20%
D\CAS	73.53%	5.2%	1%	71.60%	75.50%

a. Experiment ID = 100 (NoCh), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Success Ratio [%]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	44.00%	6.8%	1%	42.10%	45.90%
TAS	23.40%	3.3%	1%	21.50%	25.30%
TAS_DAS	49.43%	5.8%	1%	47.50%	51.40%
CAS_TAS_AAO	46.63%	5.0%	1%	44.70%	48.60%
CAS_TAS_IPS	50.53%	5.1%	1%	48.60%	52.50%
CAS_TAS_DAS_AAO	63.77%	5.5%	1%	61.80%	65.70%
CAS_TAS_DAS_IPS	64.07%	4.6%	1%	62.10%	66.00%
D\CAS	69.57%	5.8%	1%	67.60%	71.50%

a. Experiment ID = 100 (NoCh), Authentication Protocol = 3WP

Experiment ID = 250 Nodes (The "*No-Churn*" Scenario)
Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Success Ratio [%]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	77.26%	2.6%	0.4%	76.50%	78.00%
TAS	69.90%	2.0%	0.4%	69.20%	70.60%
TAS_DAS	79.43%	1.9%	0.4%	78.70%	80.10%
CAS_TAS_AAO	78.56%	1.8%	0.4%	77.90%	79.30%
CAS_TAS_IPS	80.99%	2.0%	0.4%	80.30%	81.70%
CAS_TAS_DAS_AAO	85.34%	2.0%	0.4%	84.60%	86.10%
CAS_TAS_DAS_IPS	85.77%	1.6%	0.4%	85.10%	86.50%
D\CAS	87.85%	1.7%	0.4%	87.10%	88.60%

a. Experiment ID = 250 (NoCh), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Success Ratio [%]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	76.32%	2.8%	0.4%	75.60%	77.10%
TAS	69.19%	2.1%	0.4%	68.40%	69.90%
TAS_DAS	79.47%	1.9%	0.4%	78.70%	80.20%
CAS_TAS_AAO	78.35%	2.0%	0.4%	77.60%	79.10%
CAS_TAS_IPS	80.53%	2.0%	0.4%	79.80%	81.30%
CAS_TAS_DAS_AAO	85.25%	2.2%	0.4%	84.50%	86.00%
CAS_TAS_DAS_IPS	85.68%	1.8%	0.4%	84.90%	86.40%
D\CAS	87.68%	1.7%	0.4%	86.90%	88.40%

a. Experiment ID = 250 (NoCh), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Success Ratio [%]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	74.72%	2.6%	0.4%	74.00%	75.40%
TAS	67.62%	2.1%	0.4%	66.90%	68.30%
TAS_DAS	78.66%	2.2%	0.4%	77.90%	79.40%
CAS_TAS_AAO	77.42%	1.9%	0.4%	76.70%	78.10%
CAS_TAS_IPS	79.81%	1.6%	0.4%	79.10%	80.50%
CAS_TAS_DAS_AAO	84.23%	2.0%	0.4%	83.50%	85.00%
CAS_TAS_DAS_IPS	85.06%	1.7%	0.4%	84.30%	85.80%
D\CAS	86.33%	1.9%	0.4%	85.60%	87.10%

a. Experiment ID = 250 (NoCh), Authentication Protocol = 3WP

Experiment ID = 500 Nodes (The "*No-Churn*" Scenario)
Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Success Ratio [%]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	88.28%	1.3%	0.2%	87.90%	88.60%
TAS	84.79%	1.1%	0.2%	84.40%	85.20%
TAS_DAS	89.80%	1.1%	0.2%	89.40%	90.20%
CAS_TAS_AAO	89.13%	1.0%	0.2%	88.80%	89.50%
CAS_TAS_IPS	90.47%	0.9%	0.2%	90.10%	90.80%
CAS_TAS_DAS_AAO	92.86%	1.0%	0.2%	92.50%	93.20%
CAS_TAS_DAS_IPS	93.03%	0.8%	0.2%	92.70%	93.40%
D\CAS	94.26%	0.9%	0.2%	93.90%	94.60%

a. Experiment ID = 500 (NoCh), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Success Ratio [%]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	87.70%	1.2%	0.2%	87.30%	88.10%
TAS	84.41%	1.1%	0.2%	84.00%	84.80%
TAS_DAS	89.74%	1.1%	0.2%	89.40%	90.10%
CAS_TAS_AAO	88.91%	1.1%	0.2%	88.50%	89.30%
CAS_TAS_IPS	90.26%	0.9%	0.2%	89.90%	90.60%
CAS_TAS_DAS_AAO	92.87%	1.0%	0.2%	92.50%	93.20%
CAS_TAS_DAS_IPS	92.83%	0.9%	0.2%	92.50%	93.20%
D\CAS	94.07%	0.8%	0.2%	93.70%	94.40%

a. Experiment ID = 500 (NoCh), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Success Ratio [%]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	85.95%	1.2%	0.2%	85.60%	86.30%
TAS	83.80%	1.1%	0.2%	83.40%	84.20%
TAS_DAS	89.21%	1.0%	0.2%	88.80%	89.60%
CAS_TAS_AAO	88.57%	1.1%	0.2%	88.20%	88.90%
CAS_TAS_IPS	89.84%	0.9%	0.2%	89.50%	90.20%
CAS_TAS_DAS_AAO	92.40%	1.0%	0.2%	92.00%	92.80%
CAS_TAS_DAS_IPS	92.70%	0.9%	0.2%	92.30%	93.10%
D\CAS	93.42%	0.9%	0.2%	93.10%	93.80%

a. Experiment ID = 500 (NoCh), Authentication Protocol = 3W

Round Trip Time

Experiment ID = 100 Nodes (The "*Churn*" Scenario)

Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Round Trip Time [Sec]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	25.64	6.55	3.88	17.99	33.29
TAS	NA	NA	NA	NA	NA
TAS_DAS	231.07	6.55	3.88	223.42	238.72
CAS_TAS_AAO	16.78	9.10	3.88	9.13	24.43
CAS_TAS_IPS	25.60	8.10	3.88	17.95	33.25
CAS_TAS_DAS_AAO	135.14	17.61	3.88	127.49	142.79
CAS_TAS_DAS_IPS	195.94	50.99	3.88	188.29	203.60
D\CAS	18.06	4.27	3.88	10.40	25.71

a. Experiment ID = 100 (Ch), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Round Trip Time [Sec]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	33.62	13.95	4.24	25.26	41.98
TAS	NA	NA	NA	NA	NA
TAS_DAS	230.60	5.05	4.24	222.24	238.96
CAS_TAS_AAO	16.14	7.38	4.24	7.78	24.50
CAS_TAS_IPS	33.74	10.68	4.24	25.38	42.10
CAS_TAS_DAS_AAO	138.31	19.07	4.24	129.95	146.67
CAS_TAS_DAS_IPS	214.26	54.85	4.24	205.90	222.62
D\CAS	17.80	3.59	4.24	9.44	26.16

a. Experiment ID = 100 (Ch), Authentication Protocol = 2WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Round Trip Time [Sec]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	31.23	9.84	4.22	22.91	39.56
TAS	NA	NA	NA	NA	NA
TAS_DAS	227.96	8.39	4.22	219.63	236.28
CAS_TAS_AAO	34.67	10.30	4.22	26.34	42.99
CAS_TAS_IPS	36.91	14.15	4.22	28.59	45.24
CAS_TAS_DAS_AAO	127.43	22.69	4.22	119.11	135.75
CAS_TAS_DAS_IPS	203.41	52.16	4.22	195.09	211.73
D\CAS	30.92	5.79	4.22	22.60	39.25

a. Experiment ID = 100 (Ch), Authentication Protocol = 3WP

Experiment ID = 250 Nodes (The *Churn"* Scenario)
Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Round Trip Time [Sec]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	27.47	3.99	2.30	22.94	32.01
TAS	NA	NA	NA	NA	NA
TAS_DAS	229.26	3.50	2.30	224.73	233.80
CAS_TAS_AAO	20.60	6.33	2.30	16.07	25.14
CAS_TAS_IPS	28.56	4.72	2.30	24.02	33.09
CAS_TAS_DAS_AAO	105.13	15.47	2.30	100.59	109.66
CAS_TAS_DAS_IPS	175.85	27.82	2.30	171.31	180.38
D\CAS	17.11	2.63	2.30	12.58	21.64

a. Experiment ID = 250 (Ch), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Round Trip Time [Sec]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	30.22	7.04	2.18	25.92	34.52
TAS	NA	NA	NA	NA	NA
TAS_DAS	230.76	3.66	2.18	226.45	235.06
CAS_TAS_AAO	21.34	5.07	2.18	17.03	25.64
CAS_TAS_IPS	32.07	7.54	2.18	27.77	36.37
CAS_TAS_DAS_AAO	103.85	13.82	2.18	99.55	108.16
CAS_TAS_DAS_IPS	185.69	25.64	2.18	181.39	190.00
D\CAS	17.87	2.50	2.18	13.57	22.17

a. Experiment ID = 250 (Ch), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Round Trip Time [Sec]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	35.12	7.28	2.69	29.81	40.43
TAS	NA	NA	NA	NA	NA
TAS_DAS	228.07	5.41	2.69	222.76	233.38
CAS_TAS_AAO	33.60	7.46	2.69	28.28	38.91
CAS_TAS_IPS	33.48	5.82	2.69	28.17	38.79
CAS_TAS_DAS_AAO	117.44	15.61	2.69	112.13	122.75
CAS_TAS_DAS_IPS	187.29	33.07	2.69	181.98	192.61
D\CAS	28.03	3.83	2.69	22.72	33.34

a. Experiment ID = 250 (Ch), Authentication Protocol = 3WP

Experiment ID = 500 Nodes (The "*Churn*" Scenario)
Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Round Trip Time [Sec]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	23.90	2.50	2.42	19.13	28.66
TAS	NA	NA	NA	NA	NA
TAS_DAS	140.90	30.67	2.42	136.14	145.67
CAS_TAS_AAO	22.71	3.87	2.42	17.95	27.48
CAS_TAS_IPS	36.80	6.25	2.42	32.03	41.57
CAS_TAS_DAS_AAO	48.32	7.36	2.42	43.55	53.09
CAS_TAS_DAS_IPS	82.46	13.07	2.42	77.70	87.23
D\CAS	11.88	1.36	2.42	7.11	16.64

a. Experiment ID = 500 (Ch), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Round Trip Time [Sec]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	28.27	3.51	1.88	24.56	31.98
TAS	NA	NA	NA	NA	NA
TAS_DAS	142.54	22.12	1.88	138.83	146.25
CAS_TAS_AAO	23.86	3.65	1.88	20.15	27.58
CAS_TAS_IPS	39.28	5.38	1.88	35.57	42.99
CAS_TAS_DAS_AAO	49.96	5.40	1.88	46.25	53.67
CAS_TAS_DAS_IPS	86.90	13.01	1.88	83.19	90.61
D\CAS	12.72	1.31	1.88	9.01	16.43

