Control and Management Plane Helper Functions for Information-Centric Networks

Von der Fakultät für Elektrotechnik und Informationstechnik der Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Ingenieurin

Borislava Gajic
aus Brcko, Bosnien und Herzegowina

Berichter: Univ.-Prof. Dr. Petri Mähönen
Univ.-Prof. Dr. George Xylomenos

Tag der mündlichen Prüfung: 1 Oktober 2013

Diese Dissertation ist auf den Internetseiten der Hochschulbibliothek online verfügbar.

ABSTRACT

The rapid evolution of the Internet over past decades has enabled the global communication which has fundamentally changed our lives. As an outcome, millions of users are constantly exchanging an enormous amount of information. Such a massive distribution of information and fast increase of traffic loads have posed a considerable challenge to the functionality of the current Internet. Furthermore, apart from increased communication demands and higher loads of exchanged data, the users' interests have significantly changed. Users primarily focus on finding or sharing specific information, regardless of the particular end host providing it. Thus, the current network architecture appears to be suboptimal, and perhaps even inflexible to follow the changes in communication patterns.

In order to support the emerging user requirements, different overlay data delivery structures have been proposed and implemented. Despite the perceived simplicity of such overlay approaches there are many issues, e.g., efficient resource allocation, which are considered to be suboptimal and not desirable as an architectural solution for information exchange over the Internet. Thus, it is believed that apart from overlay techniques additional powerful mechanisms are still necessary for exploiting the network resources effectively.

This thesis identifies information-centric communication as a new concept for efficient exploitation of network resources, solving the current Internet problems, and satisfying the emerging user demands. We carefully study the potential of information-centric communication based on the numerous scenarios found in the literature and adopt the most promising solutions as the framework for building our work upon.

In this thesis we focus on developing the mechanism for native support of information-centric communication at all networking layers. Having flexibility as one of the most important requirements, we aim at building a set of information-centric helper functions, each being responsible for a small portion of network functionality. A joint operation of such helper functions leads to significant improvement of network performance without increasing the complexity of network operation.

We design, implement and carefully evaluate a variety of helper functions entirely relying on information-centric concepts, carrying out the experiments through both extensive simulations and real testbed environments. We particularly focus on helper functions for the control and management plane. We design and build the modules responsible for topology discovery and creation, topology maintenance and mobility management, extending our helper functionality set with different ancillary mechanisms for optimization of data transfer, such as network coding.

We demonstrate the advantage of our approach with respect to simplicity and flexibility of the implementation, as well as the achieved resource utilization and network performance improvements. Particular attention has been paid to the extensibility and optimization of the proposed system of helper functions. Furthermore, its native integration with the fine-grained lower level network operations has been investigated. Finally, we highlight the main remaining challenges of our approach, such as deeper examination of scalability issues and the potential of wide-scale deployment.

KURZFASSUNG

Die enorme Verbreitung des Internets ist Grundlage weltweiter Kommunikation und hat unsere Lebensweise nachhaltig verändert. Millionen von Nutzern tauschen heutzutage enorme Datenmengen miteinander aus. Diese massive Verteilung von Informationen stellt eine besondere Herausforderung für die zukünftige Funktionsfähigkeit der Internetinfrastruktur dar: Jenseits des wachsenden Anspruchs nach ständiger Verfügbarkeit und hohen Datenraten haben sich die Interessen der Nutzer maßgeblich gewandelt. Heutige Internetnutzer konzentrieren sich zunehmend auf das Auffinden und Verteilen gewünschter Informationen, ungeachtet der Frage auf welchem Netzwerkknoten diese angeboten werden. Es lässt sich folgern, dass zur Befriedigung des geänderten Kommunikationsverhalten seiner Nutzer die momentane Architektur des Internets wenig geeignet, wenn nicht sogar ungeeignet ist.

Um sich den gewandelten Nutzeranforderungen anzupassen, wurden verschiedene Overlaymechanismen zur Datenverteilung vorgeschlagen und implementiert. Obwohl zuerst trivial erscheinend ist ein solcher Overlayansatz mit vielen verschiedenen Problemen behaftet, wie z.B. dem der effizienten Ressourcenallokation, welche als suboptimal und ungeeignet für den Informationsaustausch im Internet betrachtet wird. Der Overlayansatz kann daher kaum als optimale Lösung angesehen werden. Zusätzlich zur Overlaystruktur selbst scheinen weiterhin umfangreiche Maßnahmen nötig zu sein um Netzwerkressourcen effizient nutzen zu können.

Diese Arbeit identifiziert informationszentrische Kommunikation als neues Konzept für eine effiziente Nutzung von Netzwerkressourcen, welches die gegenwärtigen Probleme des Internets lösen und zükünftige Nutzeranforderungen besser befriedigen kann. Wir untersuchen in dieser Arbeit das Potenzial der informationszentrischer Kommunikation anhand verschiedener Szenarien aus der einschlägigen Literatur und adaptieren die vielversprechendsten Lösungen als Plattform auf welcher wir unsere eigene Arbeit umsetzen.

Wir haben uns auf die Entwicklung von Mechanismen zur nativen Unterstützung von informationszentrischer Kommunikation in allen Netzwerkschichten konzentriert. Ein Hauptaugenmerk liegt dabei auf einer Flexibilisierung der vorgeschlagenen Ansätze, welche durch Entwicklung atomarer informationszentrischer Hilfsfunktionen für die unterschiedlichen Aufgabenbereiche in der Netzwerkfunktionalität erreicht wird. Eine Verbindung solcher Hilfsfunktionen führt zu signifikanten Verbesserungen der Netzwerkleistung ohne die Komplexität des Netzwerkbetriebs zu erhöhen.

Im Rahmen dieser Arbeit wurden eine Vielzahl von Hilfsfunktionen entworfen, implementiert, und untersucht, wobei alle diese Funktionen ausschließlich auf informationszentrischen Konzepten basieren. Umfangreiche Experimente mittels Simulationen und innerhalb echter Testumgebungen wurden durchgeführt. Wir konzentrieren uns hierbei insbesondere auf Funktionen in der Management- und der Steuerebene. Zusatzmodule zur Topologieerkennung, -aufbau, und -verwaltung und zum Mobilitätsmanagement wurden entwickelt um unsere Hilfsfunktionen um verschiedene zusätzliche Mechanismen zur Optimierung des Datentransfers, wie zum Beispiel Netzwerkkodierung, zu erweitern.

Wir demonstrieren den Vorteil unseres Ansatzes in Bezug auf seine Einfachheit und die Flexibilität der Implementierung, und zeigen die erreichbare Netzwerkressourcenauslastung und -leistungsverbesserung. Ein besonderer Schwerpunkt wurde hierbei auf Erweiterbarkeit und Optimierung des vorgeschlagenen Hilfsfunktionensystems gelegt. Darüber hinaus wurde die Möglichkeit einer nativen Integration in die detailreichen Aufgaben des Betriebs von Netzwerken in den unteren Netzwerkschichten untersucht. Zum Abschluss dieser Arbeit heben wir die verbleibenden Herausforderungen für unseren Ansatz hervor, wie zum Beispiel eine umfassendere Untersuchung zu Skalierungseigenschaften und das Potenzial für einen großflächigen Einsatz.

Acknowledgments

This thesis has been an invaluable learning process that has helped me improving as a researcher and as a person. I am deeply thankful to all the people that have helped me during this important part of my life.

Above all, I would like to thank my supervisor, Professor Petri Mähönen, for his inexhaustible support throughout all my thesis work. His continuous help, encouragement and advices have taught me how to concentrate on the essence of a matter, which is an important lesson in research, work and in life, as well. Being always open for discussions, sharing his immense knowledge and practical experience he has created a motivating working environment. It has been an honor and a pleasure to work with him.

It has been also a pleasure to work with all my colleagues at the Institute for Networked Systems at RWTH Aachen University. Being always in the good mood and willing to help they have been a rich source of feedback and ideas.

I am also thankful to RWTH Aachen University and the European Commission, who funded the presented work through various EU-projects and the UMIC research cluster. This support has been crucial for carrying out the work throughout my thesis. Through such projects I have also gained invaluable experience while working together with most important industry leaders and highly ranked academic institutions.

Finally, I would like to thank my husband Flavio for his immense love and support. Without his continuous encouragement and patience finalizing this thesis would have been impossible. This dissertation is dedicated to him and our beloved daughter Sofia.

Contents

1

INTRODUCTION

Over only a couple of decades the Internet has become a global network and a critical part of developed countries infrastructure. Since its beginning the Internet has been developing with an incredible speed, drastically modifying our lives by facilitating global communication and business. Such a revolution in managing the information has had a tremendous impact on all spheres of our society, resulting even in the change of our habits and way of living. Today the number of Internet users is exceeding 2 billion [1], interconnected through thousands of autonomous systems [2]. Moreover, further significant growth of the Internet size is expected, mainly thanks to the widespread deployment of high-bandwidth links and proliferation of powerful mobile devices, e.g. laptops and smartphones.

However, despite its openness and flexibility the Internet has faced a variety of problems during its evolution. The new user requirements have posed considerable challenges to the original Internet design. Due to the need of fast adaptation to emerged requirements, a portion of the architectural problems were circumvented or remained only partially solved. Some of such incomplete or incremental solutions have posed additional limitations on the Internet growth and development of new applications and services [3].

1.1 PROBLEM STATEMENT AND MOTIVATION

The main guiding objective of the current Internet design was the end-to-end principle. The model was intended mainly to serve host-to-host applications supporting different extensions. The original Internet architecture was based on coupling of host's location in the network with its identity, unified in host's network address. According to such model, an arbitrary end host needs to know the network address of the desired destination host in order to establish a communication with it. The network was designed in a way to provide the minimum requirements for allowing the communication between end hosts, whereas the application specific functionality was provided using the intelligence at the edge nodes [4]. In other words, the core network's role is to provide the best effort service in transferring the data from the source to the destination.

Furthermore, the current Internet is firmly grounded on sender-oriented way of communication, i.e. the network delivers the data to the destination specified by the sender. As an outcome of such design, the Internet's core functionality remained rather simple. On the other hand, this approach introduced various problems by retaining a large space for user misbehavior, leaving the receiver without direct control over the traffic flow. Additionally, the users can have significantly different interests, some of which can even be opposed to each other or even appear as hostile. Thus, the attacks targeted to the specific end-point in the form of undesirable traffic, e.g, denial of service attacks [5], are rather simple to achieve. The later need to provide user readable (and memorizable) addresses, e.g. for web-browsing, has lead to the situation where such names need to be mapped with end-addresses. However, such mechanisms, although providing a solution, are generating new overlays and security problems into the architecture.

Moreover, the original idea of the Internet as an interconnection of mainly fixed computers has evolved over time. Today, the Internet represents a collection of a large number of often mobile users. Such changes have introduced numerous challenges in handling the mobility [6]. Particularly, the tight coupling of identity and location has created a significant obstacle to the seamless mobility support. The moving hosts change their position in the network and, thus, their network addresses alter. Nevertheless, the host identity stays unmodified. Under such circumstances the ongoing communication sessions fail, since the underlying architecture delivers the data to the destination address specified by the sender at the beginning of the session, assuming that the network address did not change.

Therefore, the Internet Protocol (IP) is commonly identified as one of the most problematic parts of Internet's original design. There are multiple reasons for such general attitude with respect to the IP. Due to the unexpectedly fast growth of the number of Internet connected devices the address space of first IP version (version 4) turned out to be insufficient. Therefore, assigning a globally unique address to every individual network interface was not possible. This stimulated the work on enlarging available address space through various mechanisms, such as introducing NAT (Network Address Translation) mechanisms and design of a new IPv6 protocol. Nevertheless, the main problem of IP addressing is in its original design and usage. The IP addresses imply not only the location of the user, but it serves as the user's identity, as well. This is the main reason that the native support of mobility and many other mechanisms is hard to achieve and this problem has not been addressed even with a new mechanisms to increase address space. Furthermore, with the increase of the number of Internet users the routing scalability has become a significant concern, since efficient handling of current number of users and corresponding routing tables has turned into a challenging task.

Additionally, over the time user demands and behavior have changed radically. New Internet applications are not anymore well served by the end-point-oriented communications paradigm. Due to the wide diffusion of information throughout the network the new applications are expected to gather the data from the disperse set of sources and to deliver it to the users according to their expressed interests. Instead of particularly targeting communications towards desired hosts, users are often more interested in specifying the data objects they are willing to retrieve, leading into content-centric communication [7, 8]. In order to satisfy users' requirements the applications need to manage a large number of data sources and handle wide variety of user demands that also frequently change.

In general, as the Internet evolved over time, the user demands have changed beyond the initial assumptions. As an outcome, the underlying Internet architecture no longer matches the current communication requirements, and it very often fails to quickly and easily adapt to the emerging user needs.

1.2 SCOPE AND FOCUS OF THE THESIS

Internet's vulnerability to unwanted data flow and tight connection between address and location are one of the fundamental concerns of the existing model. Additionally, present user requirements incline more towards the interest in data and not the data source itself. The increased number of services with a major focus on information, e.g. WWW, P2P networks, and RSS feeds demonstrate the occurred shift in the user preferences and the use of network resources.

This has stimulated research directed to modeling the Internet architecture compliant with data-oriented, i.e. *information-centric* (ICN) concepts [9, 10, 11, 12]. Within such research scope the *publish-subscribe* communication model [13] appears as one of the most promising mechanisms for supporting the idea of data-oriented networking. It has been proposed as a powerful solution to the problems of locating and retrieving the data. Due to its anonymous and loosely coupled nature it facilitates the data transfer within the highly dynamic, large-scale networks. Although the publish-subscribe concept is nearly 20 years old [14] it has attracted a significant attention only during the last few years as a flexible communication paradigm supporting the data-oriented principles.

The publish-subscribe communication model offers a number of advantages. One of the most interesting properties of the publish-subscribe pattern is that the explicit addressing between publishers and subscribers is not needed. Instead, the subscribers express their interest in receiving some particular data, whereas the publisher announces its willingness to provide that information. Such loose coupling between the communication ends allows easy extensibility of the network, since the new users can simply join or leave the network.

Additionally, the publish-subscribe communication is asynchronous, therefore the publishers and subscribers do not need to be available at the same time, and there is no strict order in issuing publications and subscriptions to the same data. Even if the subscription is issued before the corresponding publication, the data will be automatically transfered after both communication parties become available. Finally, the data transfer is not initiated before there is a clear interest for participating expressed from both communication sides. Such principles have been already utilized in numerous notification service models [15, 16, 17, 18, 19].

However, a large scale deployment of publish-subscribe communication model introduces certain concerns. Apart from existing doubts if the new architecture will uphold the performance in some specific aspects, e.g. average latency, major problem of applying such system in a clean slate fashion remains expensive and doubtful for an entire Internet structure. Nevertheless, most of the proposals are targeting redesign of just specific parts of the architecture considered as the weakest points. On the other hand, various data management techniques offer significant optimization services and improve performance of publish-subscribe systems without introducing large additional costs, e.g. caching content locally or applying coding techniques.

In this thesis we focus on building a network model that fully utilizes the publish-subscribe concept. It has been shown that the traditional publish-subscribe systems designed as overlays of the present Internet suffer from a number of issues, e.g. flexibility, scalability, etc. Therefore, we aim at modeling the network where the publish-subscribe pattern is not only used on the overlay level, but it is rather an inherent part of the fundamental architecture[1]. In particular, our interest is focused on building the lower layer information-centric network functionality as a set of independent processes, i.e. *helper functions*, that interact with a common aim of facilitating the data transfer from publishers to all interested subscribers.

In such a way the system is able to offer a high flexibility to the network operations on all protocol layers, allowing them to accurately describe their interest with powerful and expressive subscriptions. One of the biggest challenges of such an approach are implementation objectives, complexity, and efficiency, which are examined and discussed in details throughout this thesis. Furthermore, we aim at complete deployment of our architectural solutions in order to gain practical experience on the quantitative performance in real networks. The acquired knowledge from the realistic experiments is fed back to the design procedure for enhancing the proposed solution.

Due to the event-based and asynchronous character of constantly increasing number of content-oriented applications, building them around publish-subscribe pattern appears as natural approach. However, since the wider deployment of publish-subscribe-related systems remains as one of the most challenging tasks, we aim at addressing this issue with respect to our proposed solution.

[1]This does not necessary exclude the possibility to support both a new publish-subscribe architecture and the present day end-to-end packet oriented Internet concurrently.

Therefore, we examine in details the deployment requirements for building the publish-subscribe system especially focusing on designing and building the set of so called lower level helper functions for publish-subscribe framework. We investigate the impact of the helper functions to the overall system performance, with respect to the deployment complexity and costs.

1.3 THESIS CONTRIBUTIONS

Based on the discussed problems of the Internet architecture and emerging end-user changes we advocate the use of publish-subscribe, information-centric model in order to optimize the network functionality through better utilization of network resources. Furthermore, such approach appears to be more suitable for emerging new user requirements, allowing more flexibility in retrieving the desired content. Having a better insight on what has been transmitted over the network, in addition to the information about the communication end-points, the information-centric system is able to more efficiently use the network resources and deliver the data packets. Moreover, the mobility support is easier to achieve due to the split between the identity and the location of the network hosts.

In this thesis we present the results of a detailed study on the publish-subscribe helper functions on the lower layers of network protocol stack. Instead of applying the publish-subscribe paradigm on the application level for data retrieval, as already implemented in many Internet services, we aim at integrating such mechanisms directly into the control and management plane.

The work carried out throughout this thesis ranges from a general description and evaluation of the information-centric networking and publish-subscribe communication model to a design, implementation and a deep examination of the various helper functionalities that facilitate the operation of the information-centric network. Furthermore, we present and discuss all the issues hidden behind the realization of such an information-centric architecture with intrinsic support for a publish-subscribe model.

Aiming at a network-wide deployment of the information-centric concept we discuss the scalability of our helper functions. Being strongly based on the publish-subscribe model and due to the simplicity of their implementation, our helper functions are applicable to the wide-scale network setup. Another design objective that we have followed in developing our helper functions is to simplify and optimize the network operation. The underlying network architecture enables such approach by focusing on the desired data piece rather than the end-hosts as consumers or providers of the data.

Our goal is to improve user experience while satisfying predetermined requirements. We exploit the advantages of information-centric framework in order to implement the lower level operations so that the overall network flexibility increases. This way the network should become more responsive to changes in the user patterns, needs and requirements. Moreover, the network is offloaded due to the better utilization of network resources.

Another important aspect of the envisioned helper functionality model is the ease of its adoption in the overall information-centric network. Our aim is to facilitate the integration of the proposed model by building a simple, easily extensible and flexible solution. Due to its publish-subscribe basis our helper functionality model represents the native match with any information-centric network architecture, thus, the complexity of the integration and adaptation process is minimized.

The set of helper functionalities that we deploy includes the novel module for handling the topology formation and maintenance as well as the generation of the data delivery routes, i.e. topology manager helper function. Such module is, as far as we are aware of, the first ever proposal and implementation of the topology control mechanism in information-centric networks, entirely relying on publish-subscribe type of communication. The basic topology management functionality is enriched with the range of additional helper modules responsible for optimization of the topology management operation, e.g. by collecting the information about the current link properties, their values, application requirements, etc.

Such helper functionality adds not only to the efficiency of the delivery paths creation, but it optimizes the use of network resources, as well. In order to better address the challenges posed by the mobility we develop the helper functionality responsible for keeping the record of the nodes' movement, using it as an input for predicting the most likely future position of nodes. In such a way the network becomes faster responsive to the sudden changes in the topology by using the predicted information in order to facilitate the handover procedure. Such helper function-assisted handover poses significantly reduced burden to the system.

Another important helper functionality deployed in order to combat the mobility problems is the network attachment functionality. We aim at seamless merging of the naming scheme together with the topology information gathering and dissemination. In such a way the network attachment procedure and the topology discovery are automatically followed by the collection and distribution of the network knowledge. Due to the publish-subscribe nature of the attachment procedure the mobility is intrinsically supported without the need for implementing additional mechanisms for mobility handling. Finally, we discuss the network coding potential as a information-centric helper functionality. The detailed experiments show a certain benefit of using the network coding under special circumstances.

Throughout our design of helper functions and the validation we demonstrate the great potential of information-centric publish-subscribe model in handling the lower level network functionality. The proposed helper functions add to the flexibility and the simplicity of the information-centric network without imposing the burden to the system in terms of the delay and message overhead. We show that the publish-subscribe model can be efficiently used not only for the content retrieval, as already widely utilized in numerous applications, but even for the fine-grained network operations such as presented within this thesis.

1.4 OVERVIEW OF THE THESIS STRUCTURE

In the following we give an outline of the content of this thesis. In Chapter 2 we discuss the background and related work in the area of information-centric networks and publish-subscribe communication model. A high-level specification of information-centric networking idea, as well as the state-of-the-art in this field will be presented. A more specific description of publish-subscribe communication model is outlined in Chapter 3. A special focus has been devoted to the publish-subscribe model in wireless environment. The mobility issues under such circumstances are addressed, as well.

In Chapter 4 we propose a novel model for information-centric intra-domain topology manager entirely relying on publish-subscribe type of communication. We present the design and implementation of our solution followed by the thorough evaluation of its performance. An extension of the original topology management design, especially adopted to dynamic networks is discussed in Chapter 5. We present the set of mobility prediction helper functions which serve as an optimization mechanism for improving the topology management functionality. The detailed analyses of the proposed solution, along with the evaluation through a simulation study is also given.

Chapter 6 presents our original design of the network attachment and the topology discovery model, followed by the implementation description. The validation results obtained from testbed experiments are given as the illustration of the model behavior in real networks. In Chapter 7 we give an overview of the network coding helper functions in information-centric networks. We especially focus on two types of network coding approach and analyze their performance in simulation environment. Finally, we conclude the thesis in the Chapter 8, giving the brief outline of our planned future work and research directions.

2

BACKGROUND

In this chapter we provide an overview of the original design of current Internet with the focus on the control and management plane. Furthermore, we present the main concepts of the content-centric networking. We address the common approach of distinguishing between the identifier and the location, widely adopted by many information-centric architectures. Such a design natively incorporates the solutions for many mobility and security related problems. Moreover, we provide the state-of-the-art analysis on the new Internet architectures, based on the information-centric concepts. Finally, we give an overview of the most relevant new Internet architectures on which we based a large part of our experimentation work.

2.1 EVOLUTION OF ORIGINAL INTERNET ARCHITECTURE

The origins of the Internet are closely related to the development of the packet switched network ARPANET in the late 1960s [20, 21]. The design of such a network was initiated by the Defense Advanced Research Project Agency (DARPA) which was the part of the US Department of Defense. Initially, the main idea of the ARPANET was the interconnection between different existing networks [22]. Evolution of such a network has resulted in the Internet today having over 5 billion connected devices. Based on the predictions the number of Internet connected devices will grow to 15 billion by 2015 [23].

Originally, the Internet was designed according to the end-to-end principle. Such a communication model is well suited for the file transfer type of applications where the traffic flows between the pair of hosts. The most commonly used protocol stack in such architecture, TCP/IP protocol suite [24], has the internetworking layer as a central part of the architecture. The main role of the internetworking layer and its Internet Protocol (IP) [25] is to ensure the global addressing and routability, thus, enabling the communication between arbitrary pair of hosts irrespective of their placement in Internet or the domain they belong. The IP protocol design offered the simple name space of IP addresses that, apart from the identity, denoted also the location of the host. This dual-purpose of IP addressing was not seen as a problem at the time of its development. In the original architecture, we somewhat also overlaid both network address and host address (naming) into the same identifier field.

Surprisingly, the core design of the internetworking layer remained nearly unchanged over a long time: IP has kept its original structure unchanged for over a couple of decades. However, the Internet architecture has been evolving constantly, mainly due to the scaling requirements. The growing Internet introduced the hierarchical organization of the architecture as the most suitable mechanism for coping with the increased number of connected machines. The subnetting [26, 27] and the classless addressing (CIDR) [28] have brought the hierarchy into the addressing space, whereas the emerge of autonomous systems reorganized the routing in the similar fashion.

In order to create a human-readable names and to map them to the IP address space the Domain Name System (DNS) [29] has been developed, introducing the hierarchy into the naming space. The MPLS label switching mechanisms [30] improved the lookup and routing operations by directing the data between the nodes based on the short labels. The problem of exhaustion of available IP addresses has encouraged the emerge of solutions such as NAT (Network Address Translation) [31], which facilitated the address reassignment. Moreover, IPv6 [32] has been introduced in order to extend the address space and simplify the address allocation. However, such solution did not reflect the recent changes in user interests and communication patterns. Furthermore, most of the Internet still being IPv4-based implies that IPv6 has emerged very slowly to the real use.

The Border Gateway Protocol [33] has been established as the standard for inter-domain routing and providing internetworking among autonomous systems. The IP multicast [34, 35] enabled the delivery of IP packets to a group of receivers within one transmission. Furthermore, the Internet evolution has lead to other numerous solutions that have improved the Internet functionality without requiring the fundamental changes in original architecture, e.g., Mobile IP [36], VPN [37], firewalls, security protocols like IPsec [38] or TLS [39]. Recently, a large number of content distribution networks that are designed as an overly, e.g. CDN [40] and P2P [41, 42, 43, 44], have been emerged.

However, extensions of the original design of the Internet have been developed as solutions to particular problems with respect to specific requirements, often not being followed by the deep architectural considerations. In that sense, a large part of the development has been organic, or could be even described being solutions that have been deployed as ”band-aid solutions”. As an outcome, the Internet design as a whole became complex and inflexible. This fact has motivated the research in the area of the networked systems in the direction of rethinking about the alternatives to the Internet design. Furthermore, the development of the global-scale network which would natively provide the flexibility and meet the modern requirements has begun. A a new design of future architecture in a clean slate fashion has been proposed and significant work has been done during the last 5-10 years in the USA and Europe under several larger programs such as GENI [45] and FIRE [46].

2.1.1 Network control and management functions in present Internet

The control plane functions of the Internet are responsible for controlling the connection, dissemination of the connectivity information, and the calculation of the optimal path from a source to a destination. These operations provide the information necessary for the data transmission, thus, they represent the prerequisites for the actual packet forwarding. A central part in the control plane functionality is the generation of the adequate routing paths. The calculation of the routing paths is carried out through the set of algorithms starting from the BGP decision process and DNS resolution down to the OSPF [47] related operations, for example.

The OSPF protocol is responsible for intra-domain routing, i.e. withing a single administrative domain. Each router broadcasts the link-state advertisement (LSA) throughout the administrative domain. Upon receiving such announcements the routers are able to extract the connectivity information and generate the complete routing table by use of Dijkstra's algorithm. The table entries represent the shortest paths from the source to the destination nodes. Additionally, the control plane functions include a variety of signaling protocols.

The management plane functions handle the network monitoring and location of malfunctioning points. The network management and network information gathering is an important aspect since an incomplete or inaccurate understanding of the network hinders the network planning and might accordingly lead to suboptimal performance. Management functions vary widely in nature, ranging from gathering of information about present network resources, to their allocation and configuration, performing upgrades, and dynamical adaptation and optimization of network parameters for achieving maximal performance.

There are various ways of collecting required data. In the current Internet the most prevalent is the manager/agent model used in SNMP (Simple Network Management Protocol) [48] where the role of the manager is to provide interface between human and network management system while agent provides connection between management system and physical device being monitored. SNMP uses small set of messages for information exchange (GET, GET-NEXT, SET, GET-RESPONSE and TRAP). Managed system creates TRAP message as asynchronous notification of occurred event.

In addition to SNMP, other protocols (such as ICMP [49]) are in use to obtain connectivity information along paths between endpoints. The granularity of the information obtained is severely hindered by modern traffic engineering practices and limitations in the underlying protocols. However, by collecting and analyzing the measured data from the network the management plane functions are able to perform configuration optimization, at least in the local sense.

2.1.2 Network attachment in the present Internet

An important aspect of control and management plane functionality is a network attachment operation. The attachment procedure in the present IP-based networks involves a large set of authentication and authorization mechanisms before the final joining of the node to the network. The principles of the Extensive Authentication Protocol EAP [50] has been used in the 802.1X [51] protocols standard for connecting the node to the wireless or wired network. The 802.11X establishes the access control on the port level.

On the other hand, in the 3G/UMTS networks the AKA (Authentication and Key Agreement) challenge-response based protocol is used [52]. The common protocol for communicating over point-to-point links, the point to point protocol PPP [53], uses the CHAP (Challenge Handshake Authentication Protocol) [54] for the authentication. In general, the network attachment procedure comprises the certain steps, e.g. network detection, authentication, address assignment, that are the fundamental parts of the attachment protocol design. The network attachment procedure can be improved by minimizing the signaling overhead or cryptographic computation requirements. The problem of network attachment optimization can be broken down to the optimization of the building blocks of the overall procedure or it can be considered as a whole [55].

2.1.3 The Internet restructuring towards the content-centric networking principle

Following the end-to-end principle, the currently dominant internetworking model was envisioned as a platform for developing mainly end-point-oriented applications, enabling different extensions of such application model. Such original design resulted in the core functionality of the Internet remaining rather simple throughout the evolution of the Internet. On the other hand, this approach faced a number of limitations and introduced various problems by leaving a large space for user misbehavior. The sender-oriented basis of the current Internet yields the imbalanced power structure in the favor of the sender of data.

Therefore, the inherent connection between address and location, as well as the inability of avoiding the unwanted data remains one of the main concerns of the current architecture. Additional challenges relate to the efficient support for mobility, global multicast, and multi-homing. As the Internet size has grown dramatically over the past decade, the scarcity of IP addresses as well as the trustworthiness of the network has became the considerable issue, as well.

Additionally, over the time user demands have changed. Rather than the hosts containing some specific data being the main anchors in the communication, the content itself has become a central communication point. The requirements imposed by the new applications often go beyond the traditional communication paradigm. Therefore, the end-to-end communications model, retaining its original design, does not serve well to the new applications. The inflexibility of the end-to-end communication has become more evident burden with increase of information-centric services and applications, such as the sensor networks. The inadequacy of the internetworking architecture imposed the solutions that required further development of overlays and increase of network complexity. This stimulated the research towards content-centric communication paradigm [8].

The content-centric idea is based on restructuring of the Internet architecture by placing the content in the center of the design considerations. The users interested in particular data are served from the most suitable source regardless of its identity. Following such a principle the multicast and anycast types of communication become dominant. Furthermore, some of the biggest challenges in traditional networks, e.g. mobility, become easier to solve. A range of the already existing Internet applications and overlay solutions utilize the main concepts of the content-centric approach, e.g. P2P, CDN and wireless sensor networks. The information-centric approach is evident in some of the highly used web-services, such as Google and news services, today. In order to handle increasing traffic loads, need for reliability, and enable more information-centric handling of traffic the service providers have been already required to build different overlays that distribute traffic in non-host based approach. In fact, some companies are already providing content based networking services, one example being Akamai [56].

Peer-to-Peer (P2P)

Peer-to-Peer (P2P) [41, 57] represents an overly solution for efficient content distribution. By operating on content names (e.g. URLs and file names) and supporting the multi-source data retrieval the P2P protocols resemble the content-centric communication paradigm. The P2P systems have built extremely scalable services without the need of traditional end-to-end multicast. However, the P2P system relies on the current Internet architecture for name resolution, routing and transport functionality.

Content delivery networks (CDN)

Content delivery networks (CDN) [40, 58] represent a system of hosts collaborating in order to deliver data to receivers in optimal way. The strength of the CDN is in distribution of content over the network by replicating hosts of data and services. This is achieved by deploying large number of edge servers which cache the content in order to improve the data availability. On the user's request the content is retrieved from the closest data source, similar to the content-centric principle.

The CDN model relies on efficient routing mechanism which accomplishes the load balancing while optimizing different parameters, e.g. response time, data load and bandwidth costs. Hence, the user receives content of interests from the "optimal" host storing the data copy. CDNs are usually implemented as an overlay network structure operating on the top of existing IP infrastructure. As a demand for payload heavy, interactive and performance sensitive content grows dramatically, CDNs appear as a right solution for data delivery. An example of successful CDN system is Akamai [56] with more than hundred thousands of high-speed servers distributed around the world. Other major CDN architectures are Limelight [59] and Velocix [60].

Wireless sensor networks (WSN)

Wireless sensor networks (WSN) [61] have recently emerged as a promising way of gathering data especially over hardly reachable areas. The WSN represents a large collection of sensors gathering and disseminating data in order to facilitate surveillance and control of surrounding. Such an idea of small autonomous cooperative devices collecting the data of interest has a wide area of possible applications, e.g. military, environmental and hospital monitoring. The content-centric paradigm is often used as a basic communication model for gathering and dissemination of sensed data [62].

2.2 THE MAIN PRINCIPLES OF THE NEW INTERNET

Given the structure of the present day Internet and the recent evolution in technology and applications a radical redesign of the current internetworking architecture appears as a possible and tempting solution in order to meet the new challenges. The redesign of the internetworking model is a complex task, therefore, any research work needs to be done carefully, and also testing actual prototype implementations is an absolute necessity, in order to predict possible shortcomings of the design. Moreover, detailed evaluation of the bottlenecks of the current architecture is required. Most commonly the naming, addressing and routing are seen as Internet's weakest points. Therefore, redesign of these principles are envisioned as unavoidable starting point in the restructuring. Additionally, the security and privacy mechanisms represent one of the major concerns [63, 64, 65, 66].

2.2.1 The main redesign targets towards the new Internet

Initial end-to-end Internet architecture with tight connection between IP address and location has introduced various problems especially for the new applications that are not in a natural way end-to-end centric. Furthermore, due to the intrinsic conceptual mismatch with the fundamental Internet architecture the integration of mechanisms such as mobility into existing system requires large additional effort. Therefore, the naming, addressing and routing mechanisms appear as the most important redesign targets.

Naming scheme of the current Internet architecture is based on Domain Name System (DNS) with a hierarchically organized naming space. A rather simple idea of DNS has experienced numerous problems with the recent fast growth of the naming space and the number of Internet users. Along with the Internet growth and the increase of malicious behavior the DNS infrastructure has become more vulnerable to denial of service attacks. Due to the overload and limited redundancy of name-servers that are providing the name-address mapping, the lookup procedure is often negatively affected. As a result, increased delays due to the name-address resolution might be encountered.

As a possible solution the development of the fast name resolution mechanism is envisioned, e.g. by enhancing current DNS system by providing fast dynamic updates and faster resolution of DNS queries [67]. However, recent research interest has been focused mainly on modeling of new naming system with a clear separation of address and location by replacing the DNS system with flat, self-certifying naming scheme [10, 9]. The idea argues on avoiding the name-address resolution by basing entire routing on identifiers, i.e. flat labels. The advantage of this approach is in entirely distributed control over the namespace. However, such a namespace lacks a human-readable properties. Furthermore, the trusted third party certification is required in order to assure the integrity of identifiers. Although suffering from scalability problems, routing over flat labels solves identity-location binding and topology related problems.

The original addressing scheme of the current Internet was based on the existence of address classes with variable size prefixes (8, 16 and 24 bits). The inflexibility of such addressing scheme has motivated the work on CIDR (classless inter-domain routing) mechanism. Such an approach enabled better allocation of available address space and optimization of the inter-domain routing tables.

Another interesting addressing redesign approach is proposed in [68]. The main target of the model is providing accountability by assigning responsible entity for each network action. The address is of the format AD:EID (self-certifying administrative domain identifier: self-certifying host identifier) which directly incorporates a domain level identifier and the host name. Such an address design allows all network entities to confirm authenticity of traversing packets and prevent addressing misuse. Due to the uniqueness of the host name, the mobility is supported in more natural way. More radical addressing redesign model has been proposed by [10] which instead of using the IP, relies on topologically independent flat labels.

Additional concern of the current Internet is in insufficient scaling of BGP protocol and its inability to cope with the growth of the global network. According to the BGP model, every router on the inter-domain level needs to know the paths towards all possible destinations. Therefore, in order to preserve the routing consistency, the prompt updates of the routing paths changes on the global level (for the globally visible prefixes) are needed. The amount of the globally visible prefixes is constantly increasing, making the BGP route updates more complex procedure. Thus, finding more optimal routing solutions appears as one of fundamental issues. In the context of proposed addressing and naming schemes, the name based domain level routing arises as the most compelling approach. Having routing mechanisms relying on host identities, without the need of name resolution would naturally eliminate a large number of current Internet problems [10]. However, the scaling and efficiency of such model remain the main concerns.

2.2.2 Implementation considerations for new Internet architecture

According to the main redesign targets a several different design approaches for building a new Internet emerged. A great number of researchers argue on "satisfying" functionality of current Internet and propose its enhancement through the system of overlay solutions [69]. As a possible outcome of such approach the overall complexity of the network might increase significantly. Such an expectation motivated the research community to invest more in the area of complete redesign of Internet, i.e. clean slate approach on building the new system intrinsically compatible with modern demands. However, there are implementation concerns and trade-offs associated with both above mentioned design solutions [70, 71]. In the following we give a short overview of these issues.

Overlay solution

Overlay approach [69] is envisioned as a mechanism which exists on top of the present Internet infrastructure providing additional features not supported by the original Internet design. This approach has already been widely used in order to bypass limitations inherited from existing infrastructure or to provide the functional extensions of various mechanisms, e.g. mobility, quality of service, multicast, enhanced security.

Nevertheless, the deployment and resource expenses of overlay solution can be significant. Hence, the system planning requires careful formulation of various network aspects, e.g. placements of nodes and fine-tuning parameters for overlay links. The main objective is in minimizing overall network deployment costs, taking into account connectivity requirements between overlay nodes, individual user demands, and expenses of management of the traffic flows while the full coverage of all users is guaranteed.

One of the concerns in overlay design are related to the control of a large number of participants that are potentially highly scattered geographically, possibly belonging to different administrative domains and using disparate mechanisms to access the network. Moreover, overlay consumers are not persistent, which additionally emphasize its heterogeneous and dynamic nature. Taking into consideration the network growth and aforementioned characteristics of the overlay solution, it is hard to establish centralized coordination entity responsible for overlay management. Additionally, due to the heterogeneity of involved nodes they may not share the same interests of participating in an overlay transport. Thus, the overlay system needs to ensure the certain incentives for nodes in order to convince them to operate in a way which leads to overall system optimality. Recently game theory algorithms have been used to model this problem [72, 73].

Another possible issue in developing overlay networks is in choosing sustainable business model. The overlay network establishes agreement with particular underlying ISP related to nodes installation and required bandwidth purchasing. On the other hand, overlay networks recover their costs by selling its services to interested users. Thus, defining effective customer-provider relationships and rules between overlay network, underlying ISP and the users are one of the fundamentals of system development.

Recently, more doubts have emerged regarding the idea of an overlay approach as a definite solution for the Internet problem. It certainly appears as an efficient solution for different network configurations, but a number of problems still remain. The proliferation of a number of different overlay networks is also a concern, as the interplay between different overlays and scalability of approach are complex phenomena. One of the further design challenges is security, given that the security weakness of one overlay may threaten the stability of a whole system.

Clean slate solution

Furthermore, with the current growing trend of the Internet the rationality of injecting new overlays and incremental patches into existing architecture is becoming debatable. It poses the question of control and maintenance of such a complex overlay system, and initiates investments in the clean slate approach [70] as one possible solution. The clean slate aims at applying information-centric paradigm on all network layers.

However, applying redesign from the scratch over highly distributed system such as Internet implies even larger problems, e.g. adaptability and huge investments. Moreover, development of such an architecture towards commercial and feature-diverse system, as Internet is experienced today, will take at least a decade. Thus, the complete redesign of the Internet rises the concerns with respect to evolution time until the newly proposed structure becomes widely deployed and with prominent service quality. The investment is also considerable as one would need to run two networks concurrently in order to guarantee the availability of different (legacy) services.

Despite the fact that any larger change in the present Internet architecture would impose significant additional investments, a various redesign approaches argue on creating a completely new Internet architecture. Given that many problems of the present Internet appeared due to the after-thought solutions, the clean slate model might represent a promising long term solution.

In order to become intrinsically accepted the new network model needs to provide full backward compatibility with existing network and to incorporate vast majority of its widely deployed features. Finally, the trust and security issues remain as one of the most critical redesign considerations.

2.3 NEW INTERNET PROPOSALS

The research on redesigning of the current Internet has resulted in a large number of design proposals, targeting different architecture modules as main redesign objectives or having a vision of completely new model, inspired by the clean slate approach. In the following we give a short summary of selected solutions towards the new Internet framework.

2.3.1 ROFL

ROFL (Routing on Flat Labels) [10] architecture separates network location and host identity by implementing the routing mechanism relying entirely on identifiers, without resolving them to locations. By eliminating the need for separated name resolution system ROFL diminishes the major problems of current Internet while providing the better platform for implementation of mechanisms such as multihoming and mobility.

The central part of ROFL architecture is creation of the identifiers. The main aim is in making them persistent over time and without semantics, i.e. flat labels. Hence, proposed solution relies on self-certifying identifiers derived from host's public-private key pair. Namespace structure is based on successor-predecessor model, comprising a set of pointers to the last and the next hop node. In general, the set of nodes residing in the same autonomous system forms a ring of nodes' IDs having the knowledge of their successor and predecessor. However, the main concerns of this approach remain complexity and scalability related issues.

2.3.2 DONA

DONA (A Data Oriented Network Architecture) [9] focuses on the redesign of the naming space and the name resolution mechanism by replacing DNS names with flat self-certifying identifiers. Such identifiers are resolved into addresses using name based anycast network layer placed above existing IP. DONA architecture is composed of "principals", which correspond to any network entity, e.g. host, domain, data or service.

The data identifiers are derived from principal's public key and additional name chosen by the belonging principal. The form of identifiers resembles the

P:L structure, where P is hash of principal's public key and L is label selected by the principal. Moreover, the principals keep the exclusive right on their data, i.e. only hosts certified by principal P can serve the data of the form P:L. Since the data retrieved from the source contains principal's public key and signature, every receiver is able to verify authenticity of the data.

Instead of DNS servers, DONA's name resolution is based on hierarchically organized RHs (resolution handlers). Due to its rather simple name resolution scheme, DONA architecture easily supports mobility, multihoming, server selection and caching. Furthermore, it is compliant with delay-tolerant networks and deployment of middleboxes, e.g. firewalls. By making the applications relying on names and not directly on the address of data and services enhances the level of abstraction and separation of low level network protocols which, thus, can be dynamically selected. While suffering from various security issues, e.g. DoS attacks targeting RH, DONA's possible simple implementation encourages the further research in this direction and has had a lot of influence on subsequent work.

2.3.3 I3

I3 (Internet Indirection Infrastructure) [11] is an overlay solution which introduces the rendezvous system in order to find the match between the data sender and the receiver. The traditional IP addressing pattern is replaced with data-associated identifiers specified by the sender. Such identifiers are used as a basis for data distribution and reception. In this way decoupling of sender and receiver is fully enabled since knowing the identity of any of them is not necessary for the data delivery.

The network entities willing to publish an information generate the messages of the form *(id, data)*, where the *id* is identifier of the data and the *data* is the actual payload. Each identifier *id* has corresponding node, i.e. rendezvous node, responsible for managing all messages containing that particular *id*. Thus, every publication having the same identifier is routed to the same rendezvous node which performs the matching operation. Nodes interested in receiving the data with identifier *id* indicate it by sending the packets (the "triggers") of the form *(id, addr)*, where the *id* is the identifier of the data and the *addr* represents the address of receiving node.

In other words, a trigger *(id, addr)* indicates that all the messages with the identifier *id* should be forwarded to the node with the address *addr*. Therefore, the rendezvous is carried based on the identifier of the data. I3 relies on Chord [74] lookup protocol which eases the incremental deployment of the overlay since every new server can simply join the I3 system using Chord protocol. Although it provides flexible solution I3 suffers from security issues and leaves a lot of space for future improvement.

2.3.4 TRIAD

TRIAD is the Internet architecture that resembles a set of domains connected with relay agents translating packets between adjacent domains [12]. TRIAD brings notion of relay agent's internal and external IP address as a basis of identification and authentication and extends IPv4 naming system using WRAP (Wide Area Addressing Protocol) over IP. By utilizing the source routing, the WRAP protocol enables the path-based forwarding to be executed on top of existing Internet. The communication is done by implicit addressing without directly targeting the packet destination. Furthermore, the TRIAD model is easily applicable to the current Internet architecture. Relay agents, placed between a separate domains, provide the naming and routing services on inter-domain level. The operations of addressing, naming and routing within a single administrative domain is done in the same way as in current Internet. Therefore, the changes in a single domain are not required.

2.3.5 NDN

NDN [75], Named Data Networking (successor of the CCN Content-Centric Networking [76]), is an ongoing research project that focuses on the information-centric based restructuring of the current Internet following the clean slate approach. NDN proposes the hierarchical organization of the naming space [8] and the information location using the broadcast. Such a model eliminates the need for the dedicated network entity performing the rendezvous operation.

NDN relies on the identities that preserve the close relationship with the publications' content. The users broadcast the "Interest" packets containing the name of the desired content. On the other hand, any node having the requested content and willing to provide it, responds with the corresponding "Data" packets. Since the naming system is hierarchically organized the content name in the "Data" packet needs to be a suffix of the content name in the "Interest" packet. Such a prefix match ensures that the data specified in both packets belong to the same subtree. One of the main challenges of NDN is ensuring the scalability of a system with a potentially unlimited name space.

2.3.6 PURSUIT

PURSUIT [77] (successor of the PSIRP Publish-Subscribe Internet Routing Paradigm [78]) project is based on the idea that all network operations across all layers are performed on named information items. The main work in this thesis is done in the context of PURSUIT architecture and implementation framework. The architecture follows the clean slate approach natively supporting the publish-subscribe communication pattern and placing the information in the focus of all considerations. The information items as a simplest units transmitted by the network are identified by statistically unique names. The primary objective of the network is to deliver such information, regardless of its

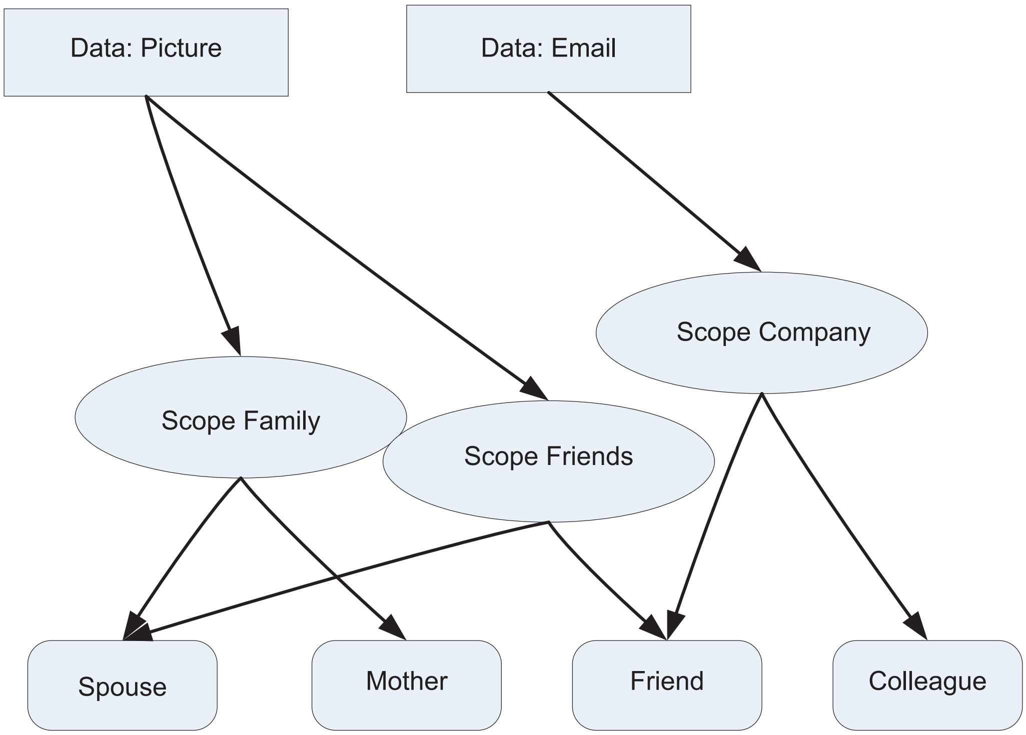

Figure 2.1: The concept of scopes.

location and the host it belongs to. Dedicated Rendezvous Identifier (RId) [79] is assigned to each information item traversing through the network. The RId can be derived from the identifier of an end-point or from application being the source of information.

A semantically related information items are grouped into scopes identified by the Scope Identifier (SId). Scoping of information improves its reachability by introducing simpler and more efficient searching mechanisms, e.g., searching area of information with particular RId is narrowed to relevant scope. Scopes represent the powerful artifact for building the social relations between the network entities and information items. Such a concept is illustrated in Figure 2.1. A certain information is generated in the context of family or friends issues, whereas the other information pieces are closely related to the business matters. The concept of scopes facilitates the grouping and reachability of such disparate information. Furthermore, it can be used as a mechanism for restricting the availability of some information to the specific network entities. As shown in Figure 2.1 the particular information can be disclosed only to some groups of users.

Each information item belongs to at least one scope. The scopes can be considered as a set of information, thus, they can be treated as information items, as well. As an outcome of such a structure, the scopes can belong to

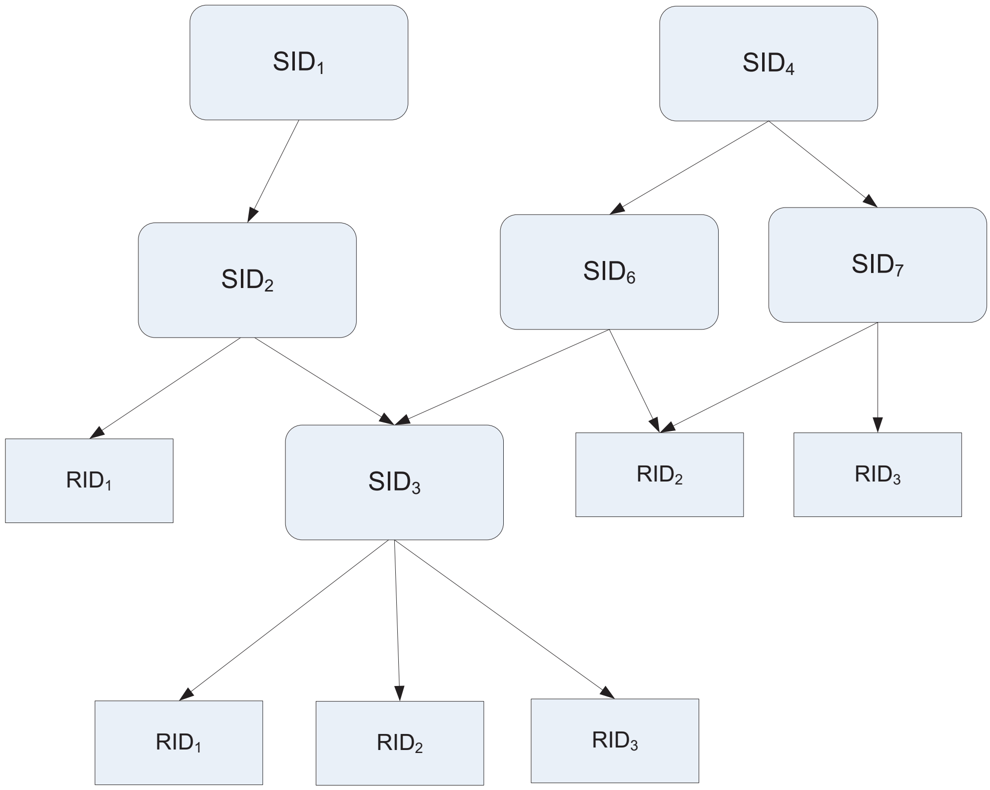

Figure 2.2: The scope structure.

another scope, building the complex (directed acyclic) graphs of information as illustrated in Figure 2.2. Each information item, identified by its rendezvous identifier (RId) is statistically unique within the scope to which it belongs.

The main building blocks of the PURSUIT architecture are: *rendezvous*, *topology*, and *forwarding* functions [80]. The rendezvous function performs the matching between publications and subscriptions to the information items, each identified via a RId/SId identifier pair. Once the match between publication and one or more subscriptions has been established the rendezvous triggers the data delivery paths creation by sending the request to the topology management. The topology management is responsible for generation of the optimal delivery paths from the publisher to the subscriber.

The functionality of topology management is separated into two operational realms: inter-domain and intra-domain topology. Intra-domain topology management is responsible for local network topology generation, i.e. within one administrative domain. Its main role is in discovering of topology information, using it as an input for computing necessary network states, and updating forwarding information to involved nodes.

Inter-domain topology formation function defines topology formation on the domain level, i.e. between administrative domains. In other words, it has a

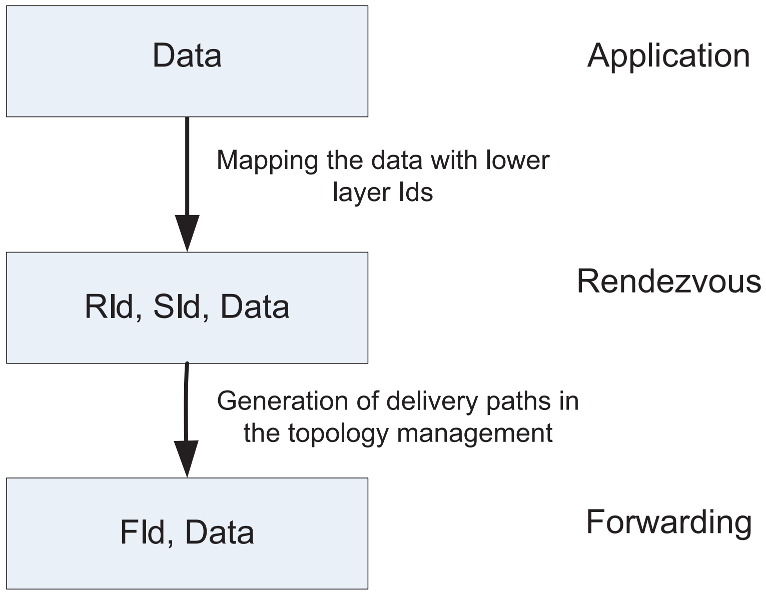

Figure 2.3: The structure of identifiers.

role of configuring and maintaining inter-domain topology states. This is further used for creating forwarding paths based on various policy compliance requirements as well. Once the optimal delivery paths are created in the topology management, the forwarding function performs the final data transfer using the label switching mechanisms. The data delivery path is generated in the form of forwarding identifier FId and it resembles the source based routing label.

The PURSUIT architecture, therefore, maintains the different identifier spaces as illustrated in Figure 2.3. On the higher levels the identifiers are assigned according to the application context. Such identifiers are further mapped to the rendezvous and scope identifiers (RId and SId) in the PURSUIT architecture. Finally, after the data delivery paths are generated in the topology manager the forwarding identifier is assigned to the information item.

Moreover, the PURSUIT architecture employs the caching functions realized natively in the architecture or executed as an external service on top of the network. The forwarding modules are capable of caching the content traversing the network; thus the lost data can be easily obtained from the closest cache [81].

Another important aspect of the PURSUIT architecture is *dissemination strategy* that underlies each scope. The dissemination strategies define the methods used for realizing the functions of rendezvous, topology management and forwarding in dedicated network entities. By the means of dissemination strategy the different parameters of the network operation are defined, e.g., data representation formats or mechanisms for identifiers creation.

Although burdened with the high scalability requirements, the efficient data structure of PURSUIT architecture enhanced with the variety of content-oriented forwarding functions shows the satisfying performance. Therefore, a large portion of the experimentation work carried out within this thesis is based on the PSIRP/PURSUIT architecture. Moreover, our solutions represent a part of the PSIRP/PURSUIT framework, and basis for further development within its architecture. Most of our modules are integrated into the PSIRP/PURSUIT prototype and are available for the larger scale testing also externally through the public release and free licensing of PSIRP/PURSUIT framework. Being a part of the PSIRP/PURSUIT team we have been continuously working on the design, development and optimization of PSIRP/PURSUIT architecture.

3

PUBLISH-SUBSCRIBE COMMUNICATION PARADIGM

As discussed in the previous chapter, the publish-subscribe communication model has recently attracted a lot of attention being envisioned as the most flexible and scalable communication solution to the present Internet problems. In general, the users having some particular piece of information to share can make it available by *publishing* it to the rest of the network. On the other hand, the users can express their interest in particular event or data by *subscribing* to it. After the match between subscriber's specification and the published data occurs, the actual data transfer begins. Hence, this model provides clear separation of publishers and subscribers in time and space [82], since all communication occurs asynchronously and independently of location and identity of publishers and subscribers. Due to its flexible, event based and asynchronous nature, publish-subscribe model satisfies better the requirements of many of the recently emