ACHIEVING QUALITY OF SERVICE IN MOBILE AD HOC NETWORKS CONTAINING PACKET FORWARDING ATTACKERS

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Engineering and Physical Sciences

2013

By
Peter J J McNerney
School of Computer Science

Contents

Word Count: 80,482

List of Tables

List of Figures

List of Algorithms

Abstract

In future, Mobile Ad Hoc Networks (MANETs) may provide access to services in the Internet. MANETs should therefore support diverse applications and data types. This introduces a need for quality of service (QoS), a process of discriminating different data types to provide them with an appropriate level of service. However, QoS can be affected by nodes performing packet forwarding attacks.

A critical analysis of the related literature shows that research into QoS and security has typically proceeded independently. However, QoS and security should be considered together as attacks may adversely affect QoS. A simulation study demonstrates this by investigating two single-path packet forwarding approaches under a range of conditions. The study shows that using single-path packet forwarding in the presence of attackers is generally insufficient to support QoS.

Based on this background research, a novel 2-Dimensional Adaptation ARChitecture (2-DAARC) and a Priority-based Multi-path Type Selection (PMTS) algorithm are proposed. 2-DAARC integrates two modes of adaptation. The single-path adaptation (SPA) mode uses adaptive bandwidth reservations over a single path for QoS in the presence of node mobility. The multi-path adaptation (MPA) mode uses duplicated data packet transmissions over multiple paths for QoS in the presence of packet forwarding attackers. Adaptation occurs within and between modes to optimize priority packet forwarding in the dynamic MANET environment. The MPA mode uses the PMTS algorithm to select a secondary path which is maximally-disjoint with the primary path. This aims to select a path which may enhance reliability whilst keeping the costs of path selection low.

Simulating 2-DAARC shows that under light loads it achieves better QoS than related work, but with a higher control packet overhead. Simulating PMTS shows that under light loads it achieves packet deliveries which are at best as good as a related approach, with lower end-to-end delays and control packet overhead.

A novel Congestion and ATtack (CAT) detection mechanism is proposed to improve the performance of 2-DAARC in heavily loaded networks. CAT detection differentiates the causes of packet loss so that adaptation can be better tailored to the network conditions. Without CAT detection, 2-DAARC uses the MPA mode in congested conditions, and this worsens QoS. Simulating 2-DAARC with CAT detection shows that it generally achieves packet deliveries which are greater than or similar to, and end-to-end delays which are less than or similar to related work, and it does so with a lower control packet overhead.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Abbreviations

2-DAARC	2-Dimensional Adaptation ARChitecture
802.11	IEEE 802.11 MAC protocol
AIMD	Additive Increase Multiplicative Decrease
AODV	Ad hoc On demand Distance Vector routing protocol
APS	Adaptive Packet Salvaging
BE	Best-Effort
bps	Bits Per Second
BQ	Base QoS
c	Speed of Light (in a vacuum)
CAT	Congestion and Attack
CBR	Constant Bit Rate
CPU	Central Processing Unit
CTS	Clear To Send
D2PD	Duplicated Data Packet Detection
DCF	Distributed Co-ordination Function
DePDR	Degraded Packet Delivery Ratio
DQoS	Denial of QoS Request Attack
DuPLR	Duplicate Packet Loss Ratio
DSSS	Direct Sequence Spread Spectrum
DSR	Dynamic Source Routing protocol
E2-DAARC	Extended 2-DAARC
ECN	Explicit Congestion Notification
EQ	Enhanced QoS
GHz	Gigahertz
HMAC	Hashed Message Authentication Code
IEEE	Institute of Electrical and Electronics Engineers

IETF	Internet Engineering Task Force
IMANET	Internet-based Mobile Ad Hoc Network
IP	Internet Protocol
IPsec	Internet Protocol Security
Kbits/s	Kilobits per second
kph	Kilometres Per Hour
m	Metres
MAC	Media Access Control *or* Message Authentication Code
MANET	Mobile Ad Hoc Network
Mbps	Mega Bits Per Second
MHz	Megahertz
MPA	Multi-Path Adaptation
mph	Miles Per Hour
m/s	Metres/Second
MuTS	Multi-path Type Selection
NDO	Node-Disjoint path-Only path selection
OLSR	Optimized Link State Routing protocol
OPLR	Original Packet Loss Ratio
PMTS	Priority-based Multi-path Type Selection
PriQueue	Priority Queue
PRNG	Pseudo Random Number Generator
QoS	Quality of Service
RePDR	Reserved Packet Delivery Ratio
RERR	Route Error Control Packet
RES	Reserve
RREP	Route Reply Control Packet
RREQ	Route Request Control Packet
RTCP	Real Time Control Protocol
RTP	Real Time Protocol
RTS	Request To Send
SPA	Single-Path Adaptation
TCP	Transmission Control Protocol
TOS	Type Of Service
UDP	User Datagram Protocol

To Mum, Dad, & M-E.

Long silence.

VLADIMIR:

That passed the time.

ESTRAGON:

It would have passed in any case.

VLADIMIR:

Yes, but not so rapidly.

—Samuel Beckett, *Waiting for Godot*, Act I.

Acknowledgements

Matthew G. and Robert L. deserve a special mention for their support.

Most of all I would like to thank my family for everything they have done for me.

This research has been supported financially by an *Engineering and Physical Sciences Research Council Doctoral Training Award*, a *University of Manchester School of Computer Science Annual Enhanced ICT Stipend*, and a *University of Manchester School of Computer Science Enhanced ICT Stipend Award for Excellent Academic Achievements*.

"*A prevention-only strategy only works if the prevention mechanisms are perfect; otherwise, someone will figure out how to get around them.*"
—Bruce Schneier, Secrets & Lies (2000)

"*There's no such thing as 'secure' any more . . . We have to build our systems on the assumption that adversaries will get in . . . We have to, again, assume that all the components of our system are not safe, and make sure we're adjusting accordingly.*"
—Debora Plunkett, Director, NSA Information Assurance Directorate (2011)

"*I think we have to go to a model where we assume that the adversary is in our networks. It's on our machines, and we've got to operate anyway.*"
—Dr James Peery, Director, Information Systems Analysis Center, Sandia National Laboratories (2012)

"*. . . in an age where there's no discernible perimeter, perimeter-oriented defences are less and less effective. So the game shifts from [the] outright prevention of breaches to [the] early detection and response to breaches.*"
—Art Coviello, Executive Chairman, RSA, The Security Division of EMC (from evidence given to the Home Affairs Committee on e-Crime, 23 April 2013)

Chapter 1

Introduction

1.1 Mobile Ad Hoc Network (MANET)

A Mobile ad hoc network (MANET) is a collection of autonomous mobile nodes which communicate with one another using wireless links [31]. They cooperate in a distributed fashion to provide network functionality in the absence of fixed network infrastructure. Nodes are typically small, light-weight, and battery powered. They usually have fewer resources and are less capable than a desktop or portable computer. The network topology is dynamic due to node mobility [45]. The autonomy and self-sufficiency of MANETs enables them to be deployed as stand alone networks which do not require network infrastructure [37].

MANETs can also play an important role in accessing networked services in the Internet. MANETs, as edge networks, could facilitate Internet service access to broader groups of users. For example, emergency responders in a disaster relief scenario may require Internet connectivity to communicate with their headquarters, or to access resources available on the Internet [143, 167]. An Internet-based Mobile Ad Hoc Network (IMANET) [30] interfaces a MANET into the wired Internet backbone by adding one or more points of attachment to the Internet, thus connecting the two network components.[1] IMANETs can provide a seamless mobile extension to the Internet and allow ubiquitous mobile access to a large number of services that are available on the Internet [174]. This implies that a greater number of applications and diverse data types should be supported in the

[1] IMANETs are known by several names in the literature, including integrated MANET [174], Hybrid MANET [174], Internet MANET [93], Tethered MANET [177], and MANET attached to the Internet [99].

MANET component of the IMANET.

1.2 Network Quality of Service (QoS)

When multiple data types are involved there is a need to address the issue of Quality of Service (QoS). QoS is the process of differentiating traffic so that a certain set of requirements can be satisfied in terms of a set of constraints [4]. In other words, QoS is a process of discriminating different packet types to provide them with an appropriate level of service. Data within a MANET may have different priority requirements, just as they do in traditional fixed networks. These requirements are typically satisfied by servicing higher-priority traffic before lower-priority traffic. For example, in existing QoS approaches, real-time voice traffic has a higher transmission priority than best-effort traffic from Websites. As a MANET can be used to extend the Internet, it should therefore aim to provide a comparable level of QoS.

However, the characteristics of a MANET may present a challenge to the way in which QoS can be provided. In traditional networks QoS is supported by the presence of infrastructure, which includes routers [15, 16]. The position of a router is fixed in such networks. In infrastructure-less MANETs, routers, which are peer communication nodes, are also mobile. The dynamic nature of MANETs therefore presents a significant challenge to QoS provisioning. In addition, the issue of QoS provisioning is further complicated by the fact that not all communication nodes may be able or willing to participate in the routing (packet forwarding) process.

1.3 Implications of Security Threats on QoS

Distributing the routing process amongst MANET nodes presents a raft of security issues which may hinder QoS. For example, a node will not forward packets if it is *broken*, *overloaded*, *selfish*, or *malicious* [123]. A broken node is one suffering from a fault, e.g., a crash, which prevents it from participating in routing operations in the network. An overloaded node may not have sufficient resources, e.g., buffer space, to serve other nodes' packet flows. A selfish node is unwilling to forward packets in order to conserve its own resources, e.g., battery. A malicious node refuses to forward packets in order to disrupt network operations; thus a

malicious node is willing to expend resources to disrupt network operations. *Node mobility* also affects packet forwarding. Node mobility causes paths between a source node and a destination node to break as nodes roam beyond one another's wireless range. These five factors can be divided into two categories: failure and misbehaviour. The occurrence of one or more of these failures (broken/overloaded nodes and node mobility) or misbehaviours (selfish/malicious nodes) during data packet forwarding will cause packet loss, reducing the QoS received by a packet flow. This reduced QoS is experienced regardless of whether these adversarial conditions are triggered unintentionally (failure) or intentionally (misbehaviour).

Supporting QoS in MANETs is difficult for a further three reasons. First, MANET nodes may be heterogeneous devices with varying resource capabilities (CPU, memory, battery life, etc.). Handling these devices introduces a number of challenges. For example, something as simple as a drained battery reduces the number of nodes available to serve as a router. This may lead to fewer available paths and routing disruption [218], and it may have an adverse effect on QoS. Additionally, genuine failure, such as that caused by a drained battery, can be difficult to differentiate from node misbehaviour [222]. This is because the effect of a genuine failure and an attack may be the same: a failed/attacked node can no longer participate in the network. However, such a failure is still considered as a security issue, even though malicious activity is not the cause of the routing disruption [79].

Second, distributing routing operations amongst MANET participant nodes opens the door to a wide range of security attacks. Attacks can occur during the route discovery phase or data packet forwarding phase of routing protocol operation. A security mechanism may be used to secure a routing protocol's route discovery phase. However, the security mechanism may itself be exploited to a malicious node's advantage. For example, the time and resource costs of deploying a security mechanism could be used as the basis of an attack to tie-up a node's limited resources and to prevent it from participating in the network [46]. This will adversely affect network operation and QoS, as time and resources will be dedicated to security processing rather than routing.

Distributing routing operations also means that malicious nodes can simply and easily subvert the data packet forwarding phase of protocol operation. For example, the *blackhole attack* [222] is a simple attack targeted at data packets: a blackhole attacker, which is a malicious intermediate node, drops all of the data

packets it is asked to forward. This has an adverse affect on QoS. A blackhole attack may also be performed inadvertently by a selfish node. For example, a node may want to conserve battery power by refusing to forward other nodes' data packets. Whilst the motive is different, the consequence of this action is the same as performing the blackhole attack. These malicious and selfish behaviours exemplify how nodes cannot be relied on to behave responsibly and altruistically.

Finally, achieving security in this context is a challenge because there is generally no single entity that all nodes trust [51]. Nodes cannot be trusted in an open MANET. In MANET research, however, it is typically assumed that nodes are trustworthy and that they will perform the specified operations [86]. This assumption may not always be valid, especially when nodes maliciously or selfishly exploit their privileged position as routers. It is therefore unreasonable to assume that all nodes are always trustworthy and that they will always properly execute routing, security, and/or QoS operations.

1.4 Research Motivation and Challenges

Security and QoS are two areas of MANET research which have so far been largely carried out separately. A number of solutions have been proposed to protect against a wide-range of security attacks, but these solutions have not typically considered their effects on QoS. Similarly, several solutions have been proposed to support QoS in the dynamic MANET environment, but these solutions do not typically take security requirements into consideration; they assume that nodes will faithfully follow protocol specifications and correctly participate in protocol and QoS operations. However, as discussed above, some of the unique characteristics of MANETs make them vulnerable to security attacks, and these attacks have a direct effect on QoS provisioning. It may therefore be less effective to support QoS without considering security, but, on the other hand, too much security provision will cost more resources and this may adversely affect the QoS which may be achieved. Thus it may be necessary to consider both QoS requirements and security requirements in an integrated manner. The hypothesis of this thesis is that integrating security and QoS may be the most effective way of providing QoS in MANETs.

One major challenge to overcome in this research is that intermediate nodes

cannot be trusted. One approach to address this issue is to apply a secure, QoS-guided route discovery process. For example, when attempting to discover a path between a source node and a destination node, only authenticated nodes which satisfy a QoS requirement will be selected to forward packets [67]. However, this approach cannot prevent an authenticated node from dropping data packets: attacks on packet forwarding cannot be prevented as a node may drop all packets passing through it [215]. An alternative approach is to impose a trust model on intermediate nodes. Trust information can be used to select trusted intermediate nodes to forward data packets. However, such an approach presents its own challenges. For example, it takes time for nodes to earn trust, and there is a significant overhead in reliably distributing this trust information around the network [217]. It is also difficult to ensure the trustworthiness of this distributed information [182], as nodes may falsely accuse one another of misbehaviour [219]. Additionally, a trusted node may still proceed to drop the data packets it is asked to forward [217]. Supporting QoS in the presence of untrustworthy intermediate nodes is therefore a challenging and non-trivial task.

1.5 Research Aim and Objectives

The aim of this research is to investigate what is the best way to support QoS in a MANET containing malicious nodes. To make the investigation feasible within the given time-frame, it is assumed in this thesis that malicious nodes only perform packet forwarding attacks on data packets. The reason for this is that data packets are not typically transmitted during the route discovery phase of routing protocol operation; and this thesis is concerned with the QoS that the data packets, rather than protocol control packets, receive. Research on supporting the delivery of routing protocol control packets in the presence of attacker nodes is a substantial research area in and of itself.

The above aim is supported by the following five objectives.

1. To investigate and analyse MANET characteristics and security threats in this environment, and understand their implications on the provision of QoS.

2. To analyse and specify design requirements for a QoS solution to support

effective and efficient QoS provisioning (including security related requirements) in a MANET containing malicious nodes.

3. To analyse critically existing MANET QoS solutions against the specified requirements to identify limitations, weaknesses, and missing features in their support for QoS provisioning.

4. To design a solution to support QoS in the presence of node mobility and malicious attacks. The design will be guided by the requirements specified in (2) and based on the existing state-of-the-art analysed in (3), harvesting their strengths and overcoming their limitations and weaknesses.

5. To perform a performance evaluation of the proposed solution, and to compare its performance with related work.

1.6 Research Hypothesis

The hypothesis of this research is that duplicating priority data packets over multiple best-effort paths may better support priority packet deliveries than the bandwidth reservation-based approach in the presence of packet forwarding attacks. However, data packet duplication is expensive in terms of bandwidth, so it should only be used when the attacker ratio is high. When the attacker ratio is low, the bandwidth reservation-based approach should be used to deliver priority data packets along a single path. This gives rise to the idea of using two modes of packet forwarding dynamically to achieve QoS: data packet duplication over multiple paths and bandwidth reservation over a single path.

1.7 Research Method

The research method used for this project consists of three key components: a literature survey and a critical analysis of the related work, theoretical work, and simulation modelling and an evaluation of the proposed solutions.

1.7.1 Literature Review

The first task carried out in this research was to study the relevant literature on the three general areas of interest: MANET routing protocols, QoS, and security. The purpose was to become familiar with MANET routing operations and to understand security threats and attacks in open MANETs and their effects on QoS. Although this is initial research, it was apparent that there was little existing work on achieving QoS in MANETs containing malicious nodes. It was discovered that existing solutions largely focus on either supporting QoS or achieving security, but not both. This gap was identified through a critical reading and analysis of the literature. Performing the literature review and critically analysing the literature satisfies Objectives (1) and (3). A design hypothesis was formulated based on the analysis of the state-of-the-art routing, QoS, and security solutions. A number of requirements were specified to guide the design of a solution to support QoS in the presence of attacker nodes. Specifying a design hypothesis and design requirements satisfies Objective (2). Theoretical work to design solutions based on the design hypothesis could then begin.

Although the literature review forms a key starting point for research it does not terminate once the research transitions into the next phase, the theoretical phase. Surveying the relevant literature is a key component of the research method. The literature was regularly reviewed throughout the duration of this research, with newly published work taken into consideration where necessary.

1.7.2 Theoretical Work

Following the initial literature review, a novel idea was formed to support QoS in MANETs containing malicious nodes. The implementation of this idea has led to two novel contributions. One is the design, implementation, and simulation evaluation of a novel routing architecture which supports dynamic adjustments of routing strategies in response to QoS statistics collected at the destination node. The other novel contribution is the design, implementation, and simulation evaluation of a path selection algorithm for multi-path routing. During the evaluations it was discovered that there was room for further improvement in the design of the routing architecture. This led to the design, implementation, and simulation evaluation of a third novel solution, a congestion and attack detection mechanism. The design of these three solutions satisfies Objective (4).

1.7.3 Simulation Study

To evaluate the designs proposed in this thesis, it was necessary to chose the most suitable evaluation methodology. Three evaluation methodologies were identified: simulation, experimental, and mathematical. Simulation was chosen, as the experimental methodology was not practicable and the mathematical methodology was too restrictive.

Prior to the simulation evaluation of the designs, it is important to ensure that the designs are correctly modelled. Two methods have been used to validate the simulation models. The first methods is to use validation scripts packaged in NS-2 (as Network Simulator version 2 [138] is used as the evaluation tool for this thesis). The simulator output is compared with reference output to validate if the simulator and simulator plug-ins have been correctly installed. The second method is to derive a mathematical model of a simplified simulation model and compare the results with those of simulation runs of the simplified simulation model. This method helps to ensure that the simulation models constructed are valid, i.e., that they are accurate representations of the designs. On completion of the simulation model validation, one could be confident of the accuracy of the simulation models and simulation results. The simulation-based performance analysis of the designed solutions could then be performed, satisfying Objective (5).

The final phase of the research method was to evaluate the results collected from the simulation runs. The results were analysed and compared with those from the related work to determine the efficacy of the proposed solutions. Conclusions were drawn from the evaluations of the proposed solutions, and directions for future research were identified.

1.8 Novel Contributions

The work presented in this thesis has led to four novel contributions.

Novel contribution 1: a simulation study of state-of the art approaches to packet forwarding in adversarial network conditions [124, 126]

The simulation study is performed to confirm empirically that integrated security and QoS is required to support data packet forwarding in MANETs containing malicious nodes. The study compares the performance and behaviour

of a reservation-based approach and a best-effort approach to packet forwarding under a range of network conditions. The results are critically analysed to understand the effects and implications of MANET characteristics (node mobility and peer nodes as routers) and network conditions (network load, types of attack, and number of attackers) on security and QoS. The insights gained inform the designs of the integrated security and QoS solutions proposed in this thesis (novel contributions 2–4). The solutions' designs are therefore evidence-driven: they harvest the strengths of the existing approaches whilst overcoming their limitations and weaknesses.

Novel contribution 2: an architecture for 2-dimensional adaptation [125]

A 2-Dimensional Adaptation ARChitecture (2-DAARC) is proposed for adaptive QoS provisioning in MANETs containing blackhole attackers. 2-DAARC combines two dimensions (modes) of adaptation, a single-path adaptation (SPA) mode and a multi-path adaptation (MPA) mode, to support QoS for priority data packet forwarding. The SPA mode uses adaptive bandwidth reservations and packet priority assignments along a single path. It is used when network conditions are suitable for bandwidth reservations. The MPA mode uses redundant, best-effort data packet transmissions along multiple paths. It is used to resist packet forwarding attacks. Adaptations within a mode and between the two modes are performed at the source node. Adaptations are governed by feedback data, in the form of QoS statistics, calculated by and received from the destination node. Adaptation also occurs in the absence of feedback by performing adaptive actions in response to timeouts. The integrity and authenticity of the feedback data are assured through the use of an authentication code generated from a keyed-hash function.

In designing 2-DAARC, the following measures have been taken to minimize the bandwidth and performance overheads. First, it uses its underlying routing protocol (DSR) to discover multiple paths to a destination node and store them in a Route Cache. It then uses the PMTS algorithm (described in novel contribution 3 below) to compare the disjointedness of the cached paths with the primary path. Extra control packets are not injected into the network to acquire a secondary path unless the Route Cache is empty, in which case new paths are

discovered. Additionally, it enables the source node to calculate path disjointedness and to select a secondary path, and this avoids the need to use data provided by intermediate nodes. This avoids the bandwidth and computational costs of establishing trust on intermediate nodes an on the data provided by them. It also enables the source node (1) to switch from the SPA mode to the MPA mode with low delay, and (2) to change the secondary path promptly when the in-use path either breaks or offers poor QoS.

2-DAARC is evaluated against INSIGNIA [108], a bandwidth-reservation-based approach to QoS in MANETs. The simulation investigation demonstrates that 2-DAARC performs better than INSIGNIA in three ways in lightly loaded networks: it achieves higher packet delivery ratios; the SPA mode delivers a greater percentage of packets using the bandwidth-reserved forwarding service; and it achieves lower end-to-end packet delivery delays.

Novel contribution 3: a priority-based multi-path type selection algorithm [125]

2-DAARC's MPA mode uses a novel Priority-based Multi-path Type Selection (PMTS) algorithm for secondary path selection. It selects a secondary path which is maximally-disjoint with the primary path, i.e., a secondary path which shares the least number of nodes and wireless links with the primary path. It does this by first calculating the disjointedness between the primary path and candidate secondary paths. It then selects a secondary path in a priority-based manner depending on the level of disjointedness offered by the candidate paths. Disjointedness is a measure of redundancy. Node-disjoint paths offer the highest redundancy, followed by link-disjoint paths, and then non-disjoint paths. Node-disjoint paths are most preferred and non-disjoint paths are least preferred.

The PMTS approach is evaluated against a node-disjoint-path-only approach. The evaluation results show that PMTS performs better in two ways in lightly loaded networks: paths selected using PMTS have a lower normalised routing load and support lower end-to-end packet delivery delays. The packet delivery ratios along the paths selected using PMTS are at best as good as those on node-disjoint paths.

Novel contribution 4: a congestion and attack detection mechanism

2-DAARC uses a novel Congestion and ATtack (CAT) detection mechanism

(1) to identify the likely cause of packet loss (congestion and/or attacks) and (2) to use this information to select the most appropriate mode of adaptation (SPA or MPA) for the inferred network conditions. The CAT detection mechanism works by combining the arrival rate of control traffic with the QoS statistics received from the destination node to infer the likely cause(s) of packet loss. This approach has two benefits. First, congestion and attacks are detected without pouring any additional control packets into the network. Utilizing the control packets already transmitted by the underlying routing protocol removes the risk of additional packets exacerbating existing network congestion. Second, intermediate nodes are not required to perform any operations other than those already prescribed by the underlying routing protocol. This means that the mechanism does not impose any trust on intermediate nodes.

An extended version of 2-DAARC with the CAT detection mechanism, hereafter referred to as E2-DAARC, is evaluated against (1) a version of 2-DAARC which uses Watchdog [123] (a misbehaviour detection system) and Explicit Congestion Notification [165] (a congestion notification system), hereafter referred to as Watchdog+ECN, (2) the original version of 2-DAARC without CAT detection, hereafter referred to as original 2-DAARC, and INSIGNIA. The comparison of the results shows the following. In static networks with both light and heavy loads, E2-DAARC achieves PDRs which are greater than or similar to those of Watchdog+ECN and INSIGNIA. In lightly loaded static and mobile networks, E2-DAATC achieves similar PDRs to original 2-DAARC. In mobile networks with both light and heavy loads, E2-DAARC achieves PDRs which are similar to those of INSIGNIA. In heavily loaded static and mobile networks, E2-DAARC achieves higher PDRs than original 2-DAARC. In both static and mobile contexts and under both light and heavy network loads E2-DAARC achieves (1) end-to-end delays which are less than or similar to those of Watchdog+ECN, original 2-DAARC, and INSIGNIA, and (2) lower normalised routing loads than Watchdog+ECN and original 2-DAARC.

1.9 Publications

Parts of the research contained in this thesis have led to the following four conference and magazine publications, all of which have been peer-reviewed.

Conference Publications

1. **Peter J. J. McNerney** and Ning Zhang, "A 2-Dimensional Approach to QoS Provisioning in Adversarial Mobile Ad Hoc Network Environments", in *15th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWIM 2012)*, 21–25 October, 2012, Paphos, Cyprus. http://dx.doi.org/10.1145/2387238.2387264. Acceptance rate: 25%.

2. **Peter J. J. McNerney** and Ning Zhang, "A Study on Reservation-Based Adaptation in Adversarial MANET Environments", in *8th International Wireless Communications and Mobile Computing Conference (IWCMC 2012)*, 27–31 August 2012, Limassol, Cyprus. http://dx.doi.org/10.1109/IWCMC.2012.6314286. Acceptance rate: 35%.

3. **Peter J. J. McNerney** and Ning Zhang, "Towards an Integration of Security and Quality of Service in IP-Based Mobile Ad Hoc Networks", *IEEE Global Communications Conference (GLOBECOM 2011)*, 5–9 December 2011, Houston, Texas, USA. http://dx.doi.org/10.1109/GLOCOM.2011.6133684. Acceptance rate: 36%.

Magazine Publications

4. **Peter J. J. McNerney** and Ning Zhang, "Smarter Cities: Making Societies Smarter". *XRDS: Crossroads* (Winter Edition), 10(3). New York, NY, USA, December, 2011. ACM. http://doi.acm.org/10.1145/2043236.2071895.

The author was one of four nominated finalists in the 'Connected World' category in the *EPSRC UK ICT Pioneers Competition* 2011 with the entry "Integrating Security and Quality of Service in a Smart City Environment". A poster presentation on the exploitation potential of this research was made to representatives of UK industry, learned societies, research councils, and government.

- **Peter J. J. McNerney** and Ning Zhang, "Integrating Security and Quality of Service in a Smart City Environment," *EPSRC UK ICT Pioneers Competition*, 23 March 2011, London, UK.

1.10 Thesis Structure

The remainder of this thesis is organised as follows.

Chapter 2 presents motivating application scenarios and a survey of the related literature. A number of observations are made on the applications scenarios. The challenges posed by the MANET environment are discussed. The literature survey reviews state-of-the art solutions to routing, QoS, and security in MANETs.

Chapter 3 presents the building blocks used in the proposed solutions and outlines the evaluation methodology used in this research. The building blocks include the Dynamic Source Routing (DSR) protocol [88], the INSIGNIA QoS framework [108], the symmetric key cryptographic system, and a Hashed Message Authentication Code (HMAC). Justifications for using a simulation-based evaluation methodology are provided. The simulation configuration is described in detail. A validation of the simulation model is presented.

Chapter 4 presents a simulation study which compares two exemplary, state-of-the-art approaches to packet forwarding in MANETs, a best-effort approach versus a reservation-based approach. The DSR protocol is used as the facilitator of the best-effort approach. INSIGNIA is used for the reservation-based approach. The results of this study are discussed to identify the best way to achieve QoS in a malicious MANET environment. The insights drawn from this study inform the designs of the solutions proposed in the following chapter.

Parts of this chapter have been published in the following two peer-reviewed conference papers: 'Towards an Integration of Security and Quality of Service in IP-Based Mobile Ad Hoc Networks' [124] and 'A Study on Reservation-Based Adaptation in Adversarial MANET Environments' [126].

Chapter 5 presents the design, implementation, and performance evaluation of the first two novel contributions of this research. These two contributions are a 2-Dimensional Adaptation ARChitecture (2-DAARC) and a Priority-based Multi-path Type Selection (PMTS) algorithm. The performance of 2-DAARC is compared with INSIGNIA. The performance of the PMTS algorithm is compared with a node-disjoint-path-only approach to path selection. Shortcomings of the 2-DAARC approach are identified. Recommendations to enhance 2-DAARC are proposed.

Parts of this chapter have been published in a peer-reviewed conference paper: 'A 2-Dimensional Approach to QoS Provisioning in Adversarial Mobile Ad Hoc

Network Environments' [125].

Chapter 6 presents the design, implementation, and performance evaluation of the third novel contribution of this research. This contribution is a Congestion and ATtack (CAT) detection mechanism. The CAT detection mechanism implements the recommendations proposed in the previous chapter. 2-DAARC extended with the CAT detection mechanism (E2-DAARC) is evaluated against Watchdog, Explicit Congestion Notification, and INSIGNIA.

Chapter 7 concludes this thesis and suggests directions for future research.

Chapter 2

Background and Literature Survey

2.1 Chapter Introduction

This chapter presents the background and motivation for this research, a survey of related literature in the areas of routing, QoS, and security, and identifies a way forward to address the issue QoS provisioning in the presence of packet forwarding attackers.

In detail, in Section 2.2 application scenarios are presented to motivate the need for integrated security and QoS. In Section 2.3 a number of observations are made on the application scenarios regarding the MANET communications environment. Section 2.4 presents a review of the main MANET routing protocols. Section 2.5 describes different approaches to QoS provisioning. Section 2.6 discusses existing approaches to QoS adaptation. Section 2.7 describes security threats to QoS. Section 2.7.3 looks at ways to stimulate node co-operation to overcome some of these security threats. Section 2.8 describes how multi-path routing can be used to support security and QoS requirements. Section 2.9 outlines existing approaches to security and QoS integration. Section 2.10 identifies what is missing in the related work for integrated security and QoS support. Section 2.11 outlines the best way forward to achieve integrated security and QoS for QoS provisioning in the presence of packet forwarding attackers. Finally, Section 2.12 presents concluding remarks.

2.2 Motivating Application Scenarios

MANETs offer an important alternative to traditional infrastructure-based networks. Their ability to support communications without infrastructure, in locations devoid of infrastructure or where it has been destroyed, makes them ideal for use in responding to natural disasters or in battlefield scenarios. A MANET can also extend the range of the Internet by serving as an edge network. This can enable mobile users to share and access data and resources on the Internet, where it would have not previously been possible. Tele-medicine is an application scenario which may benefit from MANET and Internet integration and the provision of QoS in this context.

This section presents three application scenarios—disaster relief, battlefield, and tele-medicine—to illustrate that providing QoS in MANETs is necessary and that security should be considered as an integral part of QoS provisioning.

2.2.1 Disaster Relief

Hurricane Katrina wreaked havoc across the USA and Gulf Coast in 2005. Parts of the communications infrastructure were damaged or destroyed, and this hindered the federal response [188]. Some of those affected by the disaster established a mesh network comprising a number of static nodes distributed over a large geographic area. This enabled communications between rescuers, officials, and civilians [13]. Mesh networks are related to MANETs: both are types of wireless network which may have dynamic topologies, e.g., as a sequence of lossy links, but MANET topologies are determined ad hoc and may also be affected by node mobility [10, 58].

Another use of ad hoc networks in disaster relief is to provide a communications platform for rescue robots. Autonomous and semi-autonomous rescue robots can be sent into damaged buildings which may be hazardous to rescue workers. Rescue workers operating the robotic platform in [92] use streaming video and static images to control a robot and to assess a disaster scenario remotely. These data are communicated to the robot's operator over an ad hoc network. The robot also uses an array of sensors for location tracking and obstacle detection. Thus the real-time streaming video, images, sensor data, robot control data, and routing protocol data must co-exist within the network.

2.2.2 Battlefield

Another use of robots and ad hoc networks is by the military on the battlefield. Robots can be controlled remotely over a MANET by soldiers. Using a MANET as the communications platform allows the robots to be deployed quickly and conveniently in urban environments [120]. For example, the work in [140] shows the integration of robots into a military mission. A robot is 'tele-operated' by a soldier. The soldier navigates the robot using a real-time video feed. This feed is captured by the robot and streamed over a MANET to the soldier's control unit. The video data must co-exist in the network with the robot control data and routing protocol control data. There is therefore a requirement to support the requirements of different data types simultaneously. Additionally, these data must be secured so that they cannot be accessed by an enemy.

2.2.3 Tele-Medicine

Tele-medicine is the delivery of healthcare and medical expertise using communication technologies [75]. These technologies enable medical information to be exchanged and medical and healthcare services to be delivered between geographically distributed entities (individuals and health service providers) [76]. For example, through the use of communication networks healthcare and medical treatments can be provided to patients in remote areas, or in areas where there is a lack of medical expertise or infrastructural support (e.g., for disaster relief).

Mobile wireless devices, when connected to the Internet, can allow access to medical expertise or expert treatment anywhere and at any time. For example, pre-hospital treatment can be provided or enhanced in a mobile context; and expert treatment or diagnoses are possible while a patient is en route to a hospital in an ambulance. The monitoring of the patient's health data (vital signs such as heart rate, blood pressure, etc.) can be undertaken in the ambulance and transmitted in real-time to the hospital [113, 205, 223]. Other real-time and non-real-time medical data, such as high-quality video, still images (X-rays), and electronic patient records [59], can also be transmitted, so that experts, regardless of their locations, can make a visual inspection of these data and provide a diagnosis prior to the patient's arrival at the hospital.

From the above it can be seen that communication technologies can play an important role in supporting healthcare. This applies in both mobile and static

contexts as well as in locations with and without infrastructure. There exists a need to support a wide-range of applications and provide resilient communications between different health professionals, regardless of their locations, from first contact with a patient to his or her arrival at and departure from a hospital.

2.3 Observations from Application Scenarios

Two main observations can be made from the above described application scenarios: (1) networks carry multiple data types , and (2) the diversity of devices and the dynamics of the MANET environment present a number of challenges to effective communications. This section first addresses the issue of multiple data types co-existing in a MANET before exploring in detail the aspects of the communications environment affecting QoS and security.

2.3.1 Multiple Data Types

One commonality in the above application scenarios is that several data types need to co-exist in the network. These data types can be grouped into three categories: protocol data, multimedia data, and device data. The protocol data are the control data exchanged between nodes to discover paths through the network. The multimedia data are the video, audio, and images transmitted between communicating endpoints. The device data are those captured by sensors and those used for device control, e.g., moving robots and rotating cameras.

Some of these data types have a higher priority than others, i.e., they have different QoS requirements. For example, it may be necessary to give protocol control data the highest priority. This is because these data are fundamental to network operations. The multimedia data can be divided into two categories, with each having a different priority requirement: real-time and best-effort. The video-feeds from the robots in the disaster relief and battlefield scenarios have a real-time characteristic. These data require priority treatment from the network. The still images in the disaster relief and tele-medicine scenarios do not have a real-time requirement. A best-effort service may be suitable for these data. To support effectively the QoS requirements of the multiple data types it is necessary to consider the effects that the characteristics of the MANET communication environment have on QoS provision.

2.3.2 Communication Environment and its Characteristics

In addition to the multiple data types, four further issues are observed from the application scenarios: node mobility, device (node) heterogeneity, limited wireless bandwidth, and security threats.

2.3.2.1 Node Mobility

The mobility of nodes, which are also routers, presents a challenge to smooth network operation. MANET nodes are typically mobile. They roam in and out of one another's wireless transmission range. This results in a dynamically changing network topology. This means that a path established between a source node and a destination node may be short-lived. If the underlying network is not able to respond to such dynamic topological changes in a timely manner it may result in significant periods of service disruption, e.g., delay and lost packets. Supporting QoS in this dynamic environment is therefore a challenging issue.

2.3.2.2 Device Heterogeneity, Limited Node Capabilities, and Limited Channel Bandwidth

Devices (nodes) participating in a MANET are typically heterogeneous and their specifications may vary considerably. A wireless node usually has lower resource capabilities (CPU, memory, etc.) than a desktop or laptop computer. Table 2.1 shows a comparison of processing and memory capabilities of a selection of mobile devices including a sensor [132], a PDA (Personal Digital Assistant) [64], a smartphone [65] and a laptop [35]. The range of capabilities is considerable. For example, the memory available on the laptop (3072MB) is approximately 48 times greater than that on the sensor (64KB). The specification of a device may limit the resources it is able to dedicate to the servicing of other nodes' packet flows.

In addition to potentially limited node resources, wireless networks have a lower bandwidth availability than their wired counterparts. Bandwidth availability may often fluctuate in a MANET. This may be due to (1) MANET-specific factors such as node mobility and (2) general factors such as a variable network load. A QoS solution for the wireless environment should therefore use the limited bandwidth efficiently and effectively.

Device	Processor Speed	Available Memory
Sensor	12MHz	64KB
PDA	624MHz	64MB
Smartphone	1GHz	576MB
Laptop	2GHz	3072MB
Desktop	2.93GHz	4096MB

Table 2.1: A Comparison of Processing and Memory Specifications of Heterogeneous Devices.

2.3.2.3 A Lack of Infrastructural Support

A MANET is devoid of infrastructure and does not have any central authority, coordination, or control. The responsibility of operating a network therefore falls on the communicating nodes participating in the network. This distributed approach to providing network services prompts a shift from one-hop wireless communications to multi-hop wireless communications. Nodes have the responsibility of providing routing services for other nodes. This is in addition to the communication of their own data flows. However, nodes may exploit their position as routers to affect the QoS they offer to other nodes' packet flows.

2.3.2.4 Security Threats

A number of threats to network operations are introduced as a result of delegating routing responsibilities to MANET participant nodes. Having such responsibility places these nodes in a position to launch security attacks. For example, a node could discard or delay any traffic it is required to forward. If no counter-measures are provided these security threats may significantly influence the QoS the data flows receive.

An additional security threat, although non-malicious, is the issue of device fallibility. It is significant in this environment where peer nodes act as routers. Hardware failures and depleted batteries, for example, can lead to path unavailability and routing disruption [218]. These genuine failures can be difficult to differentiate from node misbehaviour [222]. This is because failures and misbehaviour may both have the same consequence, i.e., packet loss. Whilst these failures do not result from malicious activity, maintaining path and network availability remains a security issue [79].

Security mechanisms designed to combat insecurity in a MANET may themselves be exploited to the advantage of malicious nodes. For example, the time

and resource costs of processing a security mechanism could be used as the basis of an attack to tie-up a node's limited resources and to prevent it from participating in the network [46]. This will adversely affect network operations as time and resources will be dedicated to security processing rather than routing. The level of security employed and the way the security is provided are therefore significant to the QoS that the network can support. All of these factors and considerations illustrate the association between security and QoS, and reinforce the notion that security concerns need to be addressed in order to support QoS.

2.4 MANET Routing Protocols

The routing protocols published by the Internet Engineering Task Force (IETF) MANET Working Group [78] can largely be classified into two categories: *proactive* and *reactive*. Both proactive and reactive routing protocols offer a best-effort routing service. *Hybrid routing* is a third category of routing protocol which is published in the literature; but no hybrid protocols are under consideration by the Working Group.[1] Hybrid routing protocols are therefore not considered in this research. Protocols considered by the IETF MANET Working Group are discussed in this section.

2.4.1 Proactive Routing Protocols

With a proactive routing protocol, each node in a network attempts to maintain an up-to-date view of the network topology. The main benefit of this approach is that a source node will already know a path to a destination node when it wishes to communicate with it. This minimizes the path acquisition delay. Proactive routing protocol operations are summarised as follows. Nodes attempt to maintain routing tables to other nodes in the network. Routing tables are kept fresh through the exchange of routing control packets. Control packets can be exchanged at two levels: globally and locally. Globally disseminated control packets are propagated to all nodes in a network. This allows a node to maintain complete paths from itself to each of the other nodes in the network. Locally exchanged control packets are only propagated to nodes in the wireless one-hop neighbourhood. This allows a node to determine whether a bi-directional link can

[1] Hybrid routing protocols combine proactive and reactive routing protocol mechanisms.

be maintained with the neighbouring nodes. Bi-directional links are necessary for data packet transmissions.

One of the main issues with the proactive routing approach is the high control packet overhead it generates when maintaining a node's view of the network. Thus the instant availability of paths comes at a cost: control packets must be transmitted periodically, and they must be transmitted regardless of whether or not a node is participating in routing. This will impose additional costs in terms of both individual node resources and network bandwidth. Individual nodes must also maintain internal routing tables. Entries in a routing table may regularly become stale as a consequence of node mobility. Nodes must propagate local and global control packets to inform all nodes in the network of a path break. The number of control packets required to keep paths valid and up-to-date increases as node mobility and the number of nodes in the network increases [87]. A trade-off exists between immediate path availability and the amount of control packets required to maintain these paths.

OLSR and TBRPF are two example proactive routing protocols. The Optimized Link State Routing (OLSR) protocol [27, 83] is designed to minimize control packet overhead and rapidly adapt to changes in network topology. While trying to keep each node's view of the network up-to-date, it reduces the number of control packets transmitted by using controlled flooding, i.e., control packets are broadcast to neighbouring nodes but only a subset of these nodes retransmit the control packet to their neighbouring nodes.

The Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) protocol [142] provides hop-by-hop shortest path routing to each destination node. Each node maintains a partial view of the global topology. If a node wants to know the shortest path to every node in the network it computes a minimum spanning tree rooted at itself using a modified version of Dijkstra's algorithm. A spanning tree is a tree containing each node in a graph [98]. In other words, it is a tree containing each node in the network. The Dijkstra algorithm computes the least-cost path from one node to all other network nodes [98]. This is used periodically and in response to topology changes. TBRPF attempts to reduce the volume of control packets in the network. Control packets are exchanged between neighbouring nodes to report changes in the neighbourhood. This reduces the overhead, as they are exchanged when required rather than periodically.

2.4.2 Reactive Routing Protocols

Reactive routing protocols discover a path to a destination node 'on-demand'. A path is only discovered when a node wishing to communicate does not already have a path to the intended destination. This aims to avoid the high overhead typically experienced by proactive routing protocols when maintaining complete, up-to-date, global routing tables at each node. Reactive routing protocols can be divided into two categories: *hop-by-hop routing* and *source routing*.

With the hop-by-hop routing approach each intermediate node maintains a routing table. The table contains the addresses of neighbouring nodes. An intermediate node needs to know which neighbouring node each packet should be forwarded to. A node acquires this information during the Route Discovery phase of routing protocol operation. The node creates a temporary routing table entry for the source/destination pair described in the header of a received route discovery packet, known as a Route Request. This entry is initialized when a reply to the route discovery, known as a Route Reply, is received; the address of the neighbouring node which transmitted the reply forms part of the routing table entry. The routing table entry specifies which neighbouring node a packet for a particular destination should be forwarded to. Hop-by-hop routing therefore requires intermediate nodes to dedicate resources for the creation and maintenance of routing table entries.

The Ad hoc On demand Distance Vector (AODV) routing protocol [156, 159] is a hop-by-hop routing protocol. Each node periodically transmits local control packets so that neighbouring nodes can be identified. This determines the local connectivity between nodes. Global control packet dissemination is not used. An intermediate node establishes a routing table entry for a source node during a Route Discovery process. When a data packet arrives, the intermediate node consults the routing table entry to identify the next-hop node of the path.

In contrast to the hop-by-hop approach, the source routing approach enables routing protocol operation to be fully on-demand. This means that control packets are only exchanged when a path is required. With the source routing approach the complete path between a source node and a destination node is carried in the header of a packet. This is known as the *source route*. A benefit of this approach is that intermediate nodes are not required to maintain routing information as the entire path is provided in each packet they forward. Intermediate nodes forward the packet to the next-hop node specified in the source route. A disadvantage of

this approach is that the size of a packet grows as the length of a source route increases.

The Dynamic Source Routing (DSR) protocol [88, 90] is a source routing protocol. A source node uses the controlled flooding of ROUTE REQUEST packets, as part of a Route Discovery process, to discover a path to a destination node. Each intermediate node receiving a ROUTE REQUEST packet appends its address to the source route contained in the header. Having received a ROUTE REQUEST packet the destination node unicasts a reply over the reverse of the source route contained in the packet.

The above described proactive and reactive routing protocols offer a best-effort forwarding service. A number of QoS solutions have been proposed to support QoS provisioning on top of best-effort routing protocols.

2.5 Existing Approaches to QoS Provisioning

Existing QoS solutions published in the literature largely use a number of methods to support QoS (i.e., to provide preferential treatment to higher priority traffic). These methods are resource reservation, admission control, service differentiation, QoS frameworks, and QoS routing.

2.5.1 Resource Reservation and Admission Control

With the resource reservation-based approach, a source node requests and reserves some specific level of capacity for a priority data flow at intermediate nodes, on a path to a destination node, prior to data packet transmissions. This is carried out through the use of QoS signalling and admission control.

QoS signalling is used to set-up, maintain, and tear-down resource reservations. When a reservation request is received by an intermediate node the admission control module of that node will examine the requested resources against what is currently available. If the node has sufficient spare capacity to satisfy the request the flow will be admitted, otherwise it will be rejected. A successful reservation should aim to provide the necessary level of QoS for the duration of a data flow, although there are cases where a lower quantity of resources may be offered to a flow. This latter case is useful in a congested network as it allows for a flow to be served where it otherwise would have been rejected.

Once a flow is admitted the intermediate node will typically reserve the requested resources and install a reservation state for the flow. The reservation state can be installed in one of two ways: *hard-state* or *soft-state*. A hard-state reservation is maintained until the receipt of a *teardown* control message. This message explicitly tells intermediate nodes along the path to release any reserved resources for the flow. In contrast, a soft-state reservation must be refreshed periodically before the expiration of a timeout, otherwise the reserved resources will be released. Alternatively, resources are released on receipt of a teardown message. The fundamental difference between a hard-state reservation and a soft-state reservation is that a hard-state will exist until it is signalled that it should be removed, whereas a soft-state may be removed locally by the router after a period of inactivity. The soft-state mechanism is usually employed in best-effort networks. This is because reliable, guaranteed signalling is not required, unlike the hard state mechanism [85].

There are a number of limiting factors with the resource reservation-based approach to QoS. These factors include scalability, 'starving-out' best-effort traffic, and non-optimal resource allocation due to soft-state timers. The issue of scalability is one of the most significant limiting factors. A state must be installed and maintained at each router between a source node and a destination node for every flow demanding the reserved forwarding service. This introduces additional complexity, due to the need for packet classification, scheduling, and admission control modules at each node, which can slow down the speed of packet processing [84].

Dedicating time and resources to serve resource reservations may mean that reservation-based flows are satisfied to the detriment of best-effort traffic [4, 22, 84]. Best-effort traffic is susceptible to becoming starved-out as the number of reserved flows increases. This is due to more resources being dedicated to serving the reserved traffic.

The time-out period of a soft-state connection can also lead to unnecessarily high resource consumption. If a node does not transmit an explicit teardown message to end a session gracefully, e.g., in the event of a crash, resources will still be reserved and can remain unused until the timeout occurs. These resources will not be available to other users during this time, even though they remain unused and unneeded.

Owing to these shortcomings the IETF has decided to move away from the

stateful, reservation-based, and per-flow approach to QoS towards a scalable, class-based, and all together more simple approach.

2.5.2 Service Differentiation

Service differentiation is the process of providing different grades of service to different packets [84]. QoS is provided through the aggregation of traffic classes. In other words, QoS is provided on a per-class basis rather than on the per-flow basis as used by resource reservation. Service differentiation is accomplished using two mechanisms: packet classification and per-hop behaviours. Packet classification is the process of marking a packet with the type of service it wishes to receive from the routers in a network. A class identifier is inserted into the header of an IP packet during packet classification. Routers use this identifier to apply the appropriate per-hop behaviour (PHB) necessary to meet this requirement. A PHB determines the forwarding behaviour, and therefore the QoS, that a packet is to receive. More specifically, a PHB defines a queue management technique for a router's output interface queue used to buffer data packets [135].

The per-hop forwarding behaviours from the Differentiated Services (Diffserv) framework [15] are used to explain the main classes of PHBs. Diffserv is a framework for scalable service differentiation through the aggregation of traffic classes. It is designed for the wired Internet. Diffserv has three main classes of per-hop forwarding behaviour: *default*, *expedited forwarding*, and *assured forwarding*. The default PHB services traffic using best-effort forwarding. Expedited forwarding [33] is further composed of two classes: expedited forwarding and non-expedited forwarding. The expedited forwarding class aims to make the network appear unloaded and thus provide the service quality of an unloaded network. This is achieved by servicing the relevant packets as quickly as possible to minimise the queuing time, jitter, and packet loss experienced by the packets. It is expected, however, that the majority of traffic requires non-expedited forwarding, which is handled using the best-effort service model.

With Assured Forwarding [56] packets are classified into one of four *priority classes*. Each priority class has a corresponding queue with a specified weight. A class with a higher weight receives a greater portion of forwarding resources (buffer space and bandwidth) than lower weight classes. The four Assured Forwarding classes each contain three *drop precedences*: *low*, *medium*, and *high*. These determine the relative priorities of flows within a class. When congestion occurs

traffic with a higher drop precedence will be discarded before traffic with a lower drop precedence. Twelve different packet classifications represent the combined priority classes and drop precedences. This broad range of packet classifications attempts to ensure that all traffic types can receive QoS which is tailored to their requirements.

The service differentiation approach exhibits several advantages over the resource reservation approach. First, state maintenance is not required. This makes service differentiation more scalable [161]. Additionally, this reduces the resources (memory, processing, etc.) that routers need to dedicate to servicing packet flows. Second, as routers are not required to undertake advanced set-up for each new data flow, the routers can focus their resources on routing.

The service differentiation approach has one main disadvantage. It is reliant on Service Level Agreements (SLA), and it may be difficult to enforce a SLA in a MANET. A SLA is a contract which defines performance metrics and acceptable levels of service that a network provider should support [98]. In other words, a SLA defines the level of service that a user should receive from a network provider. However, MANETs do not have a centralised network provider, i.e., they do not have a central authority and control. This makes it difficult to ensure that traffic requests and receives the correct level of service as prescribed by the PHBs and the SLA. In other words, it is difficult to ensure that the nodes which request and are entitled to priority treatment will receive it, and that those which request it unnecessarily are awarded a more appropriate lower level of service.

Both the resource reservation and service differentiation approaches were originally designed for use in static, wired networks. As the prevalence of wireless networking has increased, and as devices have become mobile, the research community has reacted by devising new QoS strategies specific to such environments.

2.5.3 QoS Frameworks

Some existing efforts on QoS support in MANETs are on the design of QoS frameworks to improve scalability and to optimize packet delivery with minimal resource consumption. For example, the Flexible QoS Model for MANETs (FQMM) [209] uses a 'hybrid provisioning' mechanism to reduce the amount of traffic served with reservations. It integrates reservation-based and class-based QoS services by providing per-flow, i.e., reservation-based, QoS only to

a small portion of high-priority traffic, and per-class QoS to the remaining non-high-priority traffic. By reducing the amount of traffic requiring per-flow QoS treatment, FQMM attempts to reduce the scalability problems associated with reservation-based approaches, where large numbers of resource reservations can tie-up routers' resources. As this hybrid provisioning can limit the processing and memory resources committed to resource reservations, it is an attractive approach to QoS provisioning in environments where resources are constrained.

INSIGNIA [108] is a QoS framework designed for MANETs. It supports two levels of QoS: reserved delivery for high priority traffic and best-effort delivery for lower priority traffic. INSIGNIA is lightweight and adaptive in response to changes in network conditions and topologies. It uses 'in-band' QoS signalling by encapsulating the signalling information in the IP Option field of every IP data packet. In addition to reducing signalling overhead, this method also allows INSIGNIA to support the fast restoration and re-routing of reserved data flows during the lifetime of a session. QoS adaptation is supported through a process known as QoS Reporting. QoS Reports are transmitted out-of-band, i.e., separately from data packets, to inform the source node of the success or failure of a reservation. These reports allows the source node to 'scale-up' or 'scale-down' QoS requirements in response to measured network conditions.

The Hybrid QoS Model for MANETs (HQMM) [53] is inspired by the ideas in FQMM and incorporates the merits of INSIGNIA. As in FQMM, HQMM provides per-flow QoS to a small portion of high-priority traffic and per-class QoS to the remaining traffic. The per-flow traffic is served using INSIGNIA.

2.5.4 QoS Routing

Another category of efforts on supporting QoS provisioning in MANETs is by designing QoS routing protocols. These protocols are designed to satisfy some specified QoS requirements by adding relevant QoS parameters into the routing table of each node. When a path is required the corresponding QoS parameter values associated with each path are calculated, and the path with the best value is chosen. End-to-end delay and bandwidth are the commonly used QoS parameters in early QoS routing protocols such as QoS-AODV [42] and Quality of Service AODV [157].

Other QoS routing protocols have started using other additional information, e.g., energy, as QoS parameters to make routing decisions energy-aware. The

Stability-based, QoS-capable AODV (SQ-AODV) protocol [195] is an example of an energy-aware QoS routing protocol. It defines a new QoS parameter, called residual node energy, and uses it to govern path selections. It uses a cross-layer approach to estimate residual node energy and feeds this estimation into the path selection and maintenance process, thus minimizing connection interruptions due to battery depletion.

2.6 Existing Approaches to QoS Adaptation

This section focuses on existing solutions which use adaptation to satisfy QoS requirements in dynamic network environments. These solutions can largely be classified into three categories depending on the adaptation parameters used: (1) traffic adaptation, (2) path adaptation, and (3) bandwidth and packet priority adaptation.

2.6.1 Traffic Adaptation

With the traffic adaptation approach, the rate at which traffic is poured into the network is adjusted using feedback mechanisms. The aim is to reduce congestion to support the delivery of priority traffic. Solutions adopting this approach may use either stateless or stateful mechanisms for QoS provisioning. Stateless Differentiation in Wireless Ad hoc Networks (SWAN) [3] is a stateless solution which uses probing to determine whether a path has sufficient resources to support the QoS requirements of a flow. A probe packet is sent towards the destination node to determine the available bandwidth on a path. Two feedback methods are used to facilitate the adaptation of an established flow. The first is an *additive increase multiplicative decrease* (AIMD) rate control algorithm. This adjusts the rate at which best-effort traffic is transmitted over a link. This traffic acts as a 'buffer zone' to absorb bursts of real-time traffic, thus the rate of best-effort traffic is adjusted to yield bandwidth to higher priority traffic. The second method is *explicit congestion notification* (ECN). Intermediate nodes set the ECN-bit in packet headers when a QoS requirement violation is detected. On receiving ECN-marked packets the destination node informs the source node to re-establish the flow along a path which meets the QoS requirements.

Another solution using stateless QoS provisioning is the scalable QoS framework proposed in [211]. The framework is designed to support QoS in large scale

MANETs containing thousands of nodes. It shares many features with SWAN, such as the use of both AIMD rate control and ECN. One key difference between this QoS framework and SWAN is that probing is not used due to the large size of the network. Instead, nodes proactively exchange bandwidth information.

In contrast to the above two stateless solutions, the QoS architecture proposed in [196] is a stateful solution. A QoS routing protocol interacts with the admission control module at each intermediate node. If resources are available a flow is admitted and resources are reserved. A rate control mechanism, distributed amongst the intermediate nodes, may be used during a session. Rate control requires intermediate nodes to know the generation rate of high-priority packets. This is learnt via periodic control packet exchange. The priority packet generation rate is used to determine whether the packet arrival rate is less than the generation rate. If so, each intermediate node can independently adjust the rate at which they transmit best-effort packets to support the priority flows.

The Adaptive reServation And Pre-allocation protocol (ASAP) [212] is another stateful solution. ASAP uses adaptive bandwidth reservations to support QoS in MANETs. Reservations are made in two phases. In the first phase a 'soft' reservation is made. In-band QoS signalling, whereby signalling information is carried in the IP options field of a data packet, is used to 'soft-reserve' resources. These resources can be used by best-effort traffic but cannot be committed to another reservation. The second phase begins once the destination node receives the reservation request: a 'hard' reserve control packet is sent back along the path to the source; each node receiving this packet forces out the traffic occupying the soft-reserved resources and commits the reservation. During the session, the destination node periodically transmits QoS information to the source node. The source node adjusts the traffic flow rate based on this information.

2.6.2 Path Adaptation

With the path adaptation approach, a node chooses a new path when the existing path no longer satisfies the QoS requirements. The Ad hoc QoS On demand Routing (AQOR) protocol [213] uses this approach. The authors of AQOR identify a number of features which should be incorporated into a MANET QoS solution. These include the instant detection of QoS violations and adaptation to the dynamic environment, both of which are embedded in the AQOR design. If a destination node observes excessive end-to-end delays or packet loss, it informs

the source node of the QoS violation. The destination node sends a *route update* control packet for this purpose. This control packet is also used to perform a QoS Route Discovery to find an alternative path which satisfies the QoS requirements. The source node then uses this new path to continue the session.

The QoS Aware Stable-path Routing (QASR) protocol [23] extends the ideas of AQOR by introducing an additional criterion, *signal stability*, for path selection. This is used with other QoS criteria to discover paths with sufficient signal strength, link stability, and resources. Signal strength and link stability are determined via periodic control packet exchange between a node and its one-hop neighbours. During a Route Discovery process, intermediate nodes forward the ROUTE REQUEST packets to a neighbouring node with a stronger signal stability. QoS violation detection during an established session leads to destination node-initiated feedback and adaptation in a similar manner to the AQOR protocol.

The QoS routing protocol in [162] uses a local recovery mechanism to respond to QoS violations close to where they occur. When a node detects a QoS violation it broadcasts an error message containing the bandwidth and delay requirements to downstream nodes. If a node can satisfy these requirements and has a path to the destination node it replies to the intermediate node which originated the error message and becomes part of the path. If a node cannot support the requirement it sends an error message upstream to the source node which then initiates a Route Discovery process.

2.6.3 Bandwidth and Packet Priority Adaptation

The third QoS adaptation approach dynamically alters the bandwidth and priority given to data packets based on network resource availability. Both source and intermediate nodes may perform the alterations. INSIGNIA [108] uses this approach. INSIGNIA, described in Section 2.5.3, is de-coupled from the underlying routing protocol and performs bandwidth and packet priority adaptations over any path discovered by the routing protocol. INSIGNIA supports two levels of bandwidth reservations: maximum-reserved and minimum-reserved. Intermediate nodes first attempt to support the maximum bandwidth requirement if sufficient resources are available. Otherwise they try to satisfy the minimum requirement. Established reservations can be 'scaled-up' and 'scaled-down' in response to network dynamics. However, if there is insufficient bandwidth to support a minimum-reserved flow the priority of the packets in the flow may be

downgraded to a best-effort delivery service. Packet priorities may be increased if sufficient resources later become available; if limited resource availability persists the packets may continue to receive the best-effort service.

There are two commonalities in the QoS solutions described in this section and the previous section. First, they apply adaptation along a single path between a pair of source and destination nodes. Second, they all require the co-operation of intermediate nodes: the solutions assume that intermediate nodes are trustworthy and that they always correctly execute routing and QoS operations. These assumptions are no longer valid in adversarial environments. One cannot always assume that the network is free of attacker nodes. Attackers may exploit the characteristics of the MANET environment for malicious reasons. In the next section security threats in MANETs are explored with a focus on the effects that the exploitation of the security threats may have on QoS.

2.7 Security Threats to QoS and Countermeasures

MANETs are vulnerable to a number of security threats, and many of these threats have direct implications on QoS. This section discusses these threats and examines how they affect network operations and QoS provision. Security threats can be classified into two broad categories: *passive* and *active*. In a passive attack a malicious node eavesdrops on communications, but it does not modify any data, or alter or impede the operation of a routing protocol. In an active attack a malicious node aims to modify or discard data, disrupt protocol operations, and/or prevent other nodes from participating in a network. Active attacks can be further divided into two categories: *routing disruption attacks* and *resource consumption attacks* [69]. The focus of this research is on active attacks, specifically routing disruption attacks. The following description of attacks is therefore restricted to routing disruption attacks. All of the described attacks focus on the network layer; attacks on other layers are outside the scope of this research.

2.7.1 Routing Disruption Attacks

Depending on the types of messages attacked, routing disruption attacks can be divided into four categories: *attacks on routing messages*, *attacks on data packets*,

attacks on routing messages and data packets, and *attacks on QoS signalling messages*. This focus of this section is on the attacks on data packets, i.e., packet forwarding attacks, and the attacks on QoS signalling messages. These are the types of attacks that this research addresses. The attacks on data packets are the *blackhole* and the *grayhole* attacks. The attack on QoS signalling messages is the *denial of QoS request* attack. Counter-measures to these attacks are described in the following section.

Blackhole Attack In the blackhole attack a malicious node lures data packets towards itself, and on receiving them it proceeds to discard them [25]. The blackhole attack operates in two stages. The first stage occurs during a Route Discovery process. A blackhole attacker subverts this process in an attempt to situate itself on a path between a source node and a destination node. On receiving a ROUTE REQUEST packet, a blackhole attacker changes the value of the hop-count field to a small value, e.g., 1. This makes the path through itself appear shorter than it is. Falsifying routing metric values is an effective method to make the malicious node appear to be a preferred router for data packet forwarding [70]. Rather than forwarding the ROUTE REQUEST packet on towards the destination node, as specified by the routing protocol, the blackhole attacker immediately sends a ROUTE REPLY packet back to the source node. This aims to make this the first ROUTE REPLY packet to be received by the source node. The source node may now be duped into selecting the path containing the blackhole attacker, as the short hop-count of the path appears favourable. The second stage of the attack begins when the source node transmits data packets along the path containing the attacker. The blackhole attacker proceeds to drop any data packets it receives. The dropping of all data packets will have a severely detrimental effect on QoS.

Grayhole Attack The grayhole attack is a variation of the blackhole attack. In this attack a node selectively drops received packets. For example, it may drop packets depending on packet type (control or data packets), or source node IP address, or even some range of IP addresses [178]. A node may also drop packets probabilistically [25].

Denial of QoS Requests A malicious node modifies or discards any reservation request messages it receives to delay or prevent fulfilment of a resource reservation between a source node and a destination node [118]. A modified request may result in too few resources being reserved to support a data flow's

QoS requirements. Supporting data flows requiring priority service will therefore become increasingly difficult. This attack is focussed on QoS signalling and does not affect the operations of the underlying routing protocol [57].

2.7.2 Countermeasures to Routing Disruption Attacks

The solutions proposed to counter the above mentioned attacks make use of either established security primitives or the characteristics of the MANET environment. They may take protective, detective, and/or reactive approaches to counter the attacks.

Countermeasures to the Blackhole Attack A blackhole attack may be countered using a reactive approach, as it is difficult to prevent attacks on packet forwarding [215]. An example reactive approach is to detect data packets as they are discarded and to react accordingly. Promiscuous overhearing can be used for this purpose. Promiscuous overhearing enables those nodes neighbouring the blackhole attacker to detect whether or not the packets they have transmitted to it are being forwarded [123]. (Further details on the operation of promiscuous overhearing to detect misbehaviour can be found in the section below on countermeasures to the denial of QoS request attack.) Other proposals to counter the blackhole attack have suggested to use multiple paths between a source node and a destination node to forward data packets [150, 193]. With multiple paths, if there are blackhole attacks along one path packets may still reach their destination along other paths. Whilst the use of multiple paths does not prevent or detect misbehaviour it does reduce the effects of an attack on communication reliability [38].

Countermeasures to the Grayhole Attack The techniques described to counter the blackhole attack can also be used to counter the grayhole attack.

Countermeasures to the Denial of QoS Request Attack Promiscuous overhearing can be used to detect the denial of QoS requests attack [118]. It enables a node to determine whether or not a downstream node has modified or dropped a data packet that they have agreed to forward. This detection mechanism works as follows. A node buffers a copy of a packet that it is forwarding. On transmission of the packet the node switches to the promiscuous receive mode. This allows the node to overhear the retransmission of the packet by the downstream node. It then compares the overheard packet with the buffered packet to determine whether or not any malicious modification has taken place. If the node

cannot overhear any retransmission it can infer that the downstream node has discarded the packet. In the case of a modified or a dropped packet the node will notify the source node which will need to retransmit that packet along a different path.

2.7.3 Countermeasures to Selfish and Malicious Behaviours

A number of security solutions exist to mitigate the effects of selfish and malicious behaviours by aiming to stimulate co-operation between nodes during the data packet forwarding phase of routing protocol operation. A selfish node is unwilling to forward packets in order conserve its own resources. A malicious node refuses to forward packets in order to disrupt network operations. The occurrence of selfish or malicious behaviours during data packet forwarding will cause packet loss, reducing the QoS received by a packet flow. Moreover, without providing security, selfish and malicious nodes may receive a superior service whilst providing or denying services to other nodes which faithfully perform network operations [18]. Countermeasures to selfishness and misbehaviour can largely be classified into two categories: *reputation-based schemes* and *credit-payment schemes*.

2.7.3.1 Reputation-Based Schemes

A number of reputation-based schemes have been proposed to stimulate node co-operation in MANETs. Reputation-based schemes can be divided into *detection schemes* and *prevention schemes*. Watchdog and Pathrater [123] is an example detection scheme, whilst CONFIDANT [19] and CORE [127] are example preventative schemes.

Watchdog and *Pathrater* is an exemplary detection scheme. Watchdog uses promiscuous overhearing to detect whether a downstream node forwards the packets it receives. Pathrater uses Watchdog's knowledge of node misbehaviour to maintain a rating for each path. This rating, combined with link reliability information, is used by the routing protocol to pick more reliable paths. One of the drawbacks of this scheme is that misbehaving nodes may be able to save their resources by being excluded from routing operations whilst still having their own packets forwarded; thus their misbehaviour may positively benefit them [74]. This occurs when a misbehaving node is detected and it is no longer given packets to forward, i.e., it no longer serves as a router for other nodes' packet flows; but the

packets that the misbehaving node originates are still forwarded by other nodes. Another drawback is that it is difficult to differentiate collisions and misbehaviour as a cause of packet loss [123, 182, 219]. For example, it is not considered a misbehaviour if a node does not retransmit a packet when a collision prevents the delivery of the packet to the next-hop node [38].

Preventative solutions are different from detective solutions in that they attempt to avoid node misbehaviour, rather than waiting for an attack to occur before reacting to it. Two exemplary preventative reputation-based solutions are CONFIDANT and CORE. CONFIDANT (Cooperation of Nodes: Fairness in Dynamic Ad hoc Networks) has four components to detect and respond to node misbehaviour: monitor, trust manager, reputation system, and path manager. The monitor detects node misbehaviour using promiscuous overhearing. The trust manager uses this information to inform other nodes of the misbehaviour. This information is used by the reputation system to rate nodes. Nodes which should be avoided are added to a blacklist. Node ratings and the blacklist are exchanged with other trusted nodes. In other words, negative information about nodes is propagated through the network. The path manager uses the list of nodes to rate paths according to their reputation. It also uses the blacklist to delete paths containing the malicious nodes.

CORE (Collaborative Reputation mechanism) uses promiscuous overhearing to identify misbehaving nodes. A node combines its own observations with reports received from other nodes. Other nodes only propagate positive information about nodes which correctly participate in routing operations. This contrasts the approach used in CONFIDANT which transmits negative information about other nodes (e.g., blacklists). Node reputations are calculated from the observations and reports. Nodes will be isolated from the network if their reputations are less than a threshold value.

Two main issues can be identified with the approaches of CONFIDANT and CORE. First, a high control packet overhead may be experienced when disseminating rating information through the network [217]. Second, the trustworthiness of the distributed rating information cannot be assured [182]. For example, malicious nodes may exploit the mechanisms by providing false accusations of misbehaviour [219]. One general drawback with preventative solutions is that unless a prevention strategy is perfect it is likely that there may exist a way around them [170]. Additionally, packet forwarding attacks, such as the blackhole attack, are

difficult to prevent, and a reactive mechanism may be best to counter the effects of the attack [215]. It can therefore be argued that detecting and responding to attacks is essential [18].

2.7.3.2 Credit-Payment Schemes

Credit-payment schemes are the second approach to stimulating node co-operation in MANETs. Credit-payment schemes are preventative mechanisms. Nuglets [20] is an exemplary credit-payment mechanism. Co-operation between nodes is stimulated by trading 'nuglets', a virtual currency. Nodes must pay other nodes for the services that they use, i.e., packet forwarding. In other words, intermediate nodes are rewarded with nuglets for forwarding packets. Paying for services motivates nodes to offer their own services so that they too can earn nuglets. One of the shortcomings of this approach is that it is necessary to have a tamper-resistant security module for nuglet storage. This is necessary so that nodes cannot maliciously increase the nuglets it owns. Another shortcoming is that some nodes may not have a chance to earn nuglets if they are not selected as routers for other nodes' traffic [73]; and this means that they may not have enough nuglets to pay for their own packets to be forwarded.

The Credit Based Routing (CBR) algorithm [5] overcomes the problem of some nodes being unable to earn credits. CBR separates the policing of best-effort forwarding and priority forwarding. It achieves this using a two-layered forwarding service: free best-effort forwarding and priced priority forwarding. Free best-effort forwarding enables nodes which have not earned credits to have their packets forwarded using a best-effort forwarding service. This overcomes one of the shortcomings identified with the Nuglets approach. Nodes requiring a priority forwarding service have to pay for that service. Intermediate nodes earn credits for providing a priority service. To earn credits an intermediate node must forward a priority packet ahead of all of the best-effort packets it may have queued. The next-hop node on the path of the priority packet uses promiscuous overhearing to confirm this.

The above reputation-based schemes and credit-payment schemes address the issue of security threats caused by selfish nodes, but they have not considered the issue of QoS. Another technique to provide security or QoS is to use multiple paths for data packet transmissions.

2.8 Multi-Path Routing

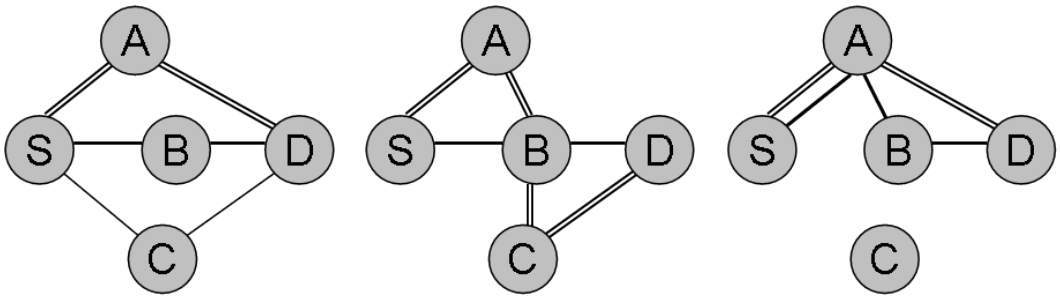
Multi-path routing in wireless networks was possibly first proposed in [49]. Since then, a large number multi-path solutions have been proposed, for example, to decrease the effects of unreliable wireless links and changes in topology [190], or to support a packet flow's QoS or security requirements. In this section related work on multi-path routing is reviewed. This includes the disjointedness of multiple paths, using multiple paths for QoS, and using multiple paths for security.

2.8.1 Path Disjointedness

One important issue in multi-path routing is the selection of paths which offer the highest level of redundancy at the network layer. Selecting paths with the highest redundancy aims to enhance the reliability of the paths. One method is to look at how multiple paths are related to one another. The paths with the highest level of redundancy are those which have the lowest level of interdependence. The level of interdependence among multiple paths can be measured by the level of *disjointedness* [130]. Disjointedness is a measure of the redundancy between paths. The greater the disjointedness between paths the lower the interdependence; greater disjointedness means that the paths offer a larger level of redundancy. Conversely, the lower the disjointedness the greater the interdependence; such paths offer lower redundancy. The level of disjointedness is therefore an important factor in determining the potential quality of a set of paths.

There are three levels of path disjointedness: node-disjoint, link-disjoint, and non-disjoint. These path types are shown in Fig. 2.1. The highest level of disjointedness is node-disjoint (shown in Fig. 2.1(a)). Node-disjoint paths do not share any nodes (except the source and the destination nodes) or links. This path type offers the highest aggregate resources and provides the highest fault tolerance [130]. Node-disjoint paths offer the highest level of reliability: a transmission between a source node and a destination node will fail only when all of the in-use node-disjoint paths fail; and the probability for all node-disjoint paths to fail is less than the probability that any individual path fails [109, 168].

Link-disjoint paths share no common links, but some nodes participate in more than one path (as shown in Fig. 2.1(b)). Link-disjoint paths are sometimes known as 'edge-disjoint' paths. The benefit of using link-disjoint paths is that if a single link fails it will cause only one path to fail. The failure of a shared node,



(a) Node Disjoint: Paths SAD, SBD and SCD have no common nodes.

(b) Link Disjoint: Paths SABCD and SBD have node B in common.

(c) Non-Disjoint: Paths SAD and SABD have link SA and node A in common.

Figure 2.1: Levels of Disjointedness Between Multiple Paths

however, can lead to the failure of multiple paths. The level of interdependence among nodes in this case is higher than that of node-disjoint paths.

Non-disjoint paths have the highest level of interdependence and lowest redundancy amongst the three path types. Non-disjoint paths contain shared nodes and shared links. If a node or a link fails in a non-disjoint path it can cause all paths which share that node or link to fail. For example, if node A in Figure 2.1(c) fails, both paths from the source node S to the destination node D will fail. This means that non-disjoint paths provide lower aggregate resources than link-disjoint and node-disjoint paths, as both nodes and links are shared.

Whilst selecting paths with the highest disjointedness aims to maximize the independence and the redundancy between paths, it is possible that the multiple paths may interfere with one another's transmissions [117, 206]. This is known as the *route coupling problem* [155, 197]. This problem occurs when paths which are disjoint at the network layer are not disjoint at the physical layer [12]. For example, the radio signals of nodes on node-disjoint paths may overlap at the physical layer. Such interference may affect the length of time it takes to forward a packet to the next-hop node, e.g., as a consequence of collisions and retransmissions.

Path disjointedness can be calculated at the source node or the destination node. The Disjoint Multipath Source Routing (DMPSR) protocol [204] is one example of a protocol which calculates path disjointedness at the source node (other example protocols include [151, 199]). A source node may receive a number of Route Reply (RREP) packets in response to the Route Request (RREQ) packets it transmits in a Route Discovery process. It is from these RREP packets that the source node learns the available paths to the destination

node. From these available paths the source node selects those which are disjoint. The DMPSR paper [204] does not state the path disjointedness level sought, but it is inferred that maximally disjoint paths are preferred. A maximally disjoint path is one which shares the lowest number of intermediate nodes with the primary path [112]. Additionally, this paper does not describe the process of determining the disjointedness between paths.

There are several solutions which perform path disjointedness calculation at the destination node [103, 110, 111, 112]. The Split Multi-path Routing (SMR) protocol [103] is one example protocol which does this. During a Route Discovery process a destination node may receive a number of RREQ packets. The first received RREQ is immediately returned to the source node. The path it traverses becomes the primary path. Rather than replying to each of the following received RREQs, the destination node collects them until the expiration of a timeout. This is so it can learn as many incoming paths as possible. The destination node then compares the paths contained in the received RREQs to find paths which are maximally disjoint with the primary path (although this paper [103] does not describe how path disjointedness is calculated).

2.8.2 Multiple Paths for QoS

Based on the literature, multi-path routing has been used to increase reliability in QoS provisioning. It has been used (1) to reserve bandwidth over multiple paths, and (2) to route packets over multiple paths in adaptation to network conditions. The QoS Multi-path Source Routing (QoS-MSR) protocol [179] uses the first approach. It uses a Multi-path Bandwidth Splitting Reservation (MBSR) mechanism [198] to harness multiple paths' bandwidth to satisfy a single reservation request. MBSR splits a request into several smaller requests, each wanting to reserve bandwidth on a single path. The benefit of this technique is that the split reservation aims to achieve a higher probability of success than a reservation along only a single path. This is because the amount requested from each path is much smaller than it would be if the overall bandwidth request had to be satisfied by a single path. Data packets are allocated to one of the paths using weighted round-robin scheduling. The work shows how the redundancy of multiple paths may be exploited to support QoS.

QoS Multi-path Source Routing-Stateless Differentiation in Wireless Ad hoc Networks (QMSR-SWAN) [221] combines the QoS-MSR protocol [179] (including

the MBSR mechanism [198]) with SWAN [3] (described in Section 2.6.1) to support reservation-based QoS across multiple paths. SWAN is used to rate-control the best-effort traffic so that real-time traffic is given priority. The real-time traffic is transmitted over multiple reserved paths using the QoS-MSR protocol.

Ticket Based Probing (TBP) [26] is a QoS routing scheme which combines resource reservations and multi-path routing. TBP selects paths which satisfy delay or bandwidth requirements. A three-level redundancy scheme is used to determine which paths to use and which to reserve resources on. With *first-level redundancy* multiple preferably node-disjoint paths are used to forward duplicated data packets. This provides the greatest redundancy for critical connections requiring QoS. *Second-level redundancy* establishes multiple paths but only transmits data packets along one path, the primary path. The secondary paths are used when the primary path fails. Best-effort traffic can be transmitted along the unused paths. Second-level redundancy is used for ordinary connections which can handle a certain level of disruption. Finally, *third-level redundancy* is used by ordinary connections not requiring QoS. This is similar to the previous level, except that resources are not reserved along the secondary paths. If the primary path fails reservation messages need to be transmitted along the secondary paths to reserve resources. In this instance, and also when any reserved paths break, packets will only be transmitted on a best-effort basis whilst a reservation is (re)established.

INORA [36] also performs resource reservations over multiple paths. INORA is a QoS support mechanism which couples INSIGNIA, which it uses for resource reservations, with the Temporally-Ordered Routing Algorithm (TORA) [152, 153].[2] INSIGNIA provides feedback to TORA during a Route Discovery process to select paths satisfying the QoS requirements. This contrasts INSIGNIA's original approach of attempting resource reservations over any discovered path. In addition to this modified version of the approach adopted by INSIGNIA, the path adaptation approach is also used: if a path cannot support a resource reservation request a new path with the necessary resources is sought; a flow may be split across multiple paths if the paths' aggregated bandwidth satisfies the resource requirement.

[2]TORA is an adaptive routing protocol which uses both proactive and reactive routing techniques over multiple paths.

Multi-Path Dynamic Source Routing (MP-DSR) [110] uses the second approach, where packets are routed over multiple paths in adaptation to network conditions, to increase reliability in QoS provisioning. MP-DSR estimates the link availabilities of all the links along a path during the route discovery process, and uses these link availabilities to estimate the end-to-end reliability of the path. Based on the estimated end-to-end reliability of each path to a destination node and an *end-to-end reliability* requirement specified by an application, MP-DSR selects the number of paths which should be used for data packet forwarding. Thus path selection is made in response to network and path conditions. MP-DSR does not prescribe a particular packet allocation scheme.

Another approach to multi-path routing is to establish multiple paths at intermediate nodes in addition to the paths maintained between the source and the destination nodes. The aim of this is to increase the fault tolerance of data packet transmissions. With the Caching and Multipath Routing Protocol (CHAMP) [194], each intermediate node caches two paths to the destination node. Intermediate nodes also cache data packets to be forwarded. In the event that a downstream node fails to forward a data packet, an upstream node will transmit a cached copy of the packet along one of the cached paths. However, if the upstream node does not have a cached packet, or does not have an alternative path to the destination node, the original source node of the packet will need to perform a Route Discovery operation. The source node will not be notified of the failed delivery until each upstream node has searched its cache to find that it does not have a copy of the packet. As a result there may be an increase in delay between the failed packet delivery and the initiation of a Route Discovery process.

The k-Redundant Multipath Routing (k-RMR) protocol [54] establishes multiple paths at intermediate nodes as part of a Route Discovery process. The number of redundant paths established at intermediate nodes can be less than or equal to the number of primary paths. Here 'primary paths' refers to multiple end-to-end paths between a source node and a destination node. k-RMR attempts to establish node-disjoint primary and redundant paths.

In the Segment Adaptive Multi-path Routing (SAMR) protocol [214], to increase delivery reliability intermediate nodes construct multiple paths to a destination node and duplicate data packets along them. To reduce potential congestion, intermediate nodes suppress duplicate data packets that they have previously forwarded. SAMR's authors remark that the extra bandwidth consumed by packet duplication is a fair sacrifice for a better QoS for real-time services.

2.8.3 Multiple Paths for Security

The solutions discussed in the previous section assume that all the nodes are trustworthy and that they always execute routing and QoS operations correctly. However, as mentioned earlier, in an open MANET environment this assumption may not always be true. In light of this, solutions have been proposed which combine security provision with multi-path routing.

Some security requirements can be considered in terms of fault tolerance (which can include security attacks). Fault tolerance is the ability of a system to handle faults such that service failures do not result [185]. Fault tolerance can be divided into two categories [14, 172]: detect-and-recover and forward-recovery. Detect-and-recover, also known as 'cold-standby', triggers error handling only after the detection of an error. This form of fault tolerance may therefore be suitable for applications where some degree of service disruption can be tolerated. Forward-recovery, also known as 'hot-standby', uses redundancy to handle faults without service disruption. Forward-recovery is considered preferable for real-time/critical applications where disruption cannot be tolerated. The focus of this research is on supporting priority packets with minimum service disruption; thus the following discussion is focussed on forward-recovery techniques.

There are two approaches to forward-recovery [14]: redundant routing and dispersity routing. Redundant routing is the process of transmitting redundant (duplicated) data packets over multiple paths. This aims to make the multi-path routing process resilient to path failures. For example, if m paths are used, data packets may still be delivered to the destination node even if $m-1$ paths fail. The main cost of this approach is an increased overhead arising from the duplicated data packet transmissions. Dispersity routing is the process of splitting a message which is normally contained within a single packet into multiple packets; these multiple packets are then transmitted over multiple paths. The message is split in such a way that each part of the message contains extra bits. The extra bits

are calculated in such a way that the original packet can be reconstructed given a subset of these packets. There are two approaches to dividing a message: diversity coding [8] and Rabin's algorithm [163]. Diversity coding is an error control-based approach for the transparent self-healing of networks. Rabin's algorithm is designed for the fault-tolerant and efficient transmission of information over networks. These two mechanisms take different approaches to divide a message and add redundancy into the divided segments, but both aim to increase fault-tolerance.

Redundant Source Routing (RSR) [199] combines redundant routing and dispersity routing to provide QoS and fault tolerance for real-time services. The scheme aims to increase the chances of successful data packet deliveries by using two node-disjoint paths for data packet transmissions: a primary path for an original packet and a secondary path for a redundant (duplicated) packet (the dispersity routing scheme used in RSR replicates each data packet in its entirety). Two agents are used to handle the creation and suppression of duplicate packets. The Packet Duplication Agent resides at a source node and duplicates outgoing data packets. The Duplicate Packet Filter, which resides at the destination, removes any received duplicated packets. Removing duplicated data packets is necessary as they can affect metrics and/or feedback mechanisms, such as the acknowledgement mechanism in TCP. If multiple paths are not available RSR reverts to a single path. However, if two paths are available RSR will always attempt to use them.

The work in [189] presents a framework for multipath routing in the presence of frequent topological changes. The framework uses the dispersity routing approach with diversity coding. Topological changes, in addition to link failures and other MANET characteristics, make the transmission of time-sensitive data a challenging issue. By splitting each data packet across multiple node-disjoint paths the probability of them arriving at the destination node within a reasonable time frame is increased. The number of smaller packets that the original packet is divided into depends on the number of paths to be used for transmission. These paths are arranged in order of 'best' to 'worst' based on a probability of successful delivery, which is defined in the work.

The Secure Message Transmission (SMT) protocol [151] uses the dispersity routing approach. The approach, named as 'data dispersion', is used to cope with attacks on data packet forwarding. Data dispersion uses Rabin's Algorithm

to divide each message into smaller messages, each with sufficient redundancy to recover the message from a subset of the smaller messages. These smaller messages are put into separate packets and are are transmitted across multiple paths. If an insufficient number of the smaller packets arrive at the destination node, the source node can retransmit them.

The work in [181] extends the SMT protocol with a misbehaving node detection mechanism. It detects misbehaviour using acknowledgements (ACKs) and binary search probing. Destination nodes transmit ACKs to the source node for each received data packet. If the number of missing ACKs exceeds a threshold the probing phase is triggered. In this phase, probes are piggy-backed on the data packets. Each node receiving a probe must send an ACK to the source node. If an attacker drops the probe/data packet, the downstream node will not receive the probe and will therefore not send an ACK. The source node can then identify the problem node. Trust levels are maintained for each link, and these are updated during probing. Links with a low trust level are avoided during path selection.

The work in [114] takes a different approach to multi-path routing and focuses on selecting paths which are the most likely to survive in the presence of attackers. It uses a fuzzy logic-based scheme to estimate a Path Survivability Level (PSL) for each path, and selects paths which have the highest PSLs. This path selection process can be summarised as follows. Control packets are periodically transmitted to collect nodes' reputations, energy levels, and certificate expiration time (as well as other criteria) along paths between a source node and a destination node. These data are fed into a fuzzy inference system which produces a PSL. As new control data come in the PSLs may change, and if they do a new set of paths with the highest PSLs will be selected. In this way paths are selected dynamically in adaptation to the underlying network conditions.

Security has been largely ignored in many routing protocols but redundant routing and dispersity routing have taken a step toward rectifying this. It enables some degree of loss to be tolerated whilst still enabling data packet deliveries to a destination node. The authors of some of the above protocols have, as already stated, recognised the positive effect that redundancy strategies can have on network performance; the packet delivery percentage can be increased at the cost of increased bandwidth consumption. Moreover, redundancy allows some of the effects of malicious behaviour and transient failures to be mitigated whilst

supporting the aims of QoS.

2.9 Existing Efforts on Integrating Security and QoS

It has been pointed out that security should be specified as a dimension of QoS [81, 116, 183] and that security is a desirable feature in QoS routing protocols [4]. It has also been pointed out that the traditional binary model of security, where something is secure or insecure, is outdated [116, 128], and that security should be specified using a range of values [40, 62, 81, 116, 144]. In light of these declarations, there have been a number of proposals on integrating security and QoS. The Variant Security Service [81] introduces a range of security services, similar to the way QoS offers levels of service. Using levels of security allows a system to adapt gracefully to fluctuating resource availability; moreover, this allows a trade-off to be sought between the security and QoS requirements, and this is something that many researchers advocate [4, 11, 46, 59, 68, 115, 144, 215]. The Variant Security Service has been extended with a translation matrix to map simple requests for 'low', 'medium', or 'high' security to specific security metrics and parameter settings [80]. The Tunable Security Service [115] has similarities with the Variant Security Service: a set of possible security configurations are specified which can be selected and adapted dynamically at run-time. The Quality of Protection (QoP) framework [144] applies the concept of the Tunable Security Service to achieve a balance between QoS and security. A Quality of Security Service (QoSS) costing framework [183] estimates the costs of security provision on QoS.

The above proposals are early attempts to integrate security with QoS in infrastructure-based networks. Their main focus is on the cost of security provision and its effects on QoS. There is little research which addresses the issue of achieving QoS in the presence of security attacks. Of what little has been done, the majority of the work focuses on the integration of security and QoS for infrastructure-based networks. A risk-aware QoS/QoP optimization algorithm [210] uses the idea of QoP to adapt dynamically to the network risk-level. This aims to balance the security and QoS requirements. The QoS^2 framework [186] also adjusts the security level in response to the network risk-level in order to achieve an acceptable QoS. Security Service RSVP [207] extends the Resource

ReserVation Protocol (RSVP) with QoSS. It does this by combining the concepts of the Variant Security Service—a component of QoSS [81, 183]—with the RSVP protocol to negotiate a suitable level of QoSS. The algorithms proposed in [24] model the delay cost of using confidentiality, integrity, and authentication to optimize security decisions with respect to QoS. The proposal in [176] aims to minimize the delay cost of secure handover for mobile nodes requiring QoS in one-hop infrastructure-based wireless networks.

Much of the related work for MANETs focuses on examining the time-costs of supporting confidentiality, integrity, and/or authentication, and the effects of these costs on the end-to-end packet delays [52, 149, 175]. Other related work in this domain is to use cryptographic techniques to protect QoS signalling information [55, 118, 172], to make the QoS signalling process resistant to DoS attacks [57], and to secure QoS-guided route discovery [67]. Multi-path routing has also been used with different forward-recovery strategies to support data packet deliveries in the presence of packet forwarding attackers [114, 151, 181, 189, 199].

2.10 What is Missing?

Providing QoS is a challenging issue in a MANET containing malicious nodes performing data packet forwarding attacks. The threat of attacks on the data packet forwarding process raises the following question: how can the effects of these attacks be mitigated whilst supporting the QoS requirements of a priority packet flow? Addressing this question is a challenging issue. The foci of existing solutions which address both security and QoS in MANETs have predominately been on (1) the delay cost of applying cryptographic primitives for the purpose of confidentiality, integrity, and authentication, (2) using cryptographic primitives to secure QoS signalling information, and (3) using multi-path routing to provide fault-tolerance. Whilst the approaches in (1) and (2) address integrated security and QoS they are still vulnerable to data packet forwarding attacks: a malicious intermediate node can drop a data packet regardless of the cryptographic primitives used to secure the payload and/or header data. An issue with the multi-path routing approaches is that they generally use a fixed number of paths and do not consider how this affects the dynamic MANET environment: this resource-expensive strategy may be too costly if the network conditions and

threat-level change such that a single path may satisfy a priority packet flow's QoS requirements. A new approach is therefore required to make a communications session resilient to attacks on priority data packets whilst still supporting their QoS requirements in dynamic MANET conditions. The following section outlines how this new approach may be realized.

2.11 The Best Way Forward

Based on the review of the related literature, the following ideas are generated for a new solution to achieve QoS in MANETs containing packet forwarding attackers. An adaptive solution should be provided which combines a multi-path routing mode and a single-path routing mode. Adaptation is necessary to respond to dynamic MANET conditions and to optimize priority packet deliveries: when the attacker ratio is high, priority data packets should be transmitted over multiple, maximally-disjoint paths; when the attacker ratio is low, data packets should be transmitted over a single, bandwidth-reserved path. To realise this solution, a routing architecture which integrates the single-path and multi-path modes of priority data packet forwarding should be provided. As part of the multi-path mode, a path selection mechanism is needed to determine and to select maximally-disjoint paths.

2.12 Concluding Remarks

The contributions of the chapter are three-fold: first, it has provided background information on the applicability of MANETs to a range of scenarios; second, these scenarios were analysed and a requirement for security and QoS provision was identified; third, the related literature was surveyed for the state-of-the-art approaches to QoS, security, and integrated security and QoS. Based on this survey, some new ideas that may help to achieve QoS in a malicious MANET environment have been proposed.

The next chapter presents the building blocks to be used to implement the novel ideas, as well as the evaluation methodology used to evaluate the novel ideas.

Chapter 3

Building Blocks and Evaluation Methodology

3.1 Chapter Introduction

This chapter describes the building blocks of the 2-Dimensional Adaptation Architecture (2-DAARC) and the evaluation methodology used to evaluate its performance.

Section 3.2 first presents a discussion of single-path routing protocols to determine which one to use in the design of 2-DAARC. Following this the chosen routing protocol, the Dynamic Source Routing (DSR) protocol, is described in detail along with the INSIGNIA QoS framework. Section 3.3 presents the cryptographic systems and primitives used in 2-DAARC. Section 3.4 discusses experimental, mathematical, and simulation-based investigation methodologies, and justifies the choice of simulation as the investigation methodology used in this thesis. Section 3.5 describes the simulation environment, simulation modelling, and the validation of the simulation model. Finally, Section 3.6 describes how statistically significant results are obtained from the simulation-based performance evaluation.

3.2 Routing Protocols and QoS Signalling Systems

This section firsts presents a comparison of the AODV routing protocol and the DSR protocol, two reactive single-path routing protocols, to justify the choice of DSR as the routing protocol to be extended in 2-DAARC. Following this the workings of the DSR protocol are described in detail. Finally the INSIGNIA QoS framework, which is another 2-DAARC building block, is described.

3.2.1 Comparing Reactive Routing Protocols

2-DAARC requires a multi-path routing functionality which could be provided using a multi-path routing protocol. However, the routing protocols which have either been published or are currently under consideration by the Internet Engineering Task Force (IEFT) MANET Working Group (MANET WG) are all single-path protocols [78]. To make 2-DAARC conform to international standards the multi-path routing functionality is built over a single-path protocol which is recognised by the IETF MANET WG. For this reason this section discusses these single-path routing protocols to determine which protocol best suits the 2-DAARC design requirements (discussed in Section 5.4).

2-DAARC uses path redundancy to maintain path availability and to provide QoS. It is therefore important to minimise the level of additional or unnecessary traffic in the underlying network when choosing a routing protocol for 2-DAARC. The proactive routing approach (described in Section 2.4.1) requires periodic exchanges of local and global control packets, and the overhead generated from these exchanges can be much greater than the reactive routing approach [17, 87]. Moreover, this overhead may further increase as mobility increases. In contrast, with the reactive routing approach, control packets are only exchanged locally or when a path is required or broken. Therefore the reactive approach may generate a lower level of routing overhead and may potentially deliver a greater number data packets than the proactive routing approach, particularly when mobility increases [17, 32, 87]. Based on these considerations a reactive source routing approach is chosen for use in 2-DAARC.

AODV and DSR are the only two reactive routing protocols published by

	AODV	**DSR**
Routing Approach	Hop-by-Hop	Source Routing
Periodic Messages	*Hello* Messages	No
Route(s) Maintained In	Route Table	Route Cache
Multiple Routes Stored	No	Yes
Routing Metric	Shortest and Freshest Path	Shortest Path or Next Route In Cache

Table 3.1: A Comparison of the AODV and the DSR Routing Protocols.

the IETF MANET Working Group (at the time of writing) [78].[1] (A general description of each of these protocols is given in Section 2.4.2.) As summarised in Table 3.1, the comparison of the two routing protocols is based on the following criteria: routing approach, control packet overheads and storage of discovered paths, and routing metrics.

AODV and DSR uses two different approaches to reactive routing. AODV uses a *hop-by-hop routing* approach whereas DSR uses a *source routing* approach. With the hop-by-hop routing approach each intermediate node between a source node and a destination node maintains a routing table. The table contains the addresses of neighbouring nodes. It also contains the next-hop node on the path to a given destination node. Hop-by-hop routing therefore requires intermediate nodes to dedicate resources for the creation and maintenance of routing table entries. With the source routing approach, the entire path from source to destination is carried in the header of each data packet. Each intermediate node is not required to maintain routing table entries. Instead each intermediate node checks the source route to determine which next-hop node it should forward the packet to. A further benefit of the source routing approach is that all nodes participating in a path are able to learn and store that path by copying it from the header of a data packet. In this way, each intermediate node is able to learn a path to all of the other nodes participating in the source route. The source routing approach is fully on-demand as control packets are only exchanged when a path is required from a source node to a destination node. However, the source route grows with the number of nodes participating in the path, and this overhead is carried in every data packet [87]; the overhead of addressing in hop-by-hop routed data packets is constant, as they only contain the source node, destination node, and

[1] A third reactive routing protocol, Dynamic MANET On-demand (AODVv2) Routing [158], is currently under consideration by the MANET WG. This protocol is still being worked on by the MANET WG, whereas the work on the original AODV and DSR proposals is complete.

next-hop node addresses.

Although AODV is a reactive routing protocol, it still relies on the periodic transmission of local control information in the form of *hello* messages. *Hello* messages are only broadcast by intermediate nodes participating in data packet forwarding. In other words, idle nodes do not transmit *hello* messages. *Hello* messages are broadcast to nodes within the one-hop wireless neighbourhood to determine local connectivity. If a reply is not received from a neighbouring node in response to a *hello* message, it is assumed that the link is broken. A link failure message is subsequently broadcast to neighbouring nodes to inform them of the broken link. A consequence of transmitting *hello* messages is that AODV transmits more control messages than DSR [17]. DSR does not transmit local control information.

The number of control packets transmitted by AODV and DSR during a Route Discovery process can differ significantly. The number of control packets transmitted is closely linked to the number of paths which can be stored at a node. With AODV only one path to a destination node is stored by the source node. AODV's Route Discovery process therefore must be performed each time the in-use path breaks. In contrast, DSR maintains multiple paths to a destination node. Multiple paths may be discovered during a single Route Discovery process. The purpose of this is to offset the expensive cost of the Route Discovery process (the cost is a consequence of broadcasting packets). The multiple paths are stored in a *Route Cache*. When the in-use path fails an alternative path may be selected from the Route Cache if it is not empty, otherwise a Route Discovery process triggered.

The routing metrics used for path selection in AODV and DSR have both similarities and differences. The routing metrics are the path *hop-count* and path *freshness*. AODV and DSR both use the hop-count metric during path selection. The length of a path is expressed in terms of its *hop-count*. A path with the shortest length is selected. If two paths have the same hop-count the first path to have been discovered will be selected. Unlike the DSR protocol, AODV is also able to determine the freshness of a path. This is achieved through the use of sequence numbers: the greater the sequence number the fresher the path.

A performance comparison of the worst-case routing time complexities of AODV and DSR is undertaken in [1]. A summary of the findings is given in Table 3.2. The theoretical time complexity of the Route Discovery and the Route

Reactive Protocols	**TC[RD]**	**TC[RM]**	**CC[RD]**	**CC[RM]**
AODV	$\mathcal{O}(D)$	$\mathcal{O}(D)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$
DSR	$\mathcal{O}(D)$	$\mathcal{O}(D)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$

TC = Time Complexity, CC = Communication Complexity, RD = Route Discovery, RM = Route Maintenance, D = Network Diameter, N = Number of Nodes in Network

Table 3.2: A Comparison of AODV and DSR Routing Protocol Complexity.

Maintenance procedures for both protocols is $\mathcal{O}(D)$, where D is the diameter of the network. The communication complexity of the route discovery and the route maintenance procedures for both protocols is $\mathcal{O}(N)$, where N is the total number of nodes that comprise the network. These worst-case complexities assume that nodes have no initial communicatory relationship with the source node and the destination node. From the table, it can be seen that AODV and DSR exhibit an equal worst-case level of performance.

However, DSR has some features which are of interest to the design of 2-DAARC (presented in Chapters 5 and 6). First, DSR can find and store multiple paths to a destination node during a single Route Discovery process. This feature makes it attractive for extension to provide a multi-path routing capability. Such a feature is not provided in the standard implementation of AODV. Second, DSR's source routing approach can be utilized during secondary path selection: the source node knows the complete paths to the destination; and these paths can be compared to determine whether they share any nodes or links, i.e., their disjointedness. The disjointedness criteria can be used in conjunction with the hop-count to select the maximally-disjoint shortest paths. With AODV, source nodes only store the next-hop node. Not knowing the complete path to the destination node means that path disjointedness cannot be calculated at the source node. Finally, DSR does not periodically transmit HELLO messages. This reduces the number of control packets contributing to network load.

3.2.2 The Dynamic Source Routing (DSR) Protocol

DSR has two main functions: *Route Discovery* and *Route Maintenance*. Route Discovery is a process by which a source node obtains a path to a destination node. This process makes use of two control messages: ROUTE REQUEST and ROUTE REPLY. To learn a path to a destination node a source node broadcasts a ROUTE REQUEST. Each intermediate node which receives the ROUTE REQUEST

appends its address to the source route contained in the packet header. The intermediate node then broadcasts the packet. On receiving the ROUTE REQUEST message the destination node will respond with a ROUTE REPLY message. This message traverses the reverse path that the ROUTE REQUEST message travelled. The ROUTE REPLY is a unicast message. A destination node may receive several copies of a ROUTE REQUEST, each of which may have traversed a different path. The destination node will respond to all such requests unless a duplicate of a previously seen request with the same packet identifier and source route is received, in which case the packet will be dropped. The path contained in each reply packet will be added to the source node's *Route Cache*. This prevents a new Route Discovery being triggered for every path breakage. This allows nodes to respond quickly to broken paths: if the primary path fails another path can be selected from the Route Cache based on a shortest-path metric. Only if the Route Cache is empty is a Route Discovery process is triggered.

The DSR protocol's second main function is Route Maintenance. This function makes use of one control message: ROUTE ERROR. The maintenance process is triggered when the link between a node and the next-hop node on the path fails. This could be due to node mobility. The node which fails to deliver a packet will send a ROUTE ERROR message to the source node to inform it of the failed link. On receipt of this packet the source node will purge all paths containing the broken link from its Route Cache. It will then either select a cached path from the Route Cache or perform a Route Discovery if it has no cached path to the destination node.

After sending a ROUTE ERROR message, the node may perform *packet salvaging*. Packet salvaging is the process by which a node attempts to retransmit data packets along an alternative path to the destination node when the next link in the current path has failed. Packet salvaging is an optional feature of DSR, but it is enabled by default. When salvaging a data packet, the intermediate node which originated the ROUTE ERROR message searches its Route Cache for a path to the destination node. If a path is found, the intermediate node will update the source route in the data packet with the new path from itself to the destination node, before forwarding the packet on to the next-hop node in the source route. If an alternative path to the destination node cannot be found in the Route Cache (the intermediate node does not initiate a Route Discovery process), the intermediate node will discard the data packet.

The DSR protocol provides a best-effort forwarding service for data packets, but it can be augmented with QoS functionality to provide preferential treatment to high-priority traffic. The following section describes the INSIGNIA QoS framework which uses DSR as its underlying routing protocol.

3.2.3 The INSIGNIA QoS Framework

INSIGNIA [2, 101, 102, 104, 105, 108] is a QoS framework designed for MANETs. It is used in 2-DAARC as it is a prominent approach to reservation-based QoS in MANETs. INSIGNIA does not include its own routing protocol and it is not coupled to a particular routing protocol: it has been designed to work with AODV, DSR, and TORA. 2-DAARC uses INSIGNIA in conjunction with DSR.

The INSIGNIA QoS framework is built around the notion that supporting real-time services in MANETs requires applications and networks to adapt to changing conditions during the lifetime of a QoS-based session, and do so as efficiently as possible. INSIGNIA is therefore designed to be lightweight and adaptive in response to dynamic changes in network conditions and topologies. The aim of INSIGNIA is to provide minimum-bandwidth assurances to data requiring reservation-based QoS, and to provide these data with enhanced, maximum-bandwidth assurances if sufficient resources to support the maximum requirement are available. INSIGNIA uses two approaches to support QoS: adaptive bandwidth reservations and adaptive packet priority assignments. Both of these approaches rely on two mechanisms: an *'in-band' QoS signalling* mechanism and a *QoS reporting* mechanism.

To reduce signalling overhead, INSIGNIA uses an in-band signalling system [105] to establish, restore, and tear-down end-to-end reservations. Signalling information is carried in the header field of a data packet, rather than using separate signalling messages as in the case of an out-of-band signalling system. The in-band approach therefore consumes less bandwidth when establishing, maintaining, and tearing-down reservations from a source node to a destination node. In MANETs, where network topologies change dynamically owing to node mobility and other dynamic features (e.g., battery depletion), many more frequent resource (re-)reservations may be necessary, and this will lead to a high level of bandwidth overhead necessary to maintain suitable QoS support for data packet deliveries. So using the in-band signalling system can significantly reduce the bandwidth overhead in comparison with an out-of-band signalling system. Also, as in-band

Option	Service Mode	Payload Type	Bandwidth Indicator	Bandwidth Request	Padding
Values	RES/BE	BQ/EQ	MAX/MIN	MAX, MIN	-
Length	1 bit	1 bit	1 bits	16 bits	13 bits

Figure 3.1: The INSIGNIA IP Options Header. 19 bits are used for the INSIGNIA IP Options (with 13 bits of padding). This figure is based on the INSIGNIA Options field in the INSIGNIA specification [108], not the NS-2 implementation [106]. The differences between the specification and the implementation of the INSIGNIA IP Options header are discussed in Section 5.5.4.1.

signalling information is carried in the header field of a data packet, reservations can be (re-)established promptly during the course of data packet transmissions. In the best case scenario (e.g., assuming that the underlying routing protocol—DSR in the case of 2-DAARC—has a cached path to the destination), the in-band signalling system can establish a reservation within the duration of a few consecutive data packets.

The process of establishing a resource reservation using the in-band signalling system is as follows. INSIGNIA supports two levels of resource (bandwidth) reservations: maximum-reserved and minimum-reserved. A source node begins a reservation process by transmitting packets which request the maximum-reserved service. The settings of the INSIGNIA IP Options field, shown in Fig. 3.1, are as follows:

- the *service mode* will be set to RES (i.e., the reserved service is requested);
- the *payload indicator* will be set to either Base QoS (BQ) or Enhanced QoS (EQ) depending on the application's bandwidth requirement (a payload indicator of BQ requires at least the minimum bandwidth requirement be met whereas a payload indicator of EQ requires the maximum bandwidth requirement be met);
- the *bandwidth indicator* will be set to MAX; and
- the *bandwidth request* fields will be set to the minimum and maximum bandwidth values necessary to support the flow.

For the remainder of this example it is assumed that the *payload indicator* of the sending node is set to BQ (a description of events when the *payload indicator* is set to EQ is provided later). Intermediate nodes first attempt to support the

MAX bandwidth requirement if sufficient resources are available. The admission control module at each intermediate node receives a packet demanding QoS service. If the node can support the QoS requirement it will admit the reservation, setting a 1 second [106] soft-state timer, and leave the IP Options field unchanged. However, if the node cannot satisfy the MAX bandwidth requirement it will try to satisfy the minimum (MIN) requirement. If the MIN bandwidth request can be supported the intermediate node will admit the reservation and set the *bandwidth indicator* in the INSIGNIA Options to MIN. This indicates to the downstream nodes that the minimum bandwidth request is the largest that can be supported by at least one of the upstream nodes. The effect of this is that the downstream nodes will attempt to allocate only the minimum requested bandwidth. However, if there is not sufficient bandwidth available to support a MIN reservation by the intermediate nodes, the resource requested will be rejected by the admission control module. The priority of the packet will then be downgraded to request a best-effort delivery service, i.e., the intermediate node will change the *service mode* option to request the best-effort (BE) service. This is known as *degraded service*. The service mode may be upgraded back to RES if sufficient resources become available later in the communications session. However, if limited resource availability persists the QoS packets may continue to receive the best-effort service.

The sequence of events is slightly different for a packet flow with the payload indicator set to EQ (enhanced QoS). A packet flow demanding EQ requires the maximum bandwidth reservation along the entire path between a source and a destination (a flow demanding BQ can be supported with a minimum resource reservation). Similar to the BQ case, a packet with the *payload indicator* set to EQ is passed to the admission control module on arrival at an intermediate node. The flow is admitted if the intermediate node can support the maximum resource request. However, if the maximum resource requirement cannot be met the *service mode* of the packet will be downgraded to request only best-effort (BE) service from the downstream nodes. This is done for two reasons. First, a packet flow demanding the EQ service requires the maximum resource allocation to be admitted in order to support the packet flow's QoS requirements. Thus a lesser resource allocation is not sufficient. It is up to the sending application to decide whether to terminate the flow and to try again later, or whether to relax the constraints of the resource request. The sending application learns the outcome

of the reservation request through INSIGNIA's QoS reporting mechanism, which is discussed below. Second, the semantics of the adaptive services in INSIGNIA give preference to BQ packets (with minimum reservations) over EQ packets.

INSIGNIA's QoS reporting mechanism is used by the destination node to inform the source node of the outcome of a reservation request and to manage periodic end-to-end adaptations. A *QoS Report* contains bandwidth information about the flow between the source node and the destination node. If the QoS Report indicates that the resource reservation is successful the source node can continue sending data packets along the reserved path. Otherwise, if the source node is informed that the reservation is unsuccessful it will set the *service mode* of outgoing packets to request the best-effort forwarding service, i.e., to 'scale-down' the request such that the *service mode* is set to BE. QoS Reports are also transmitted periodically (every 2 seconds if priority data packets are still being received) to keep the source node abreast of QoS conditions along the path to the destination. If QoS conditions are good and sufficient resources become available the source node can 'scale-up' the reservation request. A 'scale-up' process may change the *service mode* from BE to RES, or if the *service mode* is already set to RES it can scale up the *bandwidth indicator* from MIN to MAX. In this way, QoS reporting can help to assist the adaptation process.

3.3 Cryptographic Primitives

This section presents an overview of the cryptographic primitives used in 2-DAARC. These primitives are a symmetric key cryptosystem and a hashed message authentication code (HMAC).

3.3.1 Symmetric Key Cryptosystem

A symmetric key cryptosystem is also known as a secret key cryptosystem. In a symmetric key cryptosystem the same secret key is used by a sender and a receiver to perform both encryption and decryption [184]. Symmetric key cryptosystems can be used to protect message confidentiality and message authenticity. Confidentiality is provided through *encryption* by the sender and *decryption* by the receiver. A symmetric key cryptosystem can also be used to provide message authenticity. This is achieved by combining a one-way hash function with a shared symmetric key to form a hashed message authentication code (HMAC).

3.3.2 Hashed Message Authentication Code (HMAC)

A *one-way hash function* converts a variable length input string, called a *pre-image*, to a fixed-length output string, called a *hash value* [169]. Formally, a one-way hash function $H(M)$ operates on a variable-length pre-image message M, and returns a fixed-length hash value h, i.e.:

$$h = H(M), \text{where } h \text{ is of length } m. \tag{3.1}$$

A cryptographic hash function should have the properties of 'one-wayness' and collision resistance [169].

- One-way property: given M, it is easy to compute h. However, given h, it is hard to compute M such that $H(M) = h$.
- Collision-resistance: given M, it is hard to find another message M' such that $H(M) = H(M')$.

A HMAC is a ked-dependent one-way hash function. A HMAC provides message authenticity as the generated hash value can only be verified by someone with the identical symmetric key. Message authenticity provides origin authentication and message integrity: origin authentication guarantees that the source of the message is the claimed sender; and message integrity informs the message receiver whether or not the message has been altered during its transmission.

A HMAC, which is also known as a keyed-hash function, works as follows. The hash function is used with a secret key known only to the sender and the receiver of a message: the sender applies the one-way hash function over the secret key k concatenated with the message M, producing $h = H(k\|M)$, which is transmitted to the destination along with the message M; the receiver, who also holds a copy of the secret key k, can calculate $H(k\|M)$ and compare it with the received hash h. This allows the receiver to verify the authenticity and the integrity of the received message. This process is depicted in Fig. 3.2.

SHA-224 [141] is used as the HMAC in 2-DAARC. SHA stands for Secure Hash Algorithm. SHA-224 is one of five secure hash algorithms in the Secure Hash Standard (SHS) [141].[2] NIST, the National Institute of Standards and Technology, has approved SHA-2 as acceptable for all hash function applications

[2]The other four are SHA-1, SHA-256, SHA-384, and SHA-512. The latter three algorithms, when combined with SHA-224, form the SHA-2 set of cryptographic hash algorithms.

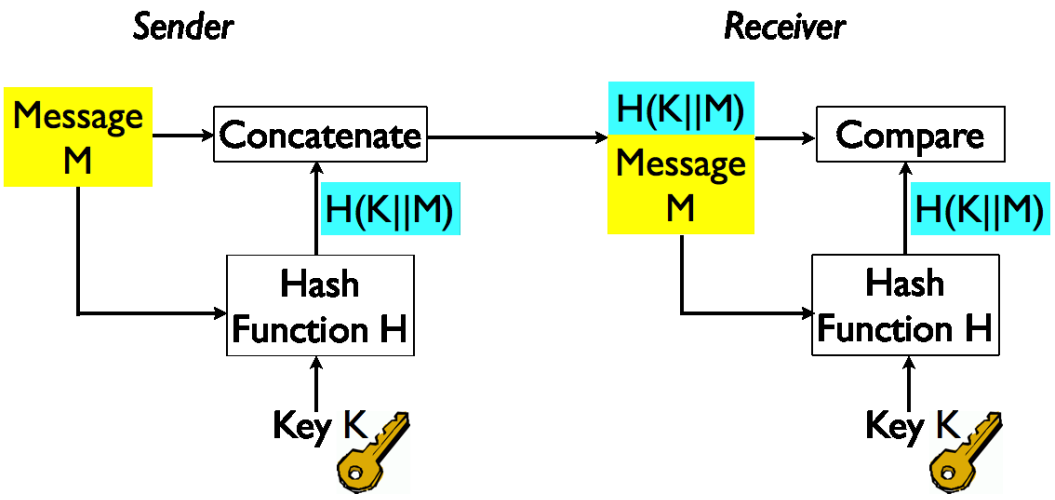


Figure 3.2: Keyed One-Way Hash Function

[9]. SHA-224 produces a 224-bit (28 byte) digest which is determined from the hash value. This is the shortest digest length of the SHA-2 algorithms. This keeps the cryptographic primitives used in 2-DAARC lightweight. The older SHA-1 hash function has a shorter message digest of 160-bits, but an attack on SHA-1 [200] and NIST's recent announcement of a new SHA-3 algorithm [137] following a five-year competition [136] suggest that SHA-2 provides a greater degree of future-proofing than SHA-1.

3.4 Evaluation Methodology

This section explores possible evaluation methodologies for the quantitative evaluation of the solutions proposed in this thesis. Three evaluation methodologies are identified and discussed: experimental, mathematical, and simulation. Simulation is chosen as the most suitable evaluation methodology, and justifications for this choice are provided. Three commonly used simulators are assessed to determine which is best suited to the evaluations performed in this thesis.

3.4.1 Experimental

The experimental evaluation methodology uses test-beds to evaluate the performance of ad hoc networking protocols on full-scale physical networks. The major benefit of generating experimental data from a test-bed is that these data are

based on realistic conditions. This is a useful step to understand the behaviour and performance of a protocol before deploying it for general use in large-scale networks [122]. Using the experimental methodology with test-beds is also likely to be the most accurate means of evaluating protocol behaviour and performance. This is because the conditions experienced during the evaluation are close to those which are likely to be experienced during a general real-world deployment.

However, the real-world characteristic of test-beds introduces a number of drawbacks. First, real-world experimental conditions are difficult to control. For example, the outdoor environment is unpredictable, as changes in weather or the movement of pedestrians and vehicles may affect results [122]. Whilst these conditions are realistic they are generally not repeatable [94]. Second, using test-beds is expensive. The costs experienced include the monetary cost of the computer equipment required to perform the experiment (one or more nodes per person), and the time cost of the people participating in the experiment. Additionally, when experimenting with mobile nodes at high speeds it may be necessary to have access to vehicles and drivers. Configuring and running a large-scale test-bed of mobile nodes is therefore a non-trivial task. The experimental evaluation methodology, whilst arguably the most accurate of the three methodologies, is not a viable means to undertake the evaluations in this thesis.

3.4.2 Mathematical

Evaluation using a mathematical model of a network is a cheaper option than using an experimental test-bed. It is possible to model mathematically many aspects of a network and network performance. However, deriving a mathematical model for a large-scale network may quickly become complicated. This is especially the case with the dynamic MANET environment, which would require complex characteristics, such as node mobility, to be factored in. Moreover, modelling the many complex factors of a network and producing meaningful results may be difficult as the complexity grows. This can be overcome by using simplifying assumptions to keep the analysis tractable. But this means that a smaller set of outcomes may be observed and analysed than is possible with an experimental or simulation-based evaluation; and this may limit the usefulness of the findings. Evaluation via mathematical model is therefore ruled-out for use in this thesis.

3.4.3 Simulation

Simulation is chosen as the evaluation methodology for this thesis. The simulation-based evaluation methodology is the process of using a *network simulator* to model computer networks, which includes the behaviour of network nodes and the communication channels [201]. The major benefits of a simulation-based evaluation methodology are that it is scalable, repeatable, and the network conditions can be controlled [94, 160]. Moreover, simulations enable the exploration of complicated scenarios which may otherwise be difficult to analyse using mathematical models, as the models may be mathematically intractable [44, 160]. Simulation-based evaluation is often used as a first-step in the evaluation of routing techniques for ad hoc networks. For example, the behaviour and performance of the DSR protocol was initially observed, analysed, and evaluated using the simulation-based evaluation methodology [89] before deploying and evaluating the protocol on an experimental test-bed [122]. It has also been demonstrated that spending time simulating protocols in a lab environment can significantly contribute to the success of later experimental testing [122]. Simulation can therefore be seen as a useful first step in the evaluation of a new protocol.

However, there are some issues with the simulation-based evaluation methodology that one should be cognizant of when interpreting simulation data. One issue is that network simulations are based on an abstract model of the world [44], and may therefore contain a number of assumptions to simplify the simulation process. For example, the two-ray ground propagation model in Network Simulator NS-2 (described below) assumes that the space between nodes is unobstructed [134]; however, obstructions in the form of both 'hard' objects, such as walls, and 'soft' objects, such as people, can influence radio propagation [119]. Another example is that NS-2 does not model signal-to-noise ratio, which means that data bits are transmitted at the same speed regardless of the proximity between two communicating nodes [166, 202]. In other words, NS-2 uses a binary transmission model. This model assumes that frame transmission is perfect when two nodes are within one another's transmission range and no collisions occur: every bit is transmitted at the same rate and is correctly received if nodes are within one another's wireless transmission range; otherwise all bits are lost if nodes are beyond one another's transmission range [134].

A more general issue is the lack of a standardised set of scenarios and simulation parameter values to evaluate MANET research [97]. This issue does not

effect the overall usefulness of the simulation-based evaluation methodology, as simulation can still be considered an inexpensive means to prototype ideas before deploying them on a larger, more expensive experimental test-bed. Moreover, the simulation-based methodology provides a compromise between cost, accuracy, and complexity. This cost-effectiveness leads to the simulation-based methodology being widely used. To overcome the lack of a standardised set of scenarios and simulation parameter values, the simulation model used in this research is based on extensive reading of the literature and takes into consideration what limited best-practice advice there is available for MANET research using simulation [97, 134, 160].

There are three popular simulation packages typically used in MANET research: *GloMoSim*, *OPNET*, and *NS*. GloMoSim is a discrete-event simulation environment for wireless network systems [48]. It is an open-source simulator and can be obtained free of charge. GloMoSim has built-in support for the DSR routing protocol. However, it does not have support for INSIGNIA, and no GloMoSim INSIGNIA plug-ins have been made available (INSIGNIA plug-ins are available for other simulators). It is of course possible to code the INSIGNIA QoS framework for the GloMoSim simulator, but this is time-consuming and may require the permission of INSIGNIA's original developers. GloMoSim is therefore not considered suitable for this research.

OPNET Modeler is a research and development solution for network modelling and discrete-event simulation [146]. The OPNET Modeller, hereafter referred to as OPNET, can be used to develop, simulate, and analyse communication networks, protocols, devices, and applications. Performing MANET simulations using the DSR protocol requires the OPNET Modeller Wireless Suite [147]. INSIGNIA, however, is not available as a part of OPNET [145]. An INSIGNIA module for OPNET has been written by the researchers who created INSIGNIA, but the source code is not available from the INSIGNIA project website (at the time of access) [106]. Additionally, OPNET is proprietary software, and issues with obtaining a licence preclude the use of this software.

The Network Simulator, NS-2 (Network Simulator version 2), is a discrete event simulator for networking research [138]. NS-2 is the most widely used network simulator for MANET research [97]. Unlike OPNET, it is an open-source simulator and can be obtained free of charge. NS was originally developed for wired networking simulations, but it has since been enhanced with support for

Hardware/Software	Specification
Machine	Dell Optiplex 755
CPU	Intel®Core™2 Duo @ 2.40 GHz
Physical Memory	2 GB
Operating System	Fedora 14, with Linux Kernel 2.6.35.6-45.fc14.i686
GCC Version	4.5.1
TCL/TK Version	8.0

Table 3.3: Specification of the Hardware and Software Used for Simulations

both infrastructure-based and ad hoc wireless networking. Additionally, the simulator can be extended with the DSR protocol and the INSIGNIA QoS framework. The DSR protocol is packaged with the wireless networking enhancements for NS-2 [29]. Source code for the INSIGNIA QoS framework is freely available online [106]. NS-2 is therefore selected as the simulator to use in this research.

3.5 Simulation Configuration

This section presents the configuration of the simulation environment and simulation model, and also describes the validation of the simulation model using a theoretical analysis.

3.5.1 Simulation Environment

Simulations are performed using the event-driven network simulator NS-2 [138] (version 2.1b3). NS-2 is extended with the CMU Monarch extensions (version 1.1.2) [29] which provide the DSR protocol and several wireless networking enhancements. The source code contains a number of optional optimizations; these are all set to their default (enabled/disabled) values. The INSIGNIA source code for NS-2 [106] is used. NS-2 is also extended with a HMAC implementation [47], as the simulator does not provide a built-in support for message authenticity. Simulations are conducted on a desktop computer running the Fedora 14 operating system. Further details about the hardware and software used on the computer running the NS-2 simulator can be found in Table 3.3.

3.5.2 Simulation Modelling

The following six components of the simulation model are described in this section: network configuration, mobility model, traffic pattern, denial of availability attacks, Watchdog [123], and Explicit Congestion Notification [165]. Table 3.4 summarises the simulation parameter values for the simulation model. A number of derived simulation parameter values are listed in Table 3.5. These are based on the formulae in [97]. The derived parameter values provide greater insight on the effects of the chosen simulation parameter values on the network.

3.5.2.1 Network Configuration

The network configuration of the simulation model is outlined as follows. The duration of each simulation is 900 seconds. 50 nodes are simulated in a flat, unobstructed 1000m x 1000m area. Using 50 nodes is common in simulation studies [17, 87, 101]. A square topology is used as it does not favour one range of motion over any other [71] (unlike, say, a rectangular topology). A priority queuing (PriQueue) mechanism [106] is used at each node with a packet buffer size of 50 packets. PriQueue is required by both DSR and INSIGNIA.

The following describes the link-layer configuration. The IEEE 802.11 MAC protocol is used with a data rate of 2Mbps. This is the MAC protocol used in the original DSR research [121] and the original INSIGNIA research [108]. The Distributed Co-ordination Function of the 802.11 protocol is used. RTS/CTS (Request To Send/Clear To Send) is enabled. A link-layer error model is not used. In other words, a packet/bit error rate of 0 is used. This is because the focus of this research is on packet loss due to attacks, node mobility, and congestion at the network layer, rather than packet corruption and loss due to the lossy wireless environment. Setting a non-zero value of error rate at the link-layer would increase the difficulty of isolating and observing the network-layer packet loss caused by the three factors of concern.

The following describes the configuration of the network model at the physical layer. An omni directional antenna is simulated with a two-ray ground propagation model. The antenna is centred in each node and is placed 1.5 metres above it. The nominal transmission range of a node is 250 metres. The wireless network interface is the hardware interface of a mobile node. It is used to access the wireless channel. The wireless shared media interface model approximates the 914MHz Lucent WaveLAN DSSS (Direct Sequence Spread Spectrum) radio

Network Configuration	
Simulation Duration	900 seconds
Number of Nodes	50
Topology (width x height)	1000m x 1000m
Queue	PriQueue
Packet Buffer Size	50
MAC Protocol	IEEE 802.11 (Distributed Co-ordination Function)
Bandwidth	2Mbps
Antenna	Omni Antenna
Propagation Model	Two-Ray Ground
Nominal Range	250 meters
Network Interface	Shared Media
Node Mobility	
Mobility Model	Random Waypoint [100, 148]
Node Speed	1–19 m/s (Mean = 10 m/s)
Pause Times	0 / 300 / 600 / 900 seconds
Traffic Model	
Traffic Type	Constant Bit Rate (CBR)
Number of Sources	10 / 20 / 30
Packet Size	512 bytes
Packet Per Second	4
INSIGNIA Configuration	
Routing Protocol	DSR (Dynamic Source Routing) [88]
Ratio of BE to QoS Packets	70:30
Ratio of BQ to EQ Packets	1:1
Adaptation Parameter	3
Adaptation Granularity	4
Attack Scenarios	
Attacks	Blackhole / Grayhole / Denial of QoS Request
Attacker Ratios	0%–50% in 10% increments
	0%–10% in 2% increments
Grayhole Drop Probabilities	0.2 / 0.4 / 0.6 / 0.8 / 1.0
Watchdog Configuration	
MIN_SIGNIFICANT_PACKETS	30
PERCENTAGE_LOSS	10%
REPORTING_PERIOD	5 seconds
STORE_PATH_TIME	30 seconds
ECN Configuration	
MAX_QUEUE_LENGTH	50 (equivalent to the *packet buffer size*)
Queue Occupancy Threshold	60%

Table 3.4: Network Configuration and Simulation Parameters

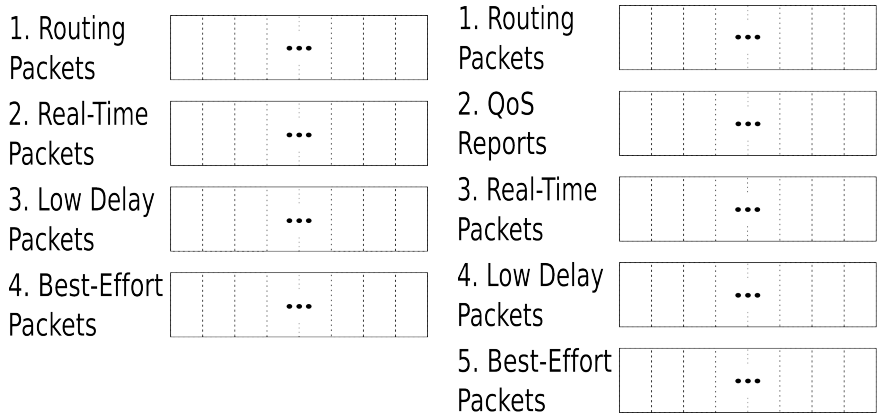
interface. This shared media interface has been used for two reasons. First, it is the default interface in NS-2. Second, the DSR and the INSIGNIA research

Parameter	Description	Formula	Parameter Value
Simulation Area	Square meter area of the topology.	$w \times h$	1,000,000 m^2
Node Density	Density of nodes in the simulation area.	$\frac{n}{(w \times h)}$	0.00005 nodes/m^2
Node Coverage	Area covered by a node transmission.	$\pi \times r^2$	196,349.54 m^2
Footprint	Percentage of the simulation area covered by a node's transmission range.	$\frac{(\pi \times r^2)}{(w \times h)} \times 100$	19.63 %
Maximum Path	The maximum linear distance a packet can travel from the source to the destination.	$\sqrt{(w^2 + h^2)}$	1414.21 m
Network Diameter	The minimum number of hops a packet can take along the maximum path from the source to the destination.	$\frac{\sqrt{w^2+h^2}}{r}$	5.66 hops
Neighbour Count	The number of neighbour nodes based on the transmission and the simulation areas. It does not account for the edge of the simulation area.	$\frac{(\pi \times r^2)}{\left(\frac{w \times h}{n}\right)}$	9.82 nodes
w = width, h = height, r = transmission range, and n = number of nodes.			

Table 3.5: Derived Simulation Scenario Parameter Values. The text of this table has been reproduced from [97]. The Parameter Value column has been added and the values it contains are calculated using the formulae in this table and the relevant parameter values from Table 3.4. π is specified to 9 decimal places. The width and height are the dimensions of the topology in Table 3.4. The radius of a node's transmission is the nominal range (250m).

do not specify which interface has been used, so it is assumed that they use the default interface.

Extending NS-2 with INSIGNIA requires some minor modifications to the original NS-2 installation, but these modifications can adversely affect the DSR protocol simulations. The two modifications causing the problems are an updated priority queuing mechanism and the introduction of a *type of service* (TOS) field to the IP-packet header. The PriQueue mechanism is used by both INSIGNIA



(a) Original PriQueue (b) INSIGNIA PriQueue

Figure 3.3: A Comparison of the Original PriQueue with the INSIGNIA PriQueue

and DSR. It is a multi-level queuing scheme supporting different priorities assigned to routing protocol control packets and data packets. The purpose of this is to prevent in-queue packet re-ordering to improve packet delivery performance [122]. Fig. 3.3(a) shows the original PriQueue and Fig. 3.3(b) shows the updated PriQueue after extending NS-2 with INSIGNIA. The difference between these two versions of PriQueue is that a new queue has been introduced for INSIGNIA QoS Reports (queue 2 in Fig. 3.3(b)). It was observed during DSR-only simulations that the updated PriQueue did not correctly place DSR's best-effort data packets into the best-effort packet queue (queue 5 in Fig. 3.3(b)). Best-effort packets appeared to be arbitrarily placed in difference queues. This detrimentally affected data packet delivery performance. When investigating this issue it was observed that the problem arose due to the incorrect setting of the newly introduced TOS bit. The TOS bit is used only by INSIGNIA, and no problems were observed during INSIGNIA only simulations. However, during the DSR-only simulations the TOS bit was occasionally randomly set. A fix was created to ensure that DSR's best-effort data packets are queued in the correct queue (queue 5) when running DSR simulations. The fix sets the TOS bit to request best-effort queuing treatment. The pseudocode for this fix is presented in Algorithm 3.1.

NS-2 does not provide built-in support for message authenticity, so it is extended with a HMAC implementation [47]. This implementation passes the IETF HMAC validation tests [139]. The tests are built into the HMAC implementation and have been run to confirm that the output of the HMAC implementation matches the output specified in [139]. This HMAC implementation has been used

Algorithm 3.1: Pseudocode for the Correct Queuing of Best-Effort Data Packets When NS-2 is Extended with DSR and INSIGNIA

```
Input: data packet
if DSR-only simulation then
    if data packet then
        type of service = best effort;
```

both in academia [187] and commercially [60].

3.5.2.2 Mobility Model

Nodes move according to the *Random Waypoint Mobility* model. This mobility model is chosen as it has arguably become a benchmark mobility model for protocol simulation. The original Random Waypoint Mobility model [17] has been considered harmful, as setting the minimum speed parameter to zero causes average node speed to decay and slowly fall toward zero [220]. An alternative implementation from a collection of Random Trip Mobility models [100, 148] is therefore used.

The Random Waypoint Mobility model works as follows. When not in motion, nodes remain stationary for a *pause time*. A node then randomly selects a location within the topology before moving to it at a speed uniformly distributed between a minimum and a maximum value. Having arrived at its destination a node again remains stationary for a *pause time* before moving to another randomly chosen location. This process is repeated for the duration of the simulation. For the simulation studies in this thesis, nodes move with a minimum speed of 1 m/s, a maximum speed of 19 m/s, and a mean speed of 10 m/s. This mean speed is consistent with the mean speed used in the original research using the Random Waypoint Mobility model [17], the DSR research [121], and the INSIGNIA research [101]. To put these speeds into a different context, they are equivalent to a minimum of 3.6 kph (2.24 mph), a maximum of 68.4 kph (42.50 mph), and a mean of 36 kph (22.37 mph). The pause times used are 0, 300, 600, and 900 seconds. A pause time of 0 seconds corresponds to continuous node movement, i.e., nodes move for the entire duration of a simulation, and a pause time of 900 seconds corresponds to nodes remaining static for the duration of a simulation. Fig. 3.4 shows an example movement pattern of a node moving in accordance with the Random Waypoint Mobility model for a 0 second pause

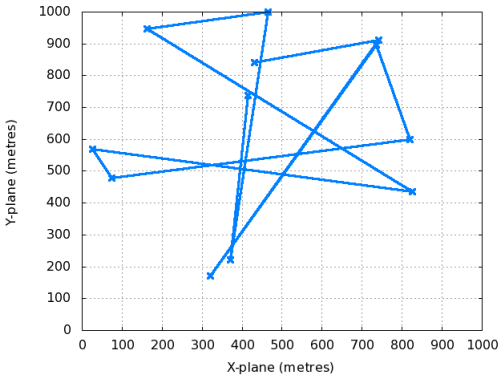


Figure 3.4: Example Node Movement Using the Random Waypoint Mobility Model

time. In the figure, the node moves between 23 waypoints over the course of a 900 second simulation.

10 different node movement trajectory scenarios have been generated for each pause time. These scenarios are used for each pause time to match the number of simulation runs performed and to test the protocols being simulated under a range of movement patterns. (Results are averaged over 10 simulation runs to reduce the effects of randomness in the simulation results; this number of simulation runs is determined in Section 3.5.3.) The same 10 scenarios for each pause time are used in all of the simulation studies performed in this research to ensure consistency and repeatability. In each scenario nodes start at a different position in the topology and their movement pattern is different from the other scenarios. For example, the scenario file named *0_SecondPause_1* has different node starting positions and node movement trajectories to the scenario named *0_SecondPause_2*. For the scenarios with a 900 second pause time the nodes are also located in a different position in the topology, but they do not move during the simulations. The scenario used for each simulation corresponds with the simulation run number. For example, for the above scenario, run 1 uses the *0_SecondPause_1* mobility pattern; run 2 for this scenario uses *0_SecondPause_2* mobility pattern, and so on.

3.5.2.3 Traffic Patterns

Nodes communicate with one another based on predefined communication patterns. Communication pattern files are generated using the traffic generation script provided in the CMU Monarch Extensions [29]. Constant bit rate (CBR)

traffic is transmitted by 10, 20, or 30 source nodes at a rate of four 512 byte packets per second. This traffic rate is consistent with simulation studies in the literature [32, 151]. When simulating using DSR, all of the traffic demands the best-effort forwarding service (hereafter referred to as *DSR best-effort traffic*). When simulating using INSIGNIA, 30% of these source nodes (referred to as *priority sources*) transmit traffic demanding the reservation-based forwarding service (referred to as *priority traffic*). The remaining 70% of source node transmit best-effort traffic (referred to as *best-effort traffic*). This best-effort traffic provides a background load in the network. In Chapters 5 and 6 this 30:70 split between the priority and best-effort sources remains, but reservation-based service is only demanded by the Single-Path Adaptation mode (adaptation dimensions are described in Chapter 5). The 30:70 split used in all the chapters is consistent with the INSIGNIA research [101]. Of the priority traffic demanding the reserved service, an equal ratio of Base QoS (minimum bandwidth requirement) and Enhanced QoS (maximum bandwidth requirement) is demanded. This is a default INSIGNIA configuration [106]). INSIGNIA's adaptation mechanism can downgrade priority traffic to receive only best-effort service: this traffic is referred to as *degraded traffic* to avoid confusion with DSR's best-effort traffic.

INSIGNIA has two further parameters which must be set: an *adaptation parameter* and an *adaptation granularity* parameter. These two parameters govern the time-scale over which adaptation occurs at a source node [108]. As these two parameters are not discussed in the literature, the following describes how they are used for bandwidth adaptations. The adaptation parameter is used to determine when to scale-up or scale-down a reserved packet flow, i.e., to determine when to increase or to decrease the bandwidth requested for resource reservations. The adaptation parameter is used in conjunction with two counters: a *scale-up counter* and a *scale-down counter*. The receipt of each QoS Report leads to either the scale-up counter being incremented if the available bandwidth is greater than the currently requested bandwidth, or the scale-down counter being incremented if the available bandwidth is less than the currently requested bandwidth. If the scale-up counter is greater than or equal to the adaptation parameter, the bandwidth requested by the flow is increased. Similarly, If the scale-down counter is greater than or equal to the adaptation parameter, the bandwidth requested by the flow is decreased.

The adaptation granularity parameter is used in a series of calculations to

determine and to adapt to the available bandwidth. Calculations are performed to determine the minimum and maximum available bandwidth. The adaptation granularity is one of several parameters used in these calculations. Additionally, a counter is maintained for the number of packets transmitted demanding Enhanced QoS (maximum bandwidth requirement). This counter is used with the calculated available bandwidth to determine whether outgoing data packets should be requesting reserved or best-effort service.

In the simulation model, the adaptation parameter value is set to 3 and the adaptation granularity parameter value is set to 4. These parameter values are default values in the INSIGNIA NS-2 extension [106]. None of the published research which uses INSIGNIA has specified the values used for these two simulation parameters; this includes the work by the original INSIGNIA researchers [2, 101, 102, 104, 105, 108] as well as other researchers whose work involves INSIGNIA [36, 39, 53, 96, 129, 179, 180, 191, 192, 198, 209, 216]. Setting these two parameters to their default values is done under the assumption that other researchers would have stated the parameter values used if they were not using the default values.

3.5.2.4 Denial of Availability Attacks

A variable percentage of network nodes perform blackhole [68], grayhole [71], and denial of QoS request [118] attacks on data packets. These attacks are referred to as *denial of availability attacks* (DoA) in this research. DoA attacks are different from denial of service (DoS) attacks. In a DoS attack, an attacker attempts to overwhelm a target node by flooding it with more traffic than it can handle. In a DoA attack, a target node's packets are removed from the network (blackhole and grayhole attacks) or are altered to limit the service that they receive (denial of QoS request attack). The blackhole attack is simulated by checking whether a received packet is a data packet and discarding it if so. Algorithm 3.2 provides pseudocode for the blackhole attack. The grayhole attack functions similarly but only drops data packets with a certain probability: a drop probability is randomly selected with a uniform distribution from the set {0.2, 0.4, 0.6, 0.8, 1.0} at simulation initiation; this drop probability is used by all grayhole attackers for the duration of a simulation. This approach to random grayhole attacker selection is influenced by the approach in [123]. Algorithm 3.3 provides pseudocode for

Algorithm 3.2: Pseudocode for the Blackhole Attack

```
if BLACKHOLE_NODE then
    if data packet then
        drop data packet;
```

Algorithm 3.3: Pseudocode for the Grayhole Attack

```
if GRAYHOLE_NODE then
    if data packet then
        if dropProbability ≠ 0 and (dropProbability == 1.0 or
        (Random.uniform() < dropProbability)) then
            drop data packet;
```

the grayhole attack. The blackhole attack and the grayhole attack do not discriminate between priority and non-priority data packets, i.e., the attacker drops both priority and non-priority data packets alike. Both the blackhole and grayhole attacks are implemented within the DSR component of NS-2. The denial of QoS requests attack is simulated by checking whether the *service mode* bit is set to RES (reserved) in the header of an INSIGNIA data packet and modifying it to BE (best-effort) if so. Algorithm 3.4 provides pseudocode for the denial of QoS request attack. The denial of QoS request attack is implemented within the INSIGNIA component of NS-2.

Two attacker ratios are used to perform the above DoA attacks. The first set of attacker ratios ranges from 0%, representing a non-malicious network environment, to 50%, in increments of 10%. While 50% of network nodes behaving maliciously may seem artificially high, it provides an opportunity to understand network performance in adverse and unreliable conditions [123]. The second set of attacker ratios ranges from 0% to 10% in increments of 2%. The lower maximum attacker ratio and smaller attacker ratio increment allow the behaviour of the simulated solutions to be observed at a more fine-grained level.

3.5.2.5 Watchdog

The performance of the Congestion and Attack (CAT) detection mechanism, presented in Chapter 6, is evaluated against Watchdog [123]. One of the metrics used in this evaluation is the normalised routing load (this and other metrics are

Algorithm 3.4: Pseudocode for the Denial of QoS Request Attack

```
if DENY_QOS_REQUEST_NODE then
    if data packet demanding reserved service then
        set service mode to best-effort;
        set payload indicator to base QoS;
        set type of service to best-effort;
```

explained in Section 3.5.2.8). The focus of this part of the evaluation is therefore on the number of control packets the detection mechanisms inject into the network. Given this focus, this simulated version of Watchdog differs from the original Watchdog proposal [123] and its AODV-based implementation [63] in the following two ways. First, intermediate nodes monitor neighbouring nodes' misbehaviour only for actions affecting priority data packets, i.e., the dropping of all non-priority data packets is ignored. This is to focus on the control packet notifications, and the overhead this generates, in response to the dropping of priority data packets. Second, to simplify the implementation, rather than using promiscuous overhearing for misbehaviour detection, each node has *a priori* knowledge of which nodes are blackhole attackers; this knowledge is used in conjunction with the source route carried in data packet headers to determine whether the next-hop node is a blackhole attacker.

The following describes how Watchdog is simulated (a description of Watchdog is given in Section 2.7.3.1). The simulation code implemented in this research is based on the Watchdog for AODV implementation [63]. Algorithm 3.5 shows the Watchdog actions performed when intermediate nodes receive priority data packets. Nodes maintain a counter for the number of packets received for each data flow traversing it. *A priori* knowledge of blackhole attackers is used to determine whether the next-hop node in the source route of a received data packet is a blackhole attacker. If the next-hop node is not an attacker a counter for the number of transmitted packets is incremented. A packet loss percentage is calculated when Watchdog has observed a significant number of data packets. A feedback packet is transmitted to the source node if both the loss percentage is greater than a threshold value and a sufficient period of time has passed since the last feedback packet was sent. The feedback is transmitted immediately if (a) this is the first feedback packet or (b) it is the first feedback packet since Watchdog's counters were reset after not receiving packets belonging to this packet flow

Algorithm 3.5: Pseudocode for Watchdog

```
Input: Priority Data Packet pkt
if (Watchdog[pkt.sourceID].time - CURRENT_TIME) >
STORE_PATH_TIME then
    resetWatchdog(pkt.sourceID);
Watchdog[pkt.sourceID].pktCount++;
if pkt.nextHopAddress ≠ blackhole node then
    Watchdog[pkt.sourceID].transmittedPkts++;
if Watchdog[pkt.sourceID].pktCount > MIN_SIGNIFICANT_PACKETS
then
    percentage = 1 - ((Watchdog[pkt.sourceID].transmittedPkts /
    Watchdog[pkt.sourceID].pktCount) * 100);
    if (percentage > PERCENTAGE_LOSS) and CURRENT_TIME >
    (Watchdog[pkt.sourceID].lastNotificationTime +
    REPORTING_PERIOD) then
        sendFeedbackPkt_watchdog(pkt.sourceID);
        Watchdog[pkt.sourceID].lastNotificationTime = CURRENT_TIME;
```

for longer than a time-interval. Any feedback to be transmitted after this first feedback packet is transmitted periodically, not immediately.

Three main parameters guide Watchdog's misbehaviour reporting mechanism. The first is *PERCENTAGE_LOSS*, the packet loss threshold. This is set to 10%. This makes this parameter consistent with 2-DAARC's packet loss adaptation threshold ρ, which is set to 10% and is discussed in Chapter 5. The second is *MIN_SIGNIFICANT_PACKETS*, the minimum number of priority data packets received before the packet loss percentage is first calculated. It is set to 30. This is to ensure that the node receives a significant number of priority data packets before the packet loss percentage is periodically calculated. The third parameter is *REPORTING_PERIOD*. This governs how often a control packet can be transmitted to the source node to inform it of routing misbehaviour. The periodicity is 5 seconds. This periodicity is consistent with 2-DAARC's reporting interval τ (described in Section 5.5.4.2).

3.5.2.6 Explicit Congestion Notification

The congestion detection component of the CAT detection mechanism, presented in Chapter 6, is evaluated against the Explicit Congestion Notification (ECN)

Algorithm 3.6: Pseudocode for ECN (Intermediate Node)

```
Input: Priority Data Packet pkt

if current queue occupancy > (MAX_QUEUE_LENGTH * 0.6) then
    pkt.congestion_experienced_flag = 1;
```

Algorithm 3.7: Pseudocode for ECN (Destination Node)

```
Input: Priority Data Packet pkt

if pkt.congestion_experienced_flag == 0 and ECN[sourceID] == 1 then
    ECN[sourceID] = 0; resetEcnData(sourceID);
else
    if pkt.congestion_experienced_flag == 1 and ECN[sourceID] == 0 then
        sendFeedbackPacket_ECN(sourceID);
        ECN[sourceID] = 1;
```

[165, 164] mechanism. This evaluation is to compare the network load introduced by the congestion detection component with that of the ECN mechanism. The following describes how ECN is simulated (a description of ECN is given in Section 2.6.1). Algorithms 3.6 and 3.7 show the ECN actions performed by the intermediate and destination nodes, respectively, on receipt of priority data packets. An intermediate node sets the congestion experienced (CE) bit of a priority data packet if its packet queue occupancy exceeds a threshold. The 60% queue occupancy threshold in Algorithm 3.6 is based on the queue length thresholds used in [66, 107]. The length of the packet queue is 50 packets, as specified in Section 3.5.2.1. On receipt of a priority data packet the destination node checks if the CE-bit is set. The destination node's response to CE-marked packets is based on the implementation of ECN for RTP over UDP [203], and works as follows. If the packet's CE-bit set and it is the first packet received with it set (either since the session started or the ECN data was last reset), a feedback packet is transmitted immediately to the source node. This feedback packet informs the source node that congestion has been experienced on the path. If the destination node continues to receive packets with the CE-bit set, all further ECN feedback information is transmitted as part of 2-DAARC's periodic feedback. If a packet for the data flow is received without the CE-bit set, the destination node resets the ECN data it stores for this node. This is because an unset CE-bit indicates an absence of congestion on the path.

3.5.2.7 Assumptions

The following seven assumptions are used in the simulation studies in this thesis.

1. Nodes executing blackhole and grayhole attacks only drop data packets. They correctly forward control packets, INSIGNIA QoS Reports, and 2-DAARC feedback packets. The justifications for this are (1) blackhole and grayhole attackers want to be included in paths, so they must participate in Route Discovery operations; and (2) the focus of this research is on the effects of blackhole and grayhole attacks on data packets. All data packets received by a blackhole or grayhole attacker node are dropped; these attacks do not discriminate between priority and non-priority data packets.

2. All attacker nodes operate independently, i.e., they do not collude. Addressing the issue of collaborative attacks is a research area in its own right.

3. The source node and the destination nodes trust each other. Threats to the data packet forwarding process are only from intermediate nodes.

4. Each pair of source-destination nodes communicating priority traffic has a security association and posses a pre-shared symmetric key.

5. All of the obtained information, such as hop counts, source routes, feedback data, etc., is considered to be a faithful representation of the network and state information.

6. Intermediate nodes faithfully and correctly perform Watchdog and ECN operations. This is so that the control traffic generated by these two mechanism can be observed without attacks on the mechanisms affecting the investigation on blackhole attacks.

7. Nodes faithfully and correctly perform adaptive packet salvaging operations. This is so that the effects of adaptation in congested and non-congested conditions can be observed.

3.5.2.8 Performance Metrics

To evaluate the performance of the traffic demanding priority service, for either INSIGNIA or 2-DAARC, only the priority data packets and control packets associated with the priority data packets are considered in the calculation of the

following metrics. In other words, all non-priority data packets and associated control packets are excluded from the calculations. To evaluate the performance of DSR traffic all data packets and control packets are included in the performance metric calculations. The following five performance metrics are used to evaluate the simulation results.

Packet delivery ratio (PDR): the total number of data packets received divided by the total number of data packets transmitted. When used in the context of INSIGNIA and 2-DAARC priority data packets, PDR refers to the priority data packets only. When used in the context of INSIGNIA best-effort traffic, PDR refers to the background traffic. In the context of DSR, PDR refers to all of the best-effort traffic.

Average end-to-end throughput: the total number of data packets received by the destination nodes divided by the duration of a simulation (in seconds).

Service quality: the ratios of the number of priority data packets which (i) are delivered with reserved service, hereafter referred to as the *reserved packet delivery ratio* (RePDR), (ii) are delivered with degraded service, referred to as the *degraded packet delivery ratio* (DePDR), and (iii) are dropped, referred to as the *dropped packet ratio* (DrPR), to the total number of priority packets. The total packet delivery ratios (PDR) for the priority traffic can be derived from the service quality values by summing-up the RePDR and DePDR values. This metric is not well defined in the literature; thus it was defined in one of the published papers resulting from this research [126].

Average end-to-end delay: the difference between the time when a packet is transmitted by the source node and the time when the packet is received at the destination node. The average end-to-end delay consists of processing, queuing, transmission, and propagation delays.

Normalised routing load [103]: also known as the 'control overhead', is the ratio of control packets transmitted by all nodes in the network to the total number of data packets received by destination nodes. 'Control packets' includes Route Request, Route Reply, and Route Error packets. When simulating INSIGNIA and 2-DAARC only the priority data packets and control packets associated with the priority data packets are considered.

In other words, all control packets associated with the non-priority data packets are excluded from the normalised routing load calculations. This is to observe the normalised routing load of only the traffic demanding priority service. When simulating INSIGNIA the normalised routing load includes QoS Reports. When simulating 2-DAARC the normalised routing load includes feedback packets and QoS Reports. When simulating Watchdog and ECN the normalised routing load also includes Watchdog Reports and ECN Reports.

Results are averaged over 10 simulation runs. This number of runs is determined in Section 3.6.

3.5.3 Simulation Model Validation

In this section the simulation model is validated. It is necessary to validate the simulation model so that the results obtained from it can be considered reliable and meaningful. The simulation model validation is carried out in two phases. In the first phase NS-2's validation scripts are used to verify that the simulator is correctly installed and behaves as expected. In the second phase a theoretical model is constructed and the results from the theoretical model are compared against the results collected from simulation runs.

3.5.3.1 Phase One: Automated Tests

NS-2 is packaged with a number of validation scripts. These are used to validate that the NS-2 binary is correctly installed and that it is generating expected output [82]. The scripts perform a series of simulations. The output of these simulations is compared against reference output packaged with the simulator. All validation tests pass. The HMAC implementation [47] has been validated using the IETF HMAC validation tests specified in [139]. The HMAC implementation has been executed using the IETF HMAC validation tests as input, and the output has been compared with the reference output in [139] to confirm that they match.

As NS-2 is extended with the CMU Monarch Extensions, it is also necessary to validate that these extensions have been correctly installed. However, no validation scripts are packaged with the simulator for this purpose. Phase two of the simulation model validation is therefore conducted with these extensions

enabled. This is to gain confidence that the simulator behaves as expected when it is used with the CMU Monarch Extensions.

3.5.3.2 Phase Two: Theoretical Model

The phase two validation is performed by comparing the simulation output with that of a theoretical model under a given set of conditions. This validation method provides an objective measure of confidence in the simulation model and in the simulation process [131]. The theoretical model is expressed as a mathematical model of a network. It shares the same set of assumptions and parameter values as the simulation model. The assumptions made are to keep the analysis tractable:

- the network contains 5 nodes which are equally spaced at 200m intervals;
- there is 1 source node and 1 destination node;
- only a single path is used between the source and the destination nodes;
- all nodes are stationary all of the time;
- the simulator transmits best-effort data packets using INSIGNIA (resource reservation is not performed);
- the data packet transmission rate is 4 packets per second, i.e., a packet is transmitted every 0.25 seconds;
- there is no background load on the network, i.e., the network is unloaded;
- no attacks are performed;
- the propagation speed used in the simulations is set to 299,709,438.5 m/s.

The theoretical model and the simulation model are compared using the *end-to-end delay* metric. The end-to-end delay consists of four delay components: *transmission delay*, *propagation delay*, *queueing delay*, and *processing delay* [98]. In the wireless environment, it is also necessary to consider the delay caused by the MAC (Medium Access Control) protocol. The end-to-end delay in the theoretical model is calculated using Eq. (3.2).

$$\begin{aligned}\text{End-to-End Delay} = \text{Transmission Delay} + \text{Propagation Delay} \\ + \text{Queueing Delay} + \text{Processing Delay} + \text{MAC Delay}\end{aligned} \tag{3.2}$$

The four components of end-to-end delay are described as follows.

Transmission Delay This is the time taken to transmit a packet on to a link. It is calculated using Eq. (3.3).

$$\text{Transmission Delay} = \frac{\text{packet size (bits)}}{\text{data transmission rate (bps)}} \tag{3.3}$$

Packet size is the total number of bits in a packet. It is 512-bytes, i.e., 4096-bits, per packet. The transmission rate is 2,000,000 bits per second, i.e., 2Mbps. This equation only takes into consideration the initial transmission delay, i.e., the source node transmitting the packet on to a link. It is also necessary to consider the additional transmission delays caused by each intermediate node's transmission of the packet on to a link. Thus the transmission delay is revised to take the intermediate nodes' transmissions into consideration. The revised transmission delay is calculated using Eq. (3.4).

$$\text{Transmission Delay} = \left(\frac{\text{packet size (bits)}}{\text{data transmission rate (bps)}}\right) \times \text{no. of hops} \tag{3.4}$$

Propagation Delay This is the time a packet takes to propagate along a link from a transmitting node to a receiving node. It is calculated using Eq. (3.5).

$$\text{Propagation Delay} = \frac{\text{distance (m)}}{\text{propagation speed (m/s)}} \tag{3.5}$$

The *distance* is the distance in metres between the node transmitting the packet and the node receiving the packet. 1–5 nodes are used in both the theoretical and simulation models. These nodes are located at a distance of 200m from one another. In this theoretical model, the distance used in the propagation delay equation is the total distance between the source node and the destination node. For example, the distance for 5 nodes is 800m ($4 \times 200m$ hops). The *propagation speed* is dependent on the physical medium of the link. MANETs use wireless links. The propagation speed of a wireless link is equivalent to the speed of light in air. This is calculated as $c/1.000277 = 299{,}709{,}438.5$ m/s, where c is the speed of light in a vacuum (299,792,458 m/s) and 1.000277 is the refractive index of air [173].

Queueing Delay A queueing delay is experienced by a packet as it waits to be transmitted on to a link. The length of the queueing delay is dependent on the

number of earlier arrived packets in the queue. No queueing delay is experienced if a queue is empty. One of the above assumptions is that the network is unloaded. Thus queueing delay is considered to be insignificant. The queueing delay is therefore set to zero in the theoretical model.

Processing Delay This refers to the time taken to examine a packet's header to determine where to direct it. The processing delay calculated using Eq. (3.6).

$$\text{Processing Delay} = 0.00095 \text{ (seconds)} \times \text{no. of intermediate nodes} \tag{3.6}$$

The value of $0.00095s$ is the processing time for each packet. The packet processing time specified in [133] is used. Each intermediate node processes a packet to examines its header [98]. Hence the packet processing time is multiplied by the number of intermediate nodes.

MAC Delay The MAC (Medium Access Control) delay is a consequence of the RTS-CTS-ACK process of the MAC protocol. The MAC protocol controls access to the shared wireless medium. Before transmitting a data packet, a node must first transmit a Request To Send (RTS) control packet to the next-hop node. The next-hop node replies to the RTS packet with a Clear To Send (CTS) control packet. After the receipt of the CTS packet, the sending node then transmits the data packet. The next-hop node acknowledges the receipt of the data packet by transmitting an ACK control packet. The MAC delay is the combined delays of the RTS, CTS, and ACK packet transmissions. This delay is calculated using Eq. (3.7).

$$\text{MAC delay} = 0.000339 \text{ (seconds)} \times \text{no. of hops} \tag{3.7}$$

The MAC delay value of $0.000339s$ is an averaged value obtained from the simulator. The MAC delay is experienced for each wireless hop. Hence the MAC delay value is multiplied by the number of hops.

Comparing the Theoretical Results with the Simulation Model

The values of the theoretical end-to-end delays are compared with the results obtained from the simulation runs. The comparison is shown in Fig. 3.5 for paths containing 1, 2, 3, and 4 wireless hops. From the figure it can be seen that the results from the theoretical model successfully validate the simulation model. The results provide confidence in the simulation model; hence any further results collected via simulation can be considered reliable.

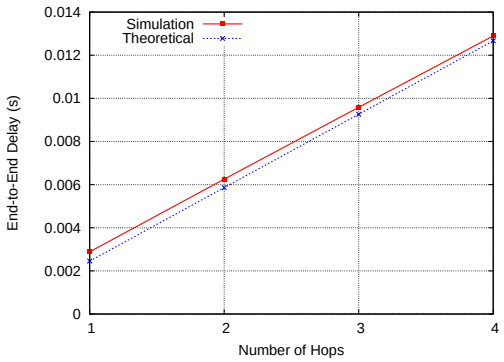


Figure 3.5: Comparing Theoretical and Simulated Results for the End-to-End Delay for Paths with 1, 2, 3, and 4 Wireless Hops.

3.6 Generating Statistically Significant Simulation Results

A number of factors should be considered to produce statistically significant results when using a simulation-based evaluation methodology. These factors are the proper seeding of the simulator's pseudo-random number generator, the number of independent simulation runs data are collected over, the duration of each simulation run, and the use of diverse simulation scenarios.

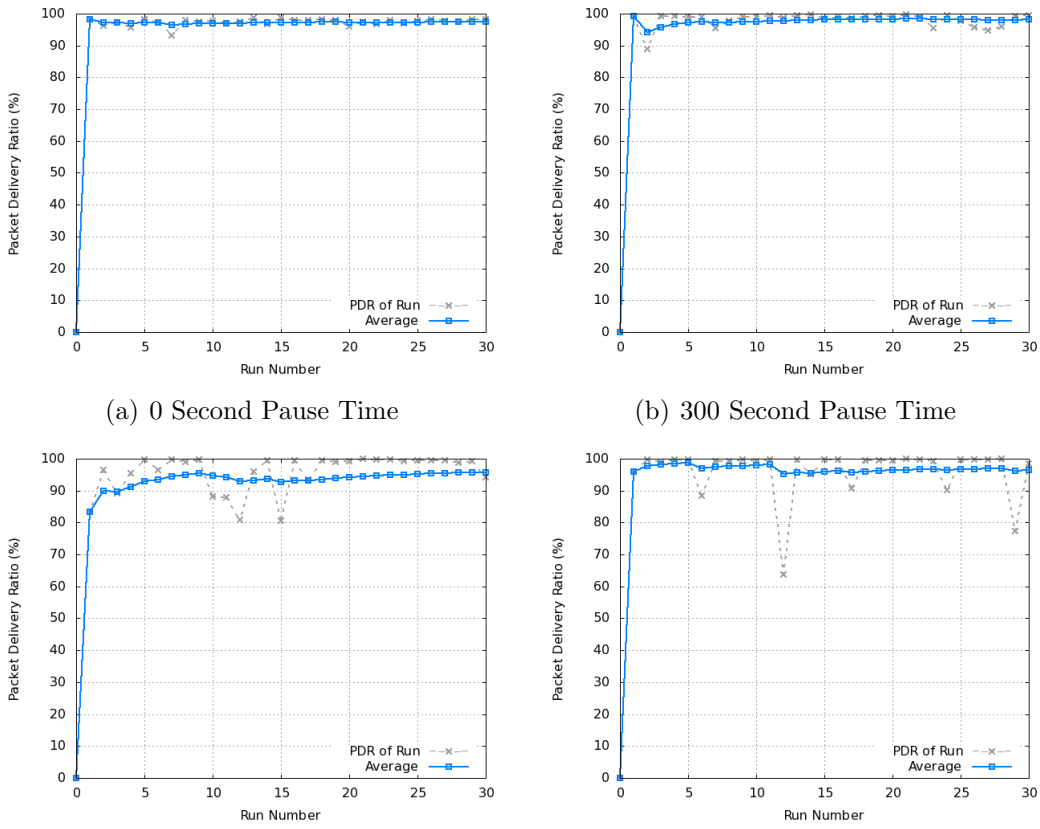
The proper seeding of a simulator's pseudo-random number generator (PRNG) is the first important factor for the production of statistically significant simulation results [97, 154]. NS-2 uses a PRNG which generates a periodic sequence of pseudorandom numbers with a period of $2^{31} - 2$. (This information is obtained from the *rng.cc* file of the CMU Monarch Extensions for ns-2.1b3.) Properly setting the PRNG seed requires the seed to be changed for each simulation run. The value of the seed is set using the simulation run number. This means that each of the 10 simulation runs performed for each scenario are initialised with a different seed. For example, the first simulation with a 900 second pause time, 10 traffic sources, and a 0% attacker ratio has a seed of 1; the second simulation uses the same configuration but has a seed of 2, and so on and so forth. When simulating a different protocol the same seed values can be re-used to ensure that the network conditions are the same. Using the time of the system clock as the seed would produce non-deterministic behaviour, but this is not repeatable and it may be difficult to compare fairly the produced data sets: it is recommended that this method is not used when the seed needs to be set for independent replications

[43].

The second important factor in the production of statistically significant simulation results is to run a sufficient number of independent simulations. Running several simulations of each scenario and taking an average of the results provides credibility in the final simulation results [154]. Moreover, collecting a sufficiently large data set can be considered as important as running simulations for a longer duration [154]. There is a trade-off to consider when choosing the number of runs over which to take an average: using fewer runs requires less storage space for the simulation output data and reduces the time required to collect these data; however, these benefits may be offset by a decreased statistical significance in the final data set if it is too small.

Simulations are conducted to determine the number of simulation runs over which data should be averaged to produce statistically significant results. Nodes transmit only best-effort traffic using INSIGNIA. INSIGNIA runs on top of DSR. Fig. 3.6 shows the packet delivery ratios (PDR) for 10 traffic sources (all best-effort) averaged over 1–30 simulation runs for the 0, 300, 600, and 900 second pause times. This is shown with the solid blue curve. The PDR value obtained during each run is also shown with the dashed gray curve. In other words, the dashed line shows the PDR achieved during that particular run whereas the solid line shows the averaged PDR of all runs up to that point. It can be seen from Fig. 3.6(a) that with a 0 second pause time the differences among the PDRs averaged over 10, 20, or 30 simulation runs are minor: the PDR for 10 runs is 95.47%, for 20 runs is 96.48% , and for 30 runs is 96.85%. In Fig. 3.6(b), with a 300 second pause time, the differences are again minor, with a PDR of 97.77% after 10 runs, 98.43% after 20 runs, and 98.16% after 30 runs. The PDRs obtained with the 0 second and 300 second pause times are largely flat, or have a minor increase as the number of runs for 30 runs increases.

With the 600 second and 900 second pause times, however, the average PDR decreases slightly as the number of runs the average is taken over increases. Fig. 3.6(c) shows a PDR of 94.71% after 10 runs, 94.12% after 20 runs, and 95.73% after 30 runs for the 600 second pause time. The variation of approximately 1 percentage point PDR is minor for the 600 second pause time. With the 900 second pause time, shown in Fig. 3.6(d), the decrease in PDR as the number of runs increases is greater than with the 600 second pause time. After 10 runs the average PDR is 98.52%, after 20 runs the PDR is 96.61%, and



(a) 0 Second Pause Time

(b) 300 Second Pause Time

(c) 600 Second Pause Time

(d) 900 Second Pause Time

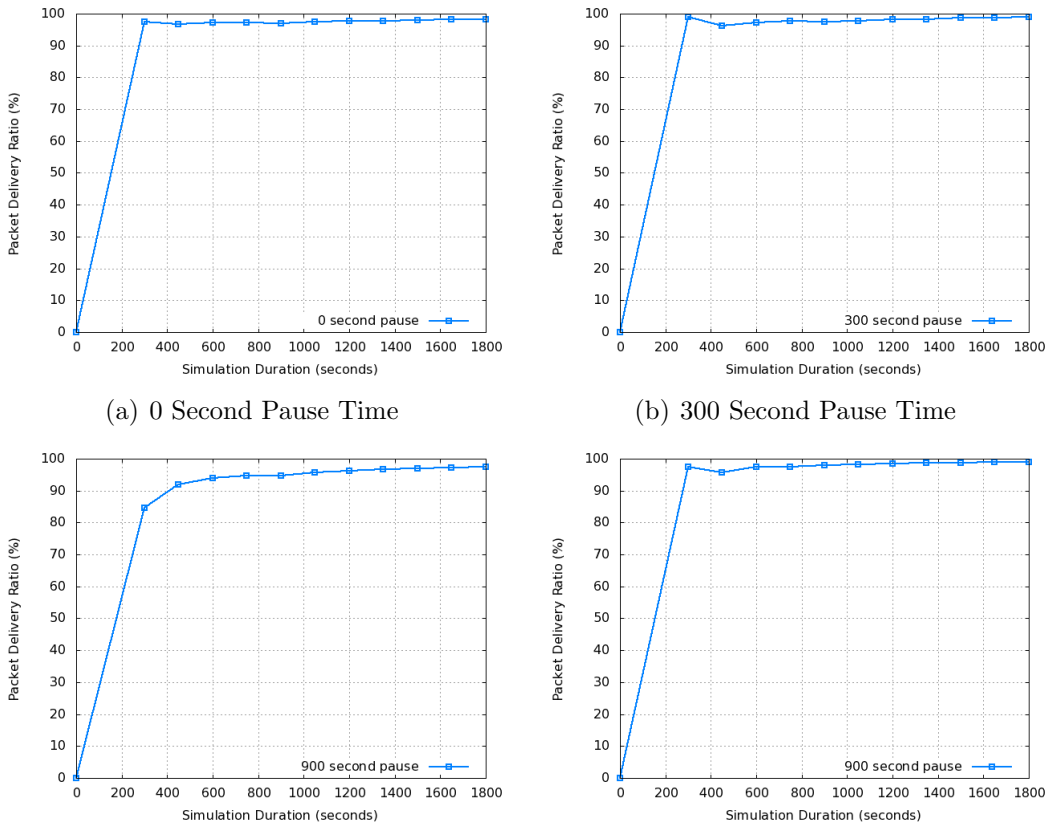
Figure 3.6: Comparing the Accuracy of Simulation Results Averaged Over 1 to 30 Simulation Runs with 10 Source Nodes

after 30 runs the PDR is 96.51%. The downward trend is a consequence of the occasional large downward spikes in PDR. These spikes are mainly due to nodes discarding packets after queueing them for longer than a timeout value. (In this case one-third of the data packets are dropped, which suggests that one of the three nodes is unable to transmit its packets to the the intended destination.) With this pause time all nodes are always stationary; thus is it possible for any negative effects, such as this queueing issue, to be experienced for a significant duration of a simulation, hence the lower PDR values. Removing the largest spike in PDR (run 12) from the average value of PDR eliminates the downward trend in the average PDR value. Thus when the average PDR value is considered without these anomalous results the PDR after 10 simulation runs is similar to the PDR after 30 simulation runs. Based on the above results, a decision is made to collect simulation data over 10 simulation runs. This provides a balance between the statistical significance of the generated simulation data and the storage space

required for the data generated by the thousands of simulations conducted in this research.

The third factor affecting the statistical significance of results is the use of a simulation duration which allows the simulator to reach a steady state. The purpose of running a simulation for a sufficient length of time is (1) to minimise any influence of the initial simulation state [154], and (2) to ensure a steady state is reached. A simulation duration of 900 seconds is commonly used in the literature for NS-2-based evaluations of MANET protocols [17, 32]. Simulations are conducted to check that a 900 second simulation duration leads to a steady-state being reached with the version and configuration of the NS-2 simulator used in this research. PDRs obtained using simulation durations of 300–1800 seconds in 150 second increments are shown in Fig. 3.7. The simulator generally reaches a steady-state by the 900 second simulation duration. The curves are relatively flat from the 450 second simulation duration and beyond for the 0, 300, and 900 second pause times (Figs. 3.7(a), 3.7(b), and 3.7(d), respectively). For the 600 second pause time, shown in Fig. 3.7(c), the curves begin to flatten from the 600 second simulation duration. The PDR achieved after 1800 seconds of simulation time is slightly higher for all pause times, with an average difference of 1.66% from the PDR at the 900 second duration. Achieving this higher accuracy with the 1800 second duration has a higher cost: a longer period of real time is needed to collect the output data from simulations with the longer duration; this output data is larger than that generated in the 900 second case, requiring more hard disk space for storage. The above observations and analysis confirm that the simulator reaches a steady-state with a 900 second simulation duration. This duration is therefore used for the simulations in this research.

The fourth factor to consider for statistically significant simulation results is the variety of simulation scenarios used for simulations. There are two different issues to consider: connection patterns and movement patterns. The connection patterns specify a source node and a destination node for each data flow. Connection patterns are generated using NS-2's constant bitrate traffic generation script (cbrgen.tcl) packaged with the CMU Monarch Extensions [29]. The output from executing this script specifies which of the network nodes are source-destination pairs for a communication session. Five connection pattern files are used. These specify the connection patters for each of the 10, 20, 30, 40, and 50 source-destination pairs. The same connection patterns are used for DSR, INSIGNIA,



(a) 0 Second Pause Time

(b) 300 Second Pause Time

(c) 600 Second Pause Time

(d) 900 Second Pause Time

Figure 3.7: Comparing the Accuracy of Simulation Results Averaged over Different Simulation Durations with 10 Source Nodes

and 2-DAARC simulations, i.e., the same source-destination pairs communicate with one another regardless of which routing approach is simulated. However, the connection patterns have minor differences due to the different types of traffic transmitted when using different routing approaches. When simulating DSR all nodes are instructed to communicate using only best-effort traffic. When simulating INSIGNIA and 2-DAARC, 30% of the nodes are instructed to communicate using reserved/priority traffic. This difference is specified in the connection pattern using different port numbers: best-effort traffic is transmitted over port #0, and reserved/priority traffic is transmitted over port #1. These 5 connection patterns are used for all of the simulations performed in this research.

In contrast, different movement patterns are used for each simulation run. Movement patterns are generated with the *Random Waypoint Mobility model* [100, 148]. One movement pattern scenario is generated for each simulation run. This makes a total of 40 scenarios, as there is one scenario for each of the 10 runs

for each pause time (0, 300, 600, and 900 seconds). These 40 movement pattern scenarios are used for all simulations, i.e., for all network loads, attacker ratios, etc.

3.7 Chapter Summary

This chapter has presented the building blocks of 2-DAARC and has discussed how it is evaluated. The building blocks of 2-DAARC are the DSR protocol, the INSIGNIA QoS framework, the symmetric key cryptosystem, and the hashed messaged authentication code. Simulation was selected as the investigation methodology, and the Network Simulator NS-2 was chosen as the simulator, given its support for DSR and INSIGNIA. The configuration of NS-2 was described in detail. A statement of assumptions was made for the simulations performed in this research. The metrics to be used for performance evaluation in the following chapters were also presented. Simulation model verification was performed by comparing the output from simulations with that of a theoretical model. Finally, a number of simulation studies were presented to determine simulation parameter values and to demonstrate the statistical significance of the generated simulation data.

Chapter 4

Reserved vs. Best-effort Packet Forwarding

4.1 Chapter Introduction

Prior to the design of new solutions to support QoS in open MANETs, a simulation study is performed to observe and critically analyse a reservation-based approach and a best-effort approach to data packet forwarding on a single path under a range of network conditions, including security attacks. As stated in Section 1.6, the hypothesis of this research is that duplicating priority data packets over multiple best-effort paths may better support priority data packet deliveries than the reservation-based approach in the presence of packet forwarding attackers. The aim of the study is (1) to investigate whether a reservation-based or best-effort approach will provide a better data packet forwarding service along a single path and the conditions under which it does so, and (2) to use the insights gained to inform the designs of the solutions presented in Chapters 5 and 6.

This chapter is organised as follows. Section 4.2 first presents the motivation behind the simulation study before presenting the results of the study. Section 4.3 presents the lessons learnt from the study. Finally, Section 4.4 presents the chapter summary.

4.2 Simulation Results

This section presents the results of the simulation study investigating the QoS achieved by the INSIGNIA QoS framework, as a facilitator of the reservation-based approach, and the DSR protocol, as a facilitator of the best-effort approach. Results are collected under a range of network conditions, including different network loads, node mobilities, types of attack, and number of attackers. This simulation study looks broadly at the blackhole, grayhole, and denial of QoS request attacks. This is so that (1) the effects of a number of different attacks on data packet forwarding can be investigated, and (2) the findings from this investigation can influence the designs of the solutions presented in Chapters 5 and 6. That said, the effects of the blackhole attack are the main focus of the results.

The results are presented in four sections. Section 4.2.1 presents the packet delivery ratios. Section 4.2.2 presents the end-to-end delays. Section 4.2.3 presents the throughputs. Finally, Section 4.2.4 presents the service quality. The following simulation parameter values are used in the simulations presented in these four sections. Three traffic loads are used: 10 source nodes are used to generate a lightly loaded network, 20 source nodes are used to generate a medium load, and 30 sources nodes are used to generate a highly loaded network. In each of these cases, 30% of the source nodes transmit data packets demanding the priority forwarding service from the network (as explained in detail in Section 3.5.2.3). Four node mobilities are used: 900, 600, 300, and 0 second pause times. The attacker ratios range from 0%–50% in 10% increments. The results have been collected with the packet salvaging optimization of the underlying routing protocol (DSR) enabled. All other parameter values are as specified in Section 3.5. Simulation parameter values remain unchanged for all of the results unless otherwise specified.

4.2.1 Packet Delivery Ratio

The first investigation studies the effects of node mobility and network load on the packet delivery ratios (PDR) in the absence of attackers. The PDR values for the INSIGNIA reserved traffic (marked 'priority' in the figures)[1], best-effort

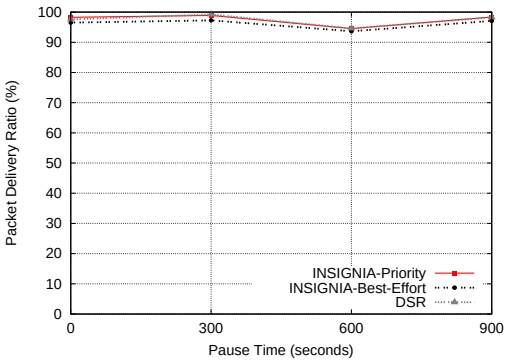
[1]The PDR of the priority traffic is the combined total of the priority traffic delivered to the destination node with (1) the reserved service and (2) the degraded (best-effort) service.

background traffic (marked 'background'), and the DSR traffic (marked 'DSR') are plotted in Fig. 4.1, for 0, 300, 600, and 900 second pause times with three distinct traffic levels: 10 sources (Fig. 4.1(a)), 20 sources (Fig. 4.1(b)), and 30 sources (Fig. 4.1(c)).

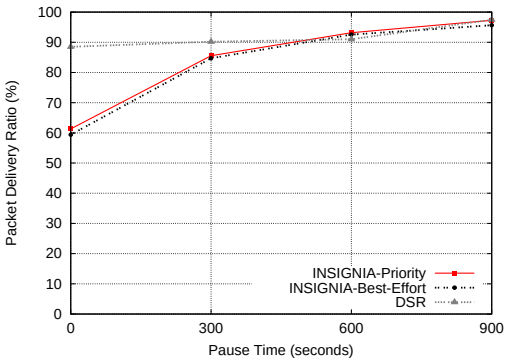
When the network is lightly loaded (10 source nodes) the PDRs of the priority traffic are similar to those of DSR and are generally maintained above 96% for all pause times. As can be seen in Fig. 4.1(a), the curves are generally flat with the exception of the 600 second pause time. The nearly 4% packet loss ratio is due to nodes dropping data packets when they do not have a path to the destination node to forward them along. When a source node performs a Route Discovery operation it buffers the outbound data packets. If no path is learnt before the expiration of a timeout the buffered packets are dropped. This means that some packets cannot be forwarded to the destination node as it has become unreachable. Additionally, if an intermediate node cannot forward a packet to the next-hop node in the source route, it will attempt to forward the packet to a different next-hop node. However, if it does not know a next-hop node located on a path to the packet's destination node the packet is dropped.

When the network load increases to that of 20 source nodes the PDRs are similar at low node mobilities, but DSR achieves higher PDRs than INSIGNIA at high node mobilities. As can be seen in Fig. 4.1(b), the PDRs are similar at the 900 and 600 second pause times, but DSR markedly outperforms INSIGNIA as the pause time continues towards 0 seconds. In other words, as node mobility and the offered load increase, the benefits brought by INSIGNIA decrease. In contrast, DSR is less sensitive to the increases in node mobility and offered load. The PDR with DSR at the 0 second pause time is 88% whereas the PDR for INSIGNIA's priority traffic is about 60%; both achieved PDRs of approximately 98% in the more lightly loaded network (Fig. 4.1(a)).

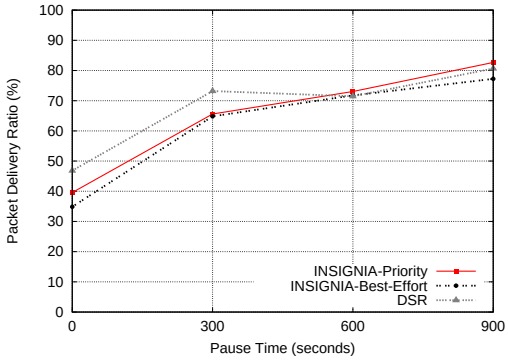
There are two related reasons for these observations: node mobility and network load. Node mobility indirectly increases the network load. This is because mobility leads to link breaks, and these reduce the effective available bandwidth. The increase in the number of source nodes places a greater load on the network. There are more packets to be transmitted with the lower available bandwidth, hence the lower PDRs. However, INSIGNIA is affected by this more than DSR. One reason is that INSIGNIA transmits more control packets than DSR, and these affect the data packet deliveries. INSIGNIA periodically transmits QoS Reports



(a) 10 Source Nodes (3 Priority Sources)



(b) 20 Source Nodes (6 Priority Sources)



(c) 30 Source Nodes (9 Priority Sources)

Figure 4.1: Comparing the PDR of INSIGNIA Priority and Best-Effort Traffic and DSR Traffic for 3, 6, and 9 Priority Sources (7, 14, and 21 Best-Effort, Background Sources) in an Attacker-Free Network Under A Range of Mobilities

in addition to the ROUTE REQUEST, ROUTE REPLY, and ROUTE ERROR control packets of its underlying routing protocol (DSR). DSR transmits only the latter three control packets. INSIGNIA therefore spends more time than DSR capturing the wireless medium for control packet, rather than data packet, transmissions. Control packets receive a higher transmission priority than data packets. Data packet queues become more heavily loaded whilst the larger number of control packets are transmitted. This, combined with the higher network load, leads to overflowing packet queues and lower PDRs.

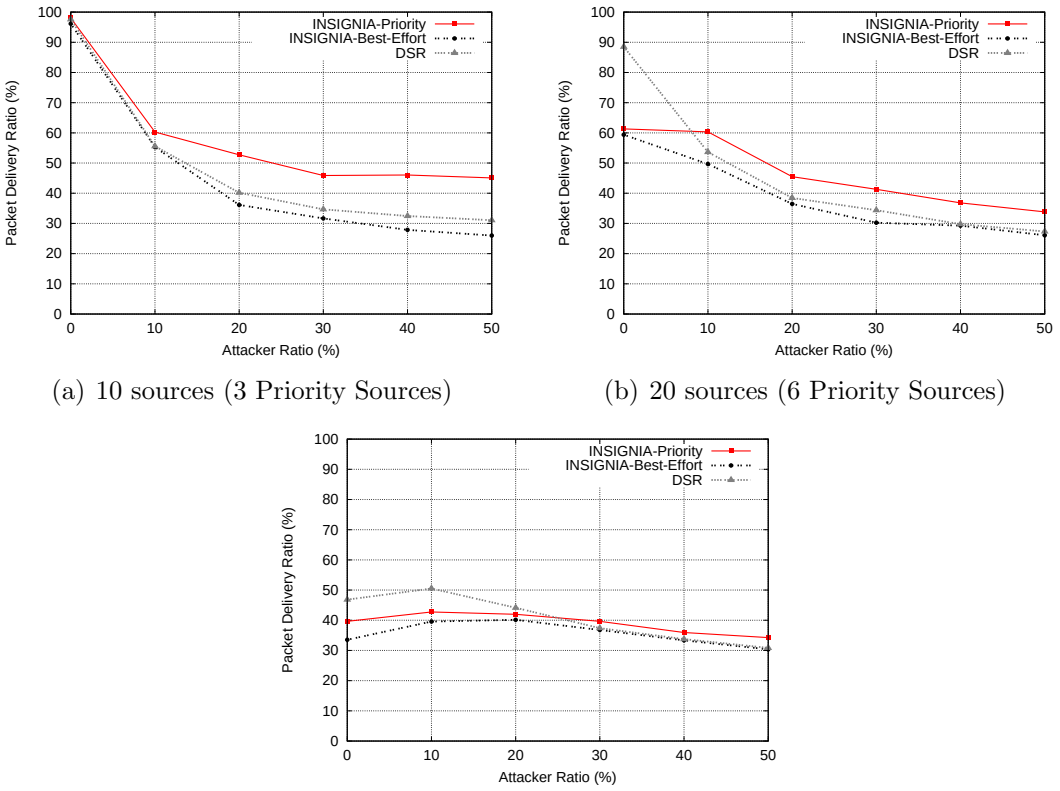
When the traffic load further increases to that of 30 source nodes the PDRs are again similar at the lower node mobilities, and DSR again achieves higher PDRs than INSIGNIA at the higher node mobilities. The reasons for these observations are similar to those presented above for the 20 source node case. At the 0 second pause time there is a marked drop in DSR's PDR: it has decreased from 88% in the

20 source node case (Fig. 4.1(b)) to 47% in the 30 source node case (Fig. 4.1(c)). The benefits brought by DSR over INSIGNIA at this pause time are reduced by the high network load and network congestion.

The remainder of the results presented in this section are obtained with highly mobile nodes performing the blackhole attack. Results for the blackhole attack are presented as they are more interesting than those collected under the grayhole and denial of QoS request attacks (the more interesting results for the grayhole and the denial of QoS request attacks are presented in Sections 4.2.3 and 4.2.4). A 0 second pause time is used to generate the high node mobility and to present challenging network conditions.

When the traffic level is light (10 source nodes) the PDRs are generally low, but INSIGNIA achieves higher PDRs than DSR at all non-zero attacker ratios. As can be seen in Fig. 4.2(a), the PDR values are approximately 98% with a 0% attacker ratio (the result for the 0% attacker ratio is the same result as that presented in the mobility-only case in Fig. 4.1(a)), before dropping to 53% for priority traffic, 40% for DSR traffic, and 36% for best-effort traffic when the attacker ratio increases to 20%. From this point on the PDR values decrease more gradually and level-off at 45% for priority traffic, 31% for DSR traffic, and 25% for best-effort traffic with a 50% attacker ratio. At this attacker ratio, blackhole attacks are the dominant factor causing packet loss: attacks are responsible for approximately 97% of the dropped priority packets and approximately 98% of the dropped best-effort and DSR packets. One observation on Fig. 4.2(a) is that fewer priority packets are dropped than best-effort or DSR packets. One reason for this is that there are fewer priority packets in the network to be attacked, i.e., the best-effort traffic is more plentiful. This observation is confirmed by the observation on the following figure, where a larger percentage of the priority packets are dropped when the number of priority packets in the network increases.

When the number of source nodes is increased to 20, INSIGNIA again achieves higher PDRs than DSR at the non-zero attacker ratios. As can be seen in Fig. 4.2(b), there is a marked difference in the PDRs at the 0% attacker ratio, but there are smaller differences in the PDRs with the non-zero attacker ratios. This is because congestion affects INSIGNIA and DSR differently at the 0% attacker ratio (as described for Fig. 4.1(b)), but they are affected more similarly by the blackhole attacks. The attacks reduce congestion as they become the dominant cause of packet loss. The reduced congestion increases the available bandwidth.



(a) 10 sources (3 Priority Sources)

(b) 20 sources (6 Priority Sources)

(c) 30 sources (9 Priority Sources)

Figure 4.2: Comparing the PDR of INSIGNIA Priority and Best-Effort Traffic and DSR Traffic for a 0 Second Pause Time and 3, 6, and 9 Priority Sources (7, 14, and 21 Best-Effort, Background Sources) Under Blackhole Attacks

INSIGNIA can take advantage of this by servicing more priority packets with the bandwidth reserved service. Another observation is that the PDRs of the priority data packets are lower compared with the 10 source node case (Fig. 4.2(a)), whereas the PDRs for DSR are similar. The decreased PDRs of the priority traffic are a consequence of the larger number of priority packets in the network (due to the increased number of source nodes). A greater percentage of the priority packets are being attacked as they are forwarded along more paths containing blackhole attackers.

When the number of source nodes is further increased to 30, INSIGNIA and DSR achieve more similar PDRs as the attacker ratio increases. As can be seen in Fig. 4.2(c), when the attacker ratio increases from 0% to 50%, the PDR values for DSR and INSIGNIA fluctuate within the region of 31% to 50% and 30% to 43%, respectively. The higher PDR of DSR at the 0%–20% attacker ratios is because

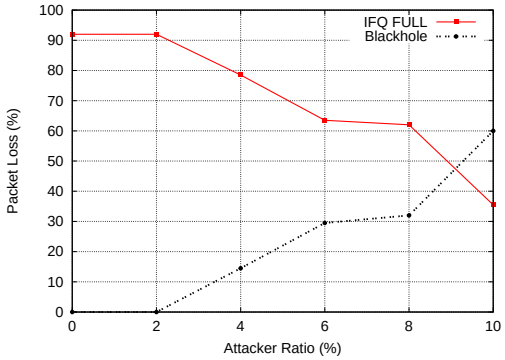


Figure 4.3: Comparing the Percentage of Packets Lost to Overflowing Packet Queues with Blackhole Attacks for DSR Traffic with a 0 Second Pause Time and 30 Best-Effort Sources Under Blackhole Attacks

it suffers from the effects of congestion less than INSIGNIA (as explained for Fig. 4.1(c)). An interesting observation is that the PDR values for both DSR and INSIGNIA increase when the attacker ratio is increased from 0% to 10%. To explain this observation the results are looked at in further detail. The focus here is on DSR, although the explanation is also applicable to INSIGNIA. The PDR values of DSR increase from 47% to 50% as the attacker ratio increases from 0% to 10%. At a 0% attacker ratio, 92% of the packet loss is caused by overflowing packet queues (due to network congestion). However, when the attacker ratio increases to 10% the PDR is 50%, of which 60% of the packet loss is due to blackhole attacks and 35% is due to overflowing packet queues. In other words, when the network traffic level is high and the attacker ratio is small (less than 8%) the main cause for packet loss is network congestion, so a small number of blackhole attacks actually serves to reduce the traffic burden in the network. This results in a slight increase in PDR.

To validate this finding a further investigation is undertaken for a 0%–10% attacker ratio in increments of 2%, whilst keeping all other simulation parameter values unchanged. The results for DSR are shown in Fig. 4.3. The results show that the percentage of packets lost due to blackhole attacks increases significantly from 14% at the 4% attacker ratio to 60% at the 10% attacker ratio. However, the percentage of packets lost due to overflowing packet queues decreases more significantly, thus offsetting the increase in packet loss caused by blackhole attacks. The blackhole attacks lead to reduced congestion, hence the above observed increase in PDRs between the 0% and 10% attacker ratios.

By comparing the PDR values of DSR and INSIGNIA shown in Figs. 4.2(a)–4.2(c) the following three further observations can be made. First, when the network load is at a medium (Fig. 4.2(b)) to high (Fig. 4.2(c)) level and when the attacker ratio is below a certain threshold value, the PDR values of DSR are higher than those of the INSIGNIA priority traffic. In other words, as the network load increases the priority source nodes suffer a greater rate of decrease in PDR than the DSR source nodes. This is because the available bandwidth decreases when the traffic load increases, and this will lead to admission control rejecting resource reservation requests. The priority packets with rejected reservations are downgraded to receive the best-effort service, i.e., INSIGNIA's degraded service. The high number of control packets required to provide the bandwidth-reserved service to the priority traffic negatively affect data packet transmissions and lead to lower PDRs. However, when the blackhole attacker ratio goes beyond the threshold value, DSR's PDR values are lower than those of the INSIGNIA priority traffic. This means that, as the attacker ratio increases, the rate of decrease in DSR's PDR values is higher than the rate of decrease in the PDR values of INSIGNIA's priority traffic. This may be due to the network becoming less overloaded as more packets are discarded (due to the attacker ratio increases), and INSIGNIA making better use of the available bandwidth. For example, as resources are freed as a consequence of packet drops, INSIGNIA will be able to admit flows demanding a resource reservation. The packets from the admitted flows are placed into the priority packet queue. This spreads the traffic load across more packet queues, and reduces the number of degraded priority packets in the best-effort queue which are lost as a consequence of congestion.

The second observation is that the attacker ratio threshold value increases as the traffic load increases. In other words, up to a threshold value, DSR achieves higher PDRs than INSIGNIA when both the attacker ratio and the traffic load increase. The higher the traffic load the more blackhole attacks are necessary to offset the packet loss caused by network congestion. The third observation is that with DSR and INSIGNIA, PDR values are markedly reduced when the blackhole attacks are in action. This means that in the presence of blackhole attacks data packet forwarding along a single-path cannot effectively support the deliveries of priority traffic. In addition, in both the DSR and INSIGNIA cases the PDRs are more sensitive to the changes in attacker ratios when the network load is lighter. This means that under light network load blackhole attacks are the main causes

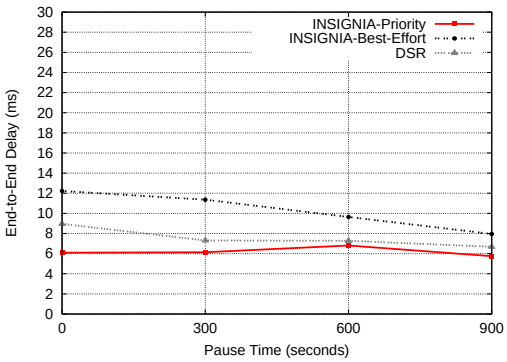
of packet loss, and as soon as the network load increases congestion becomes the major cause of packet loss.

4.2.2 End-to-End Delay

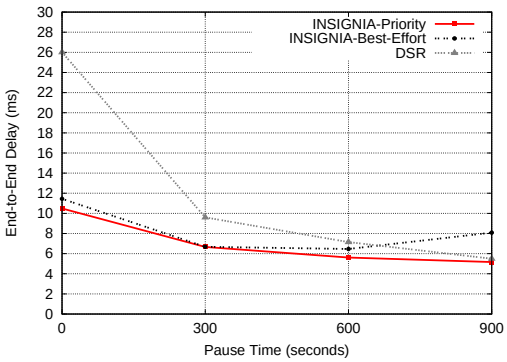
The main purpose of the simulations presented in this section is to investigate the effects of node mobility and blackhole attacks on the end-to-end delays of the two packet forwarding approaches. In the figures, the curves for the priority traffic are the end-to-end delays of both the priority traffic receiving the bandwidth-reserved service and the priority traffic degraded to receive the best-effort forwarding service.

In a lightly loaded (10 source node) attacker-free network, the priority traffic generally experiences lower end-to-end delays than the best-effort traffic under a range of node mobilities. As can be seen in Fig. 4.4(a), the priority traffic experiences end-to-end delays of approximately 6–8ms whereas the delays of DSR are approximately 7–9ms. DSR experiences longer delays than INSIGNIA because it queues all of the data packets in a single, best-effort queue; this provides all of the data packets with the same best-effort service. In contrast, INSIGNIA queues the data packets receiving the bandwidth reserved service in a high priority queue and all other data packets in a lower priority best-effort queue; INSIGNIA therefore distributes the load of the data packets across multiple queues. The priority queuing mechanism of the underlying routing protocol (described in Section 3.5.2.1) services the higher priority traffic before the lower priority traffic. The priority traffic is granted an earlier transmission opportunity than the lower priority best-effort traffic. This leads to shorter queueing delays and therefore shorter end-to-end delays for the priority traffic. Another reason for the priority traffic's lower delays is that there are fewer priority packets in the network than best-effort packets. This places a lighter burden on the priority packet queue compared with the best-effort packet queue, and it leads to shorter queueing delays.

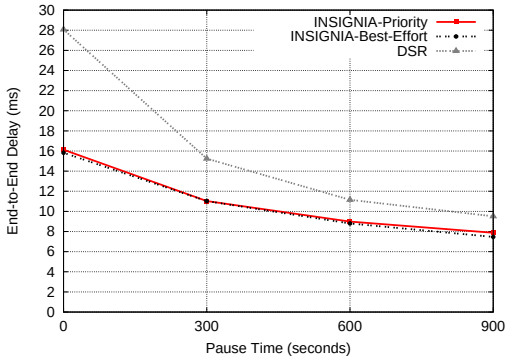
When the load increases to that of 20 source nodes, the delays of both approaches increase significantly as node mobility increase. As can be seen in Fig. 4.4(b), the delays of the priority traffic remain lower than those of the best-effort traffic. At the 0 second pause time, the delay of the priority traffic increases to approximately 10ms, which is approximately 2.6 times less than the 26ms delay of the best-effort traffic. The increased number of traffic sources means that there are more data packets in the network, and these place a greater burden



(a) 10 sources (3 Priority Sources)



(b) 20 sources (6 Priority Sources)



(c) 30 sources (9 Priority Sources)

Figure 4.4: Comparing the End-to-End Delay of INSIGNIA Priority and Best-Effort Traffic and DSR Traffic for 3, 6, and 9 Priority Sources (7, 14, and 21 Best-Effort, Background Sources) in an Attacker-Free Network Under A Range of Mobilities

on network resources. The available bandwidth reduces due to the higher traffic load, mobility-induced path breaks, and the retransmission of packets when they fail to be delivered. These retransmissions lead to interference and further retransmissions. Consequently, the network becomes increasingly busy, and this leads to longer delays. The failed packet deliveries trigger DSR's packet salvaging mechanism. With DSR, packet salvaging occurs at the 0 second pause time approximately 4.5 times more often than at the 300 second pause time, and 10 times more often than at the 900 second pause time. This process contributes to the higher queue occupancy as packets are re-queued. The high occupancy often leads to the queue overflowing. For example, DSR's packet queue at the 0 second pause time exceeds 50% buffer occupancy 5 times more often than at the 300 second pause time and 63 times more often than at the 900 second pause time. The high occupancy contributes to longer queuing delays and therefore longer

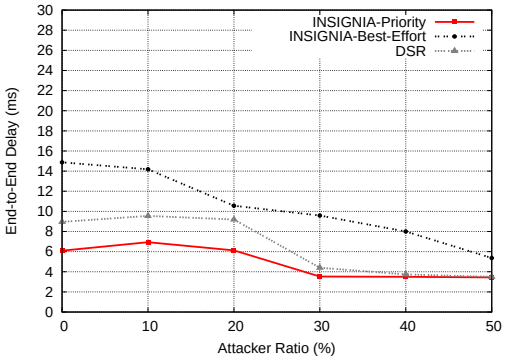
end-to-end delays.

When the network load increases to that of 30 source nodes, INSIGNIA again achieves lower end-to-end delays than DSR, although the delays of both approaches have increased compared with the 20 source node case. Fig. 4.4(c) shows the delays for this higher network load. The larger number of source nodes leads to increased delays for both the reserved and the best-effort approaches. This is due to the congestion caused by the higher network load and node mobility. The delay of DSR is now 1.75 times that of INSIGNIA. This difference is less than the 2.6 times difference between DSR and INSIGNIA in the 20 source node case. The reduced difference in delays is because INSIGNIA's delays have increased more than those of DSR. This is due to the higher network load reducing the benefits that INSIGNIA offers over DSR.

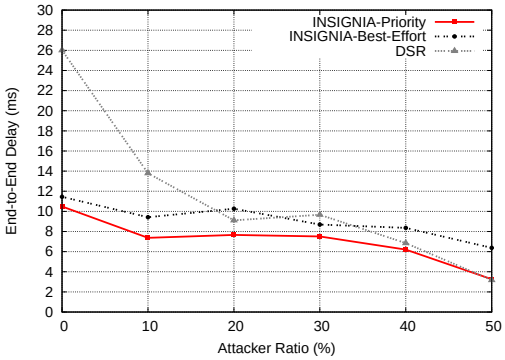
The remainder of the results presented in this section are obtained with highly mobile nodes (0 second pause time) performing blackhole attacks. The end-to-end delays of both the reservation-based and best-effort approaches generally decrease as the attacker ratio increases. This can be seen in the light (Fig. 4.5(a)), medium (Fig. 4.5(b)), and heavily loaded (Fig. 4.5(c)) networks. The reason for this is that blackhole attackers remove data packets from the network. This reduces the network load and increases bandwidth availability. Queueing delays are reduced, and this also reduces the end-to-end delays.

Another observation is that the rate of decrease in end-to-end delay is higher when an increasing number of attackers is introduced into more heavily loaded networks. For example, with 30 source nodes (Fig. 4.5(c)), the delay for DSR decreases from 28ms at the 0% attacker ratio to 4ms at the 50% attacker ratio. In contrast, with 10 source nodes (Fig. 4.5(a)) the delay decreases from 9ms at the 0% attacker ratio to 4ms at the 50% attacker ratio. The reason for these observations is that when the network is more highly loaded and is in a congested state, the blackhole attacks gradually lift the network out of the congested state; whereas in the more lightly loaded network, the network is not congested and the blackhole attacks discarding data packets does not make significantly more resources available for packet forwarding.

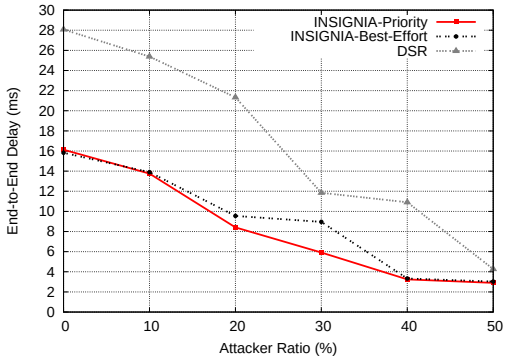
As the network load increases, the difference in the delays experienced by the INSIGNIA priority traffic and the INSIGNIA best-effort background traffic decreases. With the lower load (Fig. 4.5(a)), the priority traffic experiences significantly lower delays than the best-effort traffic; but with the higher load



(a) 10 sources (3 Priority Sources)



(b) 20 sources (6 Priority Sources)



(c) 30 sources (9 Priority Sources)

Figure 4.5: Comparing the End-to-End Delay of INSIGNIA Priority and Best-Effort Traffic and DSR Traffic for a 0 Second Pause Time and 3, 6, and 9 Priority Sources (7, 14, and 21 Best-Effort, Background Sources) Under Blackhole Attacks

(Fig. 4.5(c)), there is little difference between them. This is because INSIGNIA queues priority packets in the best-effort queue when they are not receiving the bandwidth-reserved service. Fewer packets therefore receive the earlier described advantages of being queued in the priority packet queue, and the end-to-end delays of the priority and best-effort traffic exhibit greater similarity. These results demonstrate that it is necessary to prevent the network reaching a congested state to support lower end-to-end delays for priority traffic.

4.2.3 Throughput

This section presents the throughputs achieved with the reservation-based and best-effort approaches under the blackhole, grayhole, and denial of QoS request attacks. Fig. 4.6 shows the throughput vs. offered load for INSIGNIA (combined priority and best-effort sources) and DSR under the setting of 10%, 30%, and

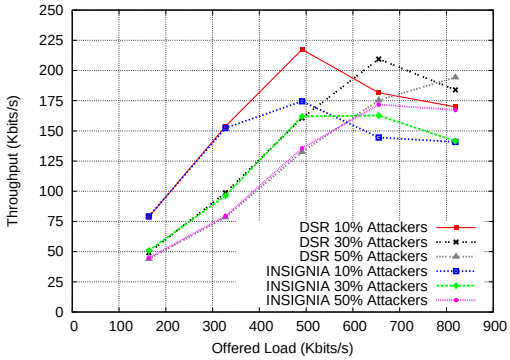


Figure 4.6: Comparing the Throughput and Offered Load of INSIGNIA and DSR for a 0 Second Pause Time Under Blackhole Attacks

50% blackhole attacker ratios. With the 10% attacker ratio, when the offered load increases from approximately 164Kbits/s (10 source nodes) to 328Kbits/s (20 source nodes), DSR and INSIGNIA achieve similar throughputs. At this point the network is not too heavily loaded and there are sufficient resources to support the reservation-based and best-effort forwarding services. However, as the offered load increases, INSIGNIA achieves lower throughputs than DSR. INSIGNIA begins to reach a saturation state. It is becoming less able to support priority packet deliveries as the offer load increases. DSR does not reach a saturation state until the offered load increases to 492Kbits/s (30 source nodes). DSR is therefore able to serve a higher volume of traffic than INSIGNIA at the higher offered loads. INSIGNIA's lower throughput is due to it transmitting more control packets than DSR: as described in Section 4.2.1, these detrimentally affect data packet deliveries.

When the attacker ratio increases to 30% and 50%, the offered load at which DSR begins to achieve higher throughputs than INSIGNIA also increases. For example, with the 30% attacker ratio, DSR begins to outperform INSIGNIA above an offered load of 492Kbits/s; with the 50% attacker ratio DSR begins to outperform INSIGNIA above an offered load of 655Kbits/s (40 source nodes). Thus the offered load at which the two approaches achieve different throughputs increases as a consequence of the increasing attacker ratios: the increasing number of blackhole attackers leads to an increasing number of data packets being dropped; this reduces the congestion and enables INSIGNIA to utilize the available bandwidth to support priority data packet deliveries. However, it is clear from the simulation results that increasing the blackhole attacker ratio decreases

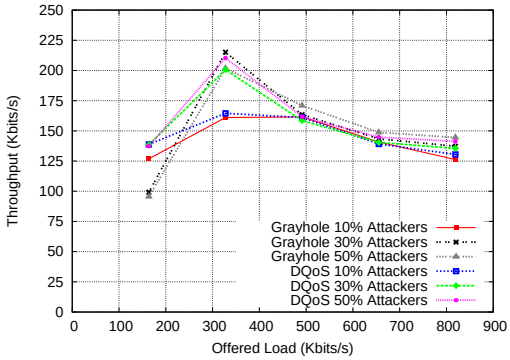


Figure 4.7: Comparing the Throughput and Offered Load of INSIGNIA for a 0 Second Pause Time Under Grayhole and DQoS Attacks

the throughput achieved by both approaches. Additional measures are therefore required to support data packet deliveries in the presence of blackhole attackers.

Fig. 4.7 shows the throughput vs. offered load for INSIGNIA under the grayhole attack and the denial of QoS request (DQoS) attack. Attacker ratios of 10%, 30%, and 50% are used along with a 0 second pause time. As can be seen in the figure, throughput increases as the offered load increases from approximately 164Kbits/s to 328Kbits/s, but it generally decreases as the offered load increases beyond this. The initial upward trend is mainly due to the network having resources available to support the offered load. One observation is that with an offered load of 328Kbits/s, the throughput is lower for both the grayhole and DQoS curves with a 10% attacker ratio than it is for higher attacker ratios. The increase in offered load and the constant node mobility reduce the effective available bandwidth, and this leads to congestion. The 10% attacker ratio means that few nodes perform the grayhole attack. Thus fewer packets are dropped compared with the higher attacker ratios. Consequently, congestion persists as fewer resources are made available to increase the throughput. The DQoS attack does not reduce the number of packets in the network: DQoS attackers downgrade the priority of the packets they forward to request only the best-effort service from downstream nodes. The low throughput arises from packets continuing to suffer the effects congestion which arise from the higher network load.

Another observation is that the throughput increases when the offered load increases from approximately 164Kbits/s to 328Kbits/s and the DQoS attacker ratio increases. Degrading the priority packets to request only the best-effort service means that fewer QoS Reports (a control packet) are transmitted. This

is because a destination node only transmits QoS Reports when packets are delivered to it using the bandwidth-reserved service. The reduced number of QoS Reports means that less time is spent capturing the wireless medium for control packet transmissions. Consequently, more data packets can be delivered to the destination nodes, hence the higher throughput. This finding illustrates that it is important for destination nodes not to inject a large number of control packets containing QoS information into the network as these can adversely affect data packet forwarding. The throughput decreases as the offered load is increased beyond 328Kbits/s. This is because there are insufficient resources available to support priority packet deliveries in this highly loaded and highly mobile network.

4.2.4 Service Quality

Having used the PDR metric to investigate the overall percentages of priority data packets delivered using the reservation-based approach, the next step is to investigate the extent to which this approach delivers these packets using the high-priority, bandwidth-reserved forwarding service. The service quality metric is used for this purpose. Investigating service quality makes it possible to identify the percentages of priority data packets which are delivered to their intended destinations using the bandwidth-reserved and best-effort forwarding services (priority data packets can be degraded to receive only the best-effort service when insufficient bandwidth is available to admit a reservation request). In other words, investigating service quality enables the effectiveness and efficacy of the bandwidth-reserved approach to be determined.

The service quality provided to the priority traffic is plotted in Figs. 4.8–4.11. Fig. 4.8 shows the service quality measured in terms of RePDR (reserved PDR) versus mobility and offered load in an attacker-free environment. These results give an indication of the best achievable service quality in such an environment under the given set of conditions. Figs. 4.9–4.11 show the service quality performances versus various attacker ratios of the blackhole, grayhole, and denial of QoS request attacks, respectively.

Fig. 4.8 plots RePDRs against four pause times for three traffic levels. To put these RePDR values into context the PDR values for this figure are presented in Table 4.1. From Fig. 4.8 it can be seen that when the network load is low (10 sources), the RePDR can largely be maintained above 30%. In other words, approximately 30% of the priority data packets are delivered using the reserved

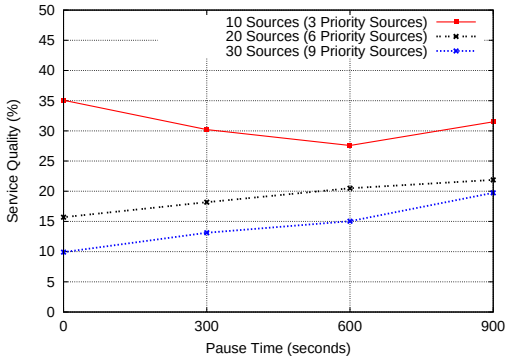


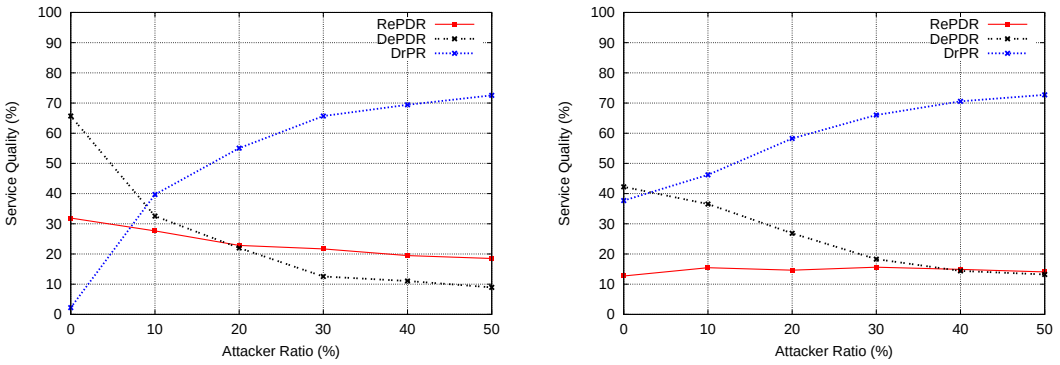
Figure 4.8: The Effects of Mobility and Offered Load on RePDR for INSIGNIA with 3, 6, and 9 Priority Sources (7, 14, and 21 Best-Effort, Background Sources) in an Attacker-Free Network Under a Range of Node Mobilities

forwarding service, whilst the rest of the priority data packets are either degraded and delivered with the best-effort forwarding service or are dropped. There is a slight dip in service quality with the 600 second pause time. The reason for this is the same as for the reduced PDR with this pause time: source nodes are unable to find paths to the destination nodes and this leads to them dropping the outbound priority data packets that they are buffering. As the pause time decreases further to 0 seconds (i.e., when all the nodes are constantly in motion) RePDR reaches the highest level of service quality at 36%. This RePDR is greater than the RePDR achieved when the network is static (900 second pause time). This higher RePDR is due to the nodes making use of the available bandwidth as they move around the network, and the bandwidth from previously reserved flows is freed. This means that the reservation-based adaptation approach is effective in maintaining the service quality in the presence of node mobility, provided that the network is lightly loaded.

However, the effectiveness of this approach weakens as the network load increases. This result is intuitive: when the network gets busier there will be fewer unused paths and less available bandwidth to adapt to. Furthermore, the network load is the major factor influencing the level of QoS supported by the network. For example, when network nodes are static RePDR is just over 32%, on average, for the 10 source node case; this value drops to 23% when the load is doubled to 20 source nodes, and drops to 20% with 30 source nodes. The largest drop in service quality occurs at the highest mobility level. For example, the RePDR value for the 10 source node case is 36%, versus 16% and 10% for the 20 and

Table 4.1: PDRs for the Curves Presented in Fig. 4.8

Pause Time (sec)	10 Sources (%)	20 Sources (%)	30 Sources (%)
0	96	60	34
300	97	86	54
600	95	93	73
900	98	98	79



(a) 10 Sources (3 Priority Sources)

(b) 30 Sources (9 Priority Sources)

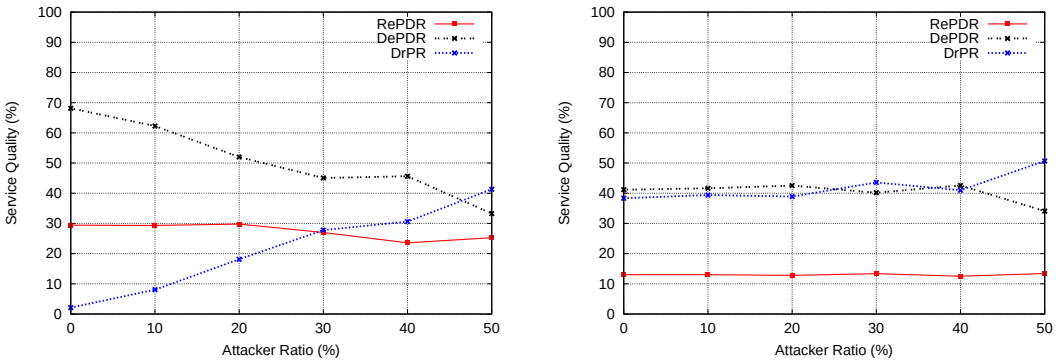
Figure 4.9: Comparing the Effects of Blackhole Attacks on Service Quality for INSIGNIA with 3 and 9 Priority Sources (7 and 21 Best-Effort, Background Sources) with a 300 Second Pause Time

30 source node cases, respectively. This implies that ensuring the network is not overloaded should be one of the important design requirements for MANET QoS solutions. Given these findings, the best RePDR value one could hope for in an adversarial MANET environment is about 30% under the set conditions. This value of 30% will be used as a benchmark value against which the results obtained in adversarial conditions (Figs. 4.9–4.11) are compared.

The results presented in Fig. 4.9–4.11 are for the 300 second pause time in an adversarial network environment containing 10 and 30 source nodes. The justification for this pause time is that the results collected under this condition are the most interesting, as they best demonstrate both the strengths and weaknesses of the reservation-based approach in adversarial network conditions. In particular, one of the observations made under this pause time exposes a shortcoming of the reservation-based adaptation mechanism; and this shortcoming influences one of the design decisions of the solution proposed in Chapter 5. The justification for using 10 and 30 source nodes is that these two settings best demonstrate the effects of light and heavy network loads on service quality.

With a low network load (10 source nodes) and with blackhole attackers, the RePDR decreases as the blackhole attacker ratio increases. As can be seen in Fig. 4.9(a), the RePDR decreases from 32% to 18% as the attacker ratio increases from 0% to 50%. As the attacker ratio is the only factor being altered, the decreasing RePDR is a direct consequence of the blackhole attacks. Another observation on this figure is that there is a marked decrease in the DePDR (i.e., the number of priority data packets receiving the degraded forwarding service) as the attacker ratio increases. In other words, an increasing percentage of the degraded data packets are being dropped. This is reflected in the increasing DrPR (dropped packet ratio). The reason for the higher loss of the degraded data packets is similar to the reason given earlier for the lower PDRs of the traffic receiving the best-effort forwarding service: there are more packets in the network requesting the best-effort forwarding service than there are packets requesting the reserved forwarding service, and this leads to greater contention for resources to service the best-effort traffic; consequently, more best-effort data packets are dropped than reserved packets. As can be seen in the figure, the RePDR appears to be less sensitive than the DePDR to the increasing blackhole attacker ratios. This is because there are fewer data packets receiving the reserved service than the best-effort service, thus there are fewer reserved packets to be affected by the attacks (which is similar to the earlier discussion on the PDRs of the best-effort data packets being affected more than those of the reserved data packets).

When the number of source nodes is increased to 30 in Fig 4.9(b) the service quality of the RePDR curve is largely flat at approximately 14% for all attacker ratios. This is lower than the service quality achieved in the 10 source node case (Fig. 4.9(a)). Thus an effect of the increased network load is that not all of the priority data packets demanding the bandwidth-reserved forwarding service receive it. The reason for this is that some of the data packets requesting resource reservations fail admission control. Admission control may fail due to a lack of available bandwidth, but another reason is related to the occupancy of the priority data packet queue. When a priority data packet arrives at an intermediate node, if the priority data packet queue occupancy exceeds 70%, admission control fails and the packet is instead placed in the best-effort packet queue. An additional effect of this higher network load at the 0% attacker ratio is that the DrPR curve is approximately 39% rather than 2% as in the 10 source node case. The



(a) 10 Sources (3 Priority Sources)

(b) 30 Sources (9 Priority Sources)

Figure 4.10: Comparing the Effects of Grayhole Attacks on Service Quality for INSIGNIA with 3 and 9 Priority Sources (7 and 21 Best-Effort, Background Sources) with a 300 Second Pause Time

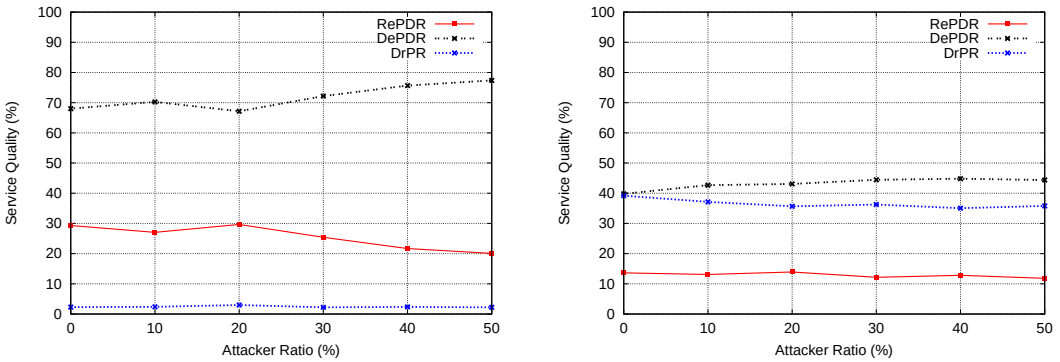
larger percentage of priority data packets being degraded to receive the best-effort service places additional strain on the packet queue servicing the best-effort traffic (which is the same packet queue servicing the best-effort background traffic). The packet queue, and by extension the network, becomes overloaded, and a significant percentage of the priority data packets are dropped. In other words, with no blackhole attacks the service quality, and therefore the QoS, still suffers because of network overloading. As the attacker ratio increases the service quality (and QoS) suffers further, with both the DePDR and the DrPR reaching their worst values: 13% for the DePDR and 73% for the DrPR at the 50% attacker ratio. Under these conditions 73% of the priority data packets are discarded, 13% are delivered with the best-effort service, and 14% are delivered with the reserved service. The main implications of these observations on Fig. 4.9 is that additional measures are needed to support data packet delivery in the presence of blackhole attackers, as single-path adaptation no longer suffices.

Fig. 4.10 presents the results of the grayhole attacks on service quality for 10 sources (Fig. 4.10(a)) and 30 sources (Fig. 4.10(b)), respectively. By comparing Fig. 4.10(a) with Fig. 4.9(a) it can be seen that the grayhole attack reduces service quality less than the blackhole attack. There are two reasons for this. The first is that the grayhole attack only selectively drops data packets (it uses a random drop probability which may be less than 1.0, as described in Section 3.5.2.4), whereas the blackhole attack drops all data packets. The second reason is that fewer dropped priority data packets leads to INSIGNIA using its adaptation facility

more frequently than under the blackhole attack. INSIGNIA's adaptation facility uses *QoS Reports*. These are control packets containing bandwidth information. They are transmitted from a destination node to a source node. Based on these simulation results, 29% more QoS Reports are sent in the presence of grayhole attackers than in the presence of blackhole attackers. This is because a grayhole attacker only selectively drops data packets, thus the destination node can receive some of the packets transmitted to it. The destination node is therefore able to detect QoS fluctuations and will transmit QoS Reports so that the source node can adapt to the measured network conditions. A blackhole attacker, however, will drop all the data packets that it receives. Flow set-up cannot be completed along a new path; the destination node will not have a chance to find out that a stream of data packets dispatched to it from the source node has actually been dropped by a blackhole attacker on the path. It will therefore not transmit QoS Reports, and the source node will not be able to perform end-to-end adaptations.

This finding demonstrates that an important pre-requisite for the single-path adaptation method is that the source and destination nodes must be made aware of QoS fluctuations along the path. The nodes along the path cannot be relied on as (i) they are assumed to be unreliable and untrustworthy and (ii) a MAC-layer packet loss detection method is not applicable here (as this would not provide bandwidth information for adaptation). With the presence of blackhole attackers the likelihood for this pre-requisite to be satisfied is low (the higher the attacker ratio the lower this likelihood) if packets are delivered along a single-path. A multi-path solution with duplicated data packet transmissions may therefore improve this likelihood.

When the number of source nodes increases to 30, the RePDR and DePDR curves are generally flat for all attacker ratios, and they are both lower than in the 10 source node case. As can be seen in Fig. 4.10(b), the RePDR is approximately 13% at all attacker ratios. This is less than half of the RePDR achieved in the 10 source node case (Fig. 4.10(a)). This decrease in RePDR is similar to the decrease in RePDR experienced under the blackhole attack when the number of sources is increased from 10 (Fig. 4.9(b)) to 30 (Fig. 4.9(b)). Similar to the blackhole case, the main factor affecting the RePDR is the congestion and resource contention caused by the higher network load. However, unlike the blackhole case, the overall PDRs (RePDR + DePDR) with 30 source nodes under the grayhole attack are largely similar to those obtained in a non-adversarial network environment (see



(a) 10 Sources (3 Priority Sources)

(b) 30 Sources (9 Priority Sources)

Figure 4.11: Comparing the Effects of Denial of QoS Request Attacks on Service Quality for INSIGNIA with 3 and 9 Priority Sources (7 and 21 Best-Effort, Background Sources) with a 300 Second Pause Time and Blackhole Attackers

Table 4.1). This reinforces the observation that the main factor affecting packet deliveries when there are 30 source nodes is the effects of congestion resulting from the higher traffic load, even in the presence of attackers. One observation which differs from both the 10 source node case and the blackhole case is that the DePDR and DrPR remain relatively flat as the grayhole attacker ratio increases. This is because, under the given conditions, the dominant factor affecting packet deliveries is the effects of network congestion, rather than the grayhole attacks (as the drop probability may be less than 1.0). This contrasts the blackhole case, where the increasing DrPR is due to the combination of the effects of network congestion and the increasing number of blackhole attackers dropping all of the data packets they receive.

Fig. 4.11 presents the results of the DQoS attack on service quality for 10 source nodes (Fig. 4.11(a)) and 30 source nodes (Fig. 4.11(b)). An observation on Fig. 4.11(a) is that the DrPR remains constant at 2% for all attacker ratios whilst DePDR increases and RePDR decreases slightly as the attacker ratio increases beyond 20%. In other words, the DrPR is not affected by the increasing DQoS attacker ratio, whereas the DePDR and RePDR are affected by it. This is because the DQoS attackers do not drop the priority data packets; they downgraded them to request only the best-effort forwarding service. The effects of this can be seen in the figures: the level of increase in DePDR matches the level of decrease in RePDR.

When the number of source nodes is increased to 30 (Fig. 4.11(b)) the service

quality of both the RePDR and the DePDR decrease and the DrPR increases (compared with the 10 source node case). This is a consequence of the effects of congestion resulting from the increase in network load. As can be seen by comparing Fig. 4.11(a) with Fig. 4.11(b), DrPR increases from 2% to 14% when the number of source nodes increases from 10 to 30, respectively. This reinforces the notion that increasing levels of congestion are detrimental to service quality.

4.3 Lessons Learnt

The following lessons have been learnt from the simulation study of the reservation-based (INSIGNIA) and best-effort (DSR) approaches to packet forwarding in MANETs.

- When the network is lightly loaded and free of attackers, the reservation-based approach is capable of supporting a better level of packet delivery than the best-effort approach. This is due to it differentiating the priority and non-priority data packets, and providing the former with an adaptive, bandwidth-reserved forwarding service along a single path and the latter a best-effort forwarding service. In contrast, the best-effort approach provides all packets with the same best-effort forwarding service. Thus the reservation-based approach can, under certain conditions, provide a superior level of QoS to the best-effort approach.

- Using the reservation-based approach generally leads to shorter end-to-end delays than the best-effort approach. Data packets receiving the reserved forwarding service are placed in a higher priority packet queue than those receiving the best-effort forwarding service. The priority queueing mechanism provides an earlier transmission opportunity to the priority data packets. Consequently, the priority data packets experience shorter queuing delays and therefore shorter end-to-end delays than the best-effort data packets.

- As the network becomes more heavily loaded, the delays of the reservation-based approach become increasingly similar to those of the best-effort approach. Thus it is necessary to prevent the network reaching a congested state to support lower end-to-end delays for priority traffic.

- In lightly loaded networks free of attackers, using adaptive bandwidth reservations along a single path generally achieves a RePDR of approximately 30%.

- A source node uses QoS Reports, which are feedback packets, to maximise the percentage of priority data packets receiving the bandwidth-reserved forwarding service. Feedback packets are therefore an important component in the QoS provisioning process: they inform the source node of the recently measured network conditions, and this enables it to make the most appropriate adaptation decisions.

- With the reservation-based approach, a source node's adaptation process is negatively affected when it fails to receive a feedback packet from the destination node, and this affects the achievable QoS. It is therefore necessary to recognise when a feedback packet has failed to be received so that actions to support QoS provisioning can be performed in its absence.

- As the attacker ratio increases, the RePDRs of the reservation-based approach and the PDRs of both the reservation-based and best-effort approaches decrease. Thus forwarding data packets along a single path in the presence of attackers, either with the reservation-based or best-effort forwarding services, is no longer sufficient to support QoS provisioning. Thus additional facilities are required to support QoS as the attacker ratio increases.

- Measures must be taken to ensure that the additional facilities are not used when they are not required or when their use could worsen QoS. For example, in heavily loaded networks, the additional facilities to support QoS should not exacerbate any existing network congestion. This is because it was observed that network congestion hinders QoS in both adversarial and non-adversarial network conditions: the effects of high network congestion on QoS can be more deleterious than a high attacker ratio.

- The additional facilities should therefore be used adaptively in response to network dynamics. One way to guide adaptation is to use feedback from the destination node of a priority data session. However, the overheads of a feedback mechanism should be kept to a minimum. This is because the feedback packets will contribute to the network load.

4.4 Chapter Summary

This chapter presented a simulation study which investigated a reservation-based approach and a best-effort approach to data packet forwarding in MANETs. These two approaches were simulated under a range of node mobilities, network loads, and security attacks. It was observed that the approach of using adaptive bandwidth reservations over a single path provides superior QoS to the best-effort approach when the network is lightly loaded and does not contain attackers. However, as the number of attackers in the network increases, this approach is no longer sufficient to support QoS provisioning. Additional facilities are therefore required to support QoS as the number of attackers increases. Network congestion was another factor identified as having a significant influence on the achievable QoS. Measures must be taken to ensure that the facilities introduced to support QoS provisioning in the presence of attackers do not cause or exacerbate network congestion. The insights gained from this study are used to inform the designs of the QoS solutions presented in Chapters 5 and 6.

Chapter 5

A 2-Dimensional Adaptation ARChitecture (2-DAARC)

5.1 Chapter Introduction

This chapter presents the design and evaluation of a 2-Dimensional Adaptation ARChitecture (2-DAARC). 2-DAARC comprises two novel ideas. The main focus of these novel ideas is to support QoS provisioning in networks containing a variable number of data packet forwarding attackers. The first novel idea is a 2-dimensional approach to adaptation. This approach combines (1) adaptive resource reservations and packet priority assignments over a single path, and (2) data packet duplication over multiple, best-effort paths. The second novel idea is a Priority-based Multi-path Type Selection (PMTS) algorithm. The simulation-based evaluation of 2-DAARC is performed in two stages. First, the 2-dimensional approach is evaluated against INSIGNIA. Second, the PMTS approach is evaluated against a node-disjoint-path-only approach.

The chapter is organised as follows. Section 5.2 presents the 2-dimensional approach to adaptation at a high-level. Section 5.3 presents the PMTS algorithm at a high-level. Section 5.4 presents the design preliminaries for 2-DAARC, including design requirements, design assumptions, and design principles. Section 5.5 presents 2-DAARC in detail. Section 5.6 presents a simulation-based performance evaluation of 2-DAARC. This section also discusses the major findings from the evaluation and provides recommendations to enhance 2-DAARC. Finally, Section 5.7 presents a summary of the chapter.

5.2 A Novel Idea: Two Dimensions of Adaptation

The first novel idea for integrated security and QoS support is to use two dimensions of adaptation: a single-path adaptation (SPA) mode for mobility support, and a multi-path adaptation (MPA) mode to resist data packet forwarding attacks. This section presents the motivation and rationale in taking the 2-dimensional adaptation approach to QoS support. It explains the challenging issues in implementing this approach. An overview is given of the two modes used in this approach, and the need to switch between the two modes to optimize packet deliveries in the dynamic MANET environment is discussed.

5.2.1 Single-Path Adaptation (SPA) Mode for Mobility Support

One way to support QoS in MANETs is to use adaptive bandwidth reservations and packet priority assignments [108]. Node mobility introduces a requirement for bandwidth reservations to be adaptive to better support packet flows' QoS requirements. For example, if an intermediate node with an admitted bandwidth reservation moves and no longer participates in the path between the source and destination nodes of a priority packet flow, it is necessary for two things to happen: (1) the intermediate node should release the reserved resources promptly; and (2) the source and destination nodes should re-establish a bandwidth reservation along a new path as soon as possible. Packet priorities can be changed to downgrade QoS traffic to receive a best-effort forwarding service when the network can no longer support the reserved bandwidth. If sufficient resources later become available the packet priorities may be upgraded to receive the reservation-based forwarding service.

The INSIGNIA QoS framework [108] is an exemplar solution which adopts the above approach. INSIGNIA is designed for QoS provisioning in MANETs where node mobility is the main factor affecting QoS. The simulation study presented in Section 4.2 shows that INSIGNIA, as a facilitator of the above approach, can support preferential treatment for a small percentage of priority traffic in a lightly loaded network which is free of attacker nodes. INSIGNIA generally provides the reservation-based forwarding service to around 30% of priority traffic under a

range of node movement scenarios. However, as observed in Section 4.2, when blackhole attackers are introduced into the network the level of QoS supported by this approach decreases significantly. This is because the blackhole attacker nodes drop packets causing a higher level of packet loss. Additionally, blackhole attackers also affect INSIGNIA's end-to-end adaptation facility. QoS Reports, sent by a destination node to a source node in response to received data packets, are not transmitted when a blackhole attacker is located on the path between the nodes; this prevents the bandwidth adaptation facility from responding to the network conditions.

One way to mitigate the problem of blackhole attacks is to find a path around the attacker nodes. However, this requires (1) the detection of attackers, (2) notifying the source node of the attackers, and (3) the source node finding an alternative path to the destination. Tasks (1) and (2) are open to abuse by both the notifying nodes and the nodes forwarding the notification messages. The notifying nodes must be reliable and truthful. In addition, the nodes that forward the notification message must not be allowed to modify the messages without being detected. Most importantly, any countermeasures to these security concerns and the execution of tasks (1) and (2) will introduce additional computational and bandwidth overheads. The computational and bandwidth resources which would be spent on attacker detection and on processing and securing notification messages could instead be spent on data packet transmissions. A hypothesis of this research is that if there are no or few attacker nodes in the network, the use of the single-path adaptation (SPA) mode may be effective to cope with the bandwidth fluctuations caused by node mobility; however, when there are more attacker nodes and/or when the threat-level in the network is high, duplicated packet transmissions over multiple paths could be used to resist packet forwarding attacks. This latter measure is the multi-path adaptation (MPA) mode.

5.2.2 Multi-Path Adaptation (MPA) Mode for Resisting Packet Forwarding Attacks

The multi-path adaptation (MPA) mode attempts to use data packet transmission redundancy to support QoS by thwarting blackhole attacks. The questions now are (1) how many additional paths should the MPA mode use; (2) under what conditions should the MPA mode be used and under what conditions should the

SPA mode be used; and (3) how to facilitate automatic switching between MPA and SPA in adaptation to the network conditions?

When deciding how many additional paths to use in the MPA mode one should also consider the negative effects the additional load has on the QoS. The work in Section 4.2 has shown that an increase in the network load can reduce the overall percentage of priority data packets which are delivered to their intended destinations; and it can also reduce the percentage of priority data packets which are delivered using the reserved forwarding service, as a higher load means that there is less bandwidth available for resource reservations. Based on these considerations only one additional path is chosen for use in the MPA mode, i.e., when the MPA mode is invoked two paths will be used for each packet transmission. One is the primary path. This is sought using a shortest hop-count path selection mechanism. The other path is the secondary path. This path is selected with reference to the primary path. The process of selecting a secondary path is one of the novel contributions of this thesis, and is described in Section 5.3.

In addition to secondary path selection it is also necessary to consider how to make use of the multiple paths for data packet transmissions. There are two ways to do this. One is to allocate data packets to the two paths alternatively, i.e., two paths are used but not simultaneously; each data packet is only transmitted along one of the two paths in a round-robin fashion. So if one path contains a blackhole attacker then only half of the packets will be lost. The other method is packet duplication. This method duplicates each data packet and transmits the copies along the primary and the secondary paths simultaneously. In this way, even if one copy is lost or is dropped in an attack, there is still another copy to be delivered and used by the destination node. Obviously, this duplication strategy is more costly in terms of bandwidth consumption [34], but it does provide a higher level of redundancy, and thus reliability (under certain network conditions), in comparison with the first method. In light of this consideration, the data packet duplication method is used in the proposed solution.

5.2.3 Switching between the SPA and the MPA Modes

As the MPA mode introduces additional traffic load into the underlying network it should only be used when it can help to improve QoS support. This means that under certain network conditions the SPA mode can bring a better QoS performance, while under other network conditions the MPA mode does better.

This requires (1) the identification of factors affecting network conditions and an assessment of how they influence the changes in the network conditions; and (2) an investigation on how to measure network conditions and how to use these measurements to facilitate dynamic switching between the SPA and the MPA modes to optimize QoS support. At a high-level, the solution should switch from the SPA mode to the MPA mode when the attacker ratio is high and the SPA mode can no longer satisfy the QoS requirements of a packet flow. Conversely, the solution should switch back to the SPA mode when either (1) the MPA mode starts to bring negative effects on QoS or (2) when the network risk-level has reduced such that the MPA mode is no longer necessary.

5.3 A Novel Idea: Priority-based Secondary Path Selection

The second novel idea in this thesis is to select a secondary path relative to the primary path to enhance the reliability of multi-path routing. Intuitively, paths with the maximum redundancy should provide the highest reliability. Disjointedness is a measure of redundancy between two paths. There are three levels of redundancy: the highest level is *node-disjoint*, where two paths do not share any nodes or links (except the source and the destination nodes); the second level is *link-disjoint*, where two paths share nodes, but not links; and the lowest level is *non-disjoint*, where paths share nodes and links. Node-disjoint paths offer the highest aggregate resources and provide the highest fault tolerance [130]. For example, a single blackhole attacker cannot simultaneously attack more than one path between a source node and a destination node if node-disjoint paths are used, whereas with link- and non-disjoint paths a blackhole attacker may be able to attack both paths simultaneously. Additionally, node-disjoint paths offer greater fault tolerance because they typically have a longer lifetime [72], are more stable [168], and are less likely to break [111] than link- or non-disjoint paths. Node-disjoint paths should therefore be selected whenever they are available.

If node-disjoint paths are not available when needed the source node may perform a route discovery operation until one is found. This path selection approach is hereafter referred to as a *node-disjoint-path-only* (NDO) approach. The NDO approach requires the source node to perform a Route Discovery operation every time a node-disjoint path is required but is not available, even if the source node

is aware that there are link- and non-disjoint paths available to the destination node. An alternative approach is to use all the path types but in a priority-based manner. That is, when a secondary path is required the source node will first select a node-disjoint path. If this path type is not available it will select a link-disjoint path. If neither a node-disjoint nor a link-disjoint path is available it will go for a non-disjoint path. If none of these paths are available the source node will perform a Route Discovery operation. This priority-based approach means that a route discovery operation is performed only when the source node has exhausted all known paths to the destination node, and not just those paths with one particular level of disjointedness.

5.4 Design Preliminaries

Before presenting the detailed design of 2-DAARC, the design requirements, design assumptions, and design principles are described next.

5.4.1 Design Requirements

The section presents four design requirements for 2-DAARC. These requirements are divided into *security requirements* and *performance requirements*. The security requirements are as follows.

- **(S1)** To protect integrity of the feedback data. The feedback data are sent from a destination node to a source node. This protection ensures that any unauthorised modification of data can be detected.
- **(S2)** To provide origin authentication of the feedback data. This is to assure a source node that these data have originated from the claimed node.

The performance requirements are as follows.

- **(P1)** The use of path redundancy should enhance QoS performance.
- **(P2)** To minimize the number of operations imposed on intermediate nodes. This is because intermediate nodes are not trustworthy, and their involvement may be counter-productive in the solution.

5.4.2 Design Assumptions

The following three assumptions are made in the 2-DAARC design.

- **(A1)** The keyed-hash function, SHA-224 [141], is secure.
- **(A2)** The source and destination nodes of a priority data packet session trust each other, and are truthful to each other, and they have a pre-shared symmetric key.
- **(A3)** Nodes' batteries last for the duration of a session.

5.4.3 Design Principles

The following four measures are taken in the 2-DAARC design to satisfy the design requirements specified in Section 5.4.1.

- **Measure 1:** A message authenticity service is used to provide integrity protection and origin authentication of the feedback data. A keyed-hash, SHA-224 [141], is used with a 112-bit symmetric key. This key is shared between only the source and destination nodes of a priority packet flow. This Measure satisfies security requirements (S1) and (S2).
- **Measure 2:** 2-DAARC switches between two modes, the SPA mode and the MPA mode, in response to changes in network conditions. This aims to ensure that the MPA mode, and therefore multi-path routing and data packet duplication, are only enabled when necessary. This Measure satisfies performance requirement (P1).
- **Measure 3:** A source node uses feedback data received from a destination node to govern adaptive actions in response to changes in network conditions. Feedback data can be gathered in the form of QoS statistics. These statistics may indicate the likely presence of a blackhole attacker on the path(s) between the source and destination nodes. The path(s) can be changed in response to these statistics. This Measure satisfies performance requirement (P2), as the feedback and adaptation operations do not increase the number of operations performed by intermediate nodes.

- **Measure 4:** Source nodes use a timeout to perform adaptive actions in the absence of feedback data. If feedback data are not received before the expiration of a timeout a new path/paths is/are selected.

5.5 2-DAARC in Detail

The architecture, as shown in Fig. 5.1, consists of three parts: a sender entity, an intermediate node entity, and a destination node entity. The sender entity contains eight components. One of these components is an application layer component, and the rest are network components.

- The application layer component is a *Profile*. This specifies a number of parameter values used in both session initialisation and session adaptation.
- The first of the network layer components is the *Path Quantity Calculator*. This is where adaptation decisions are made to determine the number of paths required to support the QoS requirements of a packet flow. At session initialisation, the number of paths to used is determined based on the parameter values specified in the Profile.
- The *INSIGNIA* component is an extension of the INSIGNIA QoS framework. It is extended to interact with three components: the *Feedback Handler*, the Path Quantity Calculator, and the *DSR* protocol.
- The Feedback Handler determines whether or not a received feedback packet should be used in making an adaptation decision. On receipt of a feedback packet, the Feedback Handler uses the shared key stored in the Key Store and the hash function housed in the Crypto component to compute a fresh digest of the feedback data. This freshly generated digest is then compared with the one carried in the feedback packet. If the two digests are the same the feedback is regarded as authentic, and it will be used by the Path Quantity Calculator to make the adaptation decisions. Otherwise the feedback data will be dropped.
- The DSR component is an extension of the DSR protocol. The extension includes a novel secondary path selection mechanism, multi-path routing, and data packet duplication. The novel path selection mechanism is named as Priority-based Multi-path Type Selection (denoted PMTS). Primary paths

are selected using DSR's existing path selection mechanism. This mechanism selects a path with the shortest hop-count to the destination node

The intermediate node entity contains two network layer components: DSR and INSIGNIA. No changes are made to the way intermediate nodes handle data and control packets. In other words, intermediate nodes behave as specified in the DSR and INSIGNIA specifications.

The destination node entity contains seven network layer components.

- The *DSR* component handles data and control packets as specified in the original protocol.

- The *INSIGNIA* component receives data packets from the DSR component and it interfaces with the *Statistics Collector* and *Feedback Generator* components. INSIGNIA QoS Reports are passed to the Feedback Generator where they are protected using a message authenticity service. This process is described below for the *Feedback Generator* module.

- The *Statistics Collector* component monitors and logs the arrival of every data packet for use in QoS statistics calculations. QoS statistics describe the quality of the path(s) used between the source and the destination nodes. The statistics are generated periodically from the data logs maintained by this component.

- The *Duplicate Suppressor* component detects and suppresses duplicate data packets which are not required at the application layer (data packets are duplicated when the MPA mode is active, but only one copy of each packet is required at the application layer). Non-duplicate data packets are passed up to the application layer. The number of received duplicates is passed back to the Statistics Collector for use in QoS statistics calculation.

- The QoS statistics are passed to the *Feedback Generator* which protects them with a message authenticity service prior to their transmission to the source node in a feedback packet. The message authenticity service gives integrity protection and origin assurance of the feedback data. Message authenticity is provided via a keyed-hash function. The Feedback Generator interacts with the *Key Store* and the *Crypto* components to generate a keyed-hash of the feedback data. The keyed-hash is inserted into a feedback

(control) packet along with the QoS statistics/QoS Report before sending the packet to the destination node.

The following sections describe how the SPA and MPA modes of adaptation extend INSIGNIA and DSR. Following this, the architectural components are described in detail.

5.5.1 The SPA Mode: Building on INSIGNIA

The SPA component of 2-DAARC is built on INSIGNIA. The INSIGNIA 'in-band' QoS signalling system [105], which uses the IP Options header of each data packet, is extended to carry options to support dynamic adaptation. The INSIGNIA single-path adaptation process is also extended to adapt to service quality statistics: if the percentage of packets delivered using the reserved forwarding service falls below a threshold value, the path is changed. The aim of this adaptation process is for the SPA mode to deliver a greater percentage of priority data packets using the reserved forwarding service.

One of the main design considerations is on how to minimize the bandwidth and processing costs introduced by the SPA component. 2-DAARC's additional options, combined with INSIGNIA's original options, are represented in fewer than 32-bits. This is important because the IP Options field must end on a 32-bit word boundary [41]. If the options total more than 32-bits, it would be necessary to add padding up to the next 32-bit word boundary. In this case, the IP Options would have a length of 64-bits (8-bytes). This would be an inefficient use of resources if there is more padding than there are options. For example, if 2-DAARC's options exceeded the 32-bit word boundary by 1-bit, 31-bits of padding would be required, and approximately 48% of the IP Options header will be padding bits. If this is the case, every data packet would be 4-bytes larger. This will increase the bandwidth consumption of the SPA mode. Moreover, this process is even more wasteful when the MPA mode (described in the following section) is activated. The data packets transmitted in the MPA mode also contain the IP Options header. The MPA mode duplicates the data packets, thus each packet will contain the additional 4-byes, and 8-bytes for each duplicated transmission.

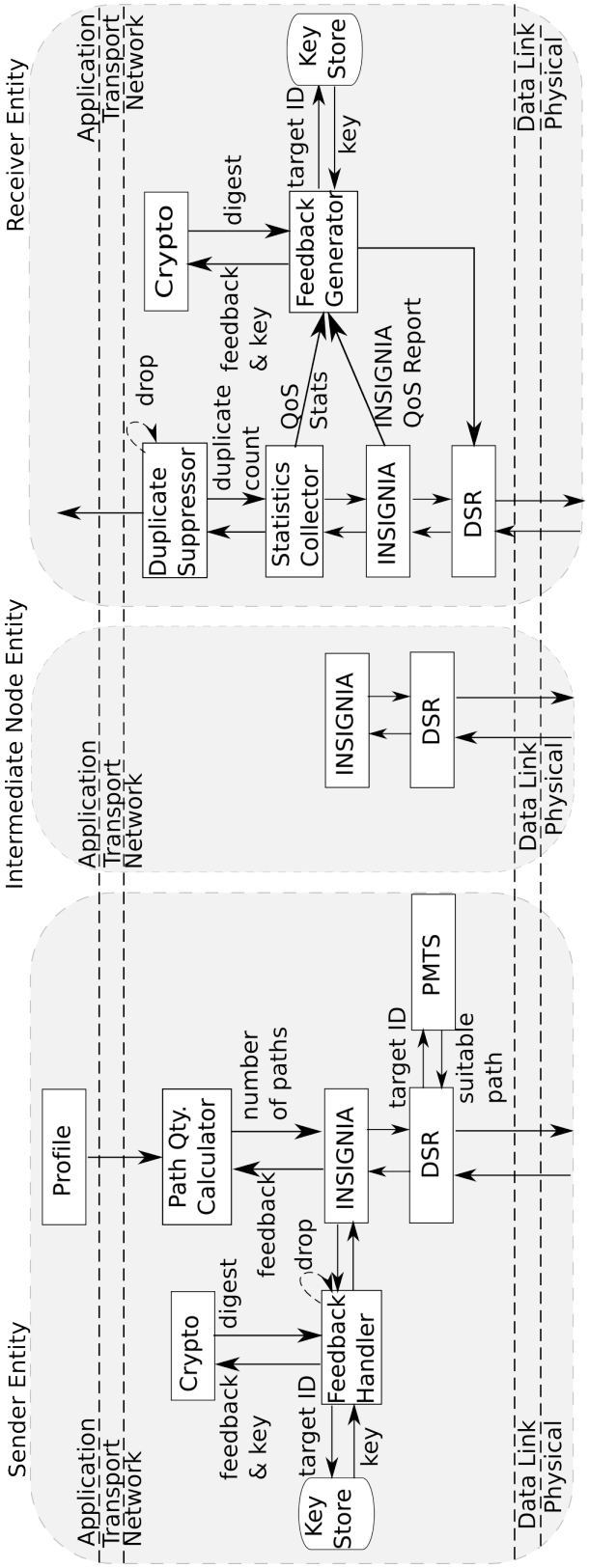


Figure 5.1: The 2-Dimensional Adaptation ARChitecture (2-DAARC)

5.5.2 The MPA Mode: Building on DSR

The MPA mode component of 2-DAARC is built on the DSR protocol. Though DSR is a single-path routing protocol (it is described in detail in Section 3.2.2), it discovers and caches multiple paths to a destination node as part of its default Route Discovery process. These additional paths can be utilized for multi-path routing. DSR is extended to support multi-path routing by designing and embedding into DSR a novel Priority-based Multi-path Type Selection (PMTS) algorithm for secondary path selection, and duplicating priority data packets along the multiple (primary and secondary) paths.

Two measures are taken to minimize the bandwidth and performance costs introduced by the MPA component. The first measure is to make use of the paths already discovered before initiating a new Route Discovery process, i.e., to let the PMTS algorithm calculate the disjointedness of the paths cached in the source node's Route Cache (which are already discovered via DSR's Route Discovery mechanism during primary path discovery). In other words, this approach to priority-based path selection does not inject additional control packets into the underlying network for secondary path selection unless the Route Cache is empty. The second measure is to let the source node calculate path disjointedness and to select the secondary path. This prevents the establishment of trust with intermediate nodes or taking additional measures to ensure such calculations are trustworthy. This measure also allows the source node to switch between the two modes (SPA and MPA) with very little delay.

5.5.3 Supporting Security and QoS Requirements

2-DAARC has three components to capture and to support security and QoS requirements. The first component is the 2-DAARC application layer profile. This is presented in Section 5.5.3.1. The second component is the PMTS algorithm, which is used when the MPA mode is active. This is described using a worked example in Section 5.5.3.2. The third component is the duplication of data packets across the primary and the secondary paths. This is described in Section 5.5.3.3.

5.5.3.1 Application Layer Profile

Source nodes use an application layer profile to specify values for QoS and adaptation parameters which are then used during session initialization and for adaptation an on-going session. (The application layer profile is the Profile component in the sender entity of the architecture diagram in Fig. 5.1.) Fig. 5.2 shows the parameters specified in the application layer profile. 7 out of the 12 parameters are original INSIGNIA parameters. The remaining 5 parameters are introduced for 2-DAARC. The INSIGNIA parameters are explained as follows. The *source port number* parameter indicates the outbound/inbound port numbers used by priority and best-effort traffic. For example, priority traffic is transmitted and received on port #1. Best-effort traffic is transmitted and received on port #0. The *packet size* (specified in bytes) and the *packet generation interval* parameters are used to calculate the minimum and maximum bandwidth demanded by a session. The packet generation interval is also used in 2-DAARC to calculate the number of priority data packets per second originated at the source node. This is required for QoS statistic calculations by the destination node (described in detail in Section 5.5.4.2). The ratios of *base QoS* (BQ) and *enhanced QoS* (EQ) specify the ratio of outbound priority packets to demand the minimum bandwidth requirement (BQ) and the maximum bandwidth requirement (EQ). The *adaptation parameter* is used to determine when to scale-up or scale-down a reserved packet flow, i.e., to determine when to increase or to decrease the bandwidth requested for resource reservations. The *adaptation granularity* parameter is used in a series of calculations to determine and to adapt to the available bandwidth during a session. The EQ, BQ, adaptation parameter, and adaptation granularity parameters are explained in detail in Section 3.5.2.3.

The five parameters used by 2-DAARC are *adaptation mode*, *message digest size*, *packet loss threshold* ρ, *SPA packet loss threshold* ϕ, and *SPA RePDR Threshold* Γ. The *adaptation mode* parameter specifies the mode of adaptation to use during a session. It takes one of two values. The first value is 0. This specifies that dynamic 2-dimensional adaptation should be used, i.e., 2-DAARC should adapt between the SPA and the MPA modes in response to network conditions. The second value is 1. This specifies that only the SPA mode should be used during a session. The *message digest size* parameter specifies the size of a digest to be generated by the keyed-hash function. The digest is used to provide integrity protection and origin authentication of feedback data transmitted from

	Profile Parameter
Original INSIGNIA Profile Options	Source Port Number
	Packet Size (bytes)
	Packet Generation Interval
	Ratio of Base QoS (BQ)
	Ratio of Enhanced QoS (EQ)
	Adaptation Parameter
	Adaptation Granularity
2-DAARC Profile Options	Adaptation Mode
	Message Digest Size
	Packet Loss Threshold (ρ)
	SPA RePDR Threshold (Γ)
	SPA Packet Loss Threshold (ϕ)

Figure 5.2: Application Layer Profile

a destination node to a source node. This process is described in Section 5.5.5.1. There are 3 adaptation parameters used for adaptation within and between the SPA and the MPA modes. The first parameter is *packet loss threshold*, ρ. This is used to adapt within and between modes in response to the level of observed packet loss on the in-use path(s). The second parameter is *SPA RePDR threshold* threshold, Γ, which is used for adaptation within the SPA mode. It is used to maximise the number of packets delivered using the reserved forwarding service. The third parameter is *SPA packet loss threshold*, ϕ. This is used to adapt to packet loss within the SPA mode. Section 5.5.5.2 describes how these parameters are used for dynamic adaptation. The values for these three adaptation parameters are determined using simulation in Section 5.6.1.

5.5.3.2 Priority-based Multi-path Type Selection (PMTS)

The MPA mode's secondary path selection mechanism uses a Priority-based Multi-path Type Selection (PMTS) algorithm to find a secondary path which is maximally disjoint with the primary path. Path disjointedness is described in Section 2.8.1. The terms *primary path* and *secondary path* are used only to distinguish the order of path selection. They are not used to differentiate between a main path and a back-up path, as both paths are used simultaneously to transmit duplicated data packets. The paths stored in the Route Cache (excluding the primary path) are referred to as *candidate paths*.

Two path selection criteria are used for secondary path selection: (1) the level

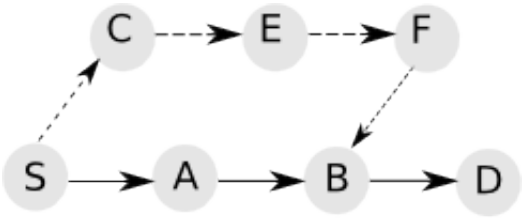
Path ID	Source Route
1	{S, A, B, D}
2	{S, C, E, F, B, D}
3	{S, H, I, J, D}
4	{S, K, A, L, F, D}

Table 5.1: Example Paths in a DSR Route Cache

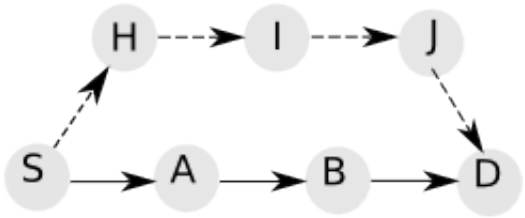
of disjointedness with the primary path and (2) the hop-count (path length). The former criteria has priority over the latter. In other words, a path with greater disjointedness is preferred to a shorter path. If there is more than one path with the same level of disjointedness, the one with shortest hop-count is preferred. The MPA mode has two components to find the most appropriate path: (1) the PMTS algorithm, which calculates the disjointedness between a candidate path and a primary path; and (2) a method for Multi-path Type Selection (MuTS), which determines whether the disjointedness and hop-count of a candidate path is the best (highest disjointedness, lowest hop-count) of the paths stored in the Route Cache.

A worked example is used to explain secondary path selection using the PMTS algorithm and the MuTS method. The example uses the paths listed in Table 5.1. The paths in the table are a selection of path entries extracted from a Route Cache. This means that the example is based on actual, rather than notional, data. These paths were obtained during a single Route Discovery operation from a source node S to a destination node D. The nodes in the paths are given alphabetical identifiers to simplify the discussion. In Table 5.1, the primary path is the first entry, where Path ID = 1. Paths 2–4 are candidate paths for the role of the secondary path. Figure 5.3 provides a graphical representation of the disjointedness of these candidate paths with the primary path. In the figure, the arrows with a solid line represent the wireless links between the nodes on the primary path. The arrows with a dashed line represent the wireless links between the nodes on the candidate paths. This example assumes that the primary path (Path ID = 1) has already been selected using the DSR protocol's shortest hop-count path selection mechanism.

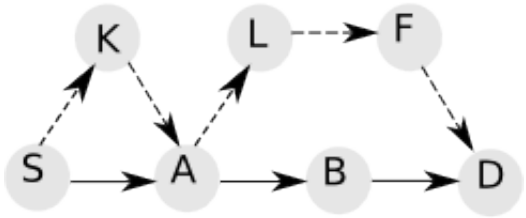
The pseudocode for PMTS is shown in Algorithm 5.1. The algorithm uses four variables: *nonDisjoint*, *linkDisjoint*, *noOfSharedNodesAndLinks*, and *preferredDisjointedness*. The *nonDisjoint* varaible is used to maintain a count of the number of hops (common nodes and link) shared between the candidate path and



(a) Path ID = 2: Non-Disjoint Path. Paths SABD and SCEFBD have node B and link $B \rightarrow D$ in common.



(b) Path ID = 3: Node-Disjoint Path. Paths SABD and SHIJD have no common nodes or links.



(c) Node ID = 4: Link-Disjoint Path. Paths SABD and SKALFD have node A in common, but no common links.

Figure 5.3: Three Types of Secondary Path Disjointedness

the primary path. The *linkDisjoint* variable is used to maintain a count of the number of common nodes shared between the candidate path and the primary path. The *noOfSharedNodesAndLinks* variable is used to maintain a count of the overall number of nodes and links shared between the candidate path and primary path. The *preferredDisjointedness* variable specifies the level of disjointedness being sought. It is initially set to 'any disjointedness'.

The PMTS algorithm works as follows. The algorithm compares the first candidate path in the Route Cache with the primary path. This candidate path is the first path in the Route Cache which is not the primary path. In this example, this is Path ID = 2 in Table 5.1. Figure 5.3(a) shows this candidate path with the primary path. PMTS compares each node of the candidate path with each node of the primary path. If a common node is found, the previous node in both paths is compared. If the same previous node is shared, this hop is non-disjoint, as the same node and link are shared. In this case, the *nonDisjoint* and *noOfSharedNodesAndLinks* counters are incremented. If the same previous

node is not shared, the hop is link-disjoint. In this case, the *linkDisjoint* and *noOfSharedNodesAndLinks* counters are incremented. In this example, node B is shared by both paths, but the previous nodes (A and F) are distinct; this hop is therefore link-disjoint. The algorithm continues until it finds the next shared node, where it repeats the above process of comparing the previous nodes. When the destination node is reached, the algorithm compares the penultimate nodes in the paths to determine whether the final hop is non-disjoint or node-disjoint: the final hop cannot be link-disjoint, as either the same node and therefore same link are used or different nodes and different links are used; a link-disjoint hop requires the same node but a different link, but this is not possible in the final hop. In the example, the last hop of both paths is $B \rightarrow D$. This hop is therefore non-disjoint, as both paths share the same previous node and wireless link. The presence of a single non-disjoint hop makes the candidate path non-disjoint with the primary path.

The next step in the secondary path selection process is to use the MuTS method to determine whether the candidate path's disjointedness and hop-count are the best of the paths retrieved from the Route Cache so far. The best candidate path is the path with the highest disjointedness, the least shared nodes and links (if not node-disjoint), and the shortest hop-count. Pseudocode for the MuTS method is shown in Algorithm 5.2. MuTS compares the candidate path with the previous best path (or default values if this is the first candidate path retrieved from the Route Cache). The candidate path is set as the best path if one of two conditions is satisfied: (1) it has a higher disjointedness than the previous best path (or higher disjointedness than the default disjointedness value); or (2) it has the same disjointedness but a shorter hop-count than the best path (or the default hop-count value). The default disjointedness value is 'non-disjoint'. The default hop-count value is 16, which is the maximum source route length specified by the DSR protocol [29]. Path disjointedness has precedence over path hop-count; thus a longer path with greater disjointedness is preferred to a shorter path with lower disjointedness. This is so that the redundancy of the path is not sacrificed for a shorter path which offers less redundancy. In the example, the candidate path (Path ID = 2) is set as the best path as it is the first path to be compared with the primary path: it has the same disjointedness but a lower hop-count than the default values. The *preferred disjointedness* variable is updated if the disjointedness of the path is greater than the disjointedness value of this

Algorithm 5.1: Pseudocode for the Priority-based Multi-path Type Selection (PMTS) Algorithm

```
Input: Candidate path from the Route Cache
Output: Disjointedness of the candidate path

for node i ∈ primaryPath do
    for node j ∈ candidatePath do
        if primaryPath(i) == candidatePath(j) and j ≠ sourceNode then
            if primaryPath(i-1) == candidatePath(j-1) then
                if preferredDisjointedness == ANY_DISJOINT then
                    nonDisjoint++;
                    noOfSharedNodesAndLinks++;
                else
                    return false ; // non-D found, node-/link-D wanted
            else
                if i ≠ primaryPathLen and j ≠ candidatePathLen then
                    if preferredDisjointedness ≠ NODE_DISJOINT then
                        linkDisjoint++;
                        noOfSharedNodesAndLinks++;
                    else
                        return false ;    // link-D found, node-D wanted
if nonDisjoint ≠ 0 then
    currentPathDisjointedness ← NON_DISJOINT;
else if nonDisjoint == 0 and linkDisjoint == 0 then
    currentPathDisjointedness ← NODE_DISJOINT;
else currentPathDisjointedness ← LINK_DISJOINT;
return true;
```

variable. In this example the variable is left unchanged as the path's disjointedness (non-disjoint) is the same as the value of the variable (the default value, i.e., non-disjoint).

The secondary path selection mechanism now uses the PMTS algorithm and the MuTS method on the remaining paths in the Route Cache. The next candidate path is Path ID = 3, and is shown in Figure 5.3(b). The PMTS algorithm calculates that this path is node-disjoint with the primary path, as the two paths do not share any nodes or links. The MuTS method sets this as the best candidate path so far, as the previous best path (ID = 2) has a lower disjointedness (non-disjoint). The *preferred disjointedness* variable is updated as a path with

Algorithm 5.2: Pseudocode for Multi-path Type Selection (MuTS) Method.

Input: pathID, currentPathDisjointedness, noOfSharedNodesAndLinks, hopCount
Output: preferredDisjointedness

```
if ((noOfSharedNodesAndLinks < lowestNoOfSharedNodesAndLinks) or
(noOfSharedNodesAndLinks == lowestNoOfSharedNodesAndLinks and
hopCount < lowestHopCountSoFar)) then
    setBestCandidatePath(pathID, hopCount,
    noOfSharedNodesAndLinks);
    if currentPathDisjointedness > preferredDisjointedness then
        preferredDisjointedness ← currentPathDisjointedness;
        lowest hop count so far ← MAXIMUM_SOURCE_ROUTE_LEN;
```

a higher disjointedness level has been found. It is set to 'node-disjointedness', which is the disjointedness of this candidate path. Consequently, only a shorter node-disjoint path can now be set as a better candidate path. The next candidate path is Path ID = 4, and is shown in Figure 5.3(c). It is link-disjoint with the primary path as node A is present in both paths but no links are shared. The PMTS algorithm ceases execution on finding this common node. This is because this path offers lower disjointedness than that which is sought by the *preferred disjointedness* variable (node-disjoint). The best path is therefore not updated. As this is the last path in this example Route Cache (Table 5.1), the previously determined best path, Path ID = 3, is selected for use by the MPA mode. The PMTS algorithm and the MuTS method are contained in the PMTS component in the sender entity of the architecture diagram shown in Fig. 5.1.

5.5.3.3 Data Packet Duplication Over Multiple Paths

Having discovered multiple paths to the destination node, data packets are duplicated and transmitted across the primary and the secondary paths. (Data packet duplication occurs in the DSR component in the sender entity of the architecture diagram in Fig. 5.1.) Data packet duplication only takes place at the source node. This design decision is made for three reasons. First, only the source node has the knowledge of the two paths. (The destination node also has knowledge of the two paths, but it does not duplicate packets). Second, it avoids having to

execute the path selection and data packet duplication mechanism at the intermediate nodes. Intermediate nodes are therefore only required to forward the data packets that they receive. They do this using the existing functionality of the underlying DSR protocol. Third, allowing intermediate nodes to duplicate data packets may lead to a growing rate of packets poured into the network; and this may lead to congestion and decreased QoS.

The payload and header data of the two copies of a packet are the same with one exception. The *packet status* option in the 2-DAARC IP Options header (described in Section 5.5.4.1) is set differently for the original and the duplicated packets: the packet transmitted along the primary path is marked as an OR (original) packet; the packet transmitted on the secondary path is marked as a DUP (duplicated) packet. This is so that destination node can observe the number of packets received along each path to capture contextual information about the quality of the paths. The 2-DAARC IP Options header and the capturing of the contextual information are described in the following section.

5.5.4 Capturing Contextual Information

2-DAARC captures contextual information to enable adaptation to measured network conditions. Contextual information is captured at the destination node in the form of QoS statistics. Data about the session configuration is required from the source node to calculate these statistics. These data are conveyed from the source node to the destination node using the IP Options field of data packets. This is described in Section 5.5.4.1. The process of calculating the QoS statistics is presented in Section 5.5.4.2. It is necessary to prevent any QoS statistics calculated at higher layers, e.g., the transport layer, from being negatively affected by the receipt of duplicate data packets at the destination node. 2-DAARC uses a duplicate data packet detection and suppression mechanism to avoid this. This mechanism is described in Section 5.5.4.3.

5.5.4.1 Conveying Parameter Values using IP Packets

To facilitate dynamic adaptation during a session it is necessary for the source node to communicate a number of parameter values to the destination node. INSIGNIA's 'in-band' QoS signalling system [105] is extended to achieve this. The signalling system makes use of the IP Options header of each data packet

Option	Service Mode	Payload Type	Bandwidth Indicator	Bandwidth Request	Padding
Value	RES/ BE	BQ/ EQ	MAX/ MIN	MAX/ MIN	-
Length	1 bit	1 bit	8 bits	16 bits	6 bits

Figure 5.4: The INSIGNIA IP Options Header in NS-2

Option	Service Mode	Payload Type	Bandwidth Indicator	Bandwidth Request	Generation Rate	Packet Type	Packet Status	Padding
Value	RES/ BE	BQ/ EQ	MAX/ MIN	MAX/ MIN	Packet Generation Rate	SPA/ MPA	OR/ DUP	-
Length	1 bit	1 bit	1 bit	16 bits	8 bits	1 bit	1 bit	3 bits

Figure 5.5: The 2-DAARC IP Options Header

to convey QoS information. INSIGNIA's original IP Options header is shown in Fig. 5.4. The 2-DAARC value setting in the IP Options header is shown in Fig. 5.5. The first four options in the 2-DAARC IP Options header are the original INSIGNIA header fields [108]. Three extra options are added. These are used to convey source node parameter values required by the destination node to calculate QoS statistics (described in the following section). The *generation rate* option holds the source node's data packet generation rate. The *packet type* option specifies if the packet is originated by the SPA mode or the MPA mode. The *packet status* option indicates if the packet is an original or duplicated data packet. Padding ensures that the IP Options field ends on the 32-bit word boundary; the IP Options field must end on a 32-bit word boundary [41].

Extending the IP Options field to support the *generation rate*, *packet type*, and *packet status* options requires modification of the original INSIGNIA Options. The INSIGNIA Options are shown in Figure 5.4 (this figure is based on the INSIGNIA Options field in the NS-2 implementation [106], not the INSIGNIA specification [108]). The INSIGNIA IP Options field is 32 bits in length. 26 bits are used for the INSIGNIA IP Options with 6 padding bits. There exists a discrepancy between the specification [108] and the NS-2 implementation [106] of the INSIGNIA Options. The specification states that 1-bit is used for the *bandwidth indicator* option, but the NS-2 implementation uses 8-bits. This changes the amount of space occupied by the INSIGNIA Options from 19-bits (as per the specification) to 26-bits (as per the implementation). Padding is necessary in both cases to fill the Options field up to the 32-bit word boundary. The three

options introduced by 2-DAARC require 10-bits: 8-bits for the *generation rate*, 1-bit for the *packet type*, and 1-bit for the *packet status*. Adding these to the existing INSIGNIA Options will overflow the 32-bit word boundary by 4-bits, as 26-bits (INSIGNIA) + 10-bits (2-DAARC) = 36 bits. This would require 28-bits of padding to reach the next 32-bit word boundary, and this level of padding is required for every data packet. In this case, approximately 40% of the IP Options header would be padded. This is clearly not ideal in the resource limited MANET environment. The INSIGNIA NS-2 implementation is therefore modified to use a 1-bit representation for the *bandwidth indicator* option. The revised INSIGNIA Options total 19-bits. This makes space for the *generation rate*, *packet type*, and *packet status* options without overflowing the 32-bit word boundary. When the *generation rate*, *packet type*, and *packet status* options are added, the size of the Options field is 29-bits. 3-bits of padding are use to fill the Options field up to the 32-bit word boundary. The revised IP Options field is shown in Figure 5.5.

5.5.4.2 Calculating QoS Statistics

Routing adaptation performed by a source node is governed by contextual information received from the destination node. The contextual information is acquired at the destination node in the form of QoS statistics. (The QoS statistics are calculated in the Statistics Calculator component in the receiver entity of the architecture diagram shown in Fig. 5.1.) Every data packet arriving at the destination node is logged so that QoS statistics can be calculated. To calculate these QoS statistics the destination node must know the source node's packet generation rate. So the packet generation rate is discussed before the QoS statistics calculations are described.

A source node inserts its packet generation rate into the *generation rate* option in the IP Options field of every data packet. There are two reasons for inserting the generation rate into data packets rather than using separate control packets. First, it can prevent the introduction of additional overhead into the underlying network. This is because additional control packet transmissions are not required. Second, it aims to provide resistance against packet loss due to accidental or malicious reasons. For example, a control packet may be lost during transmission or dropped by attackers, thus preventing the destination from calculating the statistics. Including the generation rate in each data packet provides some redundancy for this vital information, allowing it to be delivered to

the destination in a more reliable manner. The generation rate, G, is calculated as 1/Generation Interval. The Generation Interval of a packet flow is specified as part of a source node's application-layer profile. Knowledge of the generation rate enables the destination node to determine how many packets should be received over a time-interval τ.

Four QoS statistics are calculated by the destination node, and these are derived from a single metric which is used to determine the packet loss ratio. This packet loss ratio (PLR) metric is the ratio of the number of data packets lost during transmission to the total number of data packets transmitted by the source node over a time-period τ. It is calculated based on the assumption that all of the received data packets are of the same type. The PLR is calculated using Eq. 5.1,

$$PLR = \frac{G \times \tau - P}{G \times \tau} = 1 - \left(\frac{\left(\frac{P}{\tau}\right)}{G} \right) \tag{5.1}$$

where P is the number of packets received over a time-interval τ, G is the data packet generation rate conveyed from the source node, and $\frac{P}{\tau}$ is the data packet arrival rate. The value of τ is set to 5 seconds. This is the value of 2-DAARC's feedback transmission interval. This value is set to be consistent with the reporting interval recommended for the Real-Time Control Protocol [171].

Three equations are derived from Eq. 5.1, and these are used to calculate the four QoS statistics required by 2-DAARC's adaptation procedures. Table 5.2 shows the mappings between the four QoS statistics and the mode of adaptation which uses them. The four statistics are the *reserved packet delivery ratio* (RePDR), the *SPA packet loss ratio* (SPA PLR), the *original packet loss ratio* (OPLR), and the *duplicated packet loss ratio* (DuPLR). The SPA mode uses the first two statistics. The RePDR is calculated using Eq. 5.2,

$$RePDR = \frac{\left(\frac{P_{reserved}}{\tau}\right)}{G} \tag{5.2}$$

where $P_{reserved}$ is the number of packets delivered to the destination node using the reserved forwarding service during a past time-interval τ. The SPA PLR is calculated using Eq. 5.3.

$$\text{SPA PLR} = 1 - \left(\frac{\left(\frac{P_{reserved}+P_{degraded}}{\tau}\right)}{G} \right) \tag{5.3}$$

Mode of Adaptation	QoS Statistic
SPA	RePDR SPA PLR
MPA	OPLR DuPLR

Table 5.2: QoS Statistics used to Capture Contextual Information for the SPA Mode and the MPA Mode.

where $P_{degraded}$ is the number of packets delivered to the destination node using the degraded forwarding service during a past time-interval τ.

When the MPA mode is used it is necessary to monitor the packet delivery ratios of both the original and the duplicated data packets. The *packet status* indicator in the IP Options field of the data packet header is checked to determine whether a received packet is an original or a duplicate. An original packet has its *packet status* option set to OR. A duplicate packet has its *packet status* option set to DUP. The OPLR is calculated using Eq. 5.4

$$OPLR = 1 - \left(\frac{\left(\frac{P_{OR}}{\tau}\right)}{G} \right) \tag{5.4}$$

where P_{OR} is the number of original packets received during a time-interval τ. The DuPLR is calculated using the same equation, but replacing P_{OR} by P_{DUP}, where P_{DUP} is the number of duplicated packets received during time-interval τ.

Once these QoS statistics have been calculated they are transmitted to the source node in a cryptographically protected feedback packet. The feedback mechanism is described in Section 5.5.5.1. The process of using the contextual information for dynamic adaptation is described in Section 5.5.5.2.

The four QoS statistics are used by a source node as follows. For the SPA mode, the RePDR and the SPA PLR are calculated and used. The RePDR indicates the service quality of the in-use path. It is used to determine whether the path should be changed to support more packet deliveries using the reserved forwarding service. The SPA PLR is used to determine whether the packet loss is sufficient for the MPA mode to be enabled. For the MPA mode, the OPLR and the DuPLR are calculated and used. These are used to determine whether the path(s) should be changed to better support QoS or whether the packet loss is low enough to switch back to the SPA mode.

5.5.4.3 Duplicated Data Packet Detection

A destination node needs to detect and suppress the duplicated data packets of a priority data packet flow when the MPA mode is active. Priority data packets are duplicated across two paths, but only one set of these packets is required for use at the destination node's application layer. If both copies of a packet arrive at the destination node the latter received copy should be suppressed. Suppressing duplicates at the network-layer, rather than at a higher layer, can minimise any effects (including performance and storage) they may have on the higher layers. For example, receiving duplicated data packets when using the Real-time Transmission Protocol can lead to negative values for the *fraction lost* and *cumulative number of packets lost* fields [171]. Additionally, there is a performance benefit of suppressing duplicated data packets at lower layers, as the lower the layer the duplciates are suppressed the lower the costs imposed on the nodes. For example, a node will not expend resources on calculations which are of no benefit, as is the case in the previous Real-time Control Protocol example.

The task of detecting and suppressing duplicated data packets is performed by the Duplicate Suppressor component in the receiver entity of the architecture diagram shown in Fig. 5.1. This component uses a Duplicated Data Packet Detection (D2PD) algorithm to detect the received duplicate packets. The D2PD algorithm is based on the anti-replay window used in the IPsec (Internet Protocol security) protocol [95]. D2DP uses a sliding window mechanism to detect duplicate data packet sequence numbers contained within the window. It compares the sequence number of the most recently received packet with the sequence numbers of the n most recently received packets contained in the window. Here n is the size of the window. The received packet is discarded if its sequence number matches any of the sequence numbers contained within the window. The upper window edge, Uw, points to the highest sequence number seen so far. The lower window edge, Lw, points to the lowest sequence number held within the window. The window size n (where $n = Uw - Lw + 1$) refers to how far out of order the sequence number of a packet can be relative to the packet with the highest sequence number received so far [95]. (Table 5.3 lists the terms used in this section.) When a packet arrives with a sequence number S there are three cases [50] that should be considered in the design of D2PD.

Case (1): S is smaller than Lw. The destination node is unable to determine whether or not this packet's sequence number S has already been seen. The

Variable Name	Meaning
S	Sequence number of received packet
n	Window size (i.e., $n = Uw - Lw + 1$)
Lw	Lower-edge of window (lower bound of window)
Uw	Upper-edge of window (highest sequence number seen so far, upper bound of window)

Table 5.3: Definitions of Variables Used in the D2PD Algorithm

destination node discards this packet.

Case (2): S is within the window range, i.e., $Lw \leq S \leq Uw$. The destination node is able to detect whether the same packet has already been received. The packet is discarded if its sequence number S matches a sequence number in the window. Otherwise the packet is kept. The Uw is set to S, i.e., the window slides to include S. Consequently, the left-most edge of the window will also slide to the right, forcing the oldest sequence number to drop out of the window.

Case (3): S is greater than Uw. The destination node determines that this packet is an original packet. It will slide the window to include S, i.e., Uw is set to S, and Lw is shifted to the right, accordingly maintaining the condition $Uw - Lw + 1 = n$.

Case (2) can be extended to deal with the issue of out-of-order packet delivery. The following example illustrates this. In this example, a window size of $n = 5$ is used and the sequence numbers in the window are $\{1, 2, 4, 5, 6\}$. A packet with sequence number $S = 3$ may arrive at the destination node after packets with higher sequence numbers. S is within the range of the window, as $Lw \leq S \leq Uw$, although S does not already appear in the window. S is therefore not considered a duplicate. Given the procedure for Case (2), Uw should be set to S, but that will make the value of Uw less than the value of $Uw - 1$, i.e., the value of the new upper-edge is less than the value of the previous upper-edge. Case (2) therefore needs to be extended to handle out-of-order delivery. One way to do this is to set Uw to S and then, if the value of Uw is less than the value of $Uw - 1$, sort the sequence numbers so that they are ordered.

The D2PD algorithm includes the three cases above, including the extended case (2). Pseudocode for the D2PD algorithm is given in Algorithm 5.3. When a data packet arrives, the algorithm first checks for case (3) to determine if the

Algorithm 5.3: Pseudocode for the Duplicated Data Packet Detection (D2PD) Algorithm

```
Input: Packet sequence number sequenceNo
Output: sequence number is duplicate (true) or not a duplicate (false)
if sequenceNo > highest sequence number in the window then
    // Case (3)
    remove the lowest sequence number from the window;
    add sequenceNo to the right-edge of the window;
    return false ;                          // not a duplicate, use packet
else if sequenceNo < lowest sequence number in the window then
    // Case (1)
    return true ;    // not known if already seen, suppress packet
else
    // Case (2)
    for sequence number s ∈ the window do
        if s == sequenceNo then
            return true ;                   // a duplicate, suppress packet
    remove the lowest sequence number from the window;
    add sequenceNo to the right-edge of the window;
    // Extended Case (2) for out-of-order sequence numbers
    if sequenceNo < right-edge - 1 then
        sort the sequence numbers in the window;
    return false ;                          // not a duplicate, use packet
```

packet is an original, i.e., whether it has yet been received. If so, it can be used at the application layer. If not, the next step is to check for case (1) to determine whether it has been too long since this packet was generated to identify it as a duplicate. If not, the final step is to check for a packet whose sequence number is recent enough to be within the window. If the sequence number matches a sequence number in the window, the packet is discarded; otherwise it is kept for use at the application layer and Uw is set to the packet's sequence number. The extended case (2) check is then performed: if the sequence number is lower than that of $Uw - 1$, i.e, the previous value of Uw, the sequence numbers are sorted.

Specifying an appropriate window size (n) is important to balance the number of packets that may be discarded incorrectly (as they arrive too late, with sequence numbers less than Lw being discarded) against the overheads of storing and processing them. The IPsec specification [95] recommends a minimum

window size of 32 packets, although a window size of 64 packets is preferable. A default window size of 64 packets is therefore used in the D2PD algorithm. Choosing a smaller window size means that a narrower range of duplicated sequence numbers can be detected. However, this is offset by a smaller memory footprint, as fewer sequence numbers are kept in memory. Contrariwise, a larger window maintains a greater range of sequence numbers, but this comes at a higher cost: more sequence numbers have to be stored in memory; and this may incur a longer search time as the sequence number of the incoming packet is compared with a greater number of sequence numbers maintained in the window. A 16-bit integer is used for sequence numbers in MANET packet headers [28]. 128 bytes are therefore required to store the sequence numbers of 64 packets.[1]

It is worth noting that during the initial design of D2PD, the use of the *packet status* bit of the IP Options header was considered for duplicate data packet detection, but it became clear that it was not sufficient for this purpose. This is because the *packet status* bit only indicates whether a source node originated the packet as an original (*packet status* = OR) or a duplicate (*packet status* = DUP); it does not indicate the order in which the packets arrive at the destination node. Knowing the order of arrival is necessary to differentiate between which copy of the packet should be used at the application layer and which should be suppressed. For example, the end-to-end delay of the multiple in-use paths may vary and lead to the duplicate data packet arriving before the original packet. In this case, the duplicate packet should be used at the application layer and the original packet should be suppressed. To suppress the packet with *packet status* = DUP and wait for the packet with *packet status* = OR may lead to increased delays. This action may also lead to packet loss as the original packet may have been dropped by an attacker, thus this packet will never arrive at the application layer. A mechanism is therefore required to detect duplicated data packets based on the order in which the original and the duplicated packets are received, and this is provided by the D2PD algorithm.

[1]This leads to the sequence numbers wrapping after the sequence number value of 65,535, i.e., the number after 65,335 is 0 [28]. The D2PD algorithm does not deal with sequence number wrap around.

5.5.5 Achieving Dynamic Single-Path and Multi-Path Adaptations

The QoS statistics calculated by a destination node (as described in Section 5.5.4.2) are fed back to the source node for dynamic single-path and multi-path adaptations. These end-to-end feedback data are cryptographically protected to ensure their integrity and origin authentication. The process of cryptographically protecting these data is described in Section 5.5.5.1. The source node uses the received feedback data as input to its adaptation procedures. These adaptation procedures are described in Section 5.5.5.2.

5.5.5.1 Cryptographically Protected End-to-End Feedback

The QoS statistics are periodically calculated and transmitted to the source node at every time-interval τ. They are carried in a feedback (control) packet which is protected with a message authenticity service to ensure their integrity and origin authentication. (The message authenticity service is applied in the Feedback Generator component in the receiver entity of the architecture diagram shown in Fig. 5.1. The feedback Generator interacts with the Crypto and Key Store components.) The authenticity service uses a keyed-hash function, SHA-224 [141], with a 112-bit symmetric key. This key-length is the minimum length currently recommended [9]. A longer key may be used, but the storage and processing costs would be higher. The key is retrieved from the Key Store at the start of a session and is kept in the memory for the duration of the session. For each outgoing feedback packet, a message digest h is generated by keyed-hashing the QoS statistics carried in the packet. This generation process is shown in Eq. 5.5.

$$h = H(k \parallel \text{RePDR} \parallel \text{SPA PLR} \parallel \text{OPLR} \parallel \text{DuPLR}) \tag{5.5}$$

These feedback data and the message digest, shown in Fig. 5.6, are inserted into a control packet, i.e., a feedback packet, and transmitted to the source node along a single, best-effort path. The DSR protocol's default single-path selection mechanism is used to select a path with the shortest hop-count to the source node.

On receipt of the feedback packet, the source node verifies the data by computing a fresh keyed-hash of the data, which it then compares with the received

Feedback	RePDR	SPA PLR	OPLR	DuPLR	Digest
Length (bits)	32	32	32	32	224

Figure 5.6: Contents of a Feedback Packet

hash. If the two hash values are identical, then (1) the packet has not been tampered with during transit, and (2) the packet is indeed generated by the claimed destination node with which it shares the symmetric key. In this case, the feedback data will be used for adaptation. Otherwise the packet will be discarded. If the feedback packet is discarded, a new path will be selected.

5.5.5.2 Using Feedback for Dynamic Adaptation

As mentioned earlier, the source node verifies feedback data it receives, and if the verification is positive it uses these data to perform adaptation within a mode or between modes. (Adaptation decisions take place in the Path Quantity Calculator component in the sender entity of the architecture diagram shown in Fig. 5.1.) The dynamic adaptation process aims to maximise priority data packet deliveries to the destination nodes, and it also to maximise the percentages of priority data packets receiving the bandwidth-reserved forwarding service when the SPA mode is in use. The dynamic adaptation process makes use of the four QoS statistic values (RePDR, SPA PLR, OPLR, and DuPLR) in conjunction with the three adaptation threshold values specified in the application layer profile (the RePDR threshold Γ, the SPA PLR threshold ϕ, and the PLR threshold ρ), and two variables, *previous_OPLR* and *previous_DuPLR*. Γ and ϕ are used by the SPA mode. ρ is used by both the SPA and the MPA modes. The *previous_OPLR* and the *previous_DuPLR* variables are used by the MPA mode. These two variables store the values of OPLR and DuPLR experienced on the previous occasion the dynamic adaptation process was performed, and they are used to determine whether network conditions have improved since the previous time interval.

The pseudocode of the dynamic adaptation algorithm is given in Algorithm 5.4. The algorithm first checks if the communications session is currently in the SPA mode. If so, the source node compares the SPA PLR value (marked SPA_PLR in the algorithm) with the packet loss threshold ρ. If the SPA PLR value is greater than or equal to ρ the source node switches the session to the MPA mode. If the SPA PLR value is less than ρ, the source node will further examine the RePDR and SPA PLR values, i.e., it compares (1) the RePDR experienced over the last

Algorithm 5.4: Pseudocode for Dynamic Adaptation using the Path Quantity Calculator Method

```
if mode == SPA then
    if SPA_PLR ≥ ρ then
        mode = MPA;
        useTwoPaths(true);
        return;
    if RePDR ≤ Γ or SPA_PLR ≥ φ) then
        changePath(primary);
else
    // the mode is MPA
    if OPLR < previous_OPLR and OPLR < ρ and DuPLR <
    previous_DuPLR and DuPLR < ρ then
        mode = SPA;
        useTwoPaths(false);
        return;
    if OPLR ≥ ρ then
        changePath(primary);
    if DuPLR ≥ ρ then
        changePath(secondary);
    previous_OPLR = OPLR;
    previous_DuPLR = DuPLR;
```

time interval with the RePDR threshold Γ; and (2) the SPA PLR with the packet loss adaptation threshold ϕ. If either the RePDR is less than or equal to Γ or the SPA PLR is greater than or equal to ϕ the source node will choose the next-best path in the Route Cache as the primary path. This is a path with either the same hop-count or the next shortest hop-count.

If the session is in the MPA mode when the feedback data arrive, the source node will assess if the session should continue to operate in the MPA mode or switch to the SPA mode. The session should be switched if and only if the packet losses on both the primary and the secondary paths are (1) less than the packet loss threshold ρ and (2) less than previously stored values for the OPLR and DuPLR: if the current packet losses (OPLR and DuPLR) are both less than ρ and are less than the previous packet losses (previous_OPLR and previous_DuPLR), then it is likely that the network risk-level has decreased and the use of the SPA mode would be sufficient to support the QoS requirement of

the priority data traffic. If these conditions are not satisfied, the source node will assess if adaptation within the MPA mode is necessary. If the OPLR is greater than or equal to the packet loss threshold ρ the primary path should be changed. As is the case in the SPA mode, the source node will choose the next shortest path in the Route Cache as the new primary path. If the DuPLR is greater than or equal to ρ the secondary path should be changed. The source node will select the path with the greatest disjointedness with the primary path and the shortest hop-count. The current values of the OPLR and DuPLR variables are stored in the *previous_OPLR* and the *previous_DuPLR* variables for use next time the dynamic adaptation algorithm is called.

One design criterion resulting from the simulation study in Section 4.2 is that it is necessary to perform adaptive actions in the absence of a received feedback packet. The Path Quantity Calculator handles this as follows. When the SPA mode is enabled, if the source node does not receive a feedback packet before the expiration of a timeout it will infer that there is a problem with the current path. It will then select the next shortest path from the Route Cache. When the MPA mode is enabled, if no feedback is received it is clear that both paths are not functioning correctly. Two new paths are selected: the primary path is replaced with the next shortest path from the Route Cache; the secondary path is selected using the PMTS algorithm.

5.6 2-DAARC Performance Evaluation

The performance of 2-DAARC is evaluated using a simulation study. The study is carried out in three phases. In the first phase, presented in Section 5.6.1, the values for the three adaptation parameters, ρ, ϕ, and Γ, are determined. In the second phase, presented in Section 5.6.2, adaptation within the SPA mode is investigated to optimize data packet deliveries. In the third phase, presented in Section 5.6.3, the 2-DAARC approach to QoS provisioning is investigated.

5.6.1 Determining the Values of Adaptation Parameters

The three adaptation parameters, ρ, ϕ, and Γ, define either a packet loss percentage threshold (ρ and ϕ) or a service quality percentage threshold (Γ). The ρ parameter is used for two purposes: (1) it is used to determine when to switch from the SPA mode to the MPA mode (and vice versa); and (2) it is used in

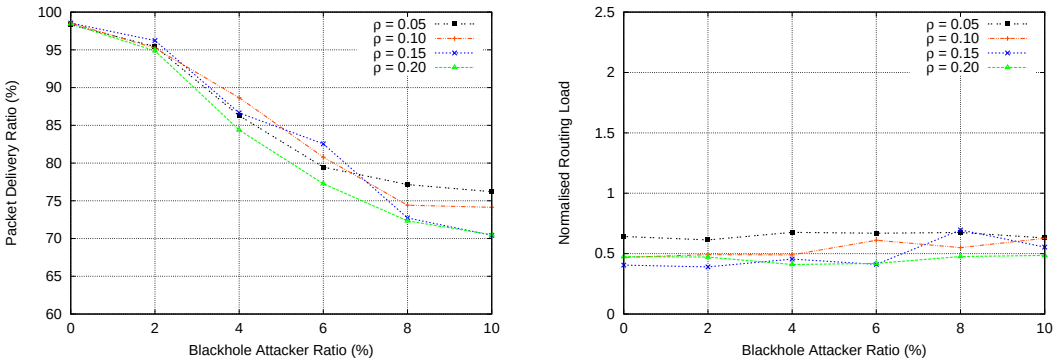
conjunction with ϕ and Γ to change the path within the SPA mode. The ϕ parameter is used within the SPA mode to determine when to change a path in response to received packet loss statistics. The Γ parameter is used within the SPA mode to determine when to change a path in response to received service quality statistics.

The following describes the simulation configurations used in Section 5.6.1.1–5.6.1.3. The simulations are conducted in a network containing 10 sources nodes, comprising 3 source nodes demanding priority treatment for data packets and 7 best-effort background sources. This number of traffic sources is used so that the effects of changing the ρ, ϕ, and Γ parameter values can be observed without the effects of other factors such as a high network load. (Adapting to packet loss caused by attacks is not the same as adapting to packet loss cased by congestion; this is discussed in Section 5.6.3.2.)

5.6.1.1 Packet Loss Parameter ρ

This first set of simulations is to determine an appropriate value for ρ to govern adaptation between the SPA and the MPA modes. Four values of the ρ parameter are investigated. They are 0.05, 0.10, 0.15, and 0.20. These values correspond with 5%, 10%, 15%, and 20% of the packet loss. The justification for using 0.20 as the largest value of ρ is that it is a high percentage of packets to have to lose before enabling the MPA mode; and having to experience high packet loss before enabling the MPA mode may worsen QoS. Source nodes learn the packet loss from the received feedback packets. The MPA mode uses the priority-based approach to secondary path selection. A blackhole attacker ratio of 0–10% in 2% increments is used. The non-zero attacker ratios are used to cause sufficient packet loss for 2-DAARC to switch to the MPA mode; the point at which the MPA mode is triggered will depend on the value of the ρ parameter. Two mobility scenarios are used: a network containing static nodes (900 second pause time) and a network where all nodes are mobile (0 second pause time).

On average, the highest PDRs are achieved with a value of $\rho = 0.05$. These PDRs are achieved with both static (Fig. 5.7(a)) and mobile (Fig. 5.8(a)) nodes. The higher PDRs are due to the MPA mode being enabled sooner than with other values of ρ. In other words, enabling the MPA after a lower percentage of packet loss leads to higher PDRs. The second highest PDRs are achieved with a value of $\rho = 0.10$. These PDRs are, on average, similar to those of $\rho = 0.05$ (they differer

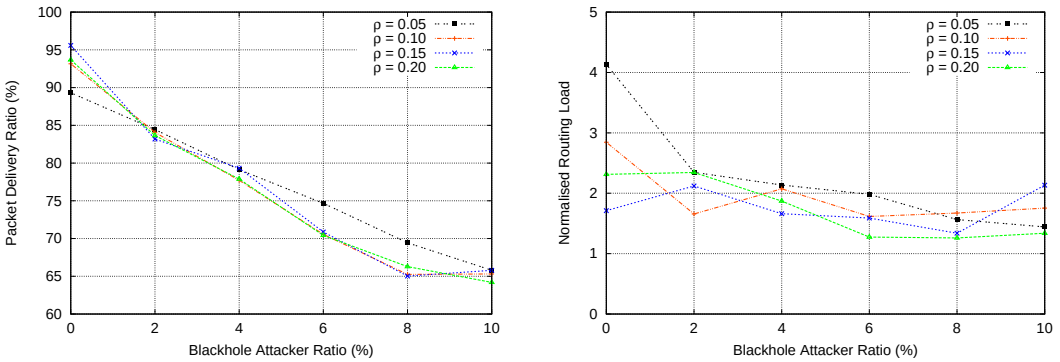


(a) PDR (b) Normalised Routing Load

Figure 5.7: Determining the value of ρ for Adaptation Between the SPA and the MPA Modes with a 900 Second Pause Time.

by less than 1 percentage point). The lowest PDRs are generally achieved when $\rho = 0.20$. This is because 20% of the packets transmitted in each time-interval τ must be lost before the MPA mode is enabled. In other words, the MPA mode is enabled later than with the other values of ρ, and this leads to lower PDRs.

Using a value of $\rho = 0.05$ generally achieves the highest normalised routing loads of the four values of ρ. This is because the MPA mode is enabled earlier and multiple paths are used more frequently. A larger number of control packet transmissions are therefore required to support the discovery and maintenance of the multiple paths compared with other values of ρ. The effects of a high normalised routing load should be considered when setting the value of ρ, as they may have a detrimental effect on QoS: control packets contribute to the network load, and, as observed in Section 4.2, the achievable QoS is sensitive to the network load. Measures should therefore be taken to minimise the number of control packets injected into the network. It is possible to do this without significantly affecting the PDR: the normalised routing load with $\rho = 0.10$ is lower than that of $\rho = 0.05$, and the PDRs it achieves are not significantly lower than those of $\rho = 0.05$. Setting the value of ρ to 0.10 therefore offers a trade-off between the PDR and the normalised routing load. Based on the above results and analysis, a value of $\rho = 0.10$ is used for adaptation between the SPA mode and the MPA mode.



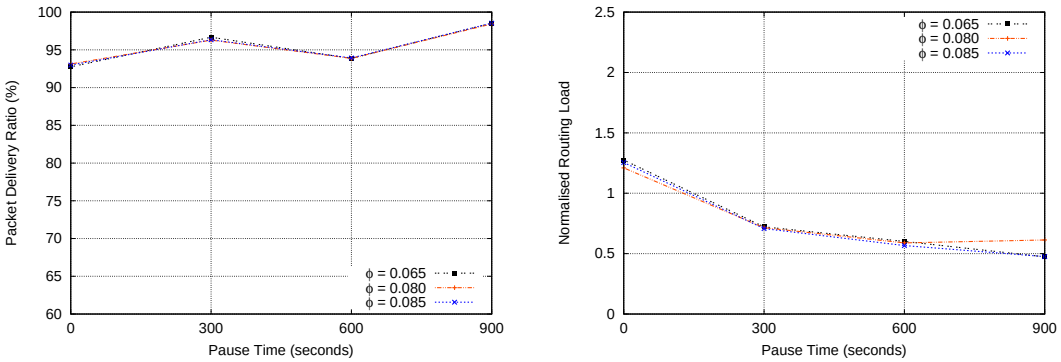
(a) PDR (b) Normalised Routing Load

Figure 5.8: Determining the value of ρ for Adaptation Between the SPA and the MPA Modes with a 0 Second Pause Time.

5.6.1.2 Packet Loss Parameter ϕ

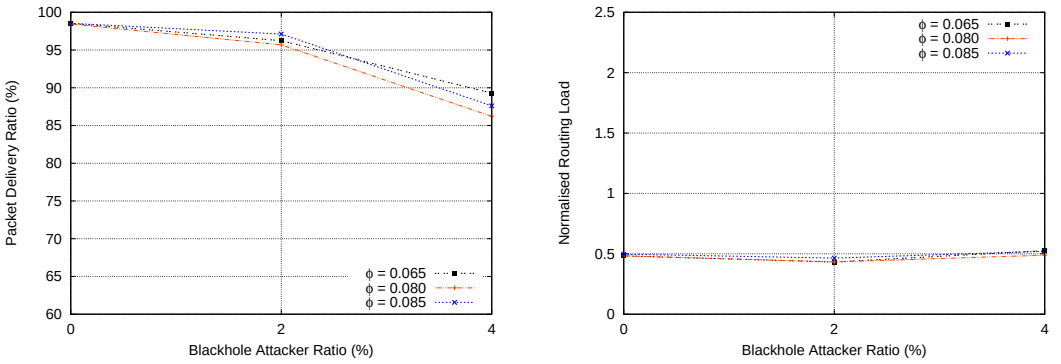
The following set of simulations are to determine a suitable value for the ϕ parameter for adaptation within the SPA mode. This parameter is used in conjunction with the ρ parameter to change the path in response to packet loss statistics (received in a feedback packet). The ϕ parameter is a lower bound on the packet loss ratio (PLR) which must be experienced before a path is changed. The ρ parameter is the upper bound on the PLR. A new path is therefore used when the PLR is within the range of $\phi \leq PLR < \rho$ (here $\rho = 0.10$, as determined in the previous section). In the simulations, values of ϕ are set to 0.050 to 0.095 in increments of 0.05. These values correspond with 5%–9.5% of the packet loss. The results presented in this section are for the three best performing values of ϕ, where $\phi = 0.065$, $\phi = 0.080$, and $\phi = 0.085$. These values correspond with 6.5%, 8%, and 8.5% of the packet loss. In these simulations, only the SPA mode is enabled, i.e., the MPA mode is not enabled even if the PLR exceeds ρ. The purpose of this is to observe the effects of adaptation only within the SPA mode.

In a network free of attackers, the PDRs and normalised routing loads achieved by the three values of ϕ are similar for 0, 300, 600, and 900 second pause times. Fig. 5.9(a) and Fig. 5.9(b) show the PDRs and the normalised routing loads, respectively. The similarity of the results for the different values of ρ is due to there being little packet loss in the network for the SPA mode to adapt to, i.e., path changes are not needed to support the QoS requirements of a priority packet flow in these network conditions. One reason for this is that the network does not contain any attackers.



(a) PDR (b) Normalised Routing Load

Figure 5.9: Determining the value of ϕ for Adaptation within the SPA Mode in a Network Free of Attackers.



(a) PDR (b) Normalised Routing Load

Figure 5.10: Determining the value of ϕ for Adaptation within the SPA Mode in the Presence of Attackers.

When a small number of blackhole attackers are introduced into the network, larger differences in the PDRs and normalised routing loads can be observed. Fig. 5.10 plots the PDRs and normalised routing loads when the attacker ratio is 0–4% and the pause time is 900 seconds. As shown in Fig. 5.10(a), the PDR is the greatest at the 2% attacker ratio when $\phi = 0.085$. At the 4% attacker ratio, the PDR is greatest when $\phi = 0.065$. This is because path changes are made after a lower level of packet loss is experienced. However, the PDRs at the 4% attacker ratio are less than 90%, i.e., the PLR is greater than 10%: if the MPA mode was not disabled, it would be used when the PLR exceeds 10% (the value of ρ). The normalised routing loads, shown in Fig. 5.10(b), are generally similar for all three values of ϕ at all attacker ratios. The simulation results do not clearly show a value of ϕ which supports the best QoS under the conditions

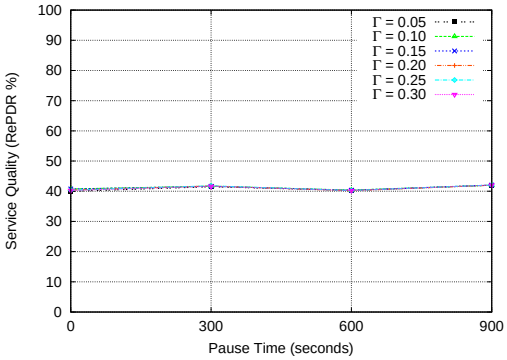
investigated, although the highest PDRs are achieved with $\phi = 0.085$ without a significant increase in the normalised routing load. Based on the above results and analysis a value of $\phi = 0.085$ is used for adaptation to packet loss statistics within the SPA mode.

5.6.1.3 Service Quality Parameter Γ

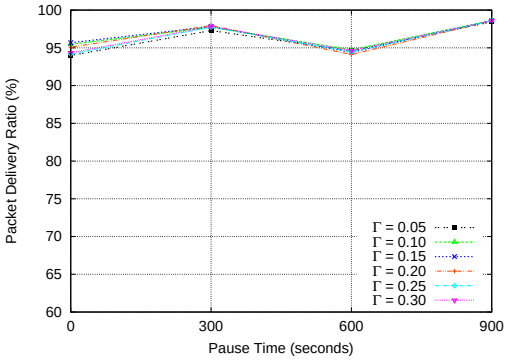
This set of simulations investigates the value of the Γ parameter for adaptation within the SPA mode to service quality statistics. This form of adaptation aims to maximize the percentage of priority data packets receiving the bandwidth-reserved forwarding service. This is achieved by comparing the RePDR (received in a feedback packet) with the threshold value Γ. If the RePDR is less than Γ, a new path is selected. The MPA mode is not used and no adaptation is performed with regard to the ρ and ϕ adaptation parameters. The purpose of this is to focus only on the effects of the SPA mode adapting to the Γ parameter.

The simulation configuration is described as follows. In the simulations, the Γ parameter is set to values ranging from 0.05 to 0.30 in increments of 0.05. These values correspond with 5%–30% of the priority data packets receiving the reserved forwarding service. These values represent the point at which the in-use path should be changed if the RePDR is less than the value of the Γ parameter. The maximum value simulated is $\Gamma = 0.30$, as it was observed in Section 4.2.4 that the RePDR can generally be maintained at this level (30%) when no attackers are present in the network. The simulations in this section do not contain attackers.

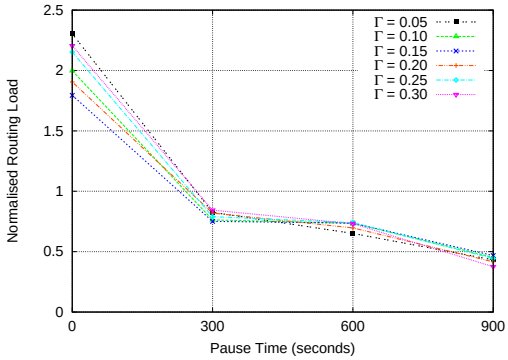
There is little difference in the service quality achieved with the different values of Γ. As can be seen in Fig. 5.11(a), the service quality, in terms of the RePDR, is similar for each value of Γ at all pause times. In the lightly loaded network conditions, intermediate nodes offer a similar level of bandwidth-reserved service regardless of when a source node changes the path. Some differences are observed when the RePDR is combined with the DePDR to form the PDR. As can be seen in Fig. 5.11(b), the differences in PDR increase as the pause time moves towards 0 seconds. At the 0 second pause time, the highest PDR is achieved with $\Gamma = 0.15$. This higher PDR is a consequence of a marginally larger percentage of packets being delivered with the degraded (best-effort) forwarding service than with other values of Γ. One reason for this is that the value of $\Gamma = 0.15$ requires fewer control packet transmissions to support data packet deliveries than the other values of Γ. Fewer control packet transmissions means that there are more resources available



(a) Service Quality



(b) PDR



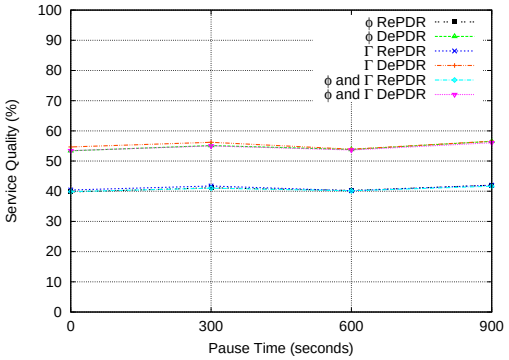
(c) Normalised Routing Load

Figure 5.11: Determining the value of Γ for Adaptation within the SPA Mode.

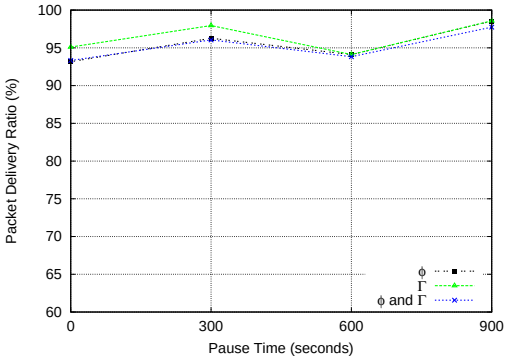
for data packet deliveries. The lower number of control packets, i.e., the lower normalised routing load, is shown in Fig. 5.11(c). Thus a value of $\Gamma = 0.15$ leads to path changes being performed to maximise packet deliveries whilst using the least number of control packets to discover and maintain the paths. The normalised routing loads generally increase as the pause time moves towards 0 seconds. This is due to paths being changed more frequently to maximise the RePDR in response to network dynamics caused by the increasing mobility. Based on these results and the analysis undertaken, a value of $\Gamma = 0.15$ is used for adaptation to the RePDR statistic within the SPA mode.

5.6.2 Revising SPA Mode Adaptation Based on Simulation Results

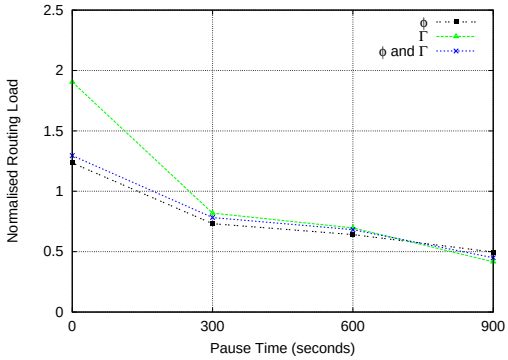
Having determined the values for the three adaptation parameters ($\rho = 0.10$, $\phi = 0.085$, and $\Gamma = 0.15$), the next step is to investigate whether a better QoS can



(a) Service Quality



(b) PDR



(c) Control Overhead

Figure 5.12: Comparing Three Path Adaptation Approaches for the SPA Mode: ϕ-based Path Adaptation, Γ-based Path Adaptation, ϕ & Γ-based Path Adaptation with a 0% Blackhole Attacker Ratio.

be achieved within the SPA mode by changing a path in response to packet loss statistics (hereafter referred to as ϕ-based path adaptation), service quality statistics (referred to as Γ-based path adaptation), or both packet loss and service quality statistics (referred to as ϕ & Γ-based path adaptation). This investigation is necessary as the ϕ and Γ parameters are not independent of one another (both are derived from the number of data packets received at the destination node). The results of these investigations are used to improve the SPA mode's adaptation process.

Fig. 5.12 shows the service quality, PDRs, and normalised routing loads of the SPA mode's three adaptation mechanisms. The ϕ-based path adaptation is marked in the figure as ϕ, the Γ-based path adaptation is marked as Γ, and the ϕ & Γ-based path adaptation is marked as ϕ and Γ. A 0% attacker ratio is used with 0, 300, 600, and 900 second pause times.

Using Γ-based path adaptation supports better priority data packet deliveries

than either ϕ-based or ϕ & Γ-based path adaptation. As shown in Fig. 5.12(a), the three adaptation approaches support a similar service quality in terms of the RePDR, but the Γ-based path adaptation supports a marginally higher DePDR as the pause time approaches 0 seconds. The positive effects of this slightly higher DePDR can be seen when it is combined with the RePDR to form the PDR. As shown in Fig. 5.12(b), the PDRs for Γ-based path adaptation are generally greater than the PDRs for the two other adaptation approaches. The PDRs for both the ϕ-based and the ϕ & Γ-based path adaptation are lower because there is little packet loss in the network to adapt to (due to the 0% attacker ratio). In other words, path changes occur more frequently when only the RePDR is considered for adaptation within the SPA mode, and this leads to higher DePDRs and PDRs. This is because path changes are performed in response to the percentages of packets successfully delivered to the destination node using the reserved forwarding service. The more frequent path changes with Γ-based path adaptation lead to more frequent Route Discovery operations and, in turn, a more up-to-date Route Cache: having a fresh Route Cache means that fewer packets are lost as a consequence of not having a path to the destination node, hence the higher PDR. The less frequent path changes with the ϕ-based and ϕ & Γ-based path adaptation leads to a larger number of stale paths in the Route Cache, and this means that more packets are lost whilst a source node tries to discover a path to the destination node. However, as can be seen in Fig. 5.12(c), the more frequent path changes of the Γ-based path adaptation means that it has the highest normalised routing loads of the three adaptation approaches at the highest node mobilities. The higher normalised routing loads are a consequence of control packets being injected into the network during Route Discovery operations.

Based on the above observations and analysis, only Γ-based path adaptation is to be used for adaptation within the SPA mode. The dynamic adaptation algorithm presented in Algorithm 5.4 can therefore be revised and simplified. The aim of this is to improve the algorithm to optimize adaptation within the SPA mode. One change is made to the algorithm: the statement "*or* SPA_PLR $\geq \phi$" is removed from the 'if' conditional on line 6. The SPA mode will now perform adaptation in response to received service quality statistics using only the Γ parameter.

5.6.3 Simulation Results

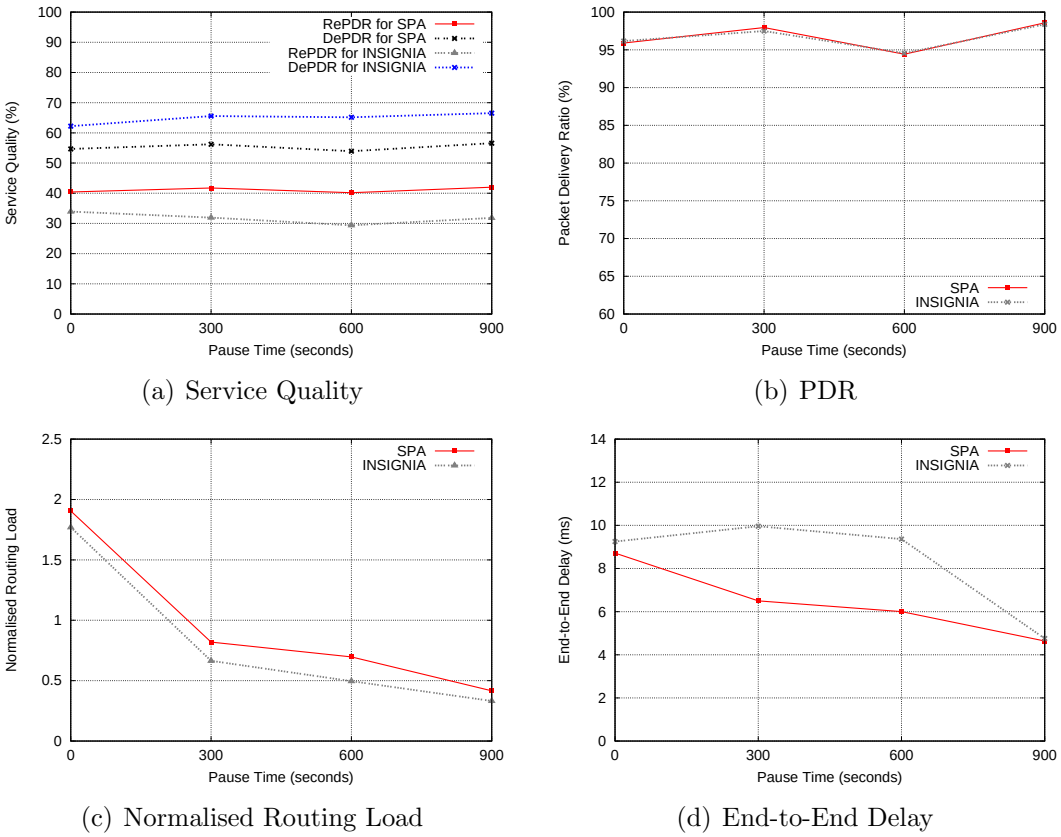
The results of this simulation study are presented in three sections. In Section 5.6.3.1, 2-DAARC's SPA mode is evaluated against INSIGNIA. In Section 5.6.3.2, dynamic adaptation between the SPA mode and the MPA mode is evaluated against INSIGNIA. In Section 5.6.3.3 the QoS achievable with the PMTS algorithm is evaluated against a node-disjoint-path-only secondary path selection mechanism.

The following simulation parameters are used in the three simulation investigations. Two levels of traffic sources are used: 10 sources nodes (3 priority sources) and 20 source nodes (6 priority sources). 10 traffic sources are used to generate a lightly loaded network. 20 traffic sources are used to generate a highly loaded network. The attacker ratio ranges from 0% to 10% in 2% increments. The simulations are performed with the DSR protocol's packet salvaging optimization enabled, unless otherwise stated. All other simulation parameters values are as specified in Section 3.5.

5.6.3.1 QoS Using Only the SPA Mode

The simulation investigations presented in this section demonstrate the effectiveness of the SPA mode's design and the extent to which its adaptation facilities support better QoS than INSIGNIA. The SPA mode extends INSIGNIA with a more comprehensive adaptation process which aims to better support QoS over a single path. The main focus of the adaptation process is on maximising the percentage of priority traffic receiving the bandwidth reserved forwarding service. To do this, the SPA mode changes paths in response to received feedback, in addition to performing bandwidth adaptation along the in-use path. The latter adaptation process is inherited from INSIGNIA.

In a lightly loaded network free of attackers, the SPA mode achieves significantly higher service quality than INSIGNIA under a range of node mobilities. As can be seen in Fig. 5.13(a), the SPA mode achieves both higher RePDRs and lower DePDRs than INSIGNIA at all pause times. This is due to the SPA mode changing paths to maximise the percentage of packets delivered using the reserved forwarding service. The SPA mode generally supports RePDRs of around 40%, whereas INSIGNIA generally supports RePDRs of around 30%. Node mobility reduces the benefits brought by the SPA mode's adaptation process. This is because mobility-induced path breaks lead to a decrease in the effective bandwidth



(a) Service Quality

(b) PDR

(c) Normalised Routing Load

(d) End-to-End Delay

Figure 5.13: Comparing the Service Quality, PDR, Normalised Routing Load, and End-to-End Delay of the SPA Mode with INSIGNIA for 3 Priority Sources (with 7 Best-Effort, Background Sources) and 0, 300, 600, and 900 Second Pause Times with Packet Salvaging Enabled.

available for resource reservations. When considering the RePDR and DePDR together in terms of the PDR, as shown in Fig. 5.13(b), the SPA mode and INSIGNIA achieve similar PDRs. The SPA mode therefore does not deliver more packets than INSIGNIA, but of the packets that the SPA mode does deliver a greater percentage receive the bandwidth-reserved forwarding service.

The better service quality of the SPA mode comes at the cost of a higher normalised routing load than INSIGNIA. As shown in Fig. 5.13(c), the normalised routing load of the SPA mode is greater than that of INSIGNIA at all pause times. This means that the SPA mode requires more control packets than INSIGNIA to support data packet deliveries. Two factors are mainly responsible for the SPA mode's higher normalised routing load. The first is the periodic transmission of feedback packets (which are control packets). They are transmitted in addition to the QoS Reports, which are part of the adaptation process inherited from

INSIGNIA. In contrast, INSIGNIA only transmits the QoS Reports. The second factor is the larger number of Route Discovery operations performed by the SPA mode. These are performed to discover new paths to the destination node to maximise the service quality.

The end-to-end delays achieved by the SPA mode are generally lower than those of INSIGNIA. As shown in Fig. 5.13(d), this is most prevalent at the 300 and 600 second pause times. The lower delays are due to the SPA mode transmitting more packets with the reserved forwarding service (by changing paths if necessary), whereas INSIGNIA may continue to forward packets along the same path which is capable of supporting only the best-effort forwarding service. The priority packets receiving the best-effort service are queued in the best-effort data packet queue, and this is the same queue used for the best-effort data packets of the background traffic. The higher occupancy of the best-effort packet queue leads to longer queueing delays and longer end-to-end delays. Another trend is that the delay generally increases as the node mobility increases, i.e., as the pause time moves towards the 0 second pause time. At this point, the delay of the SPA mode differs little from that of INSIGNIA. This can be explained as follows. For both the SPA mode and INSIGNIA, the number of mobility-induced path breaks increases as the node mobility increases. This leads to (1) more attempts to retransmit data packets, and (2) Route Error control packet transmissions to notify the source nodes of the broken links. Both of these consume bandwidth, thus reducing the effective available bandwidth. The bandwidth reserved paths experience this as well as best-effort paths, hence the similarity in delays. The retransmitted data packets and the control packets increase the load placed on the packet queues, causing them to overflow frequently: the priority packet queues for the SPA mode overflow 17 times more often at the 0 second paused time compared with the 900 second pause time. At the 900 second pause time, there is little difference between the delays of the SPA mode and INSIGNIA. With these network conditions, the SPA mode offers little advantage in delay because fewer path changes occur compared with other pause times. This is because an established path offering the desired bandwidth reserved service is less likely to break or offer a worse service over time as there is no node mobility.

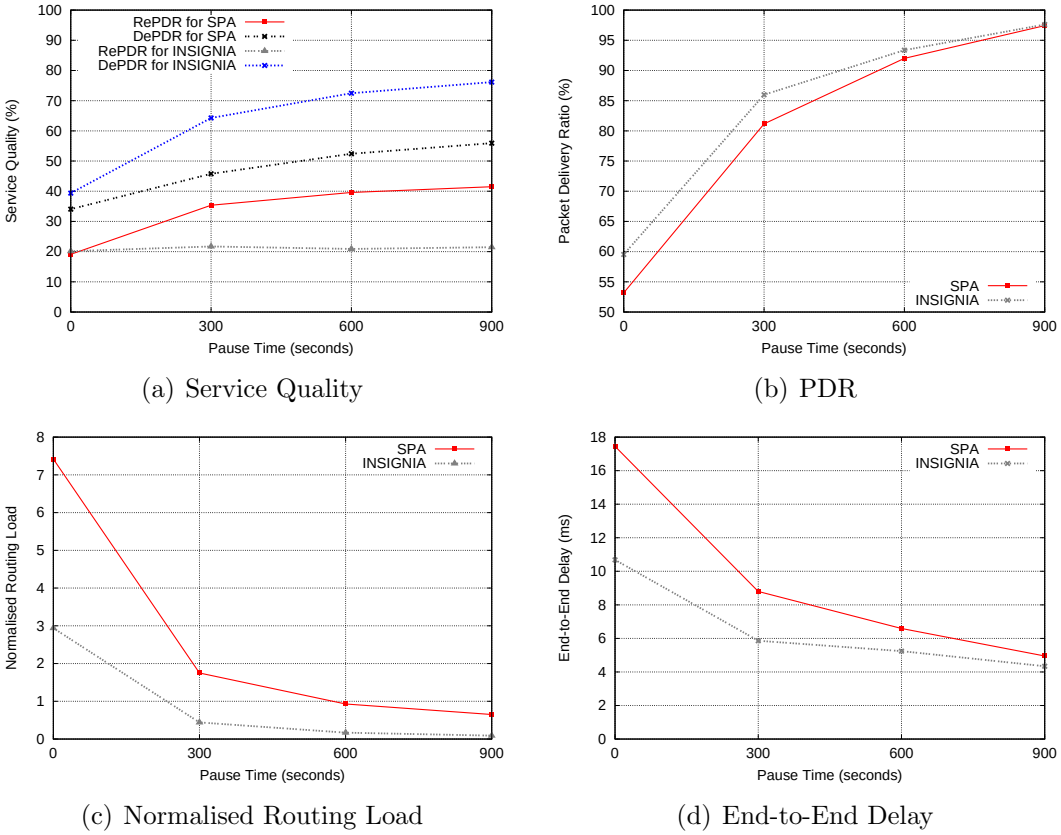
When the network load is increased to that of 20 source nodes and all other parameter values remain unchanged, the SPA mode generally achieves inferior QoS to INSIGNIA. As shown in Fig. 5.14(a), the SPA mode achieves a higher

RePDR than INSIGNIA at the 300, 600, and 900 second pause times, but they both achieve a similar RePDR at the 0 second pause time. Thus, under a high network load and with continual node mobility, the SPA mode is not able to support a better RePDR than INSIGNIA. Although the SPA mode generally supports a better RePDR than INSIGNIA, it typically has a lower PDR, as shown in Fig. 5.14(b). The inferior PDRs of the SPA mode are due to it suffering from the effects of congestion more than INSIGNIA. One reason for this is its higher normalised routing load. By comparing the normalised routing load (Fig. 5.14(c)) with the PDR (Fig. 5.14(b)), it can be seen that the PDR decreases as the normalised routing load increases. The SPA mode transmits significantly more control packets than INSIGNIA. The difference in the number of control packets transmitted also increases as node mobility increases. As can be seen in Fig. 5.14(c), the number of control packets transmitted by the SPA mode is more than double that of INSIGNIA at the 0 second pause time. This is more than triple the number of control packets transmitted by the SPA mode in the lightly loaded network (Fig. 5.13(c)). This is because the low RePDR leads to paths changes, and these require more Route Discovery operations which inject more control packets into the network.

The congested network conditions also lead to the SPA mode having longer end-to-end delays than INSIGNIA. By comparing Fig. 5.14(d) with Fig. 5.13(d), it can be seen that this is the opposite of the trend observed for the lightly loaded network. As can be seen in Fig. 5.14(d), the largest difference in delay is at the 0 second pause time. Here the delay of the SPA mode is approximately 6.8 seconds longer than INSIGNIA's delay. The reason for these observations is that the increasing normalised routing loads mean that the wireless medium is increasingly captured for control packets transmissions; this leads to increased end-to-end delays as data packets are queued for longer durations.

The packet salvaging optimization of DSR, 2-DAARC's underlying routing protocol, is another factor contributing to the inferior QoS of the SPA mode. Packet salvaging is the process by which a node attempts to retransmit a data packet along an alternative path to the destination node when the link in the current path fails. When the network is congested, the effective available bandwidth decreases. Attempting to retransmit data packets in already congested network conditions exacerbates the existing congestion.

Disabling the packet salvaging optimization leads to a marked improvement



(a) Service Quality

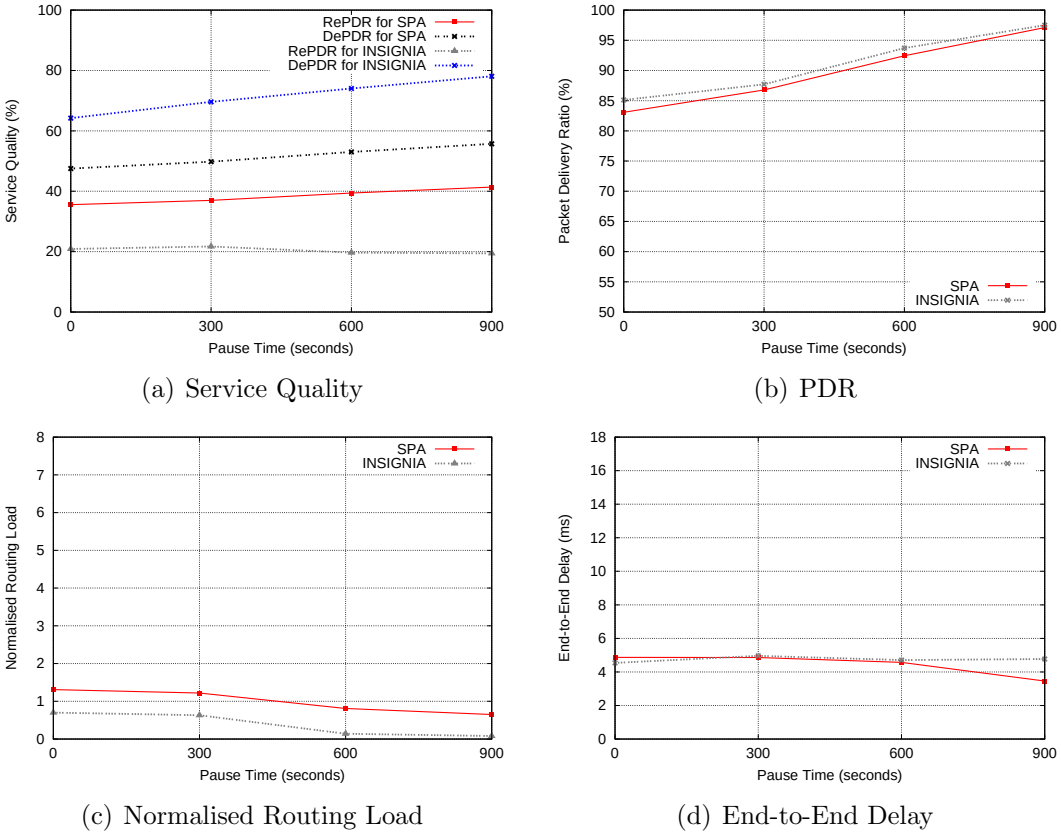
(b) PDR

(c) Normalised Routing Load

(d) End-to-End Delay

Figure 5.14: Comparing the Service Quality, PDR, Normalised Routing Load, and End-to-End Delay of the SPA Mode with INSIGNIA for 6 Priority Sources (with 14 Best-Effort, Background Sources) and 0, 300, 600, and 900 Second Pause Times with Packet Salvaging Enabled.

in the SPA mode's QoS, although it is still generally inferior to the QoS of INSIGNIA with packet salvaging disabled (DSR is its underlying routing protocol). Disabling packet salvaging reduces the congestion. This is because attempts are not made to retransmit the data packets which have failed to be delivered. As can be seen in Fig. 5.15(a), the greatest improvement in RePDR is at the 0 second paused time: the SPA mode now achieves an RePDR of 36%, compared with 21% for INSIGNIA. Another benefit of disabling the packet salvaging optimization is that the PDRs are higher for both the SPA mode and INSIGNIA at medium to high node mobilities. By comparing Fig. 5.15(b) (packet salvaging disabled) with Fig. 5.14(b) (packet salvaging enabled), it can be seen that the PDRs have increased at the 0 and 300 second pause times. This is because packet retransmissions are no longer taking place, and this increases the available bandwidth. Additionally, no data packet retransmissions means that those packets which have



(a) Service Quality

(b) PDR

(c) Normalised Routing Load

(d) End-to-End Delay

Figure 5.15: Comparing the Service Quality, PDR, Normalised Routing Load, and End-to-End Delay of the SPA Mode with INSIGNIA for 6 Priority Sources (with 14 Best-Effort, Background Sources) and 0, 300, 600, and 900 Second Pause Times with Packet Salvaging Disabled.

failed to be delivered are not queued again, and this reduces the burden on the packet queues. However, the PDRs of the SPA mode are still lower than those of INSIGNIA. This is due to the SPA mode changing paths in response to the RePDR, and is explained as follows. When the current path is no longer deemed suitable and no alternative path is available, a Route Discovery operation is performed. Whilst a new path is sought, the data packets are not forwarded along the unsuitable path and are dropped. For example, at the 0 second pause time, 78% of the SPA mode's packet loss is due to source nodes dropping their own packets because they do not have a path to the destination nodes. In contrast, when INSIGNIA cannot support the reserved forwarding service, it continues to use the same path but with the degraded forwarding service. Thus the SPA mode's path adaptation process leads to inferior PDRs in heavily loaded networks.

Disabling packet salvaging also leads to a reduction in the normalised routing

loads. As can be seen by comparing Fig. 5.15(c) with Fig. 5.14(c), the normalised routing load for the SPA mode at the 0 second pause time has reduced from 7.4 control packets with packet salvaging enabled to 1.31 control packets with packet salvaging disabled, which is an 82% decrease. The lower normalised routing load is partly due to a decrease in the number of ROUTE ERROR control packet transmissions. This is because reduced congestion leads to fewer congestion-induced path breaks, and this means that fewer ROUTE ERROR control packets are transmitted to source nodes to notify them of the broken links.

The decreased congestion resulting from disabling packet salvaging leads to the SPA mode and INSIGNIA performing similarly in terms of end-to-end delays. As can be seen by comparing Fig. 5.15(d) with Fig. 5.14(d), the delays of both the SPA mode and INSIGNIA have reduced significantly. At the 0 second pause time, the delay of the SPA mode is 3.6 times lower with packet salvaging disabled (Fig. 5.15(d)) than with packet salvaging enabled (Fig. 5.14(d)). At the 900 second pause time, the delay of the SPA mode is less than that of INSIGNIA. This is due to the SPA mode servicing twice as many priority data packets using the reserved forwarding service as INSIGNIA (Fig. 5.15(a)). The benefit of this is that the SPA mode's data packet queues are more lightly loaded than INSIGNIA's: 5% of the SPA mode's packet loss is due to buffer overflows compared with 9% for INSIGNIA. Disabling packet salvaging therefore has some positive effects on the QoS supported by the SPA mode in a heavily loaded network, but this QoS is generally lower than that of INSIGNIA.

In summary, when the network load is light the SPA mode delivers a better QoS to priority data packets than INSIGNIA. It delivers more data packets using the reserved forwarding service than INSIGNIA, and generally delivers them with a lower end-to-end delay. These benefits come at a cost. The SPA mode transmits more control packets than INSIGNIA in order to provide this better QoS. When the network load is heavy the SPA mode supports worse QoS than INSIGNIA. This is particularly the case when the packet salvaging mechanism of the underlying routing protocol (DSR) is enabled. Disabling this mechanism leads to the SPA mode offering a better QoS than when packet salvaging is enabled, but this is still generally less than the QoS supported by INSIGNIA in the given network conditions. The results presented in this section have been collected in networks free of blackhole attackers. Supporting QoS in the presence of blackhole attackers requires dynamic adaptation between the SPA mode and

the MPA mode in response to the packet loss caused by the attackers.

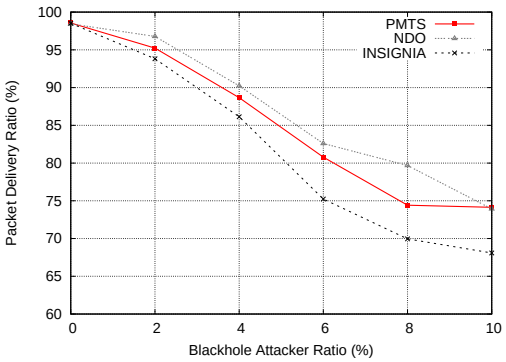
5.6.3.2 QoS Using the SPA and MPA Modes

This section investigates and compares the 2-DAARC approach against INSIGNIA in adversarial conditions. The results are first presented with a 900 second pause time (stationary nodes) under both light (10 source nodes) and heavy (20 source nodes) network loads. Following this, the results are presented for the 0 second pause time (constant node mobility) under both light and heavy loads. In all cases, an attacker ratio of 0%–10% in 2% increments is used. 2-DAARC is simulated in two ways: (1) using the PMTS approach to secondary path selection, marked PMTS in the figures, and (2) using a node-disjoint-path-only (NDO) approach to secondary path selection, marked NDO in the figures. Both of these versions of 2-DAARC are compared with INSIGNIA (Section 5.6.3.3 compares the PMTS and NDO approaches with one another).

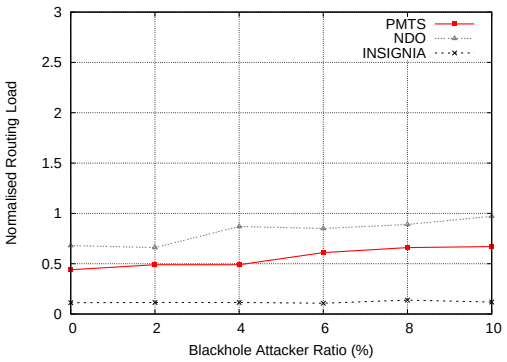
In a lightly loaded network containing stationary nodes, 2-DAARC generally supports higher PDRs than INSIGNIA in the presence of blackhole attackers. As can be seen in Fig. 5.16(a), 2-DAARC supports increasingly higher PDRs than INSIGNIA as the attacker ratio increases. This is because 2-DAARC uses the MPA mode to transmit duplicated data packets across multiple paths; using duplicated data packet transmissions helps to support data packet deliveries in the presence of attackers. The greatest difference in packet loss is at the 8% attack ratio, where 2-DAARC with NDO has a PDR of 80% and INSIGNIA has a PDR of 70%. In contrast, INSIGNIA does not adapt to the packet loss. It continues to send packets along a single path which may contain a blackhole attacker, hence its lower PDRs. When the attacker ratio is 0%, 2-DAARC and INSIGNIA achieve similar PDRs. This is because 2-DAARC does not use the MPA mode as there is little packet loss in the network to adapt to.

The PDRs achieved by 2-DAARC come at the cost of higher normalised routing loads at all attacker ratios. 2-DAARC's periodic feedback packet transmissions contribute to this higher load. As can be seen in Fig. 5.16(b), the normalised routing load generally increases as the attacker ratio increases. This is because more control packets are required to discover and maintain the multiple paths used for the data packet transmissions in the presence of attackers.

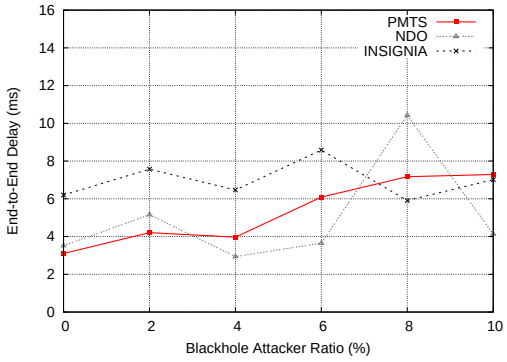
2-DAARC only supports a relatively small increase in PDR over INSIGNIA. This is due to the unintended re-use of bad paths during the adaptation process.



(a) Packet Delivery Ratio



(b) Normalised Routing Load



(c) End-to-End Delay

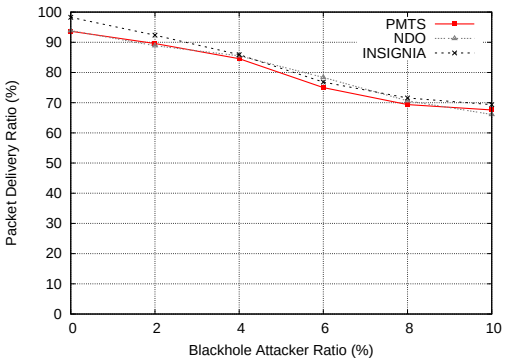
Figure 5.16: Comparing the PDR, Normalised Routing Load, and End-to-End Delay for a 900 Second Pause Time and 3 Priority Sources (with 7 Best-Effort, Background Sources) and Packet Salvaging Enabled

When the MPA mode is active, if the packet loss on a path is greater than or equal to ρ, the packet loss adaptation parameter, the next-best path in the Route Cache is selected. If the new path is for the role of primary path the shortest-path is selected; if a secondary path is required the shortest path with the greatest disjointedness with the primary path is selected. If the packet loss on the new path is greater than or equal to ρ the next-best path is again selected. However, in the case of the primary path the next-best path is likely to be the previously used path; and a change in the secondary path will likely lead to the selection of the previously used secondary path, as this offers the shortest hop-count and the highest disjointedness with the primary path. The path selection mechanism may therefore cycle between the two best paths in the Route Cache until a better path is discovered or one of the existing paths fails.

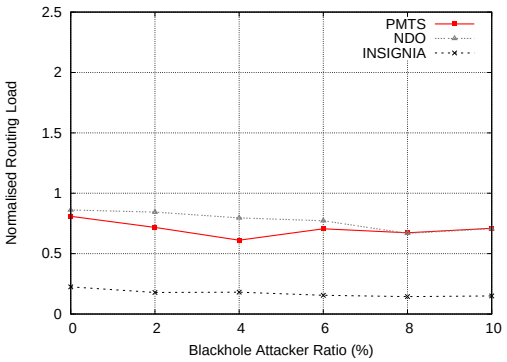
Another benefit of the 2-DAARC approach is that it generally achieves lower

end-to-end delays than INSIGNIA. As can be seen in Fig. 5.16(c), 2-DAARC generally supports lower delays at the lower attacker ratios of 0%–6%. This is because 2-DAARC's MPA mode offers an alternative path to deliver priority traffic when the packet loss is mostly being caused by attacks. However, 2-DAARC's delays generally increase as the attacker ratio increases, and they increase beyond those of INSIGNIA at the 8% and 10% attacker ratios. The increasing delays and the higher delays at the 8%–10% attacker ratios are a consequence of the increasing number of control packets required to deliver the data packets. By comparing Fig. 5.16(c) with Fig. 5.16(b), it can be seen that the delay increase as the normalised routing load increases. The increasing number of control packets required to support data packets deliveries make the network busier, increasing bandwidth consumption and requiring more time to be dedicated to control packet transmissions. Consequently, queueing delays and therefore end-to-end delays increase for the data packets.

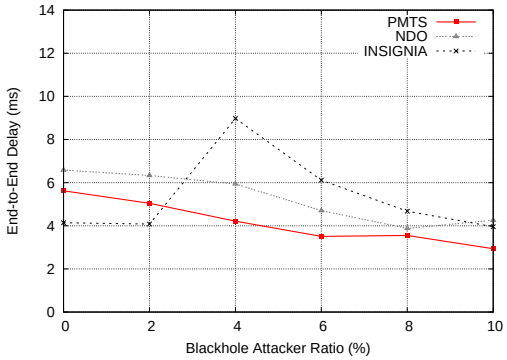
When the network load is increased to that of 20 source nodes and all other parameter values remain unchanged, 2-DAARC generally achieves lower PDRs than INSIGNIA. The increase in the number of traffic sources makes the network busier, and this reduces the PDRs which both 2-DAARC and INSIGNIA achieve. The larger number of data packets reduces the resources, e.g., bandwidth and packet queue space, available to service the data packets. As can be seen by comparing Fig. 5.17(a) with Fig. 5.16(a), 2-DAARC's lower PDRs than INSIGNIA is the opposite of the trend observed for the lightly loaded network. The lower PDRs are a consequence of 2-DAARC being affected by the higher network load more than INSIGNIA. This is a consequence of several factors. First, the MPA mode injects duplicated data packets into the already heavily loaded network. Rather than improving the QoS, the additional load generated by the duplicated data packets contributes to the network congestion, and this reduces the PDRs. Second, attempting to salvage undeliverable data packets when the MPA mode is in-use also contributes to the reduced PDRs. It is possible that both the original and the duplicated data packets may fail to be delivered and, if this happens, nodes will attempt to salvage both copies of the packet. The attempted retransmissions of these packets consumes more of the limited available bandwidth. Third, congestion-induced path breaks lead to an increased normalised routing load, and this negatively affects the PDRs. As can be seen



(a) Packet Delivery Ratio



(b) Normalised Routing Load



(c) End-to-End Delay

Figure 5.17: Comparing the PDR, Normalised Routing Load, and End-to-End Delay for a 900 Second Pause Time and 6 Priority Sources (with 14 Best-Effort, Background Sources) and Packet Salvaging Enabled

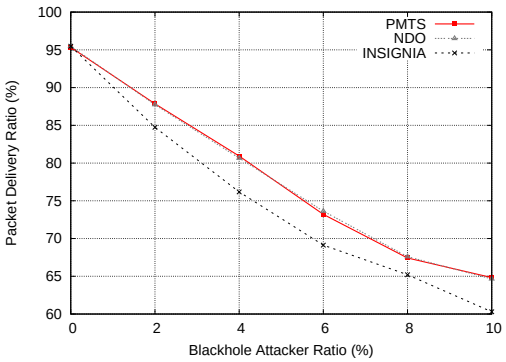
by comparing Fig. 5.17(b) with Fig. 5.16(b), the normalised routing loads of 2-DAARC have increased more than those of INSIGNIA. This is because 2-DAARC suffers more congestion-induced path breaks than INSIGNIA, and these lead to the transmission of a greater number of ROUTE ERROR packets.

The increased network load leads to 2-DAARC achieving longer end-to-end delays than in the lightly loaded network, although these are still generally less that those of INSIGNIA. The longer delays are a consequence of the network being more heavily loaded. Nodes have a larger number of data and control packets to transmit. Additionally, nodes attempt to salvage the data packets which have failed to be delivered. All of the factors contribute to the increased delays experienced by the priority data packets. From the above results, it can be seen that 2-DAARC is not able to offer significant benefits over INSIGNIA when the network traffic level is high and all nodes are stationary.

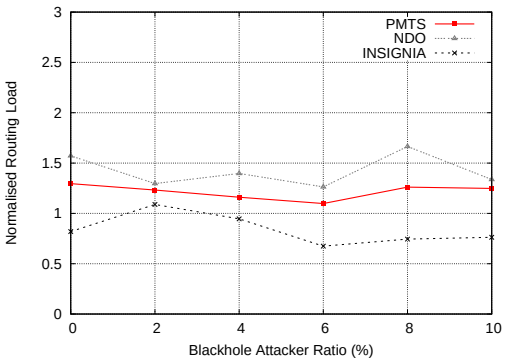
The remaining results presented in this section are for the 0 second pause time, i.e., with all nodes constantly mobile. With a low network load, 2-DAARC generally supports higher PDRs than INSIGNIA in the presence of blackhole attackers. As can be seen in Fig. 5.18(a), 2-DAARC supports higher PDRs at all non-zero attacker ratios. However, compared with the static network (Fig. 5.16(a)), 2-DAARC offers a smaller increase in PDR over INSIGNIA. The largest difference in PDR is at the 4% attacker ratio, where the PDR is 81% for 2-DAARC and 76% for INSIGNIA. This is half of the largest difference experienced when all of the nodes are static. Thus, whilst 2-DAARC still supports higher PDRs than INSIGNIA, node mobility reduces the benefits that 2-DAARC brings. The reduction in PDRs is a consequence of mobility-induced path breaks. In response to the path breaks, a source node attempts to change paths, but the continual node mobility means that the paths in its Route Cache quickly become stale. Some data packets are therefore dropped whilst the source node performs a Route Discovery operation to obtain a new path to the destination node, and hence the lower PDRs. Additionally, node mobility means that the paths may be changed at least once during 2-DAARC's feedback interval. This means that the feedback received by a source node is stale. Thus source nodes make adaptation decisions based on stale QoS statistics.

Another consequence of node mobility is that both 2-DAARC and INSIGNIA experience lower PDRs compared with the stationary network. For example, when the attacker ratio is 0% and nodes are mobile (Fig. 5.18(a)), both achieve PDRs of 95% compared with both achieving 98% PDRs when nodes are stationary (Fig. 5.16(a)). The greatest difference in PDRs between the mobile and static scenarios is at the 10% attacker ratio: with mobile nodes, 2-DAARC's PDRs are 65%, down from 74% with stationary nodes; INSIGNIA's PDRs are 60% with mobile nodes, down from 68% with stationary nodes. These reduced PDRs are a consequence of the mobility-induced path breaks and stale Route Cache reasons given above. Additionally, node mobility reduces the bandwidth in the network. There is more traffic competing for this bandwidth as nodes attempt to retransmit data packets which have failed to be delivered.

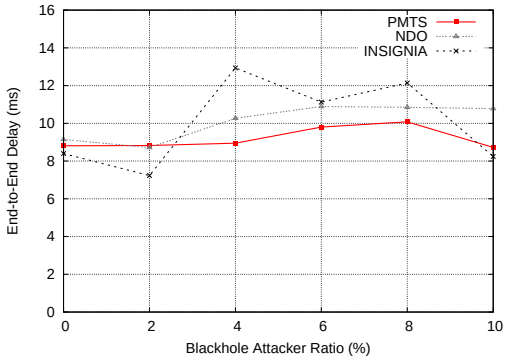
2-DAARC has higher normalised routing loads when nodes are mobile compared with a network containing stationary nodes. Although 2-DAARC achieves lower PDRs than a network with static nodes, it now requires more control packets to support the data packet deliveries. As can be seen in Fig. 5.18(b), 2-DAARC's



(a) Packet Delivery Ratio



(b) Normalised Routing Load



(c) End-to-End Delay

Figure 5.18: Comparing the PDR, Normalised Routing Load, and End-to-End Delay for a 0 Second Pause Time and 3 Priority Sources (with 7 Best-Effort, Background Sources) and Packet Salvaging Enabled

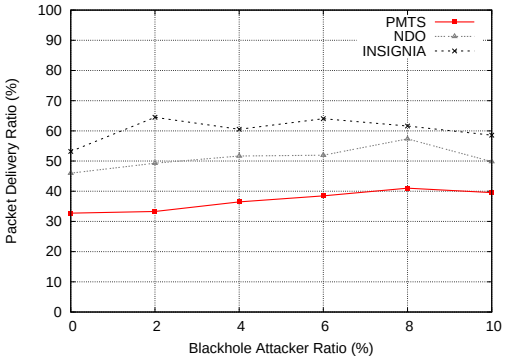
normalised routing loads are between 1 and 1.5 control packets. With stationary nodes (Fig. 5.16(b)), the normalised routing loads are between 0.5 and 1 control packets. 2-DAARC's higher normalised routing loads are a consequence of performing Route Discovery operations in response to the mobility induced path breaks: there is an increase in the number ROUTE ERROR packets transmitted to source nodes to inform them of link breaks; and, consequently, there is an increase in the number of ROUTE REQUEST and ROUTE REPLY control packets transmitted as source nodes respond with more Route Discovery operations.

2-DAARC experiences longer end-to-end delays when nodes are mobile compared with a network containing stationary nodes. As can be seen by comparing Fig. 5.18(c) with Fig. 5.16(c), the delays generally increase as the attacker ratios increase. This trend is similar to the trend observed when nodes are static. One difference between the two cases is that with mobile nodes, 2-DAARC's delays

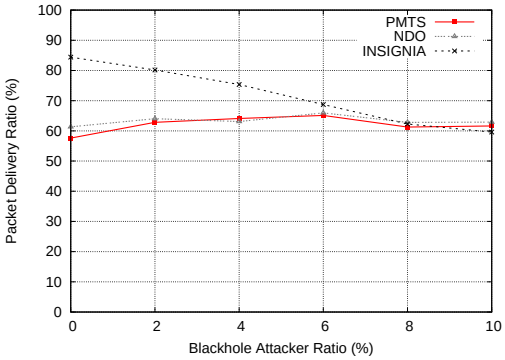
are generally greater than 8ms, whereas with stationary nodes the delays are generally less than 8ms. The longer delays are a consequence of the reduced effective bandwidth resulting from node mobility. For example, an increasing amount of bandwidth is consumed by control packet transmissions and the retransmission of data packets which have failed to be delivered. This increases the queueing delays of data packets, and in turn their end-to-end delays. Node mobility has made INSIGNIA's delays more variable than those observed in the stationary network. The range of INSIGNIA's delay is approximately 6ms, compared with approximately 2ms in the static network. INSIGNIA's variations in delay are a consequence of fewer packets receiving the reserved forwarding service: the frequent mobility-induced path changes lead to the selection of greater number of paths which can only offer the degraded (best-effort) forwarding service.

When the network load is increased to that of 20 source nodes and all other parameter values remain unchanged, 2-DAARC achieves significantly lower QoS than INSIGNIA. The PDRs for both 2-DAARC and INSIGNIA are lower than in the stationary networks and the mobile network presented above. This is because the combination of higher network load and node mobility reduce the available bandwidth and node resources, and these lead to both 2-DAARC and INSIGNIA achieving lower PDRs. As can be seen in Fig. 5.19(a), 2-DAARC achieves significantly lower PDRs than INSIGNIA. This is a consequence of 2-DAARC's adaptation algorithm switching to the MPA mode even when the network is highly congested: the congestion and attacks cause packet loss which is greater than ρ (the packet loss adaptation threshold), and 2-DAARC responds by switching to the MPA mode. Consequently, the duplicated data packets injected into the network exacerbate the existing congestion. This is a shortcoming of the 2-DAARC design, as 2-DAARC lacks the capability to use only the SPA mode when the use of the MPA mode may do more harm than good. This shortcoming is addressed in the following chapter.

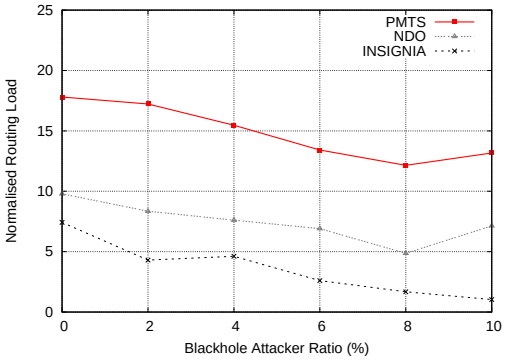
2-DAARC transmits a large number of control packets in an attempt to support the data packet deliveries, and these negatively affect the PDRs and end-to-end delays. As can be seen in Fig. 5.19(c), the normalised routing loads of 2-DAARC are significantly greater than those of INSIGNIA. The high normalised routing load is a consequence of the large number of Route Error control packets transmitted when the original and/or the duplicated data packets fail to be delivered, in addition to the control packets transmitted during the Route



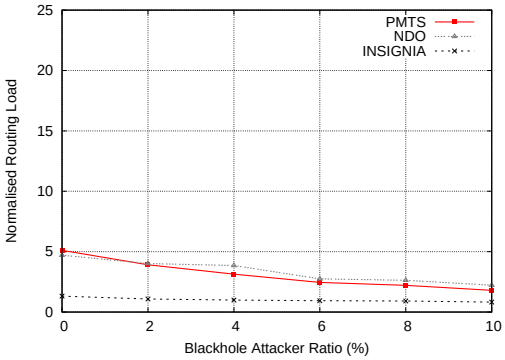
(a) PDR: Packet Salvaging On



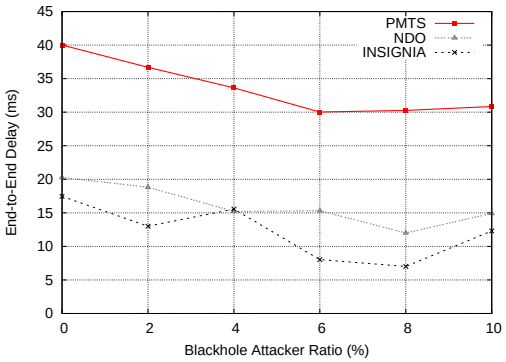
(b) PDR: Packet Salvaging Off



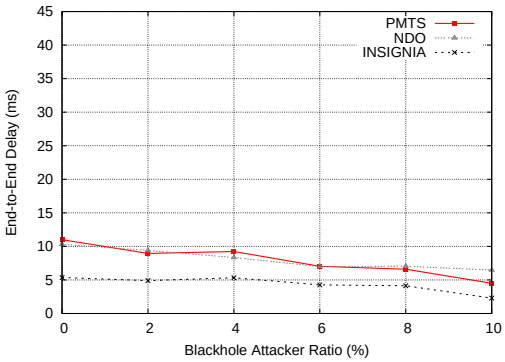
(c) Normalised Routing Load: Packet Salvaging On



(d) Normalised Routing Load: Packet Salvaging Off



(e) End-to-End Delay: Packet Salvaging On



(f) End-to-End Delay: Packet Salvaging Off

Figure 5.19: Comparing the PDR, Normalised Routing Load, and End-to-End Delay for a 0 Second Pause Time and 6 Priority Sources (with 14 Best-Effort, Background Sources), with Packet Salvaging Enabled and Disabled.

Discovery operations. INSIGNIA's normalised routing loads are lower as there is only one copy of each data packet in the network, and therefore fewer congestion-induced path breaks. One trend observed from the figure is that the normalised routing loads generally decrease as the attacker ratios increase. This is different

to the trends observed for both the stationary networks and the lightly loaded mobile network, presented above, where the normalised routing loads are either flat or increase as the attacker ratio increases. The reason for this difference is that increasing the number of blackhole attackers decreases the network congestion. This is because data packets are dropped. Fewer ROUTE ERROR packets are transmitted to notify the source nodes of broken links and, consequently, fewer Route Discovery operations are performed. This makes the network less congested and reduces the number of control packets required to support data packet deliveries.

The highly congested network conditions lead to 2-DAARC experiencing significantly longer end-to-end delays than INSIGNIA. The high volume of control packets, original and duplicated data packets, and retransmitted data packets place a significant strain on network resources. This reduces the effective network bandwidth and resources available for data packet forwarding. Data packets therefore have to wait for increasing durations to be forwarded at each hop, and this contributes to the long end-to-end delays.

Another factor significantly influencing the QoS achievable under these network conditions is the packet salvaging optimization of the underlying routing protocol (DSR). When this is enabled, nodes attempt to retransmit data packets which they fail to deliver to the next-hop node. In congested network conditions, attempting to retransmit these data packets exacerbates congestion. Both 2-DAARC and INSIGNIA use the DSR protocol, and therefore the packet salvaging optimization. However, as INSIGNIA only transmits one copy of each data packet, attempting to retransmit them does not have as significant an effect on congestion as 2-DAARC attempting to retransmit both the original and duplicated data packets.

Disabling the packet salvaging optimization, and leaving all other parameter values unchanged, leads to a marked improvement in 2-DAARC's QoS, although it is generally less than the QoS of INSIGNIA (also with packet salvaging disabled). The PDRs for both 2-DAARC and INSIGNIA with packet salvaging disabled (Fig. 5.19(b)) are significantly greater than the PDRs with packet salvaging enabled (Fig. 5.19(a)). Disabling packet salvaging leads to reduced congestion. This makes more bandwidth and resources available for packet deliveries. Consequently, there is a reduction in the number of congestion-induced path breaks.

However, at the 0%–6% attacker ratios 2-DAARC's PDRs are lower than INSIGNIA's. More packets are dropped by 2-DAARC due to congestion compared with INSIGNIA. This is due to the MPA mode being used in congested conditions. At the remaining attacker ratios, 2-DAARC and INSIGNIA achieve similar PDRs. This is because the higher blackhole attacker ratios mean that attacks become the dominant factor causing packet loss, and both 2-DAARC and INSIGNIA are affected similarly in the given network conditions.

At the 0% attacker ratio, 2-DAARC's QoS is lower than that achieved by the version of 2-DAARC which uses only the SPA mode. By comparing Fig. 5.19(b) with Fig. 5.15(b), it can be seen that the PDR of 2-DAARC with dynamic adaptation is around 60% compared with 83% for 2-DAARC using only the SPA mode. Using only the SPA mode in a congested network containing no attackers achieves higher PDRs as only the original copy of each data packet is transmitted. This places a lower burden on the network than transmitting both the original and duplicate copies using the MPA mode.

The lower congestion resulting from disabling packet salvaging leads to lower normalised routing loads and shorter end-to-end delays than 2-DAARC with packet salvaging enabled. The reduction in normalised routing loads is due to fewer Route Error control packet transmissions and fewer Route Discovery operations being performed. This occurs because there are fewer congestion-induced path breaks. The reduced normalised routing loads can be seen by comparing Fig. 5.19(d) with Fig. 5.19(c). The reduced congestion means that 2-DAARC's delays with packet salvaging disabled (Fig. 5.19(f)) are less than those with packet salvaging enabled (Fig. 5.19(e)). However, 2-DAARC's delays are still longer than INSIGNIA's delays. This is because 2-DAARC suffers from effects of congestion more than INSIGNIA.

In summary, disabling packet salvaging improves 2-DAARC's QoS beyond that of 2-DAARC with packet salvaging enabled, but 2-DAARC still generally achieves lower QoS than INSIGNIA; and in congested, mobile networks free of attackers, 2-DAARC supports better QoS when it uses only the SPA mode.

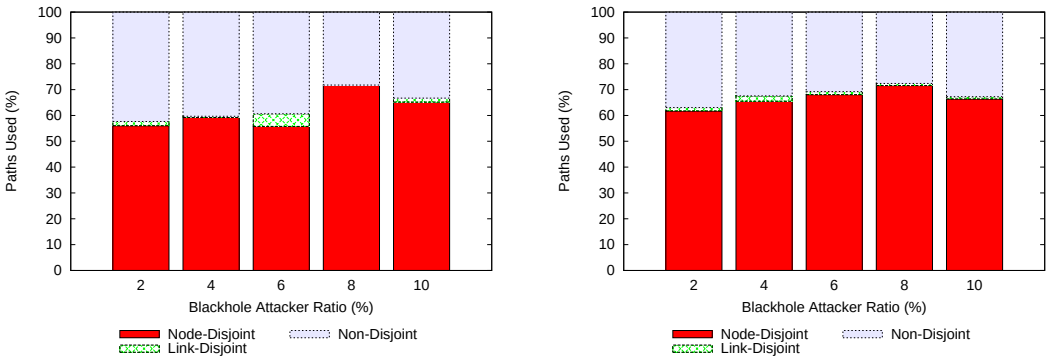
5.6.3.3 Comparing the PMTS and NDO Approaches

Having compared 2-DAARC with INSIGNIA, this section investigates and compares the QoS 2-DAARC achieves when using PMTS and NDO for secondary path selection. The aim here is to determine whether using paths which may have less

than the maximum disjointedness can offer a similar QoS and a lower control packet overhead compared with using only node-disjoint paths. The NDO path selection mechanism uses (1) the PMTS algorithm to find node-disjoint paths and (2) the MuTS method to select the shortest of the node-disjoint paths as the secondary path. The results presented in this section make references to Fig. 5.16–Fig. 5.19 presented in the previous section. The results are presented and discussed first for the lightly and heavily loaded static networks, followed by lightly and heavily loaded mobile networks.

In a lightly loaded stationary network containing attackers, NDO achieves greater PDRs than PMTS. As shown in Fig. 5.16(a), at the non-zero attacker ratios NDO generally achieves greater PDRs than PMTS. This is because the node-disjoint paths selected using NDO contain fewer blackhole attackers than the paths selected using PMTS. The reason for this is that the paths selected with PMTS may not be node-disjoint. As shown in Fig. 5.20(a), on average 38% of the paths selected using PMTS are non-disjoint and 2% are link-disjoint. 62% of the secondary paths selected using PMTS contain blackhole attackers compared with 37% of the paths selected using NDO. With PMTS, 48% of the pairs of primary and secondary paths contain blackhole attackers. Moreover, 65% of these pairs of paths contain the same attacker. With NDO, 16% of the pairs of primary and secondary paths contain attackers (the same attacker node cannot appear in both the primary and the secondary paths when using NDO, as node-disjoint paths do not share any intermediate nodes by definition). In other words, secondary paths selected using PMTS may share intermediate nodes with the primary paths, and this means that attacker nodes are more likely to affect both the primary and secondary paths, leading to lower PDRs.

In addition to the greater PDRs, using NDO for path selection in the given network conditions also generally leads to lower end-to-end delays than using PMTS. Fig. 5.16(c) shows the end-to-end delays. The paths selected using NDO are more lightly loaded than those selected using PMTS. This is because they offer greater resource redundancy than the paths selected using PMTS (which may have shared nodes). Having greater redundancy on the node-disjoint paths is beneficial because the duplicated data packets poured on to these paths increase the traffic load, and this reduces the effective bandwidth and resource availability. The bandwidth and resource consumption is spread across two paths which do not have any common nodes. When using paths which are not node-disjoint, the



(a) 900 Second Pause Time

(b) 0 Second Pause Time

Figure 5.20: Percentages of Node-Disjoint, Link-Disjoint, and Non-Disjoint Paths Selected by PMTS when using the MPA Mode with 3 Priority Sources (and 7 Best-Effort, Background Sources) and Packet Salvaging Enabled

common node(s) between the primary and secondary paths will have to forward both the original and duplicated copies of each data packet. This places a greater burden on the shared nodes' resources and can lead to congestion, which in turn causes longer delays. The congestion and longer delays may affect both the primary and secondary paths. Distributing the original and duplicated data packets across node-disjoint paths therefore leads to shorter end-to-end delays in the given network conditions.

However, as shown in Fig. 5.16(b), NDO transmits more control packets than PMTS to support the data packet deliveries. The higher normalised routing loads are due to NDO triggering Route Discovery operations when node-disjoint paths cannot be found in the Route Cache. These operations occur regardless of whether the Route Cache contains link-disjoint and non-disjoint paths to the destination node. PMTS makes better use of the paths contained in the Route Cache by using the best of the available paths. Its normalised routing loads are therefore lower than those of NDO: it performs fewer Route Discovery operations and injects fewer control packets into the network.

When the network load is increased to that of 20 source nodes, and all other parameter values remain unchanged, PMTS and NDO generally achieve similar PDRs. As shown in Fig. 5.17(a), the PDRs are similar at most attacker ratios. Using NDO to select paths which are node-disjoint therefore does not achieve higher PDRs than using PMTS to select paths which may be link- or non-disjoint. This is the opposite of the trend observed with the lower network load. The

reason for this is that with the higher load, the network is already close to being congested, and using the MPA mode to transmit duplicated data packets pushes the network into a congested state. As nodes are stationary, new paths are unlikely to be found during a session. 2-DAARC therefore continues to forward packets along the existing congested paths. The congestion in the static network therefore limits the benefits that NDO brings over PMTS. The similar PDRs are also related to the normalised routing loads of the two approaches. As can be seen by comparing Fig. 5.17(b) with Fig. 5.16(b), the normalised routing loads of PMTS and NDO are both greater than in the lightly loaded network. Moreover, the normalised routing loads exhibit greater similarity when the network load is higher. Thus PMTS and NDO now achieve similar PDRs using a similar number of control packets.

The end-to-end delays of PMTS are shorter than those of NDO at all attacker ratios. This is the opposite of the trend observed with the lower network load. This can be seen in Fig. 5.17(c). The lower delays are because some of the paths selected by PMTS are not node-disjoint: as shown in Fig. 5.21, on average, approximately 43% of the paths used by PMTS are link- and non-disjoint. These paths are generally shorter than the node-disjoint paths selected by NDO, i.e., they have a smaller hop-count. This is because link- and non-disjoint paths share at least one node with the primary path, and the primary path is the path in the Route Cache with the shortest hop-count to the destination node. Enforcing the node-disjointedness criteria generally makes the paths longer as the secondary path cannot share any nodes with the primary path. The paths selected with PMTS are, on average, 0.1 hops shorter than those selected by NDO. These shorter paths contribute to the shorter end-to-end delays.

In a lightly loaded network containing mobile nodes, PMTS and NDO achieve similar PDRs, but PMTS is the better of the two approaches when the normalised routing loads and end-to-end delays are considered. As shown in Fig. 5.18(a), PMTS and NDO achieve similar PDRs at all attacker ratios. Node mobility reduces the benefits of using only node-disjoint paths compared with the lightly loaded static network (Fig. 5.16(a)). Node mobility causes mobility-induced path breaks. Path breaks lead to frequent path changes. The NDO approach has to perform more Route Discovery operations than PMTS to find paths, as demonstrated by its higher normalised routing load in Fig. 5.18(b). This is because the paths must satisfy the node-disjointedness criteria whereas paths selected using

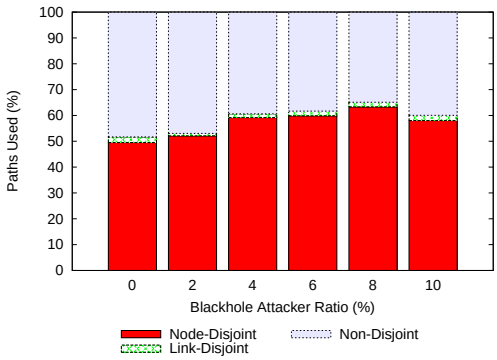


Figure 5.21: Percentages of Node-Disjoint, Link-Disjoint, and Non-Disjoint Paths Selected by PMTS when using the MPA Mode with 6 Priority Sources (and 14 Best-Effort, Background Sources), a 900 Second Pause Time, and Packet Salvaging Enabled.

PMTS may have a lower disjointedness. When nodes are static, the larger number of Route Discovery operations leads to NDO selecting paths which support higher PDRs than those paths selected using PMTS; but the introduction of node mobility means that the higher normalised routing loads are experienced without the benefits of the higher PDRs. PMTS uses the paths in the Route Cache more effectively than NDO. As shown in Fig. 5.20(b), on average approximately 34% of the paths selected by PMTS are not node-disjoint.

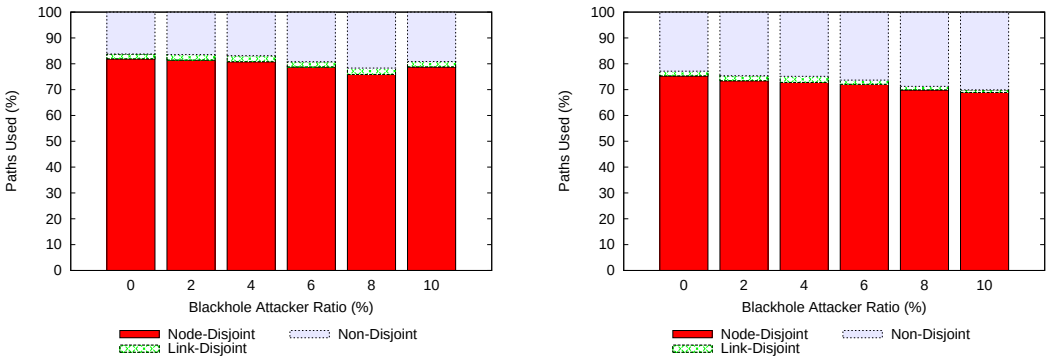
In addition to the lower normalised routing loads, PMTS achieves lower end-to-end delays than NDO. Fig. 5.18(c) shows the end-to-end delays. One reason that the delays of PMTS are lower than those of NDO is that it transmits fewer control packets, and this means that there is more bandwidth available for data packet transmissions. A benefit of this is that data packet queues are more lightly loaded as fewer control packets have to be transmitted. However, the delays of PMTS and NDO are both longer than those experienced when all nodes are stationary (Fig. 5.16(c)). This is due to the reasons given earlier, where mobility leads to decreased effective bandwidth and resources, and the MPA mode transmits duplicated data packets in this more resource-limited environment.

When the network load is increased to that of 20 source nodes, and all other parameter values remain unchanged, the QoS of PMTS is markedly worse than that of NDO. The combination of high network load and node mobility lead to high network congestion. PMTS suffers the effects of congestion more than NDO. As can be seen in Fig. 5.19(a), the PDRs of PMTS range between 33%–41% whereas the PDRs of NDO range between 46%–57%. The disjointedness of

the paths selected using PMTS contributes to network congestion and its lower PDRs. PMTS sometimes selects secondary paths which are link- or non-disjoint with the primary path. These paths share intermediate nodes. Sharing nodes leads to lower redundancy and fewer resources than node-disjoint paths. Congestion at one node on a link- or non-disjoint secondary path may affect both the primary and secondary paths simultaneously due to their interdependence. The shared nodes have the responsibility of forwarding a large number of original and duplicated data packets, and they also attempt to retransmit these packets if the link to the next-hop node fails. This makes the network increasingly busy at these nodes and exacerbates the already congested and resource constrained conditions. For example, overflowing data packet queues cause 66% of the packets loss for PMTS compared with 36% for for NDO. With NDO, paths do not share common intermediate nodes. Thus the burden of packet forwarding is distributed amongst a broader group of nodes.

One of the factors contributing to the lower PDRs of PMTS is its high normalised routing load. PMTS experiences more congestion-induced path breaks than NDO. These require a large number of ROUTE ERROR control packets transmissions to inform the source nodes of the broken links. As can be seen in Fig. 5.19(c), the normalised routing loads of PMTS are greater than those of NDO at all attacker ratios. On average, PMTS transmits 1.4 times more ROUTE ERROR control packets than NDO. The effect of this is that the wireless medium is increasingly being captured for control packet transmissions, and this exacerbates the exiting congestion. Additionally, this higher congestion leads to longer end-to-end delays. As shown in Fig. 5.19(e), the delays of PMTS are, on average, double those of NDO. This is a consequence of the higher congestion and paths sharing intermediate nodes, as described earlier.

Another factor contributing to the large differences in QoS between PMTS and NDO is the packet salvaging optimization. This is because paths selected using PMTS may share nodes, and these may have to retransmit both the original and duplicate data packets if the link to the next-hop node fails. In contrast, nodes on paths selected using NDO will only have to retransmit either the original or duplicated version of a packet if a link failure occurs. Disabling packet salvaging leads to (1) a marked improvement in the QoS achieved by both PMTS and NDO, and (2) PMTS achieving similar QoS to NDO. Disabling packet salvaging reduces congestion. This is because nodes do not attempt to retransmit data



(a) Packet Salvaging Enabled

(b) Packet Salvaging Disabled

Figure 5.22: Percentages of Node-Disjoint, Link-Disjoint, and Non-Disjoint Paths Selected by PMTS when using the MPA Mode with 6 Priority Sources (and 14 Best-Effort, Background Sources) and a 0 Second Pause Time.

packets which have failed to be delivered. Consequently, both the normalised routing loads (Fig. 5.19(d)) and the end-to-end delays (Fig. 5.19(f)) also reduce. One interesting observation on the improved QoS of PMTS is that the data packets are forwarded along fewer node-disjoint paths than in the network with packet salvaging enabled: approximately 72% of the paths are node-disjoint with packet salvaging disabled (Fig. 5.22(b)) compared with 80% with packet salvaging enabled (Fig. 5.22(a)). Using fewer node-disjoint paths means less redundancy and less resilience, thus lower PDRs. However, congestion is a significant factor affecting the packet loss: disabling packet salvaging has reduced the congestion, thus the burden placed on the nodes shared between the primary and secondary paths is greatly reduced; these nodes now have more resources available to service the priority traffic compared with having packet salvaging enabled. Thus the QoS of PMTS and NDO is similar in a highly loaded network containing mobile nodes which have their packet salvaging optimization disabled.

In this highly loaded mobile network, the number of ROUTE ERROR packets in the network varies considerably depending on whether or not the packet salvaging optimization is enabled. Figure 5.23 shows the ratio of the ROUTE ERROR packets transmitted for all priority data packets to the number priority data packets delivered to their intended destination nodes. There is a marked difference in the number of ROUTE ERROR packets transmitted when packet salvaging is enabled and when it is disabled. Four trends are observed. First, the number of ROUTE ERROR packets decreases as the attacker ratio increases.

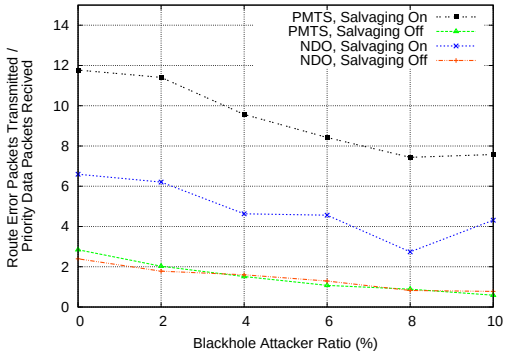


Figure 5.23: Comparing the ROUTE ERROR Packet Rate for PMTS and NDO with Packet Salvaging Enabled and Disabled for 6 Priority Sources (14 Best-Effort, Background Sources) and a 0 Second Pause Time.

This is due to the blackhole attacks. The attacks lead to a lower load on downstream nodes' data packet queues. This reduces the effects of congestion. Lower congestion means fewer failed packet deliveries and fewer ROUTE ERROR packet transmissions. Second, the number of ROUTE ERROR packets is higher when packet salvaging is enabled. This is due to intermediate nodes failing to deliver the original and salvaged versions of a data packet to a next-hop node. Third, more ROUTE ERROR packets are transmitted for PMTS than NDO when packet salvaging is enabled. This is because PMTS selects paths which are link- and non-disjoint, and these paths are more congested than node-disjoint paths (as described earlier). This higher congestion leads to a larger number ROUTE ERROR packet transmissions when data packets are not delivered to their next-hop node. The final trend is that the number of ROUTE ERROR packets transmitted is similar for both PMTS and NDO when packet salvaging is disabled. This is because disabling packet salvaging reduces the congestion in the network; reduced congestion leads to similar PDRs for both PMTS and NDO (as shown in Fig. 5.19(b)).

5.6.4 Major Findings

The simulation study of 2-DAARC has led to the following three major findings.

- 2-DAARC supports better QoS than INSIGNIA in lightly loaded networks containing blackhole attackers and in networks free of attackers. It supports better QoS in terms of greater service quality, higher PDR, and shorter,

or in the worst-case, similar end-to-end delays. These improvements in QoS are achieved using two modes of adaptation (SPA and MPA) to adapt dynamically to changes in network conditions.

- The PMTS algorithm can lead to better or similar QoS than a node-disjoint-path-only (NDO) approach. PMTS leads to better QoS than NDO in lightly loaded networks where nodes are mobile. It supports similar QoS to NDO in more heavily loaded networks.

- Adaptation decisions should also consider congestion information to ensure that the most appropriate adaptation mode is used for the network conditions. The results show that QoS cannot be supported effectively when adaptation decisions are based solely on packet loss and service quality statistics in congested networks.

2-DAARC's end-to-end feedback mechanism plays an important role in the QoS achieved in the presence of attacks. The feedback mechanism and the feedback data carried by this mechanism are exposed to the untrusted intermediate nodes which forward the feedback data.

Authenticity of feedback data. A message authenticity service is used to provide integrity protection and origin authentication of the feedback data. The integrity and authentication of these data is assured using the following three steps.

1. Feedback data are generated at the destination node and are communicated to the source node in an end-to-end manner. The destination node and source node have a security association. It is assumed that (1) they trust one another and (2) the destination node does not fabricate feedback data. Untrusted intermediate nodes are not involved in feedback data generation.

2. The integrity of feedback data is ensured using a keyed-hash function with a shared symmetric (secret) key known only to the source and destination nodes of a priority packet flow. A message integrity service is necessary because intermediate nodes, which forward the feedback data, cannot be trusted in an open MANET environment. An intermediate node could alter the feedback data in the packet, but it is not capable of generating a

valid keyed-hash of these data. This is because it does not have a copy of the symmetric key shared by the source and destination nodes.

3. The origin of the feedback data is ensured using a symmetric key with the keyed-hash function. The symmetric key is known only to the source and destination nodes. The source node can confirm the origin of a feedback packet if it can generate a matching hash of the feedback data using its symmetric key; the only other node to know this key is the destination node which has generated the initial hash of the feedback data. Using a shared symmetric key, known only to the source and destination nodes, means that it is hard for an intermediate node to forge feedback data.

From the findings presented in this section it is possible to conclude that 2-DAARC supports better QoS than INSIGNIA in a number of different network conditions, including under security attacks. However, weaknesses have also been identified in the 2-DAARC approach, i.e., adaptation in response to packet loss statistics alone is not effective.

can be taken. 2-DAARC's current approach is too coarse-grained. It assumes that the main cause of packet loss is blackhole attacks. However, network congestion can be the main cause of packet loss, even in the presence of attacks. To make 2-DAARC's adaptation process better suited to the dynamic network conditions the different causes of packet loss should be identified. The different causes of packet loss require different adaptation responses. For example, if the main cause of packet loss is blackhole attacks the MPA mode should be used to support QoS; if the main cause of packet loss is congestion the SPA mode should be used, as the MPA mode may further congest the network and worsen QoS. The following chapter presents and evaluates a novel mechanism for attack and congestion detection. The information generated by the detection mechanism is used to make adaptation decisions which are tailored to the current network conditions.

5.7 Chapter Summary

This chapter presented the design and evaluation of two novel ideas to achieve QoS in networks containing packet forwarding attackers. The 2-Dimensional Adaptation ARChitecture (2-DAARC) has been proposed which incorporates these novel

ideas. The first novel idea is a 2-dimensional approach to QoS. This 2-dimensional approach includes a single-path adaptation (SPA) mode and a multi-path adaptation (MPA) mode. The second novel idea is a priority-based approach to secondary path selection. The Priority-based Multi-path Type Selection (PMTS) algorithm has been proposed for this purpose. It is used by 2-DAARC's MPA mode.

Based on the findings in this chapter, it was observed that 2-DAARC'S current approach is too coarse-grained. 2-DAARC should be extended so that the different causes of packet loss can be identified and the appropriate adaptive actions can be taken. In other words, if the main cause of packet loss is blackhole attacks the MPA mode should be used to support QoS; if the main cause of packet loss is congestion the SPA mode should be used, as the MPA mode may further congest the network and worsen QoS. The following chapter presents and evaluates a novel mechanism for attack and congestion detection. The information generated by the detection mechanism is used to make adaptation decisions which are tailored to the current network conditions.

Chapter 6

Extended 2-DAARC (E2-DAARC)

6.1 Chapter Introduction

This chapter presents the design and evaluation of an extended version of 2-DAARC (E2-DAARC). E2-DAARC addresses the shortcomings in 2-DAARC (which have been discussed in Section 5.7). E2-DAARC is built by integrating a novel Congestion and ATtack (CAT) detection mechanism into 2-DAARC. The CAT detection mechanism infers the likely reason for packet loss (congestion and/or attacks) so that the most appropriate mode of transmission (the SPA mode or the MPA mode) is invoked for the given network conditions. Additionally, it uses an adaptive packet salvaging mechanism to minimise the effects of congestion. The main focus of these mechanisms is on supporting QoS provisioning in potentially congested networks containing a variable number of data packet forwarding attackers.

The chapter is organised as follows. Section 6.2 presents the CAT detection mechanism at a high level. Section 6.3 presents the design preliminaries for the CAT detection mechanism, including the design requirement and design principle. Section 6.4 describes the CAT detection mechanism in detail. Section 6.5 presents a simulation-based performance evaluation of E2-DAARC. In this evaluation, the effectiveness and efficiency of using the CAT detection mechanism are evaluated against those of its peers, the Explicit Congestion Notification (ECN) congestion control mechanism and the Watchdog misbehaviour detection mechanism. In addition, E2-DAARC is evaluated against INSIGNIA. This section also discusses

the major findings from the evaluation. Finally, Section 6.6 presents the chapter summary.

6.2 A Novel Idea: the Congestion and ATtack (CAT) Detection Mechanism

The third novel idea presented in this thesis, the CAT detection mechanism, is a novel mechanism used to detect whether packet loss occurs as a consequence of congestion and/or packet forwarding attacks so that the adaptation response can be tailored more effectively to the network conditions. Section 6.2.1 presents the motivation for the CAT detection mechanism, and the challenging issues faced in detecting congestion and attacks in MANETs containing untrusted intermediate nodes. In response to the outcome of the detection, an appropriate mode of transmission should be invoked to optimise packet deliveries. Section 6.2.2 discusses how the information obtained from the CAT detection mechanism can be used to perform the adaptation.

6.2.1 Detection

6.2.1.1 Design Rationale

It was observed in the previous chapter that the selection of an adaptation mode should take into consideration the cause of packet loss. This is because the mode of adaptation which is suitable for one cause of packet loss may not be suitable for another cause of loss. For example, the MPA mode is enabled when packet loss exceeds the packet loss threshold ρ, but this is only effective if the packet loss is a consequence of attacks. If the packet loss is caused by congestion or a combination of congestion and attacks, the use of the MPA mode will exacerbate the existing congestion. This will lead to further packet loss and will worsen QoS. Using the SPA mode is more appropriate in such conditions. It is therefore necessary to detect the different causes of packet loss and to discriminate between them so that the most appropriate adaptive actions can be taken.

There are two main challenges to overcome when detecting the causes of packet loss: (1) performing detection without imposing any trust on intermediate nodes and (2) performing detection with as little additional overheads as possible (i.e., using as few additional control packets as possible). In the relevant literature of

MANET research it is typically assumed that nodes are trustworthy and that they will perform the operations asked of them [86]. The two prominent approaches to misbehaviour detection both rely on intermediate nodes performing detection operations faithfully and correctly. The first approach uses promiscuous receive mode to monitor neighbouring nodes' transmissions to detect misbehaviours [21, 123]. For example, Watchdog [123] uses promiscuous overhearing to determine whether the next-hop node in a path forwards the data packets it receives. If it does not, it is considered to be misbehaving. The second approach uses probe packets to identify malicious nodes [6, 7, 91, 178, 208]. For example, the method in [91] uses a probe packet which is indistinguishable from a data packet (although it is in effect a control packet as it does not carry payload data to the destination node). If a source node A suspects node B of being malicious it will send a probe packet to node C, which is beyond node B on the path to the destination node. If node C receives the probe packet, it should transmit an acknowledgement to the source node A. If node C does not receive the probe packet, it will not transmit an acknowledgement. The source node A will determine that node B did not forward the probe packet to node C. Both of these approaches place implicit trust on the intermediate nodes. However, in an open MANET environment, intermediate nodes can neither be trusted to perform any additional operations nor to provide accurate and trustworthy information [182].

There are also two prominent approaches to congestion detection, both of which are reliant on the actions of intermediate nodes. In the first approach, each node monitors its packet buffer occupancy and informs other nodes when the occupancy exceeds a threshold value [61, 165]. The second approach is an extension of the first: packet buffer occupancy information is combined with delay information from the MAC layer to determine the extent to which the wireless medium is busy [66, 107]. This approach takes into consideration the congestion at neighbouring nodes in addition to the congestion at the node performing the congestion detection. However, as intermediate nodes cannot be trusted to perform these additional congestion control operations, a solution which minimizes the number of operations imposed on intermediate nodes is preferable in an open and untrusted MANET environment.

The second challenging issue to overcome when detecting the cause of packet loss is to perform the detection without injecting additional control packets into the underlying network. The misbehaviour detection mechanisms and some of

the congestion control mechanisms described above make use of control packets to notify other nodes of any changes in network conditions. These control packets are transmitted in addition to those already transmitted by the underlying routing protocol. The additional control packets increase the network load. In the case of the congestion detection mechanisms, the control packets are being injected into a network which is already congested. This will exacerbate congestion. Thus it is preferable to perform attack and congestion detection without having to transmit additional control packets.

The CAT detection mechanism overcomes the limitations of the existing solutions. To facilitate the detection of congestion and attacks, it does not require intermediate nodes to perform any actions other than those already defined in the underlying routing protocol (DSR), and it also makes use of the control packets which 2-DAARC already transmits.

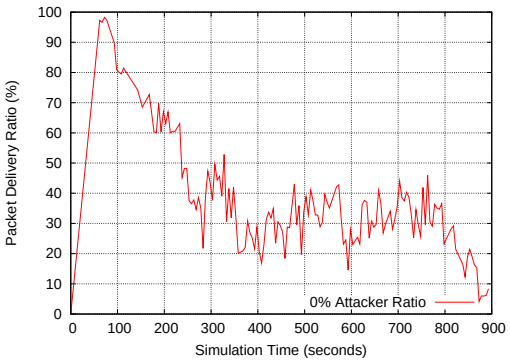
A source node uses two statistics to perform congestion and attack detection, and these statistics rely on it receiving two types of control packet. The first statistic is the packet loss ratio (PLR) which it receives from the destination node in a 2-DAARC feedback packet. The PLR is course-grained, in that it specifies the percentage of priority data packets lost on the in-use path(s) over a time-interval τ, but it does not differentiate between the different causes of packet loss. The source node generates the second statistic based on the number of Route Error (RERR) packets it receives over the same time-interval τ. RERR packets are transmitted as part of DSR's Route Maintenance procedure (as described in Section 3.2.2). They inform a source node that a link on an in-use path has broken. CAT detection uses the number of RERR packets received at a source node in conjunction with the PLR to determine whether the packet loss experienced is likely to be a consequence of broken links or attacks; and by analysing the relationship between the number of RERR packets received and the PLR it is possible to infer whether the broken links are likely to be a consequence of congestion or node mobility. The motivation for using RERR packets in CAT detection is the observation in Section 5.6.3.2 on the relationship between the number of RERR packets in a network and the packet delivery ratio (PDR) achieved. To better understand this observation, a simulation study is used to explore the relationship between the number of RERR packets and the PDR to determine how best to use the RERR packets in CAT detection. Exploring this relationship enables the threshold values of RERR packets and PDR which

indicate congestion and/or attacks to be determined.

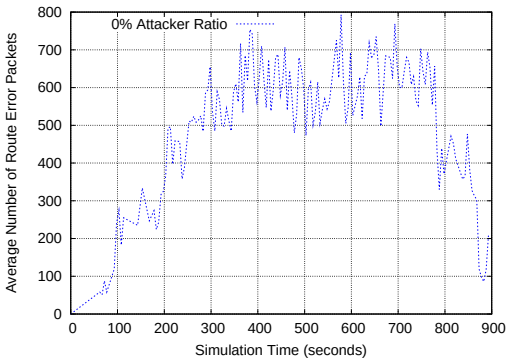
6.2.1.2 A Simulation Study: Examining the Relationship Between Route Error Packets and Packet Loss

It is necessary to examine in detail the relationship between RERR packets and packet loss to determine the extent to which it can be used to indicate congestion and attacks. Through this examination it is possible to identify levels and combinations of RERR packets and PDRs which indicate whether the packet loss is likely to be a consequence of congestion and/or attacks. Figs. 6.1–6.4 show the PDRs and the number of RERR packets for 2-DAARC with 20 source nodes (6 priority source nodes and 14 best-effort, background sources), under 0 and 900 second pause times, and with packet salvaging enabled and disabled. 20 source nodes are used as the amount of traffic generated by this number of source nodes leads to the congested conditions under which the interactions between the number of RERR packets and the PDR can be best investigated. The simulations are performed with packet salvaging enabled and disabled as it was observed in Section 5.6.3.2 that the number of RERR packets and the PDR vary significantly depending on whether or not packet salvaging is enabled (DSR's packet salvaging mechanism is described in Section 3.2.2). The curves plotted in Figs. 6.1–6.4 show the averaged values of RERR and PDR sampled every 5 seconds of the simulation. The 5 second sampling interval is consistent with τ, 2-DAARC's feedback interval. The averaged number of RERR packets shown in the figures includes (1) the RERR packets received by priority source nodes where these source nodes are the targets of the packets, and (2) the RERR packets forwarded by these nodes, i.e., where the source nodes act as intermediate nodes. The forwarded RERR packets may belong to either a priority packet flow or a best-effort packet flow. The justification for including RERR packets from both flow types is that it provides a broader view of the number of RERR packets in the network, and thus a better understanding of network congestion.

The main observation from the simulation results is that the PDR generally decreases as the number of RERR packets increases. This relationship between the PDR and the RERR packets is most prevalent in Fig. 6.1 which shows the results for a 0 second pause time with packet salvaging enabled. The relationship between the PDR (Fig. 6.1(a)) and the RERR packets (Fig. 6.1(b)) is a consequence of congestion and mobility, and is explained as follows. The network



(a) Packet Delivery Ratio

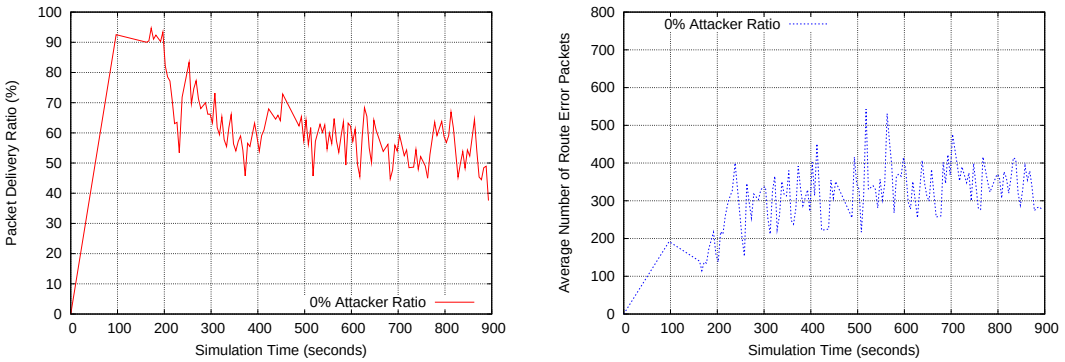


(b) Number of ROUTE ERROR Packets

Figure 6.1: Comparing the PDR and Number of ROUTE ERROR Packets for 6 Priority Sources (14 Best-Effort, Background Sources) with a 0 Second Pause Time and Packet Salvaging Enabled

congestion is a consequence of both the high network load from the 20 source nodes and the constant node mobility. The congestion means that some intermediate nodes are unable to forward data packets to the next-hop nodes because the next-hop nodes' data packet queues are full. Node mobility means that intermediate nodes are unable to forward data packets to the next-hop nodes because they have roamed out of wireless transmission range. In both cases, the intermediate nodes will transmit a RERR packet to the source node of the undelivered data packet to inform it of the failed transmission. The intermediate node will then attempt to salvage the data packet. If the node has an alternative path to the destination node in its Route Cache, it will attempt to forward the packet to the next-hop node in that path. However, network congestion may mean that the next-hop node is too congested to receive the packet. Moreover, attempting to retransmit a packet in an already congested network may exacerbate the congestion. Additionally, as the nodes in the network are mobile, the next-hop node may no longer be in wireless transmission range. In both of these cases, the data packet will be dropped, and this leads to a decrease in PDR.

Disabling packet salvaging to reduce the effects of congestion increases the PDRs and reduces the number of RERR packets in the network. As stated in Section 5.6.3.2, performing packet salvaging can lead to a high number of RERR packets and low PDRs in networks which are already congested: this is because attempting to salvage data packets in an already congested network exacerbates the congestion. The effects of disabling packet salvaging can be seen in Fig. 6.2,

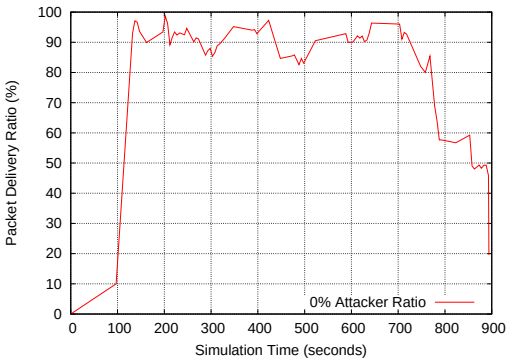


(a) Packet Delivery Ratio (b) Number of ROUTE ERROR Packets

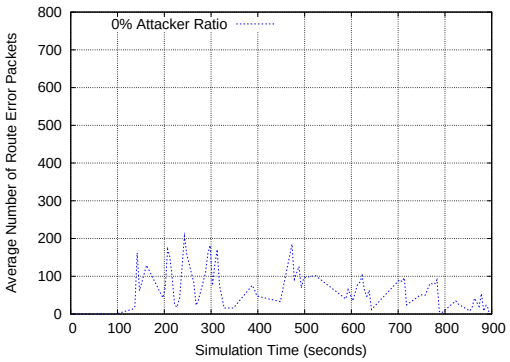
Figure 6.2: Comparing the PDR and Number of ROUTE ERROR Packets for 6 Priority Sources (14 Best-Effort, Background Sources) with a 0 Second Pause Time and Packet Salvaging Disabled

where packet salvaging has been disabled and all other parameter values remain unchanged. The original observation made when packet salvaging was enabled (Fig. 6.1) still holds: the PDR decreases as the number of RERR packets increases; however, the PDR is not as low and the number of RERR packets is not as high as in Fig. 6.1. This is because disabling packet salvaging leads to fewer RERR packet transmissions and a lower level of congestion. Lower congestion leads to fewer link breaks caused by overflowing packet queues. This, in turn, leads to fewer RERR packet transmissions. There are therefore fewer control packets in the already congested network, further reducing the load. And this leads to an increase in PDR. It is therefore necessary to disable packet salvaging to increase PDRs in congested network conditions.

The first observation—PDRs decreasing as the number of RERR packets increases—can be investigated further to identify ranges of PDR and RERR values which are related to one another. These ranges of PDR and RERR values can be used by the CAT detection mechanism to determine the network conditions. When the PDR is in the region of 80–90% the number of RERR packets is typically between 100–200 (for the given network configurations). This mainly occurs when packet salvaging is enabled (Fig. 6.1), although it also occurs less prevalently when packet salvaging is disabled (Fig. 6.2). A PDR of 90% is also a Packet Loss Ratio (PLR) of 10%. A 10% PLR is the value of ρ, the packet loss adaptation threshold. A PDR of 80% is a PLR of 20%, which can be expressed as 2ρ. Thus the PLR falls within the range of $\rho \leq PLR < 2\rho$ when the number

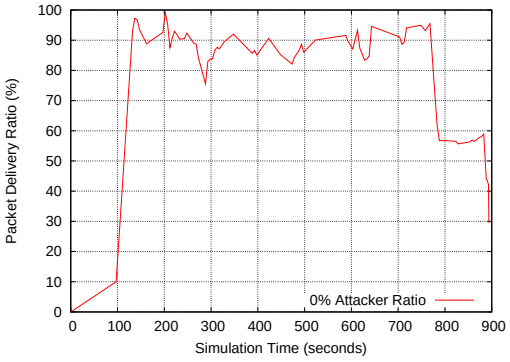


(a) Packet Delivery Ratio

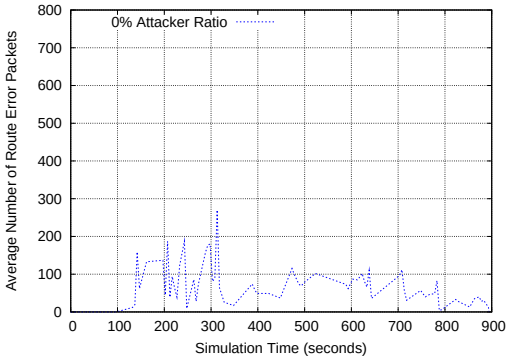


(b) Number of ROUTE ERROR Packets

Figure 6.3: Comparing the PDR and Number of ROUTE ERROR Packets for 6 Priority Sources (14 Best-Effort, Background Sources) with a 900 Second Pause Time and Packet Salvaging Enabled



(a) Packet Delivery Ratio



(b) Number of ROUTE ERROR Packets

Figure 6.4: Comparing the PDR and Number of ROUTE ERROR Packets for 6 Priority Sources (14 Best-Effort, Background Sources) with a 900 Second Pause Time and Packet Salvaging Disabled

of RERR packets is between 100 and 200. In this instance, 100 and 200 RERR packets serve as a lower and an upper threshold on the number of RERR packets received when packet loss occurs in the range of $\rho \leq PLR < 2\rho$. The upper and lower threshold of RERR packets are denoted ϵ and 2ϵ. Additionally, when the PDR is greater than 90%, i.e., the PLR is less than ρ, the number of RERR packets is less than 100, i.e., it is less than ϵ. Similarly, when the PDR is below 80%, i.e., the PLR is greater than 2ρ, the number of RERR packets is generally above 200 packets, i.e., it is greater than 2ϵ.

The main observation when all nodes are stationary (900 second pause time) is that the PDR is higher and the number of RERR packets is lower than when

all nodes are mobile. This is the case when packet salvaging is either enabled (Fig. 6.3) or disabled (Fig. 6.4). As can be seen in Fig. 6.3(a) and Fig. 6.4(a), the PDRs are generally maintained above 80%, i.e., $PLR \leq 2\rho$. The numbers of RERR packets (Fig. 6.3(b) and Fig. 6.4(b)) are generally maintained below 200, i.e., $RERR \leq 2\epsilon$. The combination of the high network load and node mobility shown for the 0 second pause time (Figs. 6.1 and 6.2) therefore has a more significant effect on PDR and the number of RERR packets compared with having all nodes stationary (Figs. 6.3 and 6.4). The combination of the PDR and the number of RERR packets can therefore be considered not only an indicator of the network load but also an indicator of whether or not nodes are mobile.

6.2.1.3 Further Discussions

Three observations are made on the above results. First, three levels of packet loss and three levels of RERR packets can be derived: low, medium, and high. Low packet loss occurs when $PLR < \rho$. Medium packet loss occurs when $\rho \leq PLR < 2\rho$. High packet loss occurs when $PLR \geq 2\rho$. The number of received RERR packets is termed the *RERR rate*. The low RERR rate occurs when $RERR < \epsilon$. The medium RERR rate occurs when $\epsilon \leq RERR < 2\epsilon$. The high RERR rate occurs when $RERR \geq 2\epsilon$. Second, the three levels of packet loss and RERR rate can be combined to determine the likely cause of packet loss. For example, high packet loss with a high RERR rate is likely to be a consequence of congestion. This is because congestion leads to high packet loss as undeliverable data packets are dropped, and RERR packets are transmitted by intermediate nodes to the source nodes to inform them of the dropped data packets. However, high packet loss with a low RERR rate is likely to be a consequence of attacks. This is because attackers do not transmit RERR packets when they drop data packets; it would not make sense for them to do so as it would alert the source node to the dropped packets. Third, packet salvaging can be an advantage or a disadvantage depending on the network conditions. With the 900 second pause time, i.e., a static network, enabling packet salvaging leads to higher PDRs. With the 0 second pause time, i.e., a highly mobile network, disabling packet salvaging leads to higher PDRs. This finding demonstrates that packet salvaging should be used adaptively in response to changes in network conditions.

6.2.2 Response

Having inferred the likely reason for packet loss (congestion and/or attacks), the next step is to make adaptation decisions based on this packet loss reason. Two adaptation decisions need to be made to determine how best to support the QoS requirements of a priority packet flow: (1) whether to use only the SPA mode (hereafter referred to as SPA-ONLY) or the adaptive 2-dimensional approach (i.e., to adapt dynamically between the SPA and the MPA modes in response to network conditions); and (2) whether to use packet salvaging. Determining the most appropriate mode of adaptation is important to support QoS without worsening network congestion. The SPA-ONLY mode should be used if either congestion or congestion and attacks are the main causes of packet loss. In other words, the MPA mode should not be enabled, even if the packet loss exceeds ρ (the packet loss threshold). This is because using the MPA mode would exacerbate the existing congestion due to the injection of duplicated data packets into an already congested network. If attacks are the main reason for packet loss and the network load is light, the adaptive 2-dimensional approach should be used to optimize packet deliveries. This enables (1) the MPA mode to be enabled to resist the packet forwarding attacks, and (2) the SPA mode to be used if network conditions change and the use of the MPA mode is counter-productive in terms of supporting the QoS requirements, e.g., when the packet flow is no longer suffering from blackhole attacks. When using the 2-dimensional approach, the dynamic adaptation algorithm (Algorithm 5.4 in Section 5.5.5.2) is therefore still used to select the SPA mode or the MPA mode dynamically in response to the changes in network conditions.

In addition to selecting the mode of adaptation, it is also necessary to determine whether DSR's packet salvaging mechanism should be used to support data packet forwarding. As was observed in Sections 5.6.3.2 and 6.2.1.2, packet salvaging can support QoS under certain conditions and it can be a hindrance to it in other conditions. For example, using packet salvaging is beneficial for QoS when the network is lightly loaded. This is because the available bandwidth can be used effectively for data packet retransmission attempts. However, when the network is more heavily loaded, using packet salvaging has a detrimental effect on QoS. This is because the lack of bandwidth available for data packet retransmissions leads to the retransmissions exacerbating the existing congestion. It is therefore advantageous to use packet salvaging only when it will support QoS.

To address this, the concept of Adaptive Packet Salvaging (APS) is proposed. APS is the process of dynamically enabling or disabling DSR's packet salvaging mechanism in response to changes in network conditions. This enables the packet salvaging mechanism to be used adaptively to support QoS. With APS, nodes are requested to enable the packet salvaging mechanism when its use is likely to support QoS, and to disable it when it is likely to worsen QoS. The DSR protocol specifies that a node either salvages all data packets or none of them for the duration of a session [88]. Thus using APS requires minor modifications to be made to the DSR protocol.

A potential issue with the APS approach is that intermediate nodes cannot be trusted to execute this additional operation, but requesting that they disable packet salvaging during periods of network congestion has three benefits which may encourage them to co-operate. First, it frees the time they would otherwise spend trying to capture the wireless medium to retransmit other nodes' data packets. In the worst case, an intermediate node may retransmit each received data packet up to the maximum number of retries allowed by the MAC protocol's data packet retry limit. With IEEE 802.11 (which is used in this research) this is determined by the *LongRetryLimit* parameter which has a value of 4, and is used when RTS/CTS is enabled [77]. If the packet remains undelivered after this time the node attempts to salvage it: the node attempts to transmit the packet to a different next-hop node, and if it is not delivered after the first attempt it will retransmit it up to the maximum number of retries allowed by the *LongRetryLimit*. The second benefit of using APS is the corollary of the first: intermediate nodes' energy consumption may be reduced. This is because fewer data packet retransmissions are required. Finally, network congestion will be reduced. This is, again, due to fewer data packet retransmissions. This also may result in a better QoS for the intermediate nodes' packet flows, as lower congestion may lead to lower end-to-end delays and fewer lost packets.

There is one main benefit which may encourage intermediate nodes to enable packet salvaging when the network load is low and there is little congestion. Enabling packet salvaging in these conditions may lead to a greater percentage of data packet deliveries. In the absence of congestion, spending time attempting to retransmit a packet to a different next-hop node when the initial packet transmission fails may lead to better QoS. Both priority and non-priority data packets are salvaged as the packet salvaging process does not discriminate between the

two packet types.

6.3 Design Requirement and Principle

The CAT detection mechanism has a design requirement, which is a performance requirement, and takes a design measure which are in addition to those presented in Section 5.4.

The additional performance requirement is as follows.

- **(P1)** To minimize the number of additional control packets injected into the underlying network. This aims to ensure that control packets do not exacerbate congestion in an already congested network.

To satisfy requirement (P1), the following measure is taken in the design of the CAT detection mechanism.

- **Measure 1:** The CAT detection mechanism uses two types of control packet already employed in 2-DAARC to infer the reason for packet loss. These two control packets are the Route Error control packet, which is transmitted as part of DSR's Route Maintenance process, and the 2-DAARC feedback packet, which contains QoS statistics from the destination node. In other words, the CAT detection mechanism does not introduce additional control packets into the network.

6.4 CAT Detection and Adaptation in Detail

The E2-DAARC architecture, which includes the CAT detection and adaptation mechanisms, is shown in Fig. 6.5. Two new components are added into the architecture (the original architecture is shown in Fig. 5.1): the *Route Error Handler* component and the *Adaptive Packet Salvaging* component. They are both located on the network layer.

- The *Route Error Handler* component is located in the source node entity. This component has three functions. First, it determines the number of RERR packets received over a time interval. This is termed the *RERR rate*. Second, it infers the likely reason for packet loss by combining the RERR

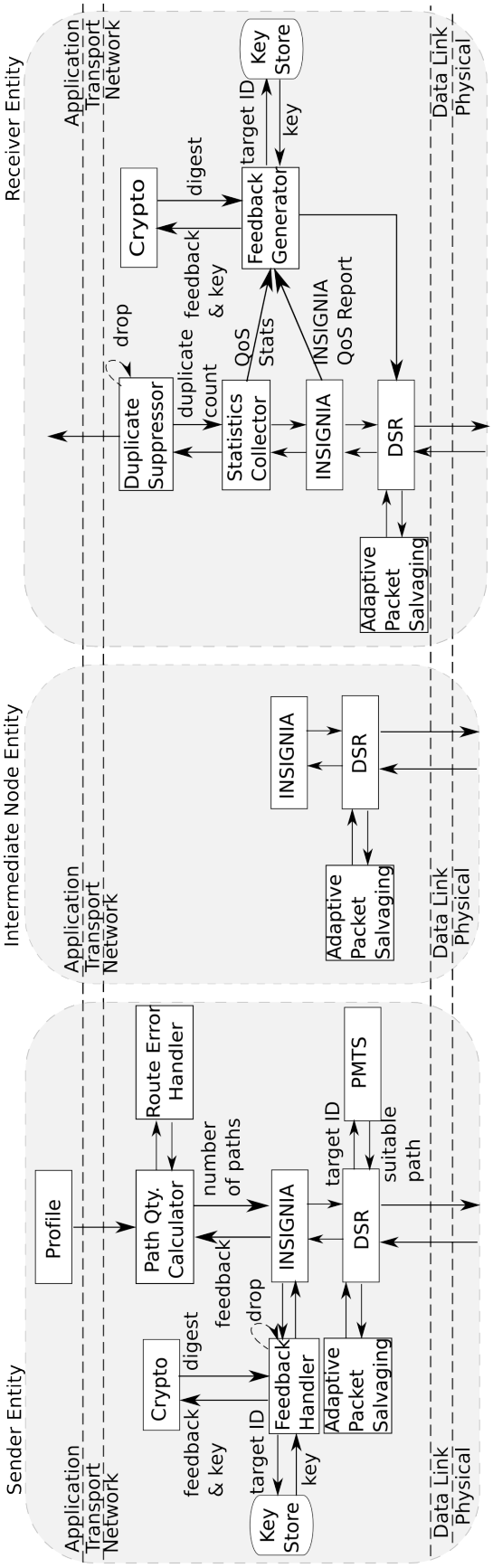


Figure 6.5: The Extended 2-Dimensional Adaptation ARChitecture (E2-DAARC)

rate with the packet loss statistics received in a feedback packet. Third, based on the packet loss reason, it (1) informs the Path Quantity Calculator component whether the SPA-ONLY mode or 2-dimensional adaptation should be used; and (2) it informs the Adaptive Packet Salvaging component whether to mark outgoing data packets to request packet salvaging at intermediate nodes.

- The *Adaptive Packet Salvaging* (APS) component is located in the source, intermediate, and destination node entities. The source node's APS component marks outgoing data packets either to request or not to request packet salvaging at intermediate nodes. It also enables or disables the packet salvaging option of the source node's DSR component to match the marking of the packets. For example, if the packets should not be salvaged, the source node will disable its salvaging option, and vice versa. The purpose of using adaptive packet salvaging is to use salvaging when the network is lightly loaded and to disable salvaging when the network is congested. This measure is also taken at the intermediate and destination nodes. If an intermediate node is unable to forward a data packet, its APS component inspects the salvaging flag carried in the packet to determine whether or not to salvage it.

The following sections describe E2-DAARC's CAT detection mechanism and the adaptation actions used to respond to the inferred cause of packet loss.

6.4.1 Detection

The CAT detection mechanism uses the RERR rate and the PLR to determine the likely cause of packet loss, and it works as follows. When an intermediate node cannot deliver a data packet to the next-hop node it transmits a RERR control packet to the source node. The source node increments a counter, ϵ, for each RERR packet received during the time-interval τ (this is the same 5 second time interval for the feedback mechanism, as specified in Section 5.5.4.2). At the end of each interval τ, the source node uses the ϵ counter value, which is the RERR rate, and the PLR from the feedback packet to infer the likely cause of packet loss. To infer the cause of loss it is necessary to use three sets of mappings between the RERR rate and the PLR.

		Packet Loss Ratio		
		Low	**Medium**	**High**
RERR Rate	**Low**	PLR $< \rho$ && RERR $< \epsilon$	$\rho \leq$ PLR $< 2\rho$ && RERR $< \epsilon$	PLR $\geq 2\rho$ && RERR $< \epsilon$
	Medium	PLR $< \rho$ && $\epsilon \leq$ RERR $< 2\epsilon$	$\rho \leq$ PLR $< 2\rho$ && $\epsilon \leq$ RERR $< 2\epsilon$	PLR $\geq 2\rho$ && $\epsilon \leq$ RERR $< 2\epsilon$
	High	PLR $< \rho$ && RERR $\geq 2\epsilon$	$\rho \leq$ PLR $< 2\rho$ && RERR $\geq 2\epsilon$	PLR $\geq 2\rho$ && RERR $\geq 2\epsilon$

Table 6.1: Mappings Between the RERR Rate and the PLR, where PLR = Packet Loss Ratio and RERR = Route Error

The mappings between the RERR rate and the PLR are established in three stages. In the first stage, the mappings for the combinations of RERR rate and PLR are expressed. In the second stage, the combinations of RERR rate and PLR from the first stage are mapped on to the likely reasons for packet loss. In the third stage, the adaptive actions to be used in response to the inferred cause of packet loss are outlined.

The first stage is to establish the mappings between the RERR rate and the PLR to determine the likely cause of packet loss. These mappings are expressed in a 3×3 matrix. This is shown in Table 6.1. This matrix maps the low, medium, and high RERR rate with the low, medium, and high PLR. The mappings result in nine combinations of RERR rate and PLR. Two adaptation parameters are used to determine the PLR and the RERR rate: ρ and ϵ. The value of ρ, the packet loss adaptation parameter, is 0.10 (as determined in Section 5.6.1.1). The value of ϵ, the RERR rate adaptation parameter, is determined using simulation in Section 6.5.1.

The second stage is to map the combinations of RERR rate and PLR expressed in Table 6.1 on to the likely reasons for packet loss. Table 6.2 shows these packet loss reasons. In the table, C refers to congestion, A refers to attacks, and M refers to node mobility. The following examples illustrate these three packet loss reasons and explains how they are derived from the RERR rate and the PLR.

- Attacks are likely to be the main cause of packet loss when the RERR rate is low and the PLR is either medium or high. Attacks do not lead to the transmission of RERR packets: a blackhole attacker does not notify the source node with RERR packets when it drops data packets. If the loss were a consequence of congestion there would be a higher rate of RERR

packets. This is because RERR packets would be transmitted in response to each failed packet delivery.

- Mobility is likely to be the main cause of packet loss when the PLR is low and the RERR rate is low, medium, or high. The low PLR suggests that attacks and congestion are not affecting data packet deliveries. Mobility-induced packet loss is low in a lightly loaded network: it was shown in Section 5.6.3.1 that the PDR is maintained above 95%, i.e., the PLR is less than 5%, when all nodes are constantly in motion; and this 5% PLR is within the low level of PLR in Table 6.1 (where the PLR is less than 10%, i.e., $PLR < \rho$). Thus when the PLR is low, mobility-induced path breaks are the main reason for RERR packet transmissions.

- Congestion and/or attacks are likely to be the main cause(s) of packet loss for the following four reasons: (1) when both the RERR rate and the PLR are medium; (2) when both the RERR rate and the PLR are high; (3) when the RERR rate is medium and the PLR is high; and (4) when the RERR rate is high and the PLR is medium. The medium-to-high RERR rate combined with the medium-to-high packet loss indicates congestion. This is because data packets are being dropped and RERR packets are being transmitted in response to their failed deliveries. This RERR rate is unlikely to be a consequence of node mobility, because the PLR is at least the medium level: this means that the PLR is greater than 10% ($\rho < PLR \leq 2\rho$), and packet loss due to mobility, as explained above, is observed to be less than 10%. The medium-to-high PLR may also be a consequence of attacks (in addition to congestion). However, it is not necessary to differentiate between congestion and attacks as the reason for packet loss when both the RERR rate and the PLR are at least at a medium level: the same adaptation response is used for the four loss reasons, as explained below.

The third stage is to specify the adaptive actions to be used in response to the inferred reason for packet loss. Table 6.3 shows these adaptive actions. In the table, *2-D* refers to the use of dynamic, 2-dimensional adaptation between the SPA mode and the MPA mode, *SPA-ONLY* refers to the use of the SPA mode only, i.e., the MPA mode will not be used, and *SALV ON/SALV OFF* refers to whether packet salvaging should be enabled or disabled. The *2-D/SALV ON* response is used when the main reason for packet loss is likely to be either

		Packet Loss Ratio		
		Low	**Medium**	**High**
RERR Rate	**Low**	M	A	A
	Medium	M	C, A	C, A
	High	M	C, A	C, A

Table 6.2: The Likely Causes of Packet Loss, where C = Congestion, A = Attack, and M = Mobility

		Packet Loss Ratio		
		Low	**Medium**	**High**
Route Errors	**Low**	2-D	2-D	2-D
		SALV ON	SALV ON	SALV ON
	Medium	2-D	SPA-ONLY	SPA-ONLY
		SALV ON	SALV OFF	SALV OFF
	High	2-D	SPA-ONLY	SPA-ONLY
		SALV ON	SALV OFF	SALV OFF

Table 6.3: Adaptive Actions to be Used for the Different Causes of Packet Loss, where *2-D* = dynamically adapt between the SPA and the MPA modes, *SPA-ONLY* = use the SPA mode even if the adaptation threshold triggering the MPA mode is exceeded, *SALV* = packet salvaging on or off.

node mobility or attacks, but not congestion. The justifications for this are two-fold. First, the 2-dimensional approach may use the MPA mode, and this is likely to support QoS, rather than cause congestion, as the network load is likely to be light. Second, using packet salvaging in a lightly loaded network may help to support data packet deliveries. This combination of *2-D/SALV ON* is used because the spare bandwidth in the lightly loaded network can be utilized (1) for duplicated data packet transmissions using the MPA mode, and (2) to perform packet salvaging on the data packets which fail to be delivered. The *SPA-ONLY/SALV OFF* response is used when congestion and/or attacks is the likely reason for packet loss. The justifications for this are two-fold: (1) congestion is likely to be at least one cause of packet loss, thus the SPA-ONLY mode should be used to avoid exacerbating any existing congestion; and (2) packet salvaging should be disabled because retransmitting data packets in a congested network leads to increased congestion and a deterioration in QoS.

Before presenting the CAT detection mechanism in detail, the four variables and two methods it uses are first explained. The four variables are the *PLR*, the *RERR rate*, the *previous value of PLR*, and the *previous value of RERR rate*.

The PLR and the previous value of PLR are compared to determine the change in PLR between two time intervals: the most recent time-interval and the time-interval immediately prior. The RERR rate and previous value of RERR rate are used for the same purpose.

The two methods are *setModeTwoDimensional()* and *setModeSPA()*. The setModeTwoDimensional() method is used to enable (1) dynamic, 2-dimensional adaptation between the SPA and the MPA modes and (2) packet salvaging. Pseudocode for this method is given in Algorithm 6.1. The method first checks whether the current adaptation mode is SPA-ONLY. If not, the 2-dimensional approach is already in-use, so there is no need to execute the remainder of the method. Otherwise, the method determines whether the PLR and the RERR rate are less than or the equal to the previous values of PLR and RERR rate. This is to ensure that 2-dimensional adaptation is activated only if the congestion has not worsened since the PLR and the RERR were measured during the previous time-interval. The method then enables 2-dimensional adaptation and packet salvaging. The SPA mode is set as the mode of adaptation to keep to a minimum the number of data packets initially injected into the network (the MPA mode can be enabled during dynamic adaptation if it is required to support QoS). Packet salvaging is enabled by setting the value of the *dsragent_salvage_with_cache* variable to 'true' (the use of the *dsragent_salvage_with_cache* variable is described in detail in Section 6.4.2). The setModeSPA() method is given in Algorithm 6.2. It first checks whether the current mode of adaptation is the SPA-ONLY mode. This is so that the remainder of the method is not unnecessarily executed if the SPA-ONLY mode is currently in use. If the SPA-ONLY mode is not currently in use, the adaptation mode is set to SPA-ONLY and packet salvaging is disabled.

The pseudocode for the CAT detection mechanism is given in Algorithm 6.3. It works as follows. CAT detection is performed by a source node at the end of each time-interval τ. If the PLR is less than ρ, i.e., the packet loss is low, the setModeTwoDimensional() method is called, regardless of the value of ϵ. If the PLR is greater than or equal to ρ, i.e., the PLR is medium or high, and the RERR rate is less than ϵ, i.e., the RERR rate is low, the setModeTwoDimensional() method is called; otherwise the setModeSPA() method is called as the RERR is greater than ϵ, i.e., the RERR rate is either medium or high.

Algorithm 6.1: Pseudocode for the setModeTwoDimensional Method

```
if adaptationMode == SPA-ONLY then
    if RERR ≤ previousRERR and PLR ≤ previousPLR then
        setAdaptationMode(SPA);
        dsragent_salvage_with_cache = TRUE;
```

Algorithm 6.2: Pseudocode for the setModeSPA Method

```
if adaptationMode ≠ SPA-ONLY then
    setAdaptationMode(SPA-ONLY);
    dsragent_salvage_with_cache = FALSE;
```

Algorithm 6.3: Pseudocode for the CAT Detection Mechanism

```
if PLR < ρ then
    setModeTwoDimensional();
else if PLR ≥ ρ then                    // i.e., ρ ≤ PLR < 2ρ or PLR ≥ 2ρ
    if RERR < ϵ then
        setModeTwoDimensional();
    else
        setModeSPA() ;                  // as ϵ ≤ RERR < 2ϵ or RERR ≥ 2ϵ
prevPLR = PLR;
prevRERR = RERR;
```

6.4.2 Response

Adaptively enabling and disabling the packet salvaging mechanism of E2-DAARC's underlying routing protocol (DSR) is another action taken to respond to fluctuations in packet loss. The APS mechanism enables or disables the packet salvaging mechanism at the source, intermediate, and destination nodes participating in a path carrying priority data packets. Performing APS at nodes other than the source node requires parameter values to be transmitted from the source node to the other nodes on the path.

The 2-DAARC IP Options header is extended in E2-DAARC to accommodate the two new options used by the APS mechanism. The E2-DAARC IP Options header is shown in Fig. 6.6 (the original 2-DAARC IP Options header is shown in Fig. 5.5). These two options specify whether or not packet salvaging should occur at the nodes receiving priority data packets. The new options are the *packet*

Option	Service Mode	Payload Type	Bandwidth Indicator	Bandwidth Request	Generation Rate	Packet Type	Packet Status	Packet Priority	Salvaging	Padding
Value	RES/ BE	BQ/ EQ	MAX/MIN	MAX/MIN	Packet Gener- ation Rate	SPA/ MPA	OR/ DUP	Priority/ Non- Priority	ON/OFF	-
Length	1 bit	1 bit	1 bit	16 bits	8 bits	1 bit	1 bit	1 bit	1 bit	1 bit

Figure 6.6: The E2-DAARC IP Options Header

priority and the *salvaging* options. The *packet priority* option specifies whether a data packet belongs to a priority or a non-priority packet flow. This option is necessary as it is not always possible to determine from the existing options whether or not a data packet belongs to a priority packet flow. For example, a packet with the *service mode* set to 'BE' could either be a *degraded* priority packet, which is a best-effort packet belonging to a priority flow, or a best-effort background packet. Only priority data packets participate in the APS process, i.e., only the values carried in the headers of priority data packets are used to change whether or not an intermediate node performs packet salvaging. Thus each priority data packet originated from a source node has its *packet priority* option marked to reflect its priority (the default value of a packet is that it is a non-priority packet).

The *salvaging* option is the second option used by the APS mechanism. This option specifies whether the nodes receiving the packet should attempt to salvage it if it cannot be delivered to the next-hop node. An intermediate node along a path first checks if a received data packet is a priority packet (using the *packet priority* option). If so, it checks the salvaging option. It will then enable or disable its packet salvaging mechanism as specified by the salvaging option, i.e., if the salvaging option is set, the node will enable its salvaging operation, and vice versa. The destination node follows a similar process on receipt of the packet. However, as it is the intended final recipient of the packet it will not attempt to salvage it. The reason for adjusting the packet salvaging mechanism at the destination node is explained below.

In detail, APS has two phases. The first phase occurs at the source node. The CAT detection mechanism specifies whether the packet salvaging mechanism should be enabled or disabled: if the setModeSPA() method is called, packet salvaging will be disabled; if the setModeTwoDimensional() method is called, packet salvaging will be enabled. If it is to be disabled, the source node sets the packet salvaging mechanism (DSR's *dsragent_salvage_with_cache* variable) to 'false'. Contrariwise, this variable is set to 'true' if packet salvaging should be enabled. The source node marks each outbound priority data packet with the current salvaging option: if *dsragent_salvage_with_cache* is set to 'false' (packet salvaging is disabled), the source node marks the packet's *salvaging* option to 'OFF'; if *dsragent_salvage_with_cache* is set to 'true' (packet salvaging enabled), the source node sets the *salvaging* option to 'ON'.

The second phase of APS occurs at the intermediate and destination nodes. On receiving a priority data packet, these nodes set their packet salvaging mechanism variable (the *dsragent_salvage_with_cache* variable) to match the *salvaging* option in the received packet's header. In other words, they set the packet salvaging mechanism to 'false' if the salvaging option is set to 'OFF', and set it to 'true' it the salvaging option is set to 'ON'. This aims to make all nodes along the path use the same salvaging option.

The source and destination nodes do not perform packet salvaging on the packets of which they are the origin or destination, but they may salvage other nodes' data packets when they serve as intermediate nodes for other nodes' packet flows. For example, if the source and/or destination nodes for packet flow A are serving as intermediate nodes for packet flow B, and they are asked to forward priority packets, the source/destination nodes of flow A will enable or disable their packet salvaging mechanism as indicated by the salvaging option in the packets received from flow B. Setting the packet salvaging mechanism to the value in the packets received from flow B may be the opposite of what is required by the priority packets originated by the source node of flow A: for example, the packets from flow A may want packet salvaging disabled whereas the packets from flow B may want packet salvaging enabled. To rectify this, the source node of flow A will set the packet salvaging mechanism to the appropriate value when it next originates a priority data packet. The value of the packet salvaging option may therefore change frequently—as the source node forwards other nodes' data packets and originates its own data packets—to best serve the needs of each priority data packet flow.

6.5 E2-DAARC Performance Evaluation

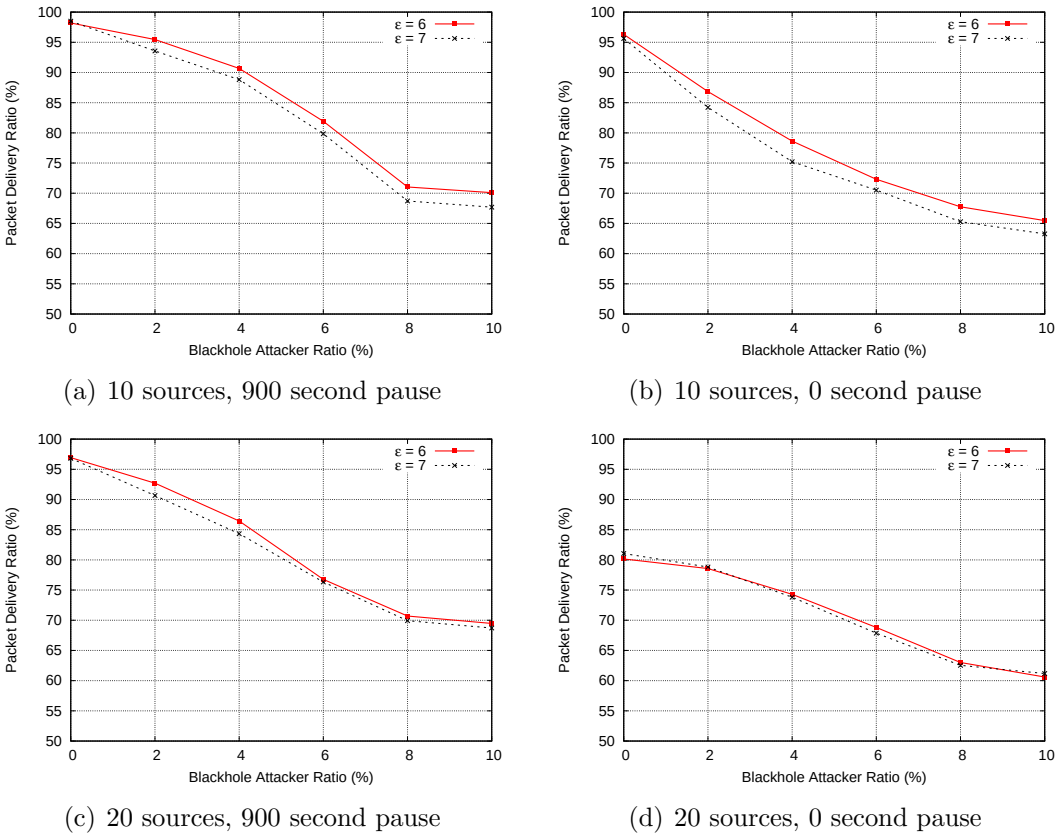
The effectiveness of the CAT detection mechanism is investigated using a simulation study. The study is performed in two phases. In the first phase, presented in Section 6.5.1, the value for ϵ is determined. In the second phase, presented in Section 6.5.2, E2-DAARC, i.e., 2-DAARC extended with the CAT detection and APS mechanisms, is evaluated against related work.

6.5.1 Determining ϵ, the Route Error Adaptation Parameter

Before evaluating the CAT detection mechanism, it is first necessary to determine a value of ϵ which leads to the highest packet delivery with the least control packet cost. The simulation results presented in this section investigate how the PDR and normalised routing load are affected by different values of ϵ. This is necessary because CAT detection is sensitive to the value of ϵ. For example, if the value of ϵ is set too low, CAT detection may incorrectly determine that the network is congested. This will lead to the use of the SPA-ONLY mode with packet salvaging disabled when the network is not congested. In other words, adaptation will occur too early. In turn, this may lead to a reduction in QoS, as the available bandwidth will not be fully utilized by the MPA mode and packet salvaging to support data packet deliveries. Conversely, if the value of ϵ is set too high, the network may be heavily congested and the CAT detection mechanism will not react to the congested conditions. Consequently, QoS may be detrimentally affected by E2-DAARC continuing to use the MPA mode and packet salvaging for longer than is suitable in the congested network. In other words, the adaptation may occur too late. A range of values of ϵ are investigated under different network conditions to determine the best value of ϵ.

The following describes the simulation configuration used in this investigation. Simulations are conducted in a network containing 10 and 20 sources nodes, i.e., 3 and 6 priority source nodes with 7 and 13 best-effort, background sources, respectively. These provide low and high network load conditions. Two pause times are used, 900 seconds (static nodes) and 0 seconds (constant node mobility). The number of blackhole attackers ranges from 0–10% in 2% increments. PMTS is used as the MPA mode's secondary path selection mechanism. 10 different values of ϵ are investigated, ranging from $\epsilon = 1$ to $\epsilon = 10$ in increments of 1. To focus the discussion, results are presented for the two best performing values of ϵ, where $\epsilon = 6$ and $\epsilon = 7$.

The main observation from the results is that $\epsilon = 6$ generally leads to PDRs which are better than or similar to those when $\epsilon = 7$ is used, and it generally does so with lower normalised routing loads. From the results in Fig. 6.7 for 10 source nodes, it can be seen that the PDR for $\epsilon = 6$ is greater than for $\epsilon = 7$ at all non-zero attacker ratios with both the 900 second (Fig. 6.7(a)) and 0 second (Fig. 6.7(b)) pause times. The higher PDRs for $\epsilon = 6$ are due to adaptive



(a) 10 sources, 900 second pause

(b) 10 sources, 0 second pause

(c) 20 sources, 900 second pause

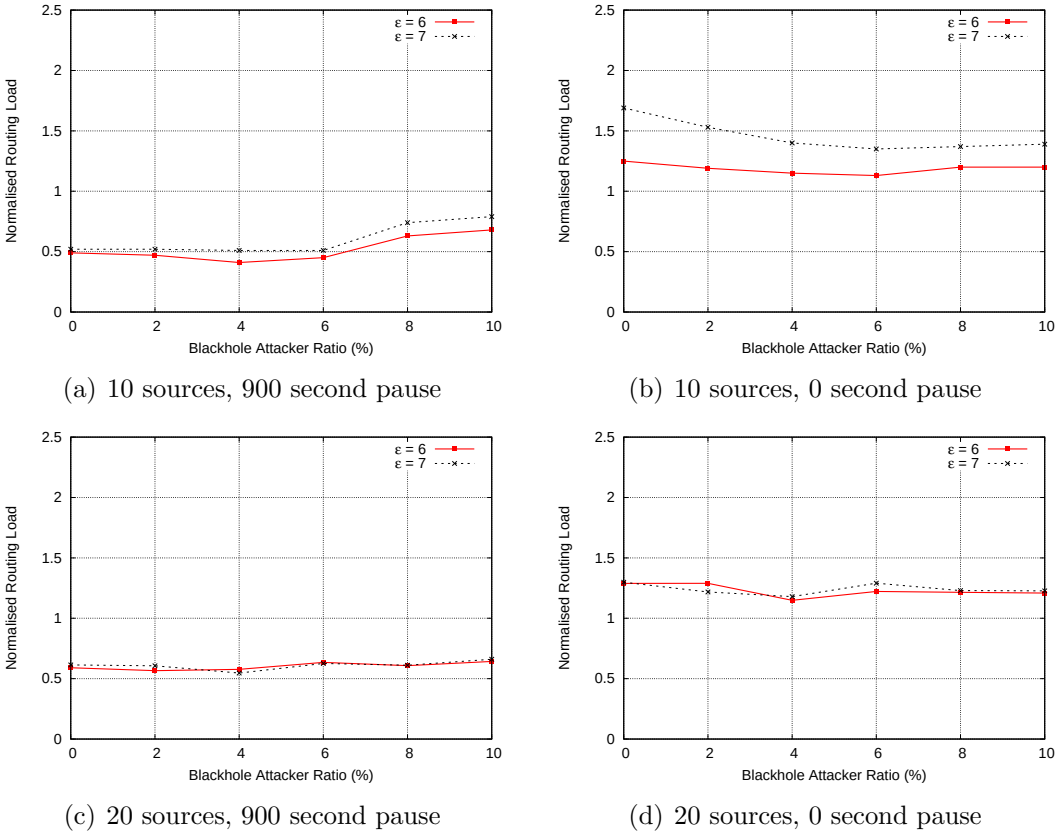
(d) 20 sources, 0 second pause

Figure 6.7: Determining a Value of ϵ for CAT Detection: PDR for 0 and 900 Second Pause Times

actions being taken one RERR packet sooner than with $\epsilon = 7$. Although the difference in the value of ϵ is only one RERR packet, performing adaptive actions sooner means that E2-DAARC can respond more quickly to changes in network conditions. In other words, using a higher value of ϵ means that conditions are allowed to deteriorate further before the adaptive actions are taken, and this leads to lower PDRs. The RERR rate generally exceeds ϵ more frequently when $\epsilon = 7$ than when $\epsilon = 6$. For example, at the 6% attacker ratio, the RERR rate exceeds ϵ twice as often when $\epsilon = 7$ than it does when $\epsilon = 6$. Moreover, adapting later, i.e., after a larger value of ϵ, leads to poor network conditions persisting for longer, and this leads to the RERR rate exceeding 2ϵ more frequently. For example, at the 6% attacker ratio, 2ϵ is exceeded approximately 1.75 times more often when $\epsilon = 7$ than when $\epsilon = 6$. Allowing poor conditions to persist for longer leads to lower PDRs. Thus adapting when $\epsilon = 6$ offers greater PDRs in more lightly loaded networks.

The earlier adaptation triggered by using $\epsilon = 6$ also has a positive effect on the normalised routing load. As can be seen in Fig. 6.8(a) and Fig. 6.8(b), the normalised routing load of $\epsilon = 6$ is clearly lower than for $\epsilon = 7$. The higher normalised routing load for $\epsilon = 7$ is due to (1) source nodes having to receive a larger number of RERR packets before adaptive actions are performed, and (2) poor network conditions persisting for longer as a consequence of the later adaptation. The normalised routing load increases between the 6%–10% attacker ratios for both values of ϵ. This occurs because the number of control packet transmissions increases whist the number of received data packets decreases. This is the opposite of the trend observed for attacker ratios of 0%–6%: the number of control packets decreases as the attacker ratio increases, thus maintaining a relatively constant normalised routing load. At the 8% attacker ratio, approximately 7% more control packets are transmitted than at the 6% attacker ratio. This arises from an increase in the number of Route Request packet transmissions. More Route Request packets are transmitted by source nodes to find new paths to the destination nodes. At the 10% attacker ratio there are approximately 2.5% more control packets transmitted than at the 6% attacker ratio. Whilst this is a smaller increase than at the 8% attacker ratio, this increase in control packets is coupled with a greater percentage of data packets failing to be delivered, and this results in an increased normalised routing load.

When nodes are mobile (Fig.6.8(b)), there is a larger difference between the normalised routing loads of $\epsilon = 6$ and $\epsilon = 7$ compared with the static case (Fig. 6.8(a)). Introducing node mobility makes the above described issues of adapting later (i.e., when using $\epsilon = 7$) more pronounced. This is because adapting later means that more path breaks occur as a consequence of mobility-related issues, such as the congestion resulting from the reduced effective bandwidth. The normalised routing load for $\epsilon = 7$ decreases up to the 6% attacker ratio. This is explained as follows. An increasing number of data packets are dropped as the attacker ratio increases, and this frees resources (bandwidth and buffer space) at intermediate nodes. Consequently, more paths can be learnt during Route Discovery operations: Route Request packets make up approximately 30% of the control packet transmissions of the underlying routing protocol (DSR) at all attacker ratios, whereas the percentage of Route Reply packet transmissions increases from 27% at the 0% attacker ratio to 33% at the 10% attacker ratio (Route Error packets make up the remainder of the routing protocol's



(a) 10 sources, 900 second pause

(b) 10 sources, 0 second pause

(c) 20 sources, 900 second pause

(d) 20 sources, 0 second pause

Figure 6.8: Determining a Value of ϵ for CAT Detection: Normalised Routing Load for 0 and 900 Second Pause Times

control traffic). Source nodes are therefore discovering paths more effectively by transmitting fewer control packets as attackers reduce the number of data packets in the network. The normalised routing loads for $\epsilon = 6$ are more consistent at all attacker ratios due to adaptation being performed sooner in response to the dynamic network conditions.

When the network load is increased to 20 source nodes (Fig. 6.7(c) and Fig. 6.7(d)), $\epsilon = 6$ generally supports marginally higher PDRs than $\epsilon = 7$. The difference between the curves has reduced compared with the 10 source node case. This is because the load from 20 source nodes reduces the availability of resources in the network, and this leaves fewer spare resources to adapt to. Thus $\epsilon = 6$ and $\epsilon = 7$ have a more similar outcome in terms of PDR than under the lower network load (10 sources). With the 0 second pause time (Fig. 6.7(d)), the PDR when $\epsilon = 6$ is only marginally higher on average than the PDR when $\epsilon = 7$. This is because the combination of continual node mobility and high network

load further erodes the benefits of adapting one RERR packet sooner (i.e., when $\epsilon = 6$ rather than $\epsilon = 7$).

The normalised routing loads under this network load are similar for both values of ϵ under all attacker ratios at both the 900 second pause time (Fig. 6.8(c)) and the 0 second pause time (Fig. 6.8(d)). The similarity of the normalised routing loads echoes the similarities exhibited by the PDRs with this network load. Like the PDRs, the combination of node mobility and high network load leads to little difference in performance for the different values of ϵ. Based on the above results, a value of $\epsilon = 6$ is to be used for the CAT detection mechanism, as it generally leads to higher PDRs and lower normalised routing loads than $\epsilon = 7$.

The value of $\epsilon = 6$ is significantly different from the initial value of ϵ observed in the preliminary simulation study presented in Section 6.2.1.2. It was observed in Section 6.2.1.2 that $\epsilon = 100$. In that simulation study, the RERR rate was determined from the cumulative number of RERR packets received by all priority source nodes in a network. The high RERR rate was due to (1) RERR packets being transmitted in response to continuing congestion and (2) no new adaptive actions being performed to reduce the congestion. In the simulation study presented in this section, the number of RERR packets is lower for two reasons. First, this simulation study looks at the number of RERR packets received by individual priority source nodes. In other words, the number of RERR packets is investigated on a per-node basis, rather than the cumulative number of RERR packets received by all priority source nodes. Second, each priority source node performs CAT detection and adaptation in response to the determined cause of packet loss. This reduces congestion which, in turn, reduces the number of RERR packet transmissions and the number of RERR packets received by each priority source node.

6.5.2 Simulation Results

The results of this simulation study are presented in two sections. The first section focuses on evaluating the effectiveness of E2-DAARC in adaptation to network congestion. The second section focuses on its effectiveness in adaptation to both congestion and attacks. In both sections, PMTS is used as the secondary path selection mechanism in the MPA mode. The results in both of these sections are compared with the results for INSIGNIA and for 2-DAARC without CAT

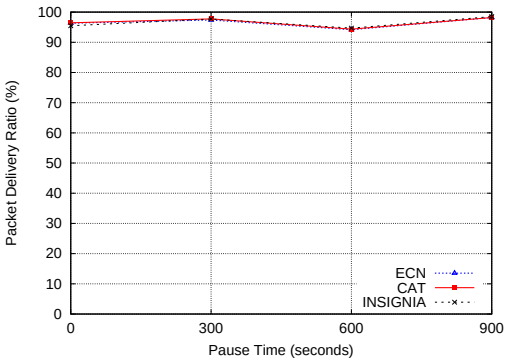
detection, i.e., the 2-DAARC configuration from Chapter 5 and its accompanying performance analysis presented in Section 5.6.3.2. Comparing E2-DAARC with 2-DAARC is done to demonstrate the merits and limitations of using CAT detection.

6.5.2.1 Adapting to Network Congestion

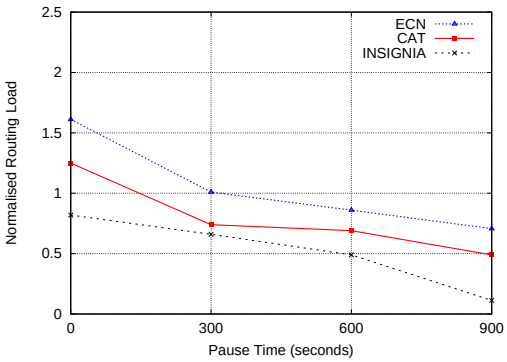
In this section, the performance of E2-DAARC is evaluated under low and high network loads and 0, 300, 600, and 900 second pause times. The performance of E2-DAARC's CAT detection mechanism, hereafter referred to as CAT detection, is compared with a version of 2-DAARC using the ECN congestion detection mechanism, hereafter referred to as ECN (integrating ECN into 2-DAARC is described in Section 3.5.2.6). In this investigation, no attackers are assumed in the network, so the Watchdog mechanism is not used. Both 2-DAARC approaches (CAT detection and ECN) use adaptive packet salvaging. As INSIGNIA does not use adaptive packet salvaging, the INSIGNIA curves in Fig. 6.9 are for packet salvaging enabled, and the INSIGNIA curves in Fig. 6.10 are for packet salvaging disabled. These results are the best results achievable by INSIGNIA under the given network conditions.

With a low network load (10 sources), CAT detection achieves similar PDRs to ECN and INSIGNIA, but it does so with lower normalised routing loads than ECN. As can be seen in Fig. 6.9(a), the PDRs are generally similar for all three curves at all pause times. The PDRs are maintained above 94%. Similar to the observation in Section 5.6.3.1, 2-DAARC does not bring benefits over INSIGNIA in terms of PDR in an attacker-free, lightly loaded network. This is because there is little congestion in the lightly loaded network for both CAT detection and ECN to adapt to.

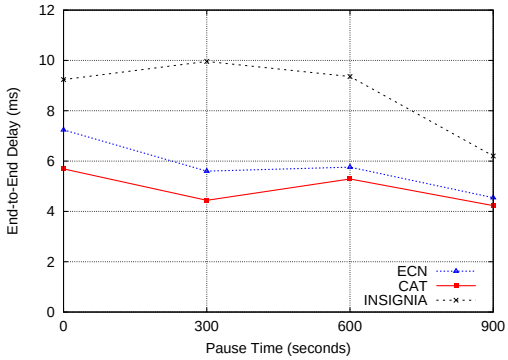
Whilst the PDRs are similar for all three approaches, CAT detection achieves these PDRs with lower normalised routing loads than ECN. The normalised routing loads for all three approaches are shown in Fig. 6.9(b). All three curves increase as the pause time moves towards 0 seconds. This is because more control packets are required to support data packet deliveries when there are more frequent paths breaks due to the increased level of node mobility. The greatest difference between the normalised routing loads of CAT detection and ECN is at the 0 second pause time. The higher normalised routing load for ECN is



(a) Packet Delivery Ratio



(b) Normalised Routing Load



(c) End-to-End Delay

Figure 6.9: Comparing the PDR, Normalised Routing Load, and End-to-End Delay of CAT Detection with ECN and INSIGNIA for 3 Priority Sources (7 Best-Effort, Background Sources) for 0, 300, 600, and 900 Second Pause Times in an Attacker-free Network.

due to two related factors: ECN control packets and node mobility. An intermediate node transmits an initial ECN control packet to a source node when its data packet queue occupancy exceeds a threshold value (the value is 60%, as specified in Section 3.5.2.6). All subsequent ECN notifications are transmitted by the destination node as part of 2-DAARC's periodic feedback (as explained in Section 3.5.2.6), rather than intermediate nodes transmitting them each time their packet queue occupancy exceeds the threshold value; this is to adhere to the ECN specification. Node mobility leads to paths breaks, and this leads to different nodes becoming intermediate nodes. These new intermediate nodes will also transmit an ECN control packet when their data packet queue occupancy initially exceeds ECN's threshold value. Thus ECN control packets and node mobility lead to ECN's higher normalised routing load at the 0 second pause time. ECN's normalised routing load is lower at non-zero second pause times

as less mobility means fewer path breaks and therefore more long-lived paths; and this leads to fewer intermediate nodes transmitting their initial ECN control packets. For example, ECN control packets account for approximately 20% of all of the control packets transmitted by ECN at the 0 second pause time compared with 12% at the 300 second pause time. CAT detection's normalised routing load is generally lower than ECN's as it utilizes the RERR control packets which are already transmitted by the underlying routing protocol to determine whether congestion is likely to be occurring. In other words, CAT detection does not inject additional control packets into the underlying network. The normalised routing loads of CAT detection and ECN are greater than INSIGNIA's. This is a consequence of 2-DAARC's periodic feedback packet transmissions.

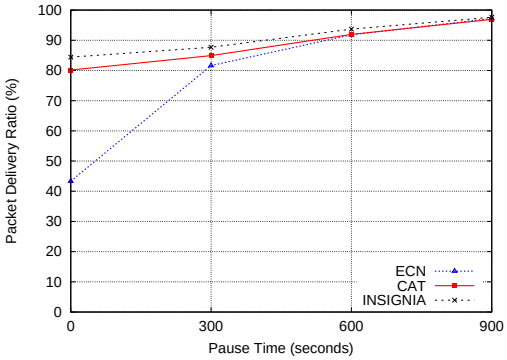
ECN's higher normalised routing loads lead to longer end-to-end data packet delays compared with CAT detection and INSIGNIA. This is shown in Fig. 6.9(c), and is explained as follows. The priority queuing mechanism (PriQueue) of the underlying routing protocol (DSR) transmits control packets before it transmits data packets. This is because the queue for control packets has a higher priority than the queue for data packets (as can be seen in Fig. 3.3). This leads to a higher data packet queue occupancy, as data packets have to wait for the control packets to be transmitted. For example, at the 0 second pause time, the data packet queue for ECN exceeds 50% occupancy approximately 4 times more often than for CAT detection. The delays of both CAT detection and ECN are lower than those of INSIGNIA. This is expected, as using multiple paths in lightly loaded networks leads to lower delays than using only a single path (as shown in Section 5.6.3.2). Additionally, when the PDR and delay results are considered together, it can be seen that, under the given network conditions, CAT detection is superior to ECN and INSIGNIA as it achieves the same PDRs but with lower end-to-end delays.

When the network load is increased to that of 20 source nodes, and all other parameter values remain unchanged, the PDRs for all three schemes decrease as the pause time approaches 0 seconds, with the PDRs of ECN decreasing the most. As can be seen in Fig 6.10(a), the decreases in PDR for CAT detection and INSIGNIA are relatively small. In a stationary network, both schemes have PDRs of 97%, but with a 0 second pause time INSIGNIA has a PDR of 84% and CAT detection has a PDR of 80%; the PDR for ECN in a stationary network is also 97%, but it decreases to 43% with a 0 second pause time. ECN's higher

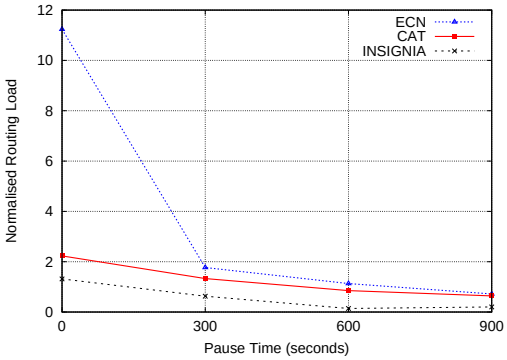
normalised routing load is the main reason for the lower PDRs as node mobility increases. As can be seen in Fig.6.10(b), the normalised routing loads for ECN are higher than for CAT detection and INSIGNIA. ECN's normalised routing load increases significantly as the pause time approaches 0 seconds: at the 0 second pause time, ECN's normalised routing load is approximately 5 times greater than that of CAT detection. This high normalised routing load is because the combination of increasing node mobility and higher network load lead to congestion, and this causes intermediate nodes to transmit ECN control packets to source nodes. As was explained for Fig. 6.9(b), node mobility leads to different intermediate nodes participating in paths between a source node and a destination node, and this leads to more ECN control packet transmissions. Additionally, as described for Fig. 6.9(c), the PriQueue mechanism and the high normalised routing load contribute to the high data packet loss: higher packet queue occupancy leads to data packet queues frequently overflowing, with 90% of ECN's packet loss at the 0 second pause time due to overflowing packet queues compared with 35% for CAT detection.

An additional effect of ECN's higher normalised routing load is longer end-to-end delays than both CAT detection and INSIGNIA. As can be seen in Fig. 6.10(c), the delays of all three schemes increase as node mobility increases, but the delays of ECN are the greatest of the three schemes at every pause time. As can be seen in the figure, ECN's delay is approximately 3.7 times greater than the delay for CAT detection at the 0 second pause time. ECN's higher delay is again a consequence of the interaction between ECN's high normalised routing load, as caused by the high network load and node mobility, and the PriQueue mechanism. The delays of both CAT detection and ECN are greater than those of INSIGNIA at every pause time. This is the opposite trend to the 10 source node case where the delays of CAT detection and ECN were less than the delays of INSIGNIA at all pause times. The reason for this is that both the CAT detection and ECN approaches are affected by congestion more than INSIGNIA. This is because the output from both CAT detection and ECN may lead to 2-DAARC enabling the MPA mode during a session; if the network is congested, the duplicated data packets injected into the network will contribute to the congestion and increase the delays. In contrast, INSIGNIA only ever uses a single copy of a data packet over a single path.

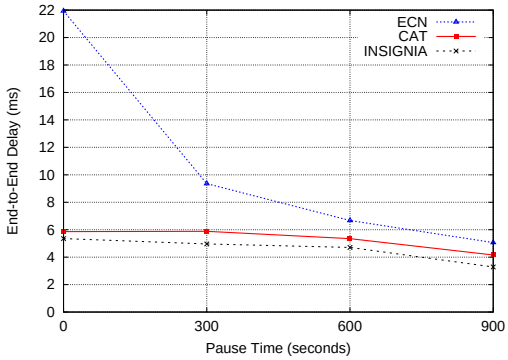
The PDRs achieved using CAT detection are significantly greater than the



(a) Packet Delivery Ratio



(b) Normalised Routing Load



(c) End-to-End Delay

Figure 6.10: Comparing the PDR, Normalised Routing Load, and End-to-End Delay of CAT Detection with ECN and INSIGNIA for 6 Priority Sources (14 Best-Effort, Background Sources) for 0, 300, 600, and 900 Second Pause Times in an Attacker-free Network.

PDRs achieved using 2-DAARC without CAT detection for a 0 second pause time and with 20 source nodes. Thus the CAT detection mechanism achieves its purpose of (1) supporting better QoS by selecting the most appropriate mode of adaptation in congested network conditions, and (2) not exacerbating the existing congestion. For example, at the 0 second pause time, the PDR for 2-DAARC without CAT detection is 33% with packet salvaging enabled (Fig. 5.19(a)) and 58% with packet salvaging disabled (Fig. 5.19(b)); the PDR is 80% when CAT detection is used (Fig. 6.10(a)). 2-DAARC achieves these improvements in QoS by using the CAT detection mechanism to detect congestion and to inform the adaptation algorithm to enable the SPA-ONLY mode and to disable the packet salvaging mechanism. Although CAT detection does not increase 2-DAARC's PDRs to the same values as those achieved by INSIGNIA, it offers a marked

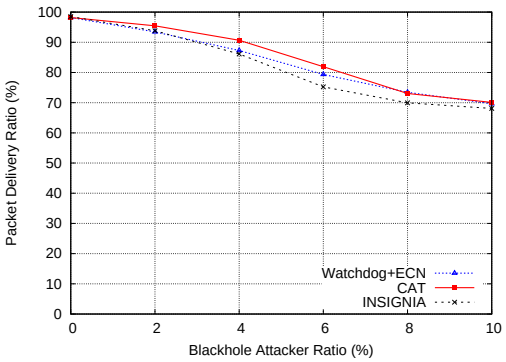
improvement in the PDRs which 2-DAARC achieves in congested networks containing nodes which are continually mobile.

The above results demonstrate that the approach of CAT detection is superior to that of ECN. With regard to PDRs, normalised routing loads, and end-to-end delays, CAT detection consistently outperforms ECN in both heavy and light network conditions. In particular, CAT detection significantly outperforms ECN in heavily loaded network conditions. In addition, it can be seen from the above results that when the network is lightly loaded, the PDRs and end-to-end delays of CAT detection are similar to or better than those of INSIGNIA, and they are only marginally worse than INSIGNIA when the network is more heavily loaded. However, as is to be explained in the following section, CAT detection is more resistant to packet forwarding attacks on QoS than INSIGNIA.

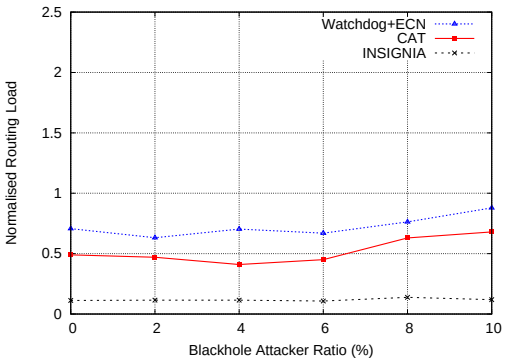
6.5.2.2 Adapting to Network Congestion and Blackhole Attackers

This section presents the results when blackhole attackers are introduced into the network. CAT detection is evaluated against INSIGNIA, a version of 2-DAARC which uses both Watchdog and ECN, hereafter referred to as Watchdog+ECN, and 2-DAARC without CAT detection (the 2-DAARC configuration from Chapter 5). Integrating Watchdog into 2-DAARC is described in Section 3.5.2.5. As INSIGNIA does not use adaptive packet salvaging, the INSIGNIA curves in Figs. 6.11 and 6.13 are with packet salvaging enabled and the INSIGNIA curves in Figs. 6.12 and 6.14 are with packet salvaging disabled. These results show the best values achievable by INSIGNIA in the given network conditions. The evaluation first focusses on a stationary network (900 second pause time), before focussing on a mobile network (0 second pause time).

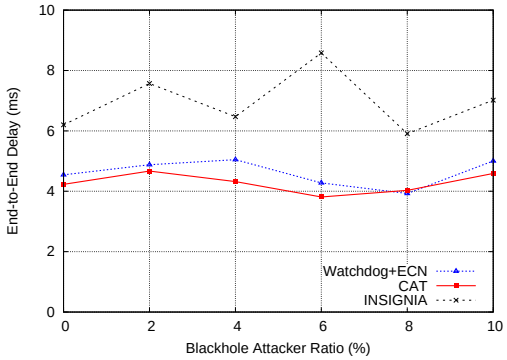
When the network load is low (10 source nodes) and all nodes are stationary, CAT detection generally supports higher PDRs than Watchdog+ECN and INSIGNIA. This can be seen in Fig. 6.11(a). The lower PDRs of Watchdog+ECN are due to its delayed initial notification of the PLR to the source node when using Watchdog: an intermediate node needs to receive a minimum significant number of priority data packets before it calculates the PLR and transmits it to the source node if it is greater than a threshold value (as explained in Section 3.5.2.5). In contrast, CAT detection uses the PLR in the 2-DAARC feedback packets which are transmitted every τ seconds. Thus, in the given network conditions, CAT detection can respond to the effects of attacks more quickly and



(a) Packet Delivery Ratio



(b) Normalised Routing Load



(c) End-to-End Delay

Figure 6.11: Comparing the PDR, Normalised Routing Load, and End-to-End Delay of CAT Detection with Watchdog+ECN and INSIGNIA for 3 Priority Sources (7 Best-Effort, Background Sources) and a 900 Second Pause Time in the Presence of Blackhole Attackers.

more effectively than the Watchdog component of Watchdog+ECN. Moreover, as shown in Fig. 6.11(b), the control packets transmitted by the Watchdog component, in addition to those transmitted by the ECN component, lead to a higher normalised load than for CAT detection. In other words, CAT detection achieves higher PDRs than Watchdog+ECN and it does so with a lower normalised routing load. The normalised routing load increases slightly at the 8% and 10% attacker ratios for both CAT detection and Watchdog+ECN. This occurs for the same reason as that described in Section 6.5.1, where the number of control packets transmissions increases whilst the PDR decreases.

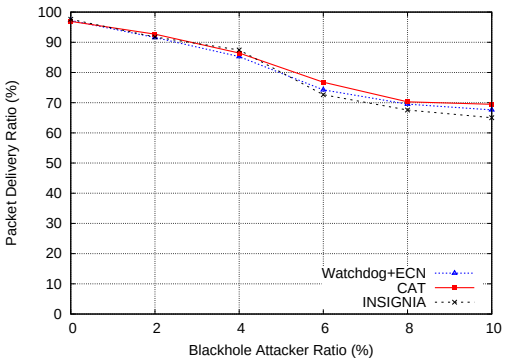
The difference in PDR that CAT detection achieves over INSIGNIA in the given network conditions is similar to the difference in PDR between 2-DAARC without CAT detection and INSIGNIA (as shown in Fig. 5.16(a)). Therefore, when the network is lightly loaded and all nodes are static, CAT detection does

not lead to an improvement in PDR compared with 2-DAARC without CAT detection; but what is important here is that the additional functionality introduced by CAT detection does not make the PDRs worse in the given network conditions. This is important because the main focus of CAT detection and adaptation is to overcome 2-DAARC's shortcomings in more highly loaded and congested networks.

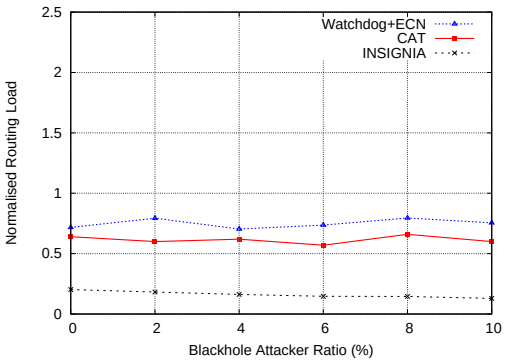
In addition to the generally higher PDRs, CAT detection achieves lower end-to-end delays than Watchdog+ECN and INSIGNIA (Fig. 6.11(c)). As described above for the PDR results, Watchdog+ECN takes longer than CAT detection to determine that attacks are the dominant factor causing packet loss. Consequently, it takes longer to adapt to the network conditions. Watchdog+ECN therefore uses a mode of adaptation which is less suited to the current network conditions, and it does so for a longer duration than CAT detection; this leads to its longer end-to-end delays. Both CAT detection and Watchdog+ECN achieve shorter delays than INSIGNIA. Additionally, their delays are less variable than INSIGNIA's. As described in Section 5.6.3.2, transmitting duplicated data packets along multiple paths leads to shorter delays than transmitting a single data packet along a single path. Moreover, combining multi-path routing with a mechanism to detect and react to attacks, as is the case with both CAT detection and Watchdog+ECN, leads to the end-to-end delays becoming less variable as the attacker ratio changes. Using 2-DAARC with CAT detection leads to superior end-to-end delay support compared with both Watchdog+ECN and INSIGNIA under the given network conditions.

When increasing the network load to that of 20 source nodes, and keeping all other parameter values unchanged, CAT detection achieves PDRs which are better than or similar to the PDRs of Watchdog+ECN and INSIGNIA. As can be seen by comparing Fig. 6.12(a) with Fig. 6.11(a), the PDRs for CAT detection are reduced compared with the 10 source node case. Additionally, there is an increase in the normalised routing load (Fig. 6.12(b) compared with Fig. 6.11(b)). These observations are due to the higher network load causing congestion. The end-to-end delays for CAT detection and Watchdog+ECN, as shown in Fig. 6.12(c), are similar to those experienced in the 10 source node case (Fig. 6.11(c)).

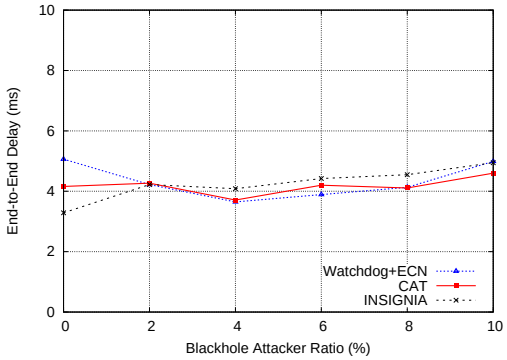
CAT detection also achieves PDRs which are greater than those achieved by 2-DAARC without CAT detection. The PDRs with CAT detection (Fig. 6.12(a))



(a) Packet Delivery Ratio



(b) Normalised Routing Load



(c) End-to-End Delay

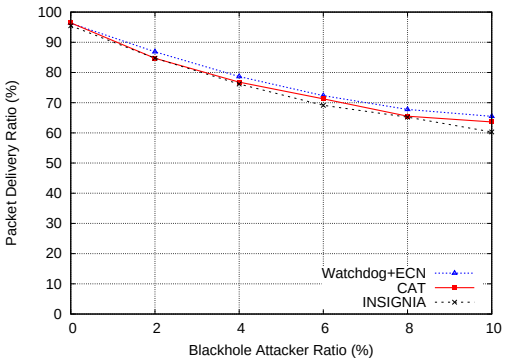
Figure 6.12: Comparing the PDR, Normalised Routing Load, and End-to-End Delay of CAT Detection with Watchdog+ECN and INSIGNIA for 6 Priority Sources (14 Best-Effort, Background Sources) and a 900 Second Pause Time in the Presence of Blackhole Attackers.

are greater than or similar to those of INSIGNIA, whereas the PDRs for 2-DAARC without CAT detection (Fig. 5.17(a)) are less than those of INSIGNIA at all attacker ratios. For example, the PDR for CAT detection is 97% with a 0% attacker ratio (Fig.6.12(a)) compared with 94% PDR for 2-DAARC without CAT detection (Fig. 5.17(a)). The 97% PDR with CAT detection is the same PDR that is achieved by INSIGNIA. CAT detection achieves higher PDRs than INSIGNIA at all of the non-zero attacker ratios investigated, whereas 2-DAARC without CAT detection achieves PDRs which are at best as good as INSIGNIA. The reason for these observations is as follows. As previously stated, the INSIGNIA results for this figure are obtained with packet salvaging enabled, as they demonstrate the best results achievable by INSIGNIA in the given network conditions. The results for 2-DAARC without CAT detection are also obtained with packet salvaging enabled. In comparison, using the CAT detection mechanism

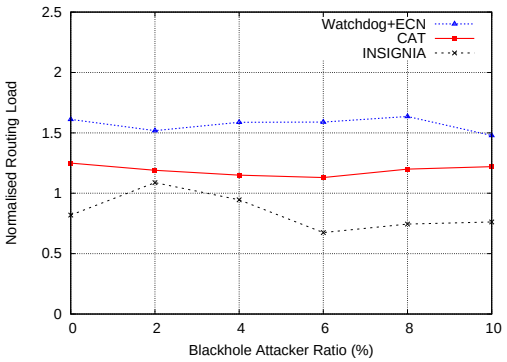
leads to E2-DAARC dynamically enabling and disabling the packet salvaging mechanism, via the APS mechanism, in response to the network conditions: with a low number of attackers, congestion is the dominant factor causing packet loss, and E2-DAARC uses the SPA-ONLY mode with packet salvaging disabled; however, when attacks are the dominant factor causing packet loss, E2-DAARC uses the MPA mode with packet salvaging enabled only when it is appropriate to do so. This leads to CAT detection achieving higher PDRs than INSIGNIA and 2-DAARC without CAT detection as they do not perform this type of adaptation.

In addition to the higher PDRs, CAT detection also has a lower normalised routing load (Fig. 6.12(b)) compared with 2-DAARC without CAT detection (Fig. 5.17(b)). The decrease in normalised routing load is, on average, approximately 12%, from 0.70 control packets for 2-DAARC without CAT detection to 0.61 control packets with CAT detection enabled. The lower normalised routing load is due to 2-DAARC using the SPA-ONLY mode with packet salvaging disabled when the network is congested. Disabling packet salvaging aims to avoid exacerbating existing congestion. Consequently, fewer link breaks occur due to congestion and fewer RERR control packets are transmitted. This also means that fewer Route Discovery operations are performed, therefore fewer Route Request and Route Reply packets are injected into the network. Thus with a high network load and when all nodes are static, using CAT detection leads to higher PDRs with lower normalised routing loads than 2-DAARC without CAT detection.

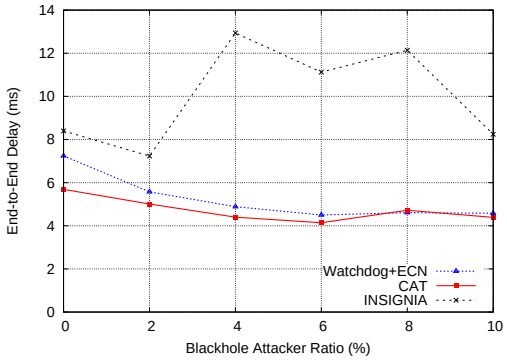
The remaining results presented in this section focus on the 0 second pause time, i.e., when all nodes are constantly in motion. With the low network load of 10 source nodes, the PDRs of CAT detection are slightly greater than or similar to the PDRs for INSIGNIA, although they are less than the PDRs of Watchdog+ECN (Fig. 6.13(a)). Watchdog+ECN achieves slightly higher PDRs because, under the given network conditions, Watchdog+ECN is able to identify more accurately that attacks are causing the packet loss. It does this via its Watchdog mechanism. Watchdog+ECN is therefore able to perform adaptive actions which are better suited to the network conditions. However, as can be seen in Fig. 6.13(b), this increased accuracy of Watchdog+ECN's detection and adaptation comes at the cost of a higher normalised routing load than CAT detection. This is due to the Watchdog and ECN control packets transmitted to a source node to notify it of attacks and congestion. The normalised routing loads



(a) Packet Delivery Ratio



(b) Normalised Routing Load



(c) End-to-End Delay

Figure 6.13: Comparing the PDR, Normalised Routing Load, and End-to-End Delay of CAT Detection with Watchdog+ECN and INSIGNIA for 3 Priority Sources (7 Best-Effort, Background Sources) and a 0 Second Pause Time in the Presence of Blackhole Attackers.

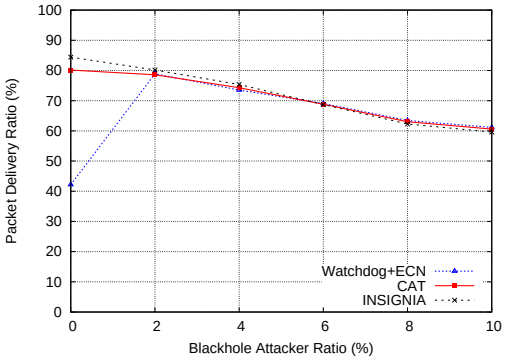
for both CAT detection and Watchdog+ECN are roughly double those exhibited in the stationary network (Fig. 6.11(b)). This is a consequence of node mobility leading to mobility-induced and congestion-induced path breaks.

CAT detection generally achieves lower end-to-end delays than both Watchdog+ECN and 2-DAARC without CAT detection. The lower delays of CAT detection compared with Watchdog+ECN (Fig. 6.13(c)) are due to it using the SPA-ONLY mode for data packet forwarding more frequently than Watchdog+ECN. The SPA-ONLY mode injects fewer data packets into the network than the MPA mode. This leads to nodes' packet queues being more lightly loaded. Queuing delays are reduced, which leads to shorter end-to-end delays. A general trend with the delays for CAT detection and Watchdog+ECN is that they decrease as the attacker ratio increases. This is a consequence of blackhole attackers dropping

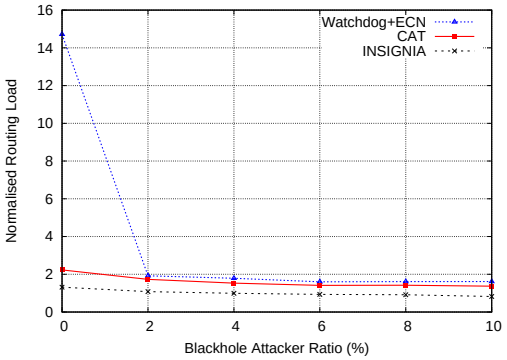
data packets, which reduces the data packet queue occupancy at nodes downstream of the attackers. However, the shorter end-to-end delays come at the cost of lower PDRs (for the reasons given above). 2-DAARC without CAT detection uses the MPA mode to achieve higher PDRs, but the combination of duplicated data packet transmissions and continual node mobility leads to it experiencing reduced effective bandwidth, greater congestion, higher packet queue occupancy, and therefore longer delays.

The end-to-end delays of both CAT detection and Watchdog+ECN are less than those of INSIGNIA, and this is especially the case with a non-zero attacker ratio (Fig. 6.13(c)). As described in Section 5.6.3.2, INSIGNIA's longer delays are a consequence of it transmitting a single data packet over a single path, whereas the 2-DAARC approach may transmit duplicated data packets over multiple paths. This indicates that, in the presence of packet forwarding attackers and under the given network conditions, dynamically changing the mode of adaptation and whether or not packet salvaging is performed can provide better QoS support than INSIGNIA's adaptation approach.

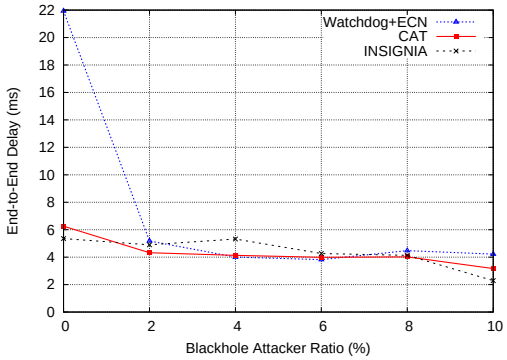
When the network load is increased to that of 20 source nodes, and all other parameter values remain unchanged, the three schemes perform similarly in terms of PDRs (Fig. 6.14(a)), normalised routing loads (Fig. 6.14(b)), and end-to-end delays (Fig. 6.14(c)), with the exception of the 0% attacker ratio. At the 0% attacker ratio, Watchdog+ECN brings the network into a congested state, resulting in a PDR of 42%. This is almost half the PDR of CAT detection at this attacker ratio. The congestion causing Watchdog+ECN's low PDR is a consequence of its high normalised routing load, and is explained as follows. As can be seen in Fig. 6.14(b), the normalised routing load for Watchdog+ECN at the 0% attacker ratio is approximately 6.6 times greater than that of CAT detection. A significant component of this normalised routing is the ECN control packets which inform the source node of congestion: ECN control packets account for 20% of the control packets at the 0% attacker ratio compared with only 4% at the 2% attacker ratio. The combination of ECN's large number of control packets and node mobility contribute to congestion in two ways. First, as described for Fig. 6.9(c), the PriQueue mechanism and the normalised routing load lead to longer delays. As can be seen in Fig 6.14(c), Watchdog+ECN's delay is approximately 3.7 times greater than the delay for CAT detection at the 0% attacker



(a) Packet Delivery Ratio



(b) Normalised Routing Load



(c) End-to-End Delay

Figure 6.14: Comparing the PDR, Normalised Routing Load, and End-to-End Delay of CAT Detection with Watchdog+ECN and INSIGNIA for 6 Priority Sources (14 Best-Effort, Background Sources) and a 0 Second Pause Time in the Presence of Blackhole Attackers.

ratio. One factor causing this is the high data packet queue occupancy, as described for Fig. 6.10(c), and this leads to overflowing packet queues and, therefore, packet loss. Second, as described for Fig. 6.9(b), more ECN control packets are transmitted as mobility increases. Thus the combination of congestion and mobility leads to Watchdog+ECN transmitting a large number of control packets, and these worsen the QoS in the already congested network conditions.

These results demonstrate that in a highly mobile and heavily loaded network, using the Watchdog+ECN approach has a detrimental effect on the achievable QoS. In contrast, the CAT detection mechanism achieves PDRs and end-to-end delays similar to those of INSIGNIA. Moreover, it does so with minimal control packet overhead, which in turn minimizes the chance of the network entering a congested state. Thus the benefits of using CAT detection over Watchdog+ECN are significant in the given network conditions.

Whilst the PDRs of CAT detection are similar to or marginally less than those of INSIGNIA, they are significantly higher than those of 2-DAARC without CAT detection. The PDRs for 2-DAARC without CAT detection and with packet salvaging enabled (Fig. 5.19(a)) range from 33% at the 0% attacker ratio to 40% at the 10% attacker ratio. With packet salvaging disabled (Fig. 5.19(b)) the PDRs for 2-DAARC without CAT detection range from 58% at the 0% attacker ratio to 62% at the 10% attacker ratio. The PDRs with CAT detection range from 80% at the 0% attacker ratio to 62% at the 10% attacker ratio. The advantage of using CAT detection is that (1) the APS response mechanism dynamically enables and disables the packet salvaging mechanism during a session, whereas it was previously configured as either enabled or disabled for the duration of a session; and (2) the MPA mode is not used when the network is congested. Thus dynamically adapting packet salvaging and the mode of adaptation (1) prevents the worsening of QoS, as was happening in Section 5.6.3.2, and (2) leads to PDRs similar to those of INSIGNIA.

In addition to the improvements in PDR, CAT detection also has lower normalised routing loads and lower end-to-end delays than 2-DAARC without CAT detection. There is a decrease in the averaged normalised routing load of approximately 62% from 3.10 control packets for 2-DAARC without CAT detection and with packet salvaging disabled (Fig. 5.19(d)) to 1.16 control packets with CAT detection (Fig. 6.14(b)). The decrease is even more significant for 2-DAARC without CAT detection with packet salvaging enabled (Fig. 5.19(c)): the decrease is approximately 92% from 14.9 control packets. The lower normalised routing load with CAT detection is because the MPA mode is used less frequently and fewer paths are required. Fewer Route Discovery operations are therefore performed, and this results in fewer control packets being injected into the network. As well as the lower normalised routing loads, the average end-to-end delay of CAT detection (Fig. 6.14(c)) is significantly reduced in comparison with 2-DAARC without CAT detection: the average delay with CAT detection is 4.3ms; the average delay without CAT detection and with packet salvaging enabled is 33.6ms (Fig. 5.19(e)), and with packet salvaging disabled it is 7.8ms (Fig. 5.19(f)). By supporting increases in PDR with decreases in the normalised routing load and end-to-end delay, the CAT detection mechanism effectively addresses the shortcomings of 2-DAARC in congested networks.

6.5.3 Major Findings

The simulation study of CAT detection and adaptation has led to the following major findings.

- In lightly or heavily loaded static networks (900 second pause time), CAT detection achieves PDRs which are better than or similar to those of INSIGNIA, Watchdog+ECN, and 2-DAARC without CAT detection.
- In lightly loaded mobile networks (0 second pause time), CAT detection achieves PDRs similar to those of INSIGNIA.
- In heavily loaded mobile networks, CAT detection achieves PDRs which are better than or similar to those of INSIGNIA and Watchdog+ECN.
- Based on the above PDR results, 2-DAARC with CAT detection generally supports the best PDRs of the schemes investigated in this chapter.
- CAT detection achieves end-to-end delays which are lower than or similar to those of INSIGNIA, Watchdog+ECN, and 2-DAARC without CAT detection in both static and mobile networks and with both light and heavy network loads. From these results it can be seen that, in terms of delay, CAT detection is the most effective of the schemes investigated.
- The normalised routing loads of CAT detection are less than those of Watchdog+ECN and 2-DAARC without CAT detection in both static and mobile networks and with both light and heavy network loads. The approach taken by CAT detection to packet loss detection is therefore more effective than the Watchdog+ECN approach. CAT detection exhibits a more effective use of the wireless bandwidth, as fewer control packets are required to deliver data packets to their intended destinations.

E2-DAARC's CAT detection and adaptation mechanism plays an important role in achieving a generally better QoS than both the Watchdog+ECN approach and 2-DAARC without CAT detection, and it does it with lower end-to-end delays and normalised routing loads under a range of network conditions. Basing adaptation decisions on the RERR rate and the feedback data can therefore be considered an effective means of adaptation to support QoS without injecting additional control traffic into a network.

6.6 Chapter Summary

This chapter presented the design and evaluation of CAT detection, a novel approach to congestion and attack detection. 2-DAARC was extended with the CAT detection mechanism (E2-DAARC) which it uses to determine the likely cause of packets loss without injecting additional control packets into the network, and without needing to impose trust on intermediate nodes. E2-DAARC adapts to the network conditions determined by CAT detection by (1) changing its mode of adaptation and (2) enabling or disabling its packet salvaging mechanism using an adaptive packet salvaging mechanism. The design of CAT detection was motivated and driven by observations made in the simulation-based performance evaluation of 2-DAARC in Chapter 5 and an additional simulation study performed in this chapter. The performance of E2-DAARC was then evaluated using simulation.

Based on the findings presented in this chapter, it was observed that E2-DAARC generally supported PDRs which are greater than or similar to those achieved by 2-DAARC without CAT detection, and it did so with lower end-to-end delays. However, E2-DAARC does not offer improvements under all network conditions: when the network load is light and all nodes are mobile, E2-DAARC achieves a smaller increase in PDR over INSIGNIA than 2-DAARC without CAT detection.

The following chapter concludes this thesis and provides recommendations for further research.

Chapter 7

Conclusion and Future Work

The focus of this thesis is on achieving QoS in MANETs containing packet forwarding attackers. This chapter summarises the work in this thesis, presents the conclusions drawn from the findings of this research, and gives recommendations for future work.

7.1 Conclusion

Background Research

At the start of this research, the related literature was read critically and analysed. MANET characteristics and security threats were investigated to understand their implications on QoS provisioning. It was observed that there is plenty of literature covering security or QoS, but there is little which addresses both issues together. To better understand the effects of MANET characteristics on packet forwarding and QoS, a simulation study of two state-of-the-art packet forwarding approaches was performed. A reservation-based approach (INSIGNIA) and a non-reservation-based approach (DSR) were simulated under a range of network conditions, including security attacks. The aim of this study was to analyse critically the existing approaches (1) to provide evidence that security and QoS should be integrated to achieve QoS, and (2) to use the insights gained from the study to inform the designs of the novel solutions presented in this thesis.

The following are four findings from the simulation study. First, the packet delivery ratio (PDR) decreases as the attacker ratio increases. This finding exposed the scope available for novel contributions to increase the PDR in the presence

of attackers. Second, a threshold value of packets delivered using the reserved forwarding service was observed. This was to serve as a benchmark against which the novel work in this research would be evaluated. Third, it was observed that a solution needs to perform adaptive actions in the absence of received feedback information. Without adaptation in the absence of feedback, a solution is not always able to respond in a timely manner to changes in network conditions. For example, the simulation study showed that INSIGNIA achieves better QoS when it performs adaptive actions in response to received feedback compared with the situations when feedback is not received and adaptive actions are not performed. Fourth, intermediate nodes cannot be trusted in an open MANET environment. A new approach to QoS provisioning should therefore delegate as many of the new QoS operations as possible to the source and destination nodes of a priority data packet flow. These findings not only demonstrate that there is scope for novel contributions to address the issue of QoS provisioning in the presence of attackers, but they also enable the designs of the novel solutions to be evidence-driven: the solutions can therefore overcome the limitations and weaknesses of the existing approaches whilst harvesting their strengths.

2-Dimensional Adaptation Architecture

Based on the findings of the simulation study, a novel routing architecture for QoS provisioning was developed. The 2-Dimensional Adaptation ARChitecture (2-DAARC) integrates two dimensions (modes) of adaptation: a single-path adaptation (SPA) mode and a multi-path adaptation (MPA) mode. The SPA mode embeds the strengths of INSIGNIA for single-path bandwidth reservations. This mode is used to support QoS for priority data packets in the presence of node mobility. The MPA mode extends the DSR protocol to provide a multi-path routing capability. This mode is used to support priority data packet deliveries in the presence of packet forwarding attacks. It does this by duplicating the priority data packets over the multiple paths. The MPA mode uses a novel path selection mechanism (described below) to select a secondary path which is maximally disjoint from the primary path. A secondary path is selected by a source node from the paths stored in its Route Cache. These paths are obtained during the Route Discovery process of 2-DAARC's underlying routing protocol (DSR). Selecting paths at the source node has four benefits: (1) no additional control packets are injected into the network to obtain a secondary path; (2) untrusted intermediate

nodes are not involved in calculating path disjointedness; (3) the source node can switch from the SPA mode to the MPA mode with low delay; and (4) the source node can promptly change the secondary path when the in-use path breaks or offers poor QoS.

To optimize packet forwarding for the dynamic MANET conditions, 2-DAARC performs adaptation both within and between the SPA and the MPA modes. Adaptation within the SPA mode is performed to maximise the percentage of priority data packets delivered using the bandwidth-reserved forwarding service. This is undertaken by changing paths when the percentage of packets receiving the bandwidth-reserved service falls below a threshold value. Adaptation within the MPA mode aims to maximise the percentage of packets delivered when the network is under attack. This is done by changing the path(s) when the PDR falls below a threshold value. This aims to ensure that the most appropriate mode of adaptation is being used for the current network conditions. Adaptation between the modes is governed by the level of packet loss: if the packet loss exceeds or is equal to a threshold value the MPA mode is enabled; if the packet loss falls below the threshold value the SPA mode is enabled.

A source node learns the packet loss and percentage of packets delivered using the reserved forwarding service from feedback packets received from the destination node. These packets contain QoS statistics which are calculated by the destination node. Based on the observation in the above described simulation study, it is also necessary to perform adaptation in the absence of feedback being received at the source node: if feedback is not received before the expiration of a timeout, the source node selects a different path for priority data packet forwarding.

The effectiveness of the 2-DAARC approach is evaluated against INSIGNIA using a simulation study. The simulation investigation demonstrates that in low network loads 2-DAARC achieves higher PDRs, lower end-to-end delays, and delivers a greater percentage of packets using the bandwidth-reserved forwarding service compared with INSIGNIA.

One of the shortcomings of the 2-DAARC approach, which was identified during the simulation investigation, is that using the MPA mode in congested networks leads to a deterioration in QoS. This is a consequence of the adaptation process used to enable the MPA mode: using only the packet loss statistic is too course-grained, as it does not indicate whether congestion and/or attacks is

the cause of packet loss. When congestion is the dominant factor causing packet loss, enabling the MPA mode exacerbates the congestion, and this worsens QoS. Based on these observations, the use of the MPA mode must be restricted to lightly loaded networks. A more tailored adaptation process is therefore required to select the most effective mode of adaptation for the network conditions. To enable this, a more fine-grained understanding of the cause of packet loss should be acquired. This needs to be determined (1) without injecting additional control packets into a network which may already be suffering the effects of congestion, and (2) without having to rely on untrusted intermediate nodes to perform congestion and attack detection operations. The CAT detection mechanism, described below, addresses this.

Priority-based Multi-path Type Selection

2-DAARC's MPA mode uses a novel Priority-based Multi-path Type Selection (PMTS) algorithm for secondary path selection. It selects paths from a source node's Route Cache in a priority-based manner depending on their disjointedness. This aims to make the best use of the paths already discovered by the source node. Disjointedness is a measure of redundancy. Node-disjoint paths offer the highest redundancy and are the most preferred path type. Link-disjoint paths offer less redundancy than node-disjoint paths, and are preferred if a node-disjoint path is not available. Non-disjoint paths offer the least redundancy of the three paths types and are the least preferred.

Intuitively, using a path with less than the maximum redundancy may lead to a blackhole attacker participating in multiple paths simultaneously. One way to mitigate this issue is to use only node-disjoint paths. Thus in addition to the PMTS approach, a node-disjoint-path-only (NDO) approach to path selection was investigated, and its performance evaluated against that of PMTS. The simulation investigation demonstrates that in lightly loaded networks PMTS has lower normalised routing loads, lower end-to-end delays, and PDRs which are at best similar to those of NDO. When network nodes are static, NDO achieves higher PDRs than PMTS.

Congestion and Attack Detection

To overcome the above identified shortcoming of the 2-DAARC approach, a novel Congestion and ATtack (CAT) detection mechanism was developed. CAT

detection infers the most likely cause of packet loss (congestion and/or attacks), and uses this to select the most appropriate mode of adaptation for the current network conditions. The likely cause of packet loss is inferred from (1) the arrival rate of ROUTE ERROR control packets, and (2) the packet loss statistics received in feedback packets from the destination node. CAT detection does not inject additional control packets into the network: the ROUTE ERROR packets are already transmitted by 2-DAARC's underlying routing protocol (DSR), and feedback packets are already transmitted as part of 2-DAARC's existing adaptation process. CAT detection therefore does not increase network load. Moreover, congestion and attacks are detected without requiring intermediate nodes to perform any operations other than those already prescribed by the DSR protocol. This means that CAT detection does not impose trust on intermediate nodes.

The performance of 2-DAARC extended with the CAT detection mechanism (E2-DAARC) is evaluated against three approaches: 2-DAARC extended with Watchdog (for attack detection) and Explicit Congestion Notification (for congestion detection), the original version of 2-DAARC, and INSIGNIA. The main findings are as follows. In light and heavy network loads with static nodes, E2-DAARC achieves PDRs which are greater than or similar to, and end-to-end delays which are less than or similar to, those of 2-DAARC with Watchdog and ECN. In light and heavy network loads with static or mobile nodes, E2-DAARC achieves lower normalised routing loads than 2-DAARC with Watchdog and ECN. When evaluated against the original version of 2-DAARC in lightly loaded networks, E2-DAARC generally achieves similar PDRs, with lower end-to-end delays and normalised routing loads. In heavily loaded networks, E2-DAARC achieves significantly higher PDRs with lower end-to-end delays and normalised routing loads. When evaluated against INSIGNIA in lightly loaded networks, E2-DAARC achieves PDRs which are greater than or similar to those of INSIGNIA and lower end-to-end delays. In heavily loaded networks, E2-DAARC achieves end-to-end delays which are less than or similar to those of INSIGNIA, along with PDRs which are greater than or similar to those of INSIGNIA in static networks, and PDRs which are less than or similar to INSIGNIA in mobile networks.

7.2 Suggestions for Future Research

The following presents four recommendations for future research.

Improving adaptation to make greater use of CAT detection

Augmenting 2-DAARC's adaptation process to use the output from CAT detection in new ways could improve QoS. Two examples of potential ways to improve QoS are outlined as follows. First, the output from CAT detection could be used to blacklist the paths which are likely to contain packet forwarding attackers. This would prevent a source node attempting to re-use these paths later in a session. Additionally, by comparing the blacklisted paths to identify common nodes it may be possible to determine which of the nodes is an attacker. The source node could use this information during path selection to avoid paths containing the suspected attacker. Second, when the MPA mode is active, the output from CAT detection could be used to choose the most appropriate secondary path selection mechanism for the current network conditions. It was observed in Section 5.6.3.3 that PMTS and NDO perform better under different network loads and node mobilities; thus the output from CAT detection could be used to choose the secondary path selection mechanism adaptively.

Determining whether redundant routing or data dispersion is a better strategy for QoS

2-DAARC's MPA mode currently uses a redundant routing approach, but it may be possible to support the necessary QoS without duplicating data packets in their entirety. Data dispersion splits every data packet into a number of smaller packets, each containing extra bits. The extra bits are calculated in such a way that the original packet can be reconstructed given a subset of these smaller packets. The smaller packets are then transmitted across multiple paths. Redundant routing and data dispersion could be investigated to determine the QoS they offer and the costs associated with supporting that level of QoS, e.g., the control packet cost and the additional delay required to deconstruct a packet at the source node and to reconstruct it at the destination node. If it is observed that the two approaches each support better QoS under different conditions, 2-DAARC's adaptation process could be improved further to select dynamically the most appropriate routing approach for the network conditions.

Investigating the energy costs of the 2-DAARC approach

The research presented in this thesis has not considered the energy costs of

the proposed solutions. The redundant routing approach of the MPA mode is likely to be expensive in terms of energy consumption. The energy consumed may lead to some nodes' batteries becoming exhausted. This will mean that these nodes will no longer be able to participate in the network, and this will lead to a reduction in the number of paths. Consequently, the available bandwidth will decrease, and this may negatively affect QoS. The following describes two examples of investigations on energy consumption which could be performed. First, the energy consumption of the redundant routing and data dispersion approaches to multi-path routing could be investigated. If it is observed that the two approaches have lower energy consumption under different network conditions whilst still supporting QoS requirements, they could be used adaptively to maximise nodes' lifetimes. Second, the energy consumption of CAT detection could be compared with the Watchdog misbehaviour detection mechanism. Unlike CAT detection, Watchdog uses the promiscuous receive mode of the wireless interface to perform misbehaviour detection. Frequently enabling the wireless interface for this purpose may have a significant energy cost. CAT detection may therefore exhibit significant energy savings over Watchdog in addition to the QoS advantages it already offers.

Applying 2-DAARC to an IMANET

The 2-DAARC approach has not been evaluated in the context of an Internet-based MANET (IMANET), where a MANET serves as an access network to the Internet. 2-DAARC could be investigated to determine whether its approach is effective in this environment. There are several possible areas of investigation. For example, is the 2-DAARC approach readily applicable to the IMANET environment? When using the MPA mode, how can data packet duplication be terminated at the gateway node, and how can this node be trusted to perform this action? How can nodes in a MANET be made addressable so that they can be contacted by nodes in the Internet?

In conclusion, the aim of this research—to achieve QoS provisioning in a MANET containing packet forwarding attackers—has been achieved, although a number of areas for future work remain.

Bibliography

[1] M. Abolhasan. A Review of Routing Protocols for Mobile Ad Hoc Networks. *Ad Hoc Networks*, 2(1):1–22, January 2004.

[2] G-S. Ahn, A. T. Cambell, S-B Lee, and X. Zhang. INSIGNIA. Internet Draft `http://tools.ietf.org/html/draft-ietf-manet-insignia-01`, 1999.

[3] Gahng-Seop Ahn, A. T. Campbell, A. Veres, and Li-Hsiang Sun. Supporting Service Differentiation for Real-Time and Best-Effort Traffic in Stateless Wireless Ad Hoc Networks (SWAN). *IEEE Transactions on Mobile Computing*, 1(3):192–207, July 2002.

[4] Jamal N. Al-Karaki and Ahmed E. Kamal. Quality of Service Routing in Mobile Ad Hoc Networks: Past and Future. In M. Ilyas and I. Mahgoub, editors, *Mobile Computing Handbook*, chapter 25, pages 569–610. CRC Press, London, 2005.

[5] S.H. Alabbad and M.E. Woodward. Localised Credit Based QoS Routing. *Communications, IEE Proceedings*, 153(6):787–796, December 2006.

[6] B. Awerbuch, R. Curtmola, D. Holmer, H. Rubens, and C. Nita-Rotaru. On the Survivability of Routing Protocols in Ad Hoc Wireless Networks. In *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*, pages 327–338, 2005.

[7] Awerbuch, Baruch and Holmer, David and Nita-Rotaru, Cristina and Rubens, Herbert. An On-Demand Secure Routing Protocol Resilient to Byzantine Failures. In *Proceedings of the 1st ACM workshop on Wireless security*, WiSE '02, pages 21–30, New York, NY, USA, 2002. ACM.

[8] E. Ayanoglu, Chih-Lin I, R.D. Gitlin, and J.E. Mazo. Diversity Coding for Transparent Self-Healing and Fault-Tolerant Communication Networks. *Communications, IEEE Transactions on*, 41(11):1677–1686, nov 1993.

[9] Elaine Barker and Allen Roginsky. NIST Special Publication 800-131A, Transitions: Recommendation for Transitioning the Use of of Cryptographic Algorithms and Key Sizes. `http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf`, January 2011. Date Retrieved: 26/07/2011.

[10] Abdussalam Baryun. [manet] All Wireless Mesh Networks are MANETs. `http://www.ietf.org/mail-archive/web/manet/current/msg13695.html`, October 2012. Date Retrieved: 22/10/2012. MANET Mailing List.

[11] Paolo Bellavista, Antonio Corradi, Corrado Federici, Rebecca Montanari, and Daniela Tibaldi. Security for Mobile Agents: Issues and Challenges. In M. Ilyas and I. Mahgoub, editors, *Mobile Computing Handbook*, chapter 39, pages 941–960. CRC Press, 2005.

[12] V. Berman and B. Mukherjee. Data Security in MANETs using Multipath Routing and Directional Transmission. In *Communications, 2006. ICC '06. IEEE International Conference on*, volume 5, pages 2322–2328, June 2006.

[13] R. Beuran. VoIP over Wireless LAN Survey. Technical Report IS-RR-2006-005, Japan Advanced Institute of Science and Technology (JAIST), Ishikawa, Japan, April 2006.

[14] T. Bheemarjuna Reddy, S. Sriram, B. S. Manoj, and C. Siva Ram Murthy. MuSeQoR: Multi-Path Failure-Tolerant Security-Aware QoS Routing in Ad Hoc Wireless Networks. *Comput. Netw.*, 50:1349–1383, June 2006.

[15] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. RFC 2475: An Architecture for Differentiated Services, December 1998.

[16] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, and S. Jamin. RFC 2205: Resource ReSerVation Protocol (RSVP) — Version 1 Functional Specification, September 1997.

[17] Josh Broch, David A. Maltz, David B. Johnson, Yih C. Hu, and Jorjeta Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 85–97, New York, NY, USA, 1998. ACM.

[18] S. Buchegger and J.-Y. Le Boudec. 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*, pages 403–410, 2002.

[19] Sonja Buchegger and Jean-Yves Le Boudec. Performance Analysis of the CONFIDANT Protocol. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '02, pages 226–236, New York, NY, USA, 2002. ACM.

[20] Levente Buttyán and Jean-Pierre Hubaux. Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks. *Mob. Netw. Appl.*, 8:579–592, October 2003.

[21] Jiwen Cai, Ping Yi, Jialin Chen, Zhiyang Wang, and Ning Liu. An Adaptive Approach to Detecting Black and Gray Hole Attacks in Ad Hoc Network. In *Proceedings of the 2010 24th IEEE International Conference on Advanced Information Networking and Applications*, AINA '10, pages 775–780, Washington, DC, USA, 2010. IEEE Computer Society.

[22] S. Chakrabarti and A. Mishra. QoS Issues in Ad hoc Wireless Networks. *Communications Magazine, IEEE*, 39(2):142–148, 2001.

[23] Giriraj Chauhan and Sukumar Nandi. QoS Aware Stable Path Routing (QASR) Protocol for MANETs. *Emerging Trends in Engineering & Technology, International Conference on*, 0:202–207, 2008.

[24] Jianyong Chen, Huawang Zeng, Cunying Hu, and Zhen Ji. Optimization Between Security and Delay of Quality-of-Service. *J. Netw. Comput. Appl.*, 34(2):603–608, March 2011.

[25] Ruiliang Chen, M. Snow, Jung-Min Park, M. T. Refaei, and M. Eltoweissy. Defense against Routing Disruption Attacks in Mobile Ad Hoc Networks.

In *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, pages 1–5, April 2007.

[26] Shigang Chen and Klara Nahrstedt. Distributed Quality-of-Service Routing in Ad-Hoc Networks. *IEEE Journal on Selected Areas in Communications*, 17(8), August 1999.

[27] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental), October 2003.

[28] Clausen, T. and Dearlove, C. and Dean, J. and Adjih, C. Generalized Mobile Ad Hoc network (MANET) Packet/Message Format. RFC 5444 (Standards Track) `http://tools.ietf.org/html/rfc5444`, February 2009.

[29] Monarch Project: Wireless and Mobility Extensions to ns. `http://www.monarch.cs.rice.edu/cmu-ns.html`, Nov. 2000. Retrieved: 23/07/2010.

[30] M. S. Corson, J. P. Macker, and G. H. Cirincione. Internet-based Mobile Ad hoc Networking. *Internet Computing, IEEE*, 3(4):63–70, August 1999.

[31] Corson, S. and Macker, J. Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. RFC 2501 (Informational) `http://www.ietf.org/rfc/rfc2501.txt`, January 1999.

[32] Samir Ranjan Das, Charles E. Perkins, and Elizabeth M. Belding-Royer. Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks. In *INFOCOM*, pages 3–12, 2000.

[33] B. Davie, A. Charny, J.C.R. Bennet, K. Benson, J.Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis. An Expedited Forwarding PHB (Per-Hop Behavior). RFC 3246 (Proposed Standard), March 2002.

[34] S. De and Chunming Qiao. Does Packet Replication Along Multipath Really Help? In *Communications, 2003. ICC '03. IEEE International Conference on*, volume 2, pages 1069–1073, 2003.

[35] Dell Inc. Inspiron^{TM}1545 Laptop. `http://www1.euro.dell.com/uk/en/home/Laptops/laptop-inspiron-1545/pd.aspx?refid=laptop-inspiron-1545\&s=dhs\&cs=ukdhs1`, 2010. Date Retrieved: 05/07/2010.

[36] Dinesh Dharmaraju, Ayan R. Chowdhury, Pedram Hovareshti, and John S. Baras. INORA—A Unified Signaling and Routing Mechanism for QoS Support in Mobile Ad hoc Networks. *Parallel Processing Workshops, International Conference on*, 2002.

[37] Shuo Ding. A Survey on Integrating MANETs with the Internet: Challenges and Designs. *Computer Communications*, 31(14):3537 – 3551, 2008.

[38] D. Djenouri, L. Khelladi, and A. N. Badache. A Survey of Security Issues in Mobile Ad Hoc and Sensor Networks. *IEEE Communications Surveys & Tutorials*, 7(4):2–28, February–April 2005.

[39] Linfang Dong, Yantai Shu, and Guanghong Wang. Independent-Set Based Neighborhood Reservation with In-Band Signaling System in Ad Hoc Networks. In *Electrical and Computer Engineering, 2006. CCECE '06. Canadian Conference on*, pages 1550–1553, May 2006.

[40] Sandrine Duflos, Brigitte Kervella, and Valerie C. Gay. Considering Security and Quality of Service in SLS to Improve Policy-Based Management of Multimedia Services. In *Proceedings of the Sixth International Conference on Networking*, ICN '07, Washington, DC, USA, 2007. IEEE Computer Society.

[41] J. Postel (Ed.). Internet Protocol, DARPA Internet Program Protocol Specification. RFC 791 `http://www.ietf.org/rfc/rfc791.txt`, September 1981.

[42] Nur I. Enzai, Farhat Anwar, and Omer Mahmoud. Evaluation Study of QoS-enabled AODV. In *Computer and Communication Engineering, 2008. ICCCE 2008. International Conference on*, pages 1254–1259, May 2008.

[43] Kevin Fall and Kannan Varadhan. *The ns Manual (formerly ns Notes and Documentation)*, January 2009.

[44] Sally Floyd and Vern Paxson. Difficulties in Simulating the Internet. *IEEE/ACM Transactions on Networking*, 9:392–403, August 2001.

[45] M. Frodigh, P. Johansson, and P. Larsson. Wireless Ad Hoc Networking—The Art of Networking Without a network. *Ericsson Review*, 4:248–263, 2000.

[46] Liljana Gavrilovska and Ramjee Prasad. *Ad-Hoc Networking Towards Seamless Communications*. Signals and Communication Technology. Springer, 2006.

[47] Olivier Gay. HMAC-SHA2. `http://www.ouah.org/ogay/hmac/`. Retrieved: 31/10/2011.

[48] About GloMoSim. `http://pcl.cs.ucla.edu/projects/glomosim/`. Retrieved: 19/01/2012.

[49] N. Gogate and S.S. Panwar. Supporting Video/Image Applications in a Mobile Multihop Radio Environment Using Route Diversity. In *Communications, 1999. ICC '99. 1999 IEEE International Conference on*, volume 3, pages 1701–1706, 1999.

[50] M. G. Gouda, C.-T. Huang, , and E. Li. Anti-Replay Window Protocols for Secure IP. In *Proceedings of the 9th IEEE International Conference on Computer Communications and Networks*, Las Vegas, October 2000.

[51] Zygmunt J. Haas and Edwards Y. Hua. Multipath Routing—A Cross-Layer Deign Tool for QoS Provisioning in MANETs. Technical report, Cornell University, Ithaca, NY, 14835, June 2004. Date Retrieved: 19/03/2012.

[52] Wenbo He and K. Nahrstedt. An Integrated Solution to Delay and Security Support in Wireless Networks. In *IEEE Wireless Communications and Networking Conference, 2006. WCNC 2006.*, pages 2211–2215. IEEE, 2006.

[53] Yan He and Hussein A. Wahab. HQMM: A Hybrid QoS Model for Mobile Ad-hoc Networks. In *Computers and Communications, IEEE Symposium on*, pages 194–200, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[54] Yang He, Juhua Pu, and Zhang Xiong. A Redundant Multipath Routing for Mobile Ad Hoc Networks. *Computer and Computational Sciences, International Multi-Symposiums on*, 0:75–82, 2008.

[55] A. Hegland and E. Winjum. Securing QoS Signaling in IP-Based Military Ad Hoc Networks. *Communications Magazine, IEEE*, 46(11):42–48, November 2008.

[56] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group. RFC 2597 (Proposed Standard), June 1999. Updated by RFC 3260.

[57] M. Hejmo, B. L. Mark, C. Zouridaki, and R. K. Thomas. Design and Analysis of a Denial-of-Service-Resistant Quality-of-Service Signaling Protocol for MANETs. *IEEE Transactions on Vehicular Technology*, 55(3):743–751, May 2006.

[58] Ulrich Herberg. Re: [manet] Discussing LOADng suggestions. `http://www.ietf.org/mail-archive/web/manet/current/msg13304.html`, July 2012. Date Retrieved: 24/08/2012. MANET Mailing List.

[59] Naftall Herscovici, Christos Christodoulou, E. Kyriacou, M. Pattichis, C. Pattichis, A. Panayides, and A. Pitsillides. m-Health e-Emergency Systems: Current Status and Future Directions - [Wireless corner]. *IEEE Antennas and Propagation Magazine*, 49(1):216–231, February 2007.

[60] Hewlett-Packard Development Company. *HP Officejet Pro 8600 e-All-in-One series User Guide*. Hewlett-Packard, 2011. Date Retrieved: 19/03/2012.

[61] Bryan J. Hogan, Michael Barry, and Sean Mcgrath. Congestion Avoidance in Source Routed Ad Hoc Networks. In *in 13th IST Mobile and Wireless Communications Summit*, pages 682–686, 2004.

[62] S. Holeman, G. Manimaran, and J. Davis. Differentially Secure Multicasting and its Implementation Methods. In *Computer Communications and Networks, 2001. Proceedings. Tenth International Conference on*, pages 212–217, August 2002.

[63] Jorge Hortelano. SafeWireless Detection Mechanisms. `http://safewireless.sourceforge.net/tools.html`. Retrieved: 06/03/2012.

[64] HP. HP iPAQ 100 Series Classic Handheld. `http://www.shopping.hp.com/shopping/pdf/iPAQ_100series.pdf`, September 2007. Date Retrieved: 05/07/2010.

[65] HTC. HTC Desire. `http://www.htc.com/www/product/desire/specification.html`, 2010. Date Retrieved: 05/07/2010.

[66] Y.-C. Hu and D.B. Johnson. Exploiting Congestion Information in Network and Higher Layer Protocols in Multihop Wireless Ad Hoc Networks. In *Distributed Computing Systems, 2004. Proceedings. 24th International Conference on*, pages 301–310, 2004.

[67] Yih C. Hu and David B. Johnson. Securing Quality-of-Service Route Discovery in On-Demand Routing for Ad Hoc Networks. In *SASN '04: Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks*, pages 106–117, New York, NY, USA, 2004. ACM.

[68] Yih C. Hu and Adrian Perrig. A Survey of Secure Wireless Ad Hoc Routing. *IEEE Security and Privacy*, 2(3):28–39, 2004.

[69] Yih-Chun Hu. *Enabling Secure High-Performance Wireless Ad Hoc Networking*. PhD Thesis, Carnegie Mellon University, May 2003.

[70] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. In *MobiCom '02*, 2002.

[71] Yin-Chun Hu, Adrian Perrig, and David B. Johnson. Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols. In *WiSe '03: Proceedings of the 2nd ACM workshop on Wireless security*, pages 30–40, New York, NY, USA, 2003. ACM.

[72] Edward Y. Hua and Zygmunt J. Haas. Path Selection Algorithms in Homogeneous Mobile Ad Hoc Networks. In *Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing*, IWCMC '06, pages 275–280, New York, NY, USA, 2006. ACM.

[73] Elgan Huang, Jon Crowcroft, and Ian Wassell. Rethinking Incentives for Mobile Ad Hoc Networks. In *Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems*, PINS '04, pages 191–196, New York, NY, USA, 2004. ACM.

[74] Jean-Pierre Hubaux, Levente Buttyán, and Srdan Capkun. The Quest for Security in Mobile Ad Hoc Networks. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '01, pages 146–155, New York, NY, USA, 2001. ACM.

[75] E. M. Husni, Y. Heryadi, W. T. H. Woon, M. S. Arifianto, D. V. Viswacheda, and L. Barukang. Mobile Ad Hoc Network and Mobile IP for Future Mobile Telemedicine System. In *Mobile ad hoc network and mobile IP for future mobile telemedicine system*, pages 5–10, August 2006.

[76] ICUcare LLC. What Is Telemedicine? `http://www.icucare.com/PageFiles/Telemedicine.pdf`, December 2009. Date Retrieved: 03/06/2010.

[77] IEEE 802.11: Wireless Local Area Networks (WLAN) Working Group. IEEE Standard for Information technology–Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pages 1–2793, 29 2012.

[78] IETF. Mobile Ad-hoc Networks (manet) Working Group. `http://www.ietf.org/dyn/wg/charter/manet-charter.html`. Date Retrieved: 25/01/2010.

[79] M. Iordanakis and G. Dilintas. ARPAM Routing Protocol Vulnerabilities in Aeronautical Mobile Ad Hoc Networks. In *2nd International Scientific Conference eRA*, September 2007.

[80] C. Irvine, T. Levin, E. Spyropoulou, and B. Allen. Security as a Dimension of Quality of Service in Active Service Environments. In *Active Middleware Services, 2001. Third Annual International Workshop on*, pages 87–93, August 2001.

[81] Cynthia Irvine and Timothy Levin. Quality of Security Service. In *NSPW '00: Proceedings of the 2000 workshop on New security paradigms*, pages 91–99, New York, NY, USA, 2000. ACM.

[82] Svilen Ivanov, André Herms, and Georg Lukas. Experimental Validation of the ns-2 Wireless Model using Simulation, Emulation, and Real Network. In *In 4th Workshop on Mobile Ad-Hoc Networks (WMAN07*, pages 433–444, 2007.

[83] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized Link State Routing Protocol for Ad Hoc Networks. In *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*, pages 62–68, August 2002.

[84] Sanjay Jha and Mahbub Hassan. *Engineering Internet QoS*. Artech House, London, 2002.

[85] Ping Ji, Zihui Ge, Jim Kurose, and Don Towsley. A Comparison of Hard-state and Soft-state Signaling Protocols. In *In: Proc. of SIGCOMM 2003*, pages 251–262, 2003.

[86] Tao Jiang and John S. Baras. Ant-Based Adaptive Trust Evidence Distribution in MANET. In *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops - W7: EC (ICDCSW'04) - Volume 7*, ICDCSW '04, pages 588–593, Washington, DC, USA, 2004. IEEE Computer Society.

[87] Per Johansson, Tony Larsson, Nicklas Hedman, Bartosz Mielczarek, and Mikael Degermark. Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 195–206, New York, NY, USA, 1999. ACM.

[88] D. Johnson, Y. Hu, and D. Maltz. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. RFC 4728 (Experimental) `http://www.ietf.org/rfc/rfc4728.txt`, February 2007.

[89] David B. Johnson and David A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In Tomasz Imielinski and Hank Korth, editors, *Mobile Computing*, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.

[90] David B. Johnson, David A. Maltz, and Josh Broch. DSR: the Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks. In Charles E. Perkins, editor, *Ad Hoc Networking*, chapter 5, pages 139–172. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, December 2001.

[91] Mike Just, Evangelos Kranakis, and Tao Wan. Resisting Malicious Packet Dropping in Wireless Ad Hoc Networks. In Samuel Pierre, Michel Barbeau, and Evangelos Kranakis, editors, *Ad-Hoc, Mobile, and Wireless Networks*, volume 2865 of *Lecture Notes in Computer Science*, pages 151–163. Springer Berlin Heidelberg, 2003.

[92] Tetsushi Kamegawa, Noritaka Sato, Michinori Hatayama, Yojiro Uo, and Fumitoshi Matsuno. Design and Implementation of Grouped Rescue Robot System Using Self-Deploy Networks. *Journal of Field Robotics*, 28(6):977–988, 2011.

[93] Ramanarayana Kandikattu and Lillykutty Jacob. Secure Internet Connectivity for Dynamic Source Routing (DSR) based Mobile Ad hoc Networks. *International Journal of Electronics, Circuits and Systems*, 2(1), 2008.

[94] Qifa Ke, David A. Maltz, and David B. Johnson. Emulation of Multi-Hop Wireless Ad Hoc Networks. In *Proceedings of the Seventh International Workshop on Mobile Multimedia Communications (MOMUC 2000)*, Tokyo, Japan, October 2000. IEEE Communications Society.

[95] Kent, S. IP Authentication Header. RFC 4302 (Standards Track) `http://tools.ietf.org/html/rfc4302`, December 2005.

[96] David Kidston. Using Modelling and Simulation to Evaluate Network Services in Maritime Networks. In Gregorio Romero Rey and Luisa Martinez Muneta, editors, *Modelling Simulation and Optimization*. InTech, February 2010. `http://www.intechopen.com/articles/show/title/using-modelling-and-simulation-to-evaluate-network-services-in-maritime-networks`.

[97] Stuart Kurkowski, Tracy Camp, and Michael Colagrosso. MANET Simulation Studies: The Incredibles. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(4):50–61, October 2005.

[98] James F. Kurose and Keith W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison Wesley, 3 edition, 2005.

[99] L. Lamont, M. Wang, L. Villasenor, T. Randhawa, and S. Hardy. Integrating WLANs & MANETs to the IPv6 based Internet. In *Communications,*

2003. ICC '03. IEEE International Conference on, volume 2, pages 1090–1095 vol.2, May 2003.

[100] J. Y. Le Boudec and M. Vojnovic. Perfect Simulation and Stationarity of a Class of Mobility Models. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 4, pages 2743–2754, 2005.

[101] S. B. Lee, G. S. Ahn, and A. T. Campbell. Improving UDP and TCP Performance in Mobile Ad Hoc Networks with INSIGNIA. *IEEE Communications Magazine*, 39(6):156–165, June 2001.

[102] S.-B. Lee and A. T. Campbell. INSIGNIA: In-Band Signaling Support For QoS in Mobile Ad Hoc Networks. In *5th International Workshop on Mobile Multimedia Communications (MoMuc'98)*, October 1998.

[103] S. J. Lee and M. Gerla. Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks. In *Communications, 2001. ICC 2001. IEEE International Conference on*, volume 10, pages 3201–3205, August 2001.

[104] Seoung-Bum Lee. *Adaptive Quality of Service for Wireless Ad hoc Networks*. PhD thesis, Columbia University, 2006. `http://www.cs.dartmouth.edu/~campbell/papers/sbl-thesis.pdf`.

[105] Seoung-Bum Lee, Gahng-Seop Ahn, Xiaowei Zhang, and Andrew T. Campbell. Evaluation of the INSIGNIA Signaling System. In *In Proceedings of the International Conference on Broadband Communications, High Performance Networking, and Performance of Communication Networks (Networking*, pages 311–324, 2000.

[106] Seoung-Bum Lee, Gahng-Seop Ahn, Xiaowei Zhang, Andrew T. Campbell, Sung-Yup Nham, and Chang-Jae Yoo. INSIGNIA NS-2 Source Code. `http://comet.columbia.edu/insignia/ns_source_code.html`. Retrieved: 23/07/2010.

[107] Seoung-Bum Lee, Jiyoung Cho, and Andrew T. Campbell. A Hotspot Mitigation Protocol for Ad Hoc Networks. *AD HOC NETWORKS*, 1:87–106, 2003.

[108] Lee, Seoung-Bum and Ahn, Gahng-Seop and Zhang, Xiaowei and Campbell, Andrew T. INSIGNIA: An IP-Based Quality of Service Framework for Mobile ad Hoc Networks. *Journal of Parallel and Distributed Computing*, 60:374–406, 2000.

[109] Roy Leung, Roy L. Jilei, Edmond Poon, Ah-lot C. Chan, and Baochun Li. MP-DSR: A QoS-aware Multi-path Dynamic Source Routing Protocol for Wireless Ad-Hoc Networks. In *In IEEE LCN'01*, pages 132–141, 2001.

[110] Roy Leung, Roy L. Jilei, Edmond Poon, Ah-lot C. Chan, and Baochun Li. MP-DSR: A QoS-aware Multi-path Dynamic Source Routing Protocol for Wireless Ad-Hoc Networks. In *In IEEE LCN'01*, pages 132–141, 2001.

[111] Xuefei Li and L. Cuthbert. Stable Node-Disjoint Multipath Routing with Low Overhead in Mobile Ad Hoc Networks. In *Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, 2004. (MASCOTS 2004). Proceedings. The IEEE Computer Society's 12th Annual International Symposium on*, pages 184–191, October 2004.

[112] Haejung Lim, Kaixin Xu, and M. Gerla. TCP Performance over Multipath Routing in Mobile Ad Hoc Networks. In *Communications, 2003. ICC '03. IEEE International Conference on*, volume 2, pages 1064–1068, may 2003.

[113] Hun-Jung Lim, Soo-Jin Jung, Jong-Hyouk Lee, Young-Ju Han, and Tai-Myoung Chung. Ad-hoc Protocol Performance Analysis Based on Emergency Medical Data. In *The 9th International Conference on Advanced Communication Technology*, pages 52–56. IEEE, February 2007.

[114] M.N. Lima, H.W. da Silva, A.L. dos Santos, and G. Pujolle. Survival Multipath Routing for MANETs. In *Network Operations and Management Symposium, 2008. NOMS 2008. IEEE*, pages 425–432, April 2008.

[115] S. Lindskog, A. Brunstrom, and Z. Faigl. Analyzing Tunable Security Service. In *Fourth Swedish National Computer Networking Workshop (SNCNW 2006)*, October 2006.

[116] Stefan Lindskog and Erland Jonsson. Adding Security to Quality of Service Architectures. In *Proceedings of the Scuola Superiore G. Reiss Romoli 2002 Summer Conference (SSGRR-2002s)*, July 2002.

[117] Wenjing Lou, Wei Liu, and Yanchao Zhang. Performance Optimization Using Multipath Routing in Mobile Ad Hoc and Wireless Sensor Networks. In M.X. Cheng, Y. Li, and D. Du, editors, *Combinatorial Optimization in Communication Networks*, Combinatorial Optimization. Springer, 2006.

[118] Bin Lu and U. W. Pooch. Security in QoS Signaling Systems for Mobile Ad Hoc Networks. In *Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005), IEEE International Conference on*, volume 3, pages 213–220, October 2005.

[119] David Lundberg. Ad hoc Protocol Evaluation and Experiences of Real World Ad Hoc Networking. Master's thesis, Department of Information Technology, Uppsala University, Sweeden, 2002.

[120] Brian B. Luu, Barry J. O'Brien, David G. Baran, and Rommie L. Hardy. A Soldier-Robot Ad Hoc Network. In *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on*, pages 558–563, April 2007.

[121] David A. Maltz, Josh Broch, Jorjeta Jetcheva, and David B. Johnson. The Effects of On-Demand Behavior in Routing Protocols for Multi-Hop Wireless Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, 17:1439–1453, 1999.

[122] David A. Maltz, Josh Broch, and David B. Johnson. Experiences Designing and Building a Multi-Hop Wireless Ad Hoc Network Testbed. Technical Report CMU-CS-99-116, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, March 1999.

[123] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 255–265, New York, NY, USA, 2000. ACM Press.

[124] Peter J. J. McNerney and Ning Zhang. Towards an Integration of Security and Quality of Service in IP-Based Mobile Ad Hoc Networks. In *Proceedings of the Global Communications Conference (GLOBECOM 2011), 2011 IEEE*, Houston, Texas, USA, December 2011.

[125] Peter J. J. McNerney and Ning Zhang. A 2-Dimensional Approach to QoS Provisioning in Adversarial Mobile Ad Hoc Network Environments. In *Proceedings of the 15th International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM 2012), 2012 ACM*, MSWiM '12, New York, NY, USA, October 2012. ACM.

[126] Peter J. J. McNerney and Ning Zhang. A Study on Reservation-Based Adaptation for QoS in Adversarial MANET Environments. In *8th International Wireless Communications and Mobile Computing Conference (IWCMC 2012), 2012*, pages 677–682, Limassol, Cyprus, August 2012.

[127] Pietro Michiardi and Refik Molva. CORE: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks. In *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security: Advanced Communications and Multimedia Security*, pages 107–121, Deventer, The Netherlands, The Netherlands, 2002. Kluwer, B.V.

[128] Dragorad Milovanovic and Zoran Bojkovic. Integration QoS and Security Technologies in 4G Mobile Networks. In *ICCOM'05: Proceedings of the 9th WSEAS International Conference on Communications*, pages 1–4, Stevens Point, Wisconsin, USA, 2005. World Scientific and Engineering Academy and Society (WSEAS).

[129] Yasser L. Morgan and Thomas Kunz. A Design Framework for Wireless MANET QoS Gateway. *Wireless Conference 2005 - Next Generation Wireless and Mobile Communications and Services (European Wireless), 11th European*, pages 1–7, April 2005.

[130] Stephen Mueller, Rose P. Tsang, and Dipak Ghosal. Multipath Routing in Mobile Ad Hoc Networks: Issues and Challenges. In *Performance Tools and Applications to Networked Systems*, pages 209–234. Springer Berlin / Heidelberg, 2004.

[131] Paul R. Muessig, Dennis R. Laack, and John J. Wrobleski. An Integrated Approach to Evaluating Simulation Credibility. In *Proceedings of the 2000 Summer Computer Simulation Conference*, pages 449–457, 2000.

[132] Lama Nachman, Ralph Kling, Robert Adler, Jonathan Huang, and Vincent Hummel. The Intel®Mote platform: a Bluetooth-based sensor network for industrial monitoring. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, pages 61–66, Piscataway, NJ, USA, 2005. IEEE Press.

[133] A. Neogi, T. Chiueh, and P. Stirpe. Performance Analysis of an RSVP-Capable Router. *IEEE Network*, 13(5):56–63, Sept 1999.

[134] Calvin Newport, David Kotz, Yougu Yuan, Robert S. Gray, Jason Liu, and Chip Elliott. Experimental Evaluation of Wireless Simulation Assumptions. *Simulation*, 83:643–661, September 2007.

[135] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. RFC 2474 (Proposed Standard), December 1998. Updated by RFCs 3168, 3260.

[136] NIST. Cryptographic Hash Algorithm Competition. `http://csrc.nist.gov/groups/ST/hash/sha-3/index.html`, April 2005. Date Retrieved: 08/08/2011. Last Updated: 13 December 2010.

[137] NIST. SHA-3 Winner. `http://csrc.nist.gov/groups/ST/hash/sha-3/winner_sha-3.html`, October 2012. Date Retrieved: 3/10/2012. Last Updated: 2/10/2012.

[138] The Network Simulator - ns-2. `http://nsnam.isi.edu/nsnam/index.php/Main_Page`. Retrieved: 28/01/2010.

[139] M. Nystrom. Identifiers and Test Vectors for HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512. RFC 4231 (Standards Track) `http://tools.ietf.org/html/rfc4231`, December 2005.

[140] Barry J. O'Brien, David G. Baran, and Brian B. Luu. Ad Hoc Networking for Unmanned Ground Vehicles: Design and Evaluation at Command, Control, Communications, Intelligence, Surveillance and Reconnaissance On-the-Move. Technical report, Army Research Laboratory, November 2006. Date Retrieved: 09/11/2009.

[141] National Institute of Standards and Technology. Federal Information Processing Standard (FIPS) 180-3, Secure Hash Standard (SHS). `http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf`, October 2008. Date Retrieved: 28/09/2011.

[142] R. Ogier, F. Templin, and M. Lewis. Topology Dissemination Based on Reverse-Path Forwarding (TBRPF). RFC 3684 (Experimental), February 2004.

[143] A. Oliveira, Zhili Sun, M. Monier, P. Boutry, D. Gimenez, A. Pietrabissa, and K.B. Juros. On Optimizing Hybrid Ad-Hoc and Satellite Betworks—The MONET Approach. In *Future Network and Mobile Summit, 2010*, pages 1–8, june 2010.

[144] C. S. Ong, K. Nahrstedt, and Y. Wanghong. Quality of Protection for Mobile Multimedia Applications. In *ICME '03: Proceedings of the 2003 International Conference on Multimedia and Expo*, pages 137–140, Washington, DC, USA, 2003. IEEE Computer Society.

[145] OPNET Technologies, Inc. Discrete Event Simulation Model Library. `http://www.opnet.com/solutions/network_rd/simulation_model_library/`, 2012. Date Retrieved: 16/01/2012.

[146] OPNET Technologies, Inc. OPNET Modeler Accelerating Network R&D. `http://www.opnet.com/solutions/network_rd/modeler.html`, 2012. Date Retrieved: 16/01/2012.

[147] OPNET Technologies, Inc. OPNET Modeler Wireless Suite. `http://www.opnet.com/solutions/network_rd/modeler_wireless.html`, 2012. Date Retrieved: 16/01/2012.

[148] S. PalChaudhuri. The Random Trip Mobility Model. `http://ica1www.epfl.ch/RandomTrip/`, November 2004. Retrieved: 17/10/2010.

[149] Emmanouil A. Panaousis, Tipu Arvind Ramrekha, Grant P. Millar, and Christos Politis. Adaptive and Secure Routing Protocol for Emergency Mobile Ad Hoc Networks. *International Journal of Wireless & Mobile Networks*, 2(2):62–78, 2010.

[150] Panagiotis Papadimitratos and Zygmunt J. Haas. Secure Data Transmission in Mobile Ad Hoc Networks. In *WiSe '03: Proceedings of the 2nd ACM workshop on Wireless security*, pages 41–50, New York, NY, USA, 2003. ACM.

[151] Panagiotis Papadimitratos and Zygmunt J. Haas. Secure Data Communication in Mobile Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, 24(2):343–356, 2006.

[152] V. Park and S. Corson. Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification. Internet Draft `http://www.ietf.org/proceedings/53/I-D/draft-ietf-manet-tora-spec-04.txt`, July 2001.

[153] Vincent D. Park and M. Scott Corson. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In *Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution*, INFOCOM '97, pages 1405–, Washington, DC, USA, 1997. IEEE Computer Society.

[154] K. Pawlikowski, H.-D.J. Jeong, and J.-S.R. Lee. On Credibility of Simulation Studies of Telecommunication Networks. *Communications Magazine, IEEE*, 40(1):132–139, January 2002.

[155] Marc R. Pearlman, Zygmunt J. Haas, Peter Sholander, and Siamak S. Tabrizi. On the Impact of Alternate Path Routing for Load Balancing in Mobile Ad Hoc Networks. In *MobiHoc '00: Proceedings of the 1st ACM International Symposium on Mobile ad hoc Networking & Computing*, pages 3–10, Piscataway, NJ, USA, 2000. IEEE Press.

[156] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental) `http://www.ietf.org/rfc/rfc3561.txt`, July 2003.

[157] C. Perkins and E. M. Belding-Royer. Quality of Service for Ad hoc On-Demand Distance Vector Routing. Internet Draft, Novemeber 2001.

[158] C. Perkins and I. Chakeres. Dynamic MANET On-demand (AODVv2) Routing. Intended status: Standards Track, March 2012.

[159] Charles Perkins and Elizabeth Royer. Ad-hoc on-demand distance vector routing. In *In Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, 1997.

[160] L.F. Perrone and Yougu Yuan. Modeling and Simulation Best Practices for Wireless Ad Hoc Networks. In *Simulation Conference, 2003. Proceedings of the 2003 Winter*, volume 1, pages 685–693, dec. 2003.

[161] L. L. Peterson and B.S. Davie. *Computer Networks A Systems Approach.* Morgan Kaufman, 4 edition, 2007.

[162] Gao Qian. An Improved On-demand QoS-Based Routing Protocol for Mobile Ad hoc Networks. In *Wireless Networks and Information Systems, 2009. WNIS '09. Int'l Conf. on*, pages 175–178, Dec. 2009.

[163] Michael O. Rabin. Efficient dispersal of information for security, load balancing, and fault tolerance. *J. ACM*, 36(2):335–348, April 1989.

[164] K. Ramakrishnan, S. Floyd, and D. Black. The Addition of Explicit Congestion Notification (ECN) to IP. RFC 3168 (Standards Track) `http://tools.ietf.org/html/rfc3168`, September 2001.

[165] K. K. Ramakrishnan and Raj Jain. A Binary Feedback Scheme for Congestion Avoidance in Computer Networks. *ACM Transactions on Computer Systems*, 8(2):158–181, May 1990.

[166] Mubashir Rehmani. Basic Rate and Data Rate. `http://mailman.isi.edu/pipermail/ns-users/2011-March/069712.html`, March 2001. Retrieved: 23/04/2012.

[167] Henning Rogge. [manet] Replacing MAANET with a simple solution. `http://www.ietf.org/mail-archive/web/manet/current/msg12657.html`, April 2012. Date Retrieved: 20/04/2012. MANET Mailing List.

[168] N. Sarma and S. Nandi. A Route Stability Based Multipath QoS Routing (SMQR) in MANETs. In *Emerging Trends in Engineering and Technology, 2008. ICETET '08. First International Conference on*, pages 193–198, july 2008.

[169] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms and Source Code in C*. John Wiley & Sons, New York, 2nd edition, 1996.

[170] Bruce Schneier. *Secrets & Lies: Digital Security in a Networked World*. John Wiley & Sons, 2000.

[171] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 2550 (Proposed Standard), July 2003.

[172] S. Sedaghat, F. Adibniya, and V. Derhami. A Secure Mechanism for QoS Routing in Mobile Ad Hoc Networks with QoS Requirements Consideration. In *Computational Intelligence and Communication Networks (CICN), 2010 International Conference on*, pages 320–324, Nov. 2010.

[173] Ruxandra M. Serbanescu. Lecture 20: Chromatic Dispersion Total Internal Reflection Optical Fibers; Applications . University Lecture `http://www.physics.utoronto.ca/~sandra/PHY238Y/Lectures/Lecture20_04.pdf`, 2004.

[174] Chen ShanShan and A. J. Kassler. Extending SWAN to Provide QoS for MANETs Connected to the Internet. In *Wireless Communication Systems, 2005. 2nd International Symposium on*, pages 503–507, December 2005.

[175] ZhengMing Shen and J. P. Thomas. Security and QoS Self-Optimization in Mobile Ad Hoc Networks. *Mobile Computing, IEEE Transactions on*, 7(9):1138–1151, July 2008.

[176] Y. Sheng, H. Cruickshank, A. Pragad, P. Pangalos, and A. Aghvami. An Integrated QoS, Security and Mobility Framework for Delivering Ubiquitous Services Across All IP-based Networks. In *IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–5, 2008.

[177] Nirmala Shenoy, Yin Pan, and Vishal G. Reddy. Quality of Service in Internet MANETs. In *IEEE 16th Internation Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1823–1829. IEEE, 2005.

[178] D. M. Shila and T. Anjali. Defending selective forwarding attacks in WMNs. In *Electro/Information Technology, 2008. EIT 2008. IEEE International Conference on*, pages 96–101, May 2008.

[179] Yan T. Shu, Guang H. Wang, Lei Wang, Oliver W. W. Yang, and Yong J. Fan. Provisioning QoS Guarantee by Multipath Routing and Reservation in Ad Hoc Networks. *J. Comput. Sci. Technol.*, 19:128–137, March 2004.

[180] Zhang Shuguang, Jin Yuehui, Cui Yidong, and Que Xirong. Research on Improved INSIGNIA Based on Network Measurement. In *Broadband Network Multimedia Technology, 2009. IC-BNMT '09. 2nd IEEE International Conference on*, pages 330–334, October 2009.

[181] R. Sivakami and G.M.K. Nawaz. Reliable Communication for MANETS in Military Through Identification and Removal of Byzantine Faults. In *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, volume 5, pages 377–381, April 2011.

[182] Chengqi Song and Qian Zhang. Protocols for Stimulating Packet Forwarding in Wireless Ad Hoc Networks. *Wireless Communications, IEEE*, 17(5):50–55, october 2010.

[183] Evdoxia Spyropoulou, Timothy Levin, and Cynthia Irvine. Calculating Costs for Quality of Security Service. In *In 15th Computer Security Applications Conference*, pages 334–343, 2000.

[184] William Stallings. *Cryptography and Network Security Principles and Practice*. Pearson Education Inc., Upper Saddle River, NY, fifth edition, 2011.

[185] James P.G. Sterbenz, David Hutchison, Egemen K. Çetinkaya, Abdul Jabbar, Justin P. Rohrer, Marcus Schöller, and Paul Smith. Resilience and Survivability in Communication Networks: Strategies, Principles, and Survey of Disciplines. *Computer Networks*, 54(8):1245–1265, 2010. Resilient and Survivable Networks.

[186] Taleb, Tarik and Hadjadj-Aoul, Yassine. QoS2: A Framework for Integrating Quality of Security with Quality of Service. *Security and Communication Networks*, pages n/a–n/a, 2012.

[187] Keren Tan. *Large-scale Wireless Local-area Network Measurement and Privacy Analysis*. PhD thesis, Dartmouth College, Hanover, New Hampshire, August 2011. Dartmouth Computer Science Technical Report TR2011-703.

[188] The Federal Response to Hurricane Kartina—Lessons Learned. `http://georgewbush-whitehouse.archives.gov/reports/katrina-lessons-learned.pdf`, February 2006. Date Retrieved: 02/06/2010.

[189] A. Tsirigos and Z. J. Haas. Multipath Routing in the Presence of Frequent Topological Changes. *Communications Magazine, IEEE*, 39(11):132–138, 2001.

[190] A. Tsirigos and Z.J. Haas. Analysis of Multipath Routing—Part I: The Effect on the Packet Delivery Ratio. *Wireless Communications, IEEE Transactions on*, 3(1):138–146, jan. 2004.

[191] K. K. Vadde and V. R. Syrotiuk. Quantifying Factors Affecting Quality of Service in Mobile Ad Hoc Networks. *Simulation*, 81(8):547–560, 2005.

[192] K.K. Vadde and V.R. Syrotiuk. Factor Interaction on Service Delivery in Mobile Ad Noc Networks. *Selected Areas in Communications, IEEE Journal on*, 22(7):1335–1346, September 2004.

[193] Binod Vaidya, Mieso K. Denko, and Joel J. P. C. Rodrigues. Secure Framework for Voice Transmission over Multipath Wireless ad-hoc Network. In *GLOBECOM'09: Proceedings of the 28th IEEE conference on Global telecommunications*, pages 4299–4304, Piscataway, NJ, USA, 2009. IEEE Press.

[194] Alvin Valera, Winston Seah, and S. V. Rao. Cooperative Packet Caching and Shortest Multipath Routing In Mobile Ad hoc Networks. In *in Proceedings of IEEE INFOCOM, March-April 2003*, pages 260–269, 2003.

[195] Mallapur Veerayya, Vishal Sharma, and Abhay Karandikar. SQ-AODV: A novel energy-aware stability-based routing protocol for enhanced QoS in wireless ad-hoc networks. In *Military Communications Conference, 2008. MILCOM 2008. IEEE*, pages 1–7, November 2008.

[196] S. Venkatasubramanian and N. P. Gopalan. A Quality of Service Architecture for Resource Provisioning and Rate Control in Mobile Ad Hoc Networks. *International Journal of Ad hoc, Sensor, & Ubiquitous Computing (IJASUC)*, 1(3):106–120, September 2010.

[197] S. Waharte and R. Boutaba. Totally Disjoint Multipath Routing in Multihop Wireless Networks. In *Communications, 2006. ICC '06. IEEE International Conference on*, volume 12, pages 5576 –5581, june 2006.

[198] Guanghong Wang, Yantai Shu, Yongjie Fan, Lei Wang, and O.W.W. Yang. Multipath Bandwidth Splitting Reservation in Ad Hoc Networks. In *Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th IEEE Proceedings on*, volume 3, pages 2621–2625, sept. 2003.

[199] Lei Wang, Seungho Jang, and Tae Y. Lee. Redundant Source Routing for Real-Time Services in Ad Hoc Networks. *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, 0:7–87, 2005.

[200] Xiaoyun Wang, Yiqun Yin, and Hongbo Yu. Finding Collisions in the Full SHA-1. In Victor Shoup, editor, *Advances in Cryptology CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 17–36. Springer Berlin / Heidelberg, 2005.

[201] E. Weingartner, H. vom Lehn, and K. Wehrle. A Performance Comparison of Recent Network Simulators. In *Communications, 2009. ICC '09. IEEE International Conference on*, pages 1–5, june 2009.

[202] Matthias Wellens. Distance vs. Data Rate in 802.11b. `http://mailman.isi.edu/pipermail/ns-users/2004-January/038440.html`, January 2004. Retrieved: 23/04/2012.

[203] M. Westerlund, I. Johansson, P. Perkins, C. O'Hanlon, and K. Carlberg. Explicit Congestion Notification (ECN) for RTP over UDP draft-ietf-avtcore-ecn-for-rtp-08. Internet Draft `http://tools.ietf.org/html/draft-ietf-avtcore-ecn-for-rtp-08`, May 2012.

[204] N. Wisitpongphan and O.K. Tonguz. Disjoint Multipath Source Routing in Ad Hoc Networks: Transport Capacity. In *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, volume 4, pages 2207–2211, oct. 2003.

[205] K.D. Wong, T.J. Kwon, and V. Varma. Towards Commercialization of Ad Hoc Networks. In *Networking, Sensing and Control, 2004 IEEE International Conference on*, volume 2, pages 1207–1211, 2004.

[206] Kui Wu and Janelle Harms. On-Demand Multipath Routing for Mobile ad hoc Networks. In *Networks EPMCC 2001, Vienna, 20th-22nd February 2001*, volume 4, pages 1–7, 2001.

[207] ZhengYou Xia and YunAn Hu. Extending RSVP for Quality of Security Service. *IEEE Internet Computing*, 10(2):51–57, 2006.

[208] B. Xiao, B. Yu, and C. Gao. CHEMAS: Identify Suspect Nodes in Selective Forwarding Attacks. *Journal of Parallel and Distributed Computing*, 67(11):1218–1230, November 2007.

[209] Hannan Xiao, Winston K. G. Seah, Anthony Lo, and Kee C. Chua. A Flexible Quality of Service Model for Mobile Ad-Hoc Networks. In *IEEE VTC2000-spring*, volume 1, pages 445–449, 2000.

[210] Yanping Xiao, Chuang Lin, Yixin Jiang, Xiaowen Chu, and Shengling Wang. Risk-Aware QoP/QoS Optimization for Multimedia Applications in Wireless Networks. In Novella Bartolini, Sotiris Nikoletseas, Prasun Sinha, Valeria Cardellini, and Anirban Mahanti, editors, *Quality of Service in Heterogeneous Networks*, volume 22, chapter 3, pages 34–51. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[211] Kaixin Xu, Ken Tang, R. Bagrodia, M. Gerla, and M. Bereschinsky. Adaptive Bandwidth Management and QoS Provisioning in Large Scale ad hoc Networks. In *Military Communications Conference, 2003. MILCOM 2003. IEEE*, volume 2, pages 1018–1023, October 2003.

[212] Jiaiibo Xue, Patrick Stuedi, and Gustavo Alonso. ASAP: An Adaptive QoS Protocol for Mobile Ad Hoc Networks. In *In IEEE Intl. Symposium on Personal Indoor and Mobile Radio Communications (PIMRC2003)*, pages 2616–2620, September 2003.

[213] Qi Xue and Aura Ganz. Ad hoc QoS On-demand Routing (AQOR) in Mobile ad hoc Networks. *J. Parallel Distrib. Comput.*, 63:154–165, February 2003.

[214] Yan Zhen and Mu-qing Wu and Da-peng Wu and Qin-juan Zhang and Chun-xiu Xu. Toward Path Reliability by Using Adaptive Multi-Path Routing Mechanism for Multimedia Service in Mobile Ad-Hoc Network. *The Journal of China Universities of Posts and Telecommunications*, 17(1):93–100, 2010.

[215] Hao Yang, Haiyun Luo, Fan Ye, Songwu Lu, and Lixia Zhang. Security in Mobile Ad Hoc Networks: Challenges and Solutions. *Wireless Communications, IEEE*, 11(1):38–47, August 2004.

[216] Tan Yang, Shuguang Zhang, Yidong Cui, and Yuehui Jin. Improved INSIGNIA Based on Network Measurement. Internet Draft `http://tools.ietf.org/html/draft-yang-manet-improvedinsignia-00`, July 2009.

[217] Po-Wah Yau and C.J. Mitchell. Reputation Methods for Routing Security for Mobile Ad Hoc Networks. In *Mobile Future and Symposium on Trends in Communications, 2003. SympoTIC '03. Joint First Workshop on*, pages 130–137, oct. 2003.

[218] Zhenqiang Ye, Srikanth V. Krishnamurthy, and Satish K. Tripathi. A Framework for Reliable Routing in Mobile Ad Hoc Networks. In *IEEE INFOCOM*, pages 270–280, 2003.

[219] Younghwan Yoo and Dharma P. Agrawal. Why Does It Pay to Be Selfish in a MANET? *Wireless Communications, IEEE*, 13(6):87–97, dec. 2006.

[220] J. Yoon, M. Liu, and B. Noble. Random Waypoint Considered Harmful. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1312–1321. IEEE, 2003.

[221] Qiao Yu-lan, Su Bing, Qiao Xiao-ping, Wang Hong-yuan, and Lu Jie-ru. An Improved Service Scheme Combinated SWAN with QoS-MSR in Ad-Hoc Networks. In *Logistics Systems and Intelligent Management, 2010 International Conference on*, volume 1, pages 465–468, January 2010.

[222] Manel G. Zapata and N. Asokan. Securing Ad Hoc Routing Protocols. In *WiSE '02: Proceedings of the 1st ACM Workshop on Wireless Security*, pages 1–10, New York, NY, USA, 2002. ACM.

[223] Yan Zhang, Nirwan Ansari, and Hiroshi Tsunoda. Wireless Telemedicine Services Over Integrated IEEE 802.11/WLAN and IEEE 802.16/WiMAX Networks. *IEEE Wireless Communications*, 17(1):30–36, February 2010.

"It was curious that he seemed not merely to have lost the power of expressing himself, but even to have forgotten what it was that he had originally intended to say ... The actual writing would be easy. All he had to do was to transfer to paper the interminable restless monologue that had been running inside his head, literally for years. At this moment, however, even the monologue had dried up ... He was conscious of nothing except the blankness of the page in front of him ... Suddenly he began writing in sheer panic, only imperfectly aware of what he was setting down."
—George Orwell, *1984*

"It is most gratifying that your enthusiasm for our planet continues unabated. As a token of our appreciation, we hope you will enjoy the two thermonuclear missiles we've just sent to converge with your craft. To ensure ongoing quality of service, your death may be monitored for training purposes. Thank you."
—Douglas Adams, *The Hitchhiker's Guide to the Galaxy*

"I realized that procrastination can rule our lives, yet not provide us with any arguments in its defence."
—Françoise Sagan, *Bonjour Tristesse*

"That's my role. To be ridiculous. That's the part I get to play in this little farce aspiring to tragedy."
—Jay McInerney, *Model Behaviour*

"You common cry of curs, whose breath I hate
As reek o'th'rotten fens, whose loves I prize
As the dead carcasses of unburied men
That do corrupt my air—I banish you.
And here remain with your uncertainty!
Let every feeble rumour shake your hearts;
Your enemies, with nodding of their plumes,
Fan you into despair! Have the power still
To banish your defenders, till at length
Your ignorance—which finds not till it feels,
Making but reservation of yourselves
Still your own foes—deliver you
As most abated captives to some nation
That won you without blows! Despising
For you the city, thus I turn my back.
There is a world elsewhere."

—Shakespeare, *Coriolanus*, III.3