a. Experiment ID = 500 (Ch), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Round Trip Time [Sec]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	29.877	3.94750	2.481	24.986	34.769
TAS	NA	NA	NA	NA	NA
TAS_DAS	183.508	31.40983	2.481	178.616	188.399
CAS_TAS_AAO	29.241	3.77743	2.481	24.350	34.132
CAS_TAS_IPS	42.682	6.56217	2.481	37.791	47.573
CAS_TAS_DAS_AAO	56.293	8.10870	2.481	51.402	61.185
CAS_TAS_DAS_IPS	96.638	12.80901	2.481	91.747	101.529
D\CAS	17.720	1.75118	2.481	12.829	22.611

a. Experiment ID = 500 (Ch), Authentication Protocol = 3WP

Experiment ID = 100 Nodes (The "*No-Churn*" Scenario)
Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Round Trip Time [Sec]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	23.58	4.39	2.45	18.75	28.40
TAS	38.51	9.13	2.45	33.68	43.33
TAS_DAS	131.78	16.78	2.45	126.95	136.60
CAS_TAS_AAO	22.67	5.92	2.45	17.85	27.50
CAS_TAS_IPS	50.10	13.97	2.45	45.27	54.92
CAS_TAS_DAS_AAO	78.48	13.61	2.45	73.65	83.30
CAS_TAS_DAS_IPS	126.21	25.13	2.45	121.38	131.03
D\CAS	15.58	2.82	2.45	10.76	20.41

a. Experiment ID = 100 (NoCh), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Round Trip Time [Sec]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	25.99	4.45	2.56	20.94	31.03
TAS	40.82	9.25	2.56	35.77	45.86
TAS_DAS	137.71	19.29	2.56	132.66	142.75
CAS_TAS_AAO	24.28	5.60	2.56	19.24	29.33
CAS_TAS_IPS	54.41	11.82	2.56	49.36	59.45
CAS_TAS_DAS_AAO	80.96	13.41	2.56	75.91	86.00
CAS_TAS_DAS_IPS	131.16	27.20	2.56	126.11	136.20
D\CAS	16.46	2.36	2.56	11.42	21.51

a. Experiment ID = 100 (NoCh), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Round Trip Time [Sec]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	29.10	4.48	2.46	24.25	33.94
TAS	45.79	9.14	2.46	40.95	50.63
TAS_DAS	145.73	13.95	2.46	140.89	150.57
CAS_TAS_AAO	28.06	5.47	2.46	23.22	32.90
CAS_TAS_IPS	57.24	13.78	2.46	52.39	62.08
CAS_TAS_DAS_AAO	82.98	11.68	2.46	78.14	87.83
CAS_TAS_DAS_IPS	137.36	28.06	2.46	132.51	142.20
D\CAS	20.97	3.05	2.46	16.12	25.81

a. Experiment ID = 100 (NoCh), Authentication Protocol = 3WP

Experiment ID = 250 Nodes (The "*No-Churn*" Scenario)
Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Round Trip Time [Sec]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	14.29	1.79	0.73	12.86	15.72
TAS	16.03	1.56	0.73	14.60	17.46
TAS_DAS	42.37	5.25	0.73	40.93	43.80
CAS_TAS_AAO	12.38	1.68	0.73	10.95	13.82
CAS_TAS_IPS	22.69	3.88	0.73	21.26	24.12
CAS_TAS_DAS_AAO	30.02	3.78	0.73	28.59	31.45
CAS_TAS_DAS_IPS	46.55	7.75	0.73	45.12	47.99
D\CAS	9.79	1.22	0.73	8.35	11.22

a. Experiment ID = 250 (NoCh), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Round Trip Time [Sec]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	16.77	1.71	0.76	15.27	18.26
TAS	17.99	1.57	0.76	16.49	19.49
TAS_DAS	46.27	4.97	0.76	44.77	47.76
CAS_TAS_AAO	14.07	1.62	0.76	12.57	15.56
CAS_TAS_IPS	26.26	3.46	0.76	24.76	27.75
CAS_TAS_DAS_AAO	31.82	4.24	0.76	30.32	33.32
CAS_TAS_DAS_IPS	51.59	8.59	0.76	50.09	53.08
D\CAS	10.88	1.32	0.76	9.39	12.38

a. Experiment ID = 250 (NoCh), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Round Trip Time [Sec]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	21.40	1.82	0.76	19.90	22.90
TAS	22.24	1.76	0.76	20.75	23.74
TAS_DAS	52.61	4.72	0.76	51.12	54.11
CAS_TAS_AAO	17.08	1.57	0.76	15.59	18.58
CAS_TAS_IPS	34.08	4.14	0.76	32.58	35.57
CAS_TAS_DAS_AAO	34.73	4.61	0.76	33.24	36.23
CAS_TAS_DAS_IPS	59.16	8.23	0.76	57.66	60.66
D\CAS	13.99	1.03	0.76	12.50	15.49

a. Experiment ID = 250 (NoCh), Authentication Protocol = 3WP

Experiment ID = 500 Nodes (The "*No-Churn*" Scenario)
Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Round Trip Time [Sec]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	11.01	0.81	0.38	10.27	11.75
TAS	11.89	0.61	0.38	11.14	12.63
TAS_DAS	24.25	2.62	0.38	23.51	24.99
CAS_TAS_AAO	9.07	0.65	0.38	8.33	9.81
CAS_TAS_IPS	15.73	1.66	0.38	14.99	16.48
CAS_TAS_DAS_AAO	17.89	2.13	0.38	17.15	18.63
CAS_TAS_DAS_IPS	27.25	4.25	0.38	26.51	28.00
D\CAS	7.97	0.66	0.38	7.22	8.71

a. Experiment ID = 500 (NoCh), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Round Trip Time [Sec]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	14.23	1.20	0.43	13.38	15.07
TAS	13.75	0.63	0.43	12.90	14.60
TAS_DAS	26.77	2.80	0.43	25.92	27.61
CAS_TAS_AAO	10.48	0.62	0.43	9.64	11.33
CAS_TAS_IPS	19.57	1.74	0.43	18.72	20.41
CAS_TAS_DAS_AAO	19.82	2.39	0.43	18.97	20.67
CAS_TAS_DAS_IPS	31.88	5.02	0.43	31.04	32.73
D\CAS	8.94	0.65	0.43	8.09	9.78

a. Experiment ID = 500 (NoCh), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Round Trip Time [Sec]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	19.93	1.25	0.47	19.00	20.86
TAS	17.81	0.86	0.47	16.88	18.74
TAS_DAS	31.32	2.48	0.47	30.39	32.25
CAS_TAS_AAO	13.60	0.72	0.47	12.67	14.53
CAS_TAS_IPS	27.95	2.47	0.47	27.02	28.88
CAS_TAS_DAS_AAO	22.56	2.34	0.47	21.63	23.49
CAS_TAS_DAS_IPS	40.11	5.70	0.47	39.18	41.04
D\CAS	11.53	0.69	0.47	10.60	12.46

a. Experiment ID = 500 (NoCh), Authentication Protocol = 3WP

Communication Overhead

Experiment ID = 100 Nodes (The "*Churn*" Scenario)
Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Communication Overhead [Bytes]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	2749	195	241	2273	3225
TAS	NA	NA	NA	NA	NA
TAS_DAS	21225	176	241	20750	21701
CAS_TAS_AAO	10548	1568	241	10072	11024
CAS_TAS_IPS	2705	210	241	2230	3181
CAS_TAS_DAS_AAO	18611	1333	241	18135	19087
CAS_TAS_DAS_IPS	11785	2728	241	11309	12261
D\CAS	10498	658	241	10022	10974

a. Experiment ID = 100 (Ch), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Communication Overhead [Bytes]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	3820	281	238	3352	4289
TAS	NA	NA	NA	NA	NA
TAS_DAS	21300	156	238	20831	21768
CAS_TAS_AAO	11495	1376	238	11027	11963
CAS_TAS_IPS	3809	218	238	3340	4277
CAS_TAS_DAS_AAO	19283	1253	238	18815	19751
CAS_TAS_DAS_IPS	13204	2814	238	12735	13672
D\CAS	10444	561	238	9975	10912

a. Experiment ID = 100 (Ch), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Communication Overhead [Bytes]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	4406	263	251	3911	4900
TAS	NA	NA	NA	NA	NA
TAS_DAS	22741	549	251	22246	23236
CAS_TAS_AAO	14413	1517	251	13918	14908
CAS_TAS_IPS	4498	343	251	4003	4993
CAS_TAS_DAS_AAO	20013	1459	251	19518	20508
CAS_TAS_DAS_IPS	13443	2757	251	12949	13938
D\CAS	12650	839	251	12156	13145

a. Experiment ID = 100 (Ch), Authentication Protocol = 3WP

Experiment ID = 250 Nodes (The "*Churn*" Scenario)
Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Communication Overhead [Bytes]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	2738	114	146	2450	3026
TAS	NA	NA	NA	NA	NA
TAS_DAS	21292	92	146	21004	21580
CAS_TAS_AAO	11385	1098	146	11097	11673
CAS_TAS_IPS	2765	152	146	2477	3053
CAS_TAS_DAS_AAO	17210	991	146	16922	17498
CAS_TAS_DAS_IPS	10687	1447	146	10399	10975
D\CAS	10695	395	146	10407	10983

a. Experiment ID = 250 (Ch), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Communication Overhead [Bytes]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	3796	162	116	3566	4025
TAS	NA	NA	NA	NA	NA
TAS_DAS	21418	106	116	21188	21647
CAS_TAS_AAO	12574	861	116	12344	12803
CAS_TAS_IPS	3850	187	116	3620	4079
CAS_TAS_DAS_AAO	18656	608	116	18427	18886
CAS_TAS_DAS_IPS	12029	1231	116	11800	12259
D\CAS	11147	383	116	10917	11376

a. Experiment ID = 250 (Ch), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Communication Overhead [Bytes]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	4460	169	151	4163	4757
TAS	NA	NA	NA	NA	NA
TAS_DAS	22511	284	151	22214	22808
CAS_TAS_AAO	14672	1141	151	14375	14969
CAS_TAS_IPS	4399	169	151	4102	4696
CAS_TAS_DAS_AAO	20242	668	151	19945	20539
CAS_TAS_DAS_IPS	12703	1625	151	12406	13000
D\CAS	12658	486	151	12361	12955

a. Experiment ID = 250 (Ch), Authentication Protocol = 3WP

Experiment ID = 500 Nodes (The "*Churn*" Scenario)

Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Communication Overhead [Bytes]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	2622	67	122	2381	2862
TAS	NA	NA	NA	NA	NA
TAS_DAS	17660	1284	122	17419	17900
CAS_TAS_AAO	12270	608	122	12030	12510
CAS_TAS_IPS	3503	408	122	3263	3743
CAS_TAS_DAS_AAO	14135	621	122	13895	14375
CAS_TAS_DAS_IPS	6047	704	122	5807	6287
D\CAS	11328	218	122	11088	11569

a. Experiment ID = 500 (Ch), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Communication Overhead [Bytes]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	3787	73	100	3589	3985
TAS	NA	NA	NA	NA	NA
TAS_DAS	20039	462	100	19841	20237
CAS_TAS_AAO	14675	942	100	14477	14873
CAS_TAS_IPS	4742	491	100	4544	4940
CAS_TAS_DAS_AAO	18373	464	100	18175	18571
CAS_TAS_DAS_IPS	7370	698	100	7172	7568
D\CAS	12929	264	100	12730	13127

a. Experiment ID = 500 (Ch), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Communication Overhead [Bytes]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	4348	106	125	4101	4595
TAS	NA	NA	NA	NA	NA
TAS_DAS	22272	367	125	22025	22519
CAS_TAS_AAO	15580	1387	125	15333	15827
CAS_TAS_IPS	5466	609	125	5219	5713
CAS_TAS_DAS_AAO	20920	479	125	20673	21167
CAS_TAS_DAS_IPS	8576	714	125	8329	8823
D\CAS	15000	335	125	14753	15247

a. Experiment ID = 500 (Ch), Authentication Protocol = 3WP

Experiment ID = 100 Nodes (The "*No-Churn*" Scenario)

Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Communication Overhead [Bytes]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	2605	108	163	2284	2926
TAS	13971	1058	163	13650	14292
TAS_DAS	17387	785	163	17066	17708
CAS_TAS_AAO	12736	973	163	12415	13057
CAS_TAS_IPS	4392	903	163	4071	4713
CAS_TAS_DAS_AAO	15904	897	163	15582	16225
CAS_TAS_DAS_IPS	8327	1364	163	8006	8649
D\CAS	10766	447	163	10445	11087

a. Experiment ID = 100 (NoCh), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Communication Overhead [Bytes]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	3782	127	168	3451	4113
TAS	18437	1087	168	18106	18768
TAS_DAS	20028	667	168	19696	20359
CAS_TAS_AAO	16118	1013	168	15786	16449
CAS_TAS_IPS	6087	924	168	5755	6418
CAS_TAS_DAS_AAO	18455	906	168	18124	18786
CAS_TAS_DAS_IPS	9815	1503	168	9483	10146
D\CAS	11286	431	168	10955	11617

a. Experiment ID = 100 (NoCh), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Communication Overhead [Bytes]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	4393	130	186	4026	4759
TAS	21021	1113	186	20654	21387
TAS_DAS	22193	552	186	21827	22559
CAS_TAS_AAO	18240	1135	186	17873	18606
CAS_TAS_IPS	6918	1186	186	6552	7284
CAS_TAS_DAS_AAO	20346	968	186	19979	20712
CAS_TAS_DAS_IPS	10931	1661	186	10565	11298
D\CAS	12808	590	186	12442	13175

a. Experiment ID = 100 (NoCh), Authentication Protocol = 3WP

Experiment ID = 250 Nodes (The "*No-Churn*" Scenario)

Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Communication Overhead [Bytes]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	2359	47	66	2230	2489
TAS	12492	191	66	12362	12622
TAS_DAS	13519	254	66	13389	13649
CAS_TAS_AAO	12374	567	66	12244	12503
CAS_TAS_IPS	2917	243	66	2787	3047
CAS_TAS_DAS_AAO	13343	560	66	13213	13473
CAS_TAS_DAS_IPS	4168	422	66	4038	4297
D\CAS	11470	255	66	11341	11600

a. Experiment ID = 250 (NoCh), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Communication Overhead [Bytes]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	3564	55	68	3429	3699
TAS	17901	264	68	17766	18036
TAS_DAS	18404	194	68	18270	18539
CAS_TAS_AAO	18069	559	68	17934	18204
CAS_TAS_IPS	4336	259	68	4201	4471
CAS_TAS_DAS_AAO	18611	573	68	18477	18746
CAS_TAS_DAS_IPS	5622	484	68	5488	5757
D\CAS	13480	268	68	13345	13614

a. Experiment ID = 250 (NoCh), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Communication Overhead [Bytes]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	4275	72	70	4137	4414
TAS	20979	277	70	20841	21118
TAS_DAS	21298	254	70	21159	21436
CAS_TAS_AAO	21086	530	70	20947	21224
CAS_TAS_IPS	5497	370	70	5359	5636
CAS_TAS_DAS_AAO	21495	521	70	21357	21634
CAS_TAS_DAS_IPS	6696	496	70	6558	6835
D\CAS	15842	324	70	15703	15980

a. Experiment ID = 250 (NoCh), Authentication Protocol = 3WP

Experiment ID = 500 Nodes (The "*No-Churn*" Scenario)

Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Communication Overhead [Bytes]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	2278	22	62	2156	2400
TAS	12262	97	62	12140	12384
TAS_DAS	12759	121	62	12636	12881
CAS_TAS_AAO	12161	607	62	12039	12283
CAS_TAS_IPS	2579	96	62	2457	2701
CAS_TAS_DAS_AAO	12647	634	62	12525	12769
CAS_TAS_DAS_IPS	3184	225	62	3062	3307
D\CAS	11905	259	62	11783	12027

a. Experiment ID = 500 (NoCh), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable :Communication Overhead [Bytes]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	3508	42	59	3391	3625
TAS	17877	145	59	17760	17994
TAS_DAS	18098	96	59	17981	18215
CAS_TAS_AAO	18393	577	59	18276	18510
CAS_TAS_IPS	3945	141	59	3828	4062
CAS_TAS_DAS_AAO	18687	593	59	18570	18804
CAS_TAS_DAS_IPS	4563	289	59	4446	4680
D\CAS	14307	158	59	14190	14424

a. Experiment ID = 500 (NoCh), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Communication Overhead [Bytes]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	4285	53	56	4175	4395
TAS	20824	217	56	20714	20934
TAS_DAS	21057	156	56	20947	21167
CAS_TAS_AAO	21638	453	56	21528	21748
CAS_TAS_IPS	5090	209	56	4979	5200
CAS_TAS_DAS_AAO	21810	505	56	21700	21920
CAS_TAS_DAS_IPS	5721	365	56	5611	5831
D\CAS	16852	194	56	16742	16962

a. Experiment ID = 500 (NoCh), Authentication Protocol = 3WP

Failure Frequency
Experiment ID = 100 Nodes (The "*Churn*" Scenario)

Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Failure Frequency [Node/min]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	1.03	0.11	0.02	0.99	1.06
TAS	1.17	0.09	0.02	1.14	1.21
TAS_DAS	0.97	0.10	0.02	0.93	1.01
CAS_TAS_AAO	1.00	0.09	0.02	0.96	1.04
CAS_TAS_IPS	1.01	0.11	0.02	0.97	1.05
CAS_TAS_DAS_AAO	0.78	0.11	0.02	0.74	0.82
CAS_TAS_DAS_IPS	0.87	0.10	0.02	0.83	0.91
D\CAS	0.68	0.12	0.02	0.65	0.72

a. Experiment ID = 100 (Ch), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Failure Frequency [Node/min]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	1.06	0.12	0.02	1.02	1.10
TAS	1.17	0.09	0.02	1.14	1.21
TAS_DAS	0.98	0.09	0.02	0.94	1.02
CAS_TAS_AAO	1.00	0.09	0.02	0.96	1.03
CAS_TAS_IPS	1.02	0.09	0.02	0.98	1.06
CAS_TAS_DAS_AAO	0.78	0.11	0.02	0.75	0.82
CAS_TAS_DAS_IPS	0.90	0.09	0.02	0.87	0.94
D\CAS	0.69	0.13	0.02	0.65	0.72

a. Experiment ID = 100 (Ch), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Failure Frequency [Node/min]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	1.05	0.11	0.02	1.01	1.08
TAS	1.17	0.09	0.02	1.14	1.21
TAS_DAS	1.02	0.10	0.02	0.98	1.06
CAS_TAS_AAO	1.03	0.09	0.02	0.99	1.07
CAS_TAS_IPS	1.03	0.09	0.02	0.99	1.06
CAS_TAS_DAS_AAO	0.87	0.08	0.02	0.83	0.90
CAS_TAS_DAS_IPS	0.90	0.09	0.02	0.87	0.94
D\CAS	0.82	0.14	0.02	0.79	0.86

a. Experiment ID = 100 (Ch), Authentication Protocol = 3WP

Experiment ID = 250 Nodes (The "*Churn*" Scenario)
Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Failure Frequency [Node/min]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	1.03	0.05	0.01	1.00	1.06
TAS	1.22	0.06	0.01	1.20	1.24
TAS_DAS	0.99	0.07	0.01	0.96	1.01
CAS_TAS_AAO	1.02	0.07	0.01	0.99	1.04
CAS_TAS_IPS	1.01	0.07	0.01	0.99	1.04
CAS_TAS_DAS_AAO	0.72	0.10	0.01	0.69	0.75
CAS_TAS_DAS_IPS	0.78	0.08	0.01	0.75	0.81
D\CAS	0.62	0.08	0.01	0.59	0.65

a. Experiment ID = 250 (Ch), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Failure Frequency [Node/min]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	1.06	0.07	0.01	1.04	1.09
TAS	1.22	0.06	0.01	1.20	1.24
TAS_DAS	1.00	0.07	0.01	0.97	1.02
CAS_TAS_AAO	1.03	0.07	0.01	1.00	1.05
CAS_TAS_IPS	1.03	0.07	0.01	1.00	1.06
CAS_TAS_DAS_AAO	0.70	0.07	0.01	0.67	0.72
CAS_TAS_DAS_IPS	0.80	0.07	0.01	0.77	0.83
D\CAS	0.62	0.08	0.01	0.59	0.65

a. Experiment ID = 250 (Ch), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Failure Frequency [Node/min]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	1.07	0.06	0.01	1.05	1.10
TAS	1.22	0.06	0.01	1.20	1.24
TAS_DAS	1.05	0.06	0.01	1.02	1.08
CAS_TAS_AAO	1.04	0.07	0.01	1.02	1.07
CAS_TAS_IPS	1.04	0.08	0.01	1.02	1.07
CAS_TAS_DAS_AAO	0.80	0.08	0.01	0.78	0.83
CAS_TAS_DAS_IPS	0.83	0.06	0.01	0.80	0.85
D\CAS	0.75	0.09	0.01	0.73	0.78

a. Experiment ID = 250 (Ch), Authentication Protocol = 3WP

Experiment ID = 500 Nodes (The "*Churn*" Scenario)
Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Failure Frequency [Node/min]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	0.91	0.08	0.01	0.89	0.94
TAS	1.23	0.05	0.01	1.21	1.25
TAS_DAS	0.87	0.05	0.01	0.84	0.89
CAS_TAS_AAO	0.92	0.08	0.01	0.90	0.95
CAS_TAS_IPS	0.93	0.07	0.01	0.90	0.95
CAS_TAS_DAS_AAO	0.62	0.07	0.01	0.59	0.64
CAS_TAS_DAS_IPS	0.73	0.07	0.01	0.70	0.75
D\CAS	0.54	0.07	0.01	0.52	0.57

a. Experiment ID = 500 (Ch), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Failure Frequency [Node/min]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	0.96	0.10	0.01	0.94	0.99
TAS	1.23	0.05	0.01	1.21	1.25
TAS_DAS	0.86	0.05	0.01	0.83	0.89
CAS_TAS_AAO	0.93	0.07	0.01	0.90	0.95
CAS_TAS_IPS	0.94	0.07	0.01	0.91	0.97
CAS_TAS_DAS_AAO	0.63	0.08	0.01	0.60	0.65
CAS_TAS_DAS_IPS	0.74	0.05	0.01	0.71	0.76
D\CAS	0.53	0.07	0.01	0.51	0.56

a. Experiment ID = 500 (Ch), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Failure Frequency [Node/min]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	0.97	0.09	0.01	0.94	1.00
TAS	1.23	0.05	0.01	1.21	1.25
TAS_DAS	0.94	0.05	0.01	0.91	0.96
CAS_TAS_AAO	0.97	0.09	0.01	0.95	1.00
CAS_TAS_IPS	0.95	0.07	0.01	0.92	0.97
CAS_TAS_DAS_AAO	0.68	0.06	0.01	0.65	0.70
CAS_TAS_DAS_IPS	0.76	0.06	0.01	0.73	0.79
D\CAS	0.64	0.08	0.01	0.61	0.67

a. Experiment ID = 500 (Ch), Authentication Protocol = 3WP

Experiment ID = 100 Nodes (The "*No-Churn*" Scenario)

Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Failure Frequency [Node/min]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	0.82	0.16	0.02	0.77	0.86
TAS	1.01	0.09	0.02	0.97	1.06
TAS_DAS	0.75	0.12	0.02	0.71	0.80
CAS_TAS_AAO	0.80	0.11	0.02	0.75	0.84
CAS_TAS_IPS	0.80	0.12	0.02	0.75	0.84
CAS_TAS_DAS_AAO	0.59	0.13	0.02	0.55	0.64
CAS_TAS_DAS_IPS	0.67	0.10	0.02	0.62	0.71
D\CAS	0.52	0.13	0.02	0.47	0.56

a. Experiment ID = 100 (NoCh), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Failure Frequency [Node/min]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	0.82	0.16	0.02	0.78	0.86
TAS	1.02	0.09	0.02	0.98	1.06
TAS_DAS	0.75	0.11	0.02	0.71	0.79
CAS_TAS_AAO	0.80	0.11	0.02	0.76	0.84
CAS_TAS_IPS	0.80	0.12	0.02	0.76	0.84
CAS_TAS_DAS_AAO	0.60	0.12	0.02	0.56	0.64
CAS_TAS_DAS_IPS	0.67	0.10	0.02	0.62	0.71
D\CAS	0.52	0.13	0.02	0.48	0.56

a. Experiment ID = 100 (NoCh), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Failure Frequency [Node/min]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	0.83	0.16	0.02	0.78	0.87
TAS	1.02	0.09	0.02	0.97	1.06
TAS_DAS	0.78	0.09	0.02	0.73	0.82
CAS_TAS_AAO	0.81	0.11	0.02	0.76	0.85
CAS_TAS_IPS	0.80	0.11	0.02	0.76	0.84
CAS_TAS_DAS_AAO	0.61	0.10	0.02	0.57	0.65
CAS_TAS_DAS_IPS	0.68	0.10	0.02	0.64	0.72
D\CAS	0.58	0.13	0.02	0.54	0.62

a. Experiment ID = 100 (NoCh), Authentication Protocol = 3WP

Experiment ID = 250 Nodes (The "*No-Churn*" Scenario)
Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Failure Frequency [Node/min]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	0.77	0.09	0.02	0.67	0.74
TAS	0.97	0.10	0.02	0.93	1.00
TAS_DAS	0.74	0.11	0.02	0.70	0.77
CAS_TAS_AAO	0.74	0.11	0.02	0.71	0.78
CAS_TAS_IPS	0.79	0.11	0.02	0.75	0.82
CAS_TAS_DAS_AAO	0.57	0.11	0.02	0.53	0.61
CAS_TAS_DAS_IPS	0.66	0.11	0.02	0.62	0.69
D\CAS	0.53	0.10	0.02	0.50	0.57

a. Experiment ID = 250 (NoCh), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Failure Frequency [Node/min]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	0.71	0.10	0.02	0.67	0.74
TAS	0.96	0.10	0.02	0.93	1.00
TAS_DAS	0.74	0.11	0.02	0.70	0.77
CAS_TAS_AAO	0.74	0.10	0.02	0.70	0.77
CAS_TAS_IPS	0.79	0.11	0.02	0.75	0.83
CAS_TAS_DAS_AAO	0.57	0.10	0.02	0.53	0.60
CAS_TAS_DAS_IPS	0.65	0.10	0.02	0.62	0.69
D\CAS	0.54	0.09	0.02	0.51	0.58

a. Experiment ID = 250 (NoCh), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Failure Frequency [Node/min]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	0.69	0.10	0.02	0.65	0.72
TAS	0.96	0.09	0.02	0.93	1.00
TAS_DAS	0.75	0.11	0.02	0.72	0.79
CAS_TAS_AAO	0.75	0.09	0.02	0.71	0.78
CAS_TAS_IPS	0.77	0.10	0.02	0.74	0.81
CAS_TAS_DAS_AAO	0.59	0.10	0.02	0.55	0.62
CAS_TAS_DAS_IPS	0.66	0.10	0.02	0.62	0.69
D\CAS	0.58	0.10	0.02	0.55	0.62

a. Experiment ID = 250 (NoCh), Authentication Protocol = 3WP

Experiment ID = 500 Nodes (The "*No-Churn*" Scenario)

Authentication Protocol = 1WP

Statistics Description[a]

Dependent Variable: Failure Frequency [Node/min]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	0.75	0.10	0.02	0.71	0.79
TAS	0.96	0.09	0.02	0.92	1.00
TAS_DAS	0.70	0.10	0.02	0.67	0.74
CAS_TAS_AAO	0.76	0.12	0.02	0.72	0.80
CAS_TAS_IPS	0.81	0.14	0.02	0.77	0.85
CAS_TAS_DAS_AAO	0.56	0.12	0.02	0.52	0.60
CAS_TAS_DAS_IPS	0.63	0.10	0.02	0.59	0.67
D\CAS	0.49	0.11	0.02	0.45	0.53

a. Experiment ID = 500 (NoCh), Authentication Protocol = 1WP

Authentication Protocol = 2WP

Statistics Description[a]

Dependent Variable: Failure Frequency [Node/min]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	0.73	0.08	0.02	0.69	0.77
TAS	0.95	0.10	0.02	0.91	0.99
TAS_DAS	0.70	0.08	0.02	0.66	0.74
CAS_TAS_AAO	0.75	0.12	0.02	0.72	0.79
CAS_TAS_IPS	0.82	0.13	0.02	0.78	0.85
CAS_TAS_DAS_AAO	0.55	0.10	0.02	0.51	0.59
CAS_TAS_DAS_IPS	0.65	0.11	0.02	0.61	0.69
D\CAS	0.50	0.10	0.02	0.46	0.54

a. Experiment ID = 500 (NoCh), Authentication Protocol = 2WP

Authentication Protocol = 3WP

Statistics Description[a]

Dependent Variable: Failure Frequency [Node/min]

Server Architecture	Mean	Std. Deviation	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
CAS	0.65	0.08	0.02	0.62	0.69
TAS	0.94	0.09	0.02	0.91	0.98
TAS_DAS	0.72	0.09	0.02	0.68	0.75
CAS_TAS_AAO	0.75	0.12	0.02	0.71	0.78
CAS_TAS_IPS	0.80	0.12	0.02	0.77	0.84
CAS_TAS_DAS_AAO	0.57	0.10	0.02	0.54	0.60
CAS_TAS_DAS_IPS	0.64	0.09	0.02	0.60	0.67
D\CAS	0.53	0.09	0.02	0.50	0.56

a. Experiment ID = 500 (NoCh), Authentication Protocol = 3WP

RTT- STDev

Security architecture	100-Node (No-Churn)	250-Node (No-Churn)	500-Node (No-Churn)	100-Node (Churn)	250-Node (Churn)	500-Node (Churn)	AVG (RTT-STDev)
1WP_CAS	30.83	21.93	16.42	26.55	32.48	31.52	26.62
2WP_CAS	32.35	23.09	19.07	40.03	37.73	36.06	31.39
3WP_CAS	32.94	25.71	23.48	34.25	39.45	35.12	31.82
1WP_TAS	35.91	20.98	14.25	NA	NA	NA	23.72
2WP_TAS	36.18	21.14	14.76	NA	NA	NA	24.03
3WP_TAS	37.01	22.61	16.93	NA	NA	NA	25.51
1WP_TAS_DAS	106.62	75.96	53.69	20.63	27.54	103.58	64.67
2WP_TAS_DAS	103.75	77.63	54.55	20.36	30.30	102.80	64.90
3WP_TAS_DAS	102.16	80.24	55.69	22.48	26.41	79.32	61.05
1WP_CAS_TAS_AAO	30.67	19.77	13.89	22.36	29.00	31.31	24.50
2WP_CAS_TAS_AAO	31.01	20.47	14.27	20.59	30.15	31.67	24.69
3WP_CAS_TAS_AAO	31.94	20.99	15.27	43.42	39.84	34.87	31.05
1WP_CAS_TAS_IPS	78.88	48.54	36.45	28.40	34.84	61.96	48.18
2WP_CAS_TAS_IPS	82.74	51.33	39.45	42.46	39.55	61.76	52.88
3WP_CAS_TAS_IPS	82.37	58.12	47.85	39.60	38.54	61.94	54.74
1WP_CAS_TAS_DAS_AAO	97.24	63.71	45.48	108.56	104.78	81.96	83.62
2WP_CAS_TAS_DAS_AAO	97.94	63.73	46.35	108.54	104.55	81.84	83.83
3WP_CAS_TAS_DAS_AAO	96.41	63.41	45.91	101.21	103.49	83.43	82.31
1WP_CAS_TAS_DAS_IPS	170.44	108.36	78.06	200.85	198.68	147.85	150.71
2WP_CAS_TAS_DAS_IPS	171.90	111.37	80.23	202.61	199.87	149.59	152.59
3WP_CAS_TAS_DAS_IPS	172.65	114.32	84.74	198.03	197.49	152.59	153.30
1WP_D/CAS	24.42	16.33	12.46	27.25	26.40	20.15	21.17
2WP_D/CAS	24.61	16.49	12.26	26.46	26.25	19.89	20.99
3WP_D/CAS	27.06	18.23	13.37	37.08	34.48	24.90	25.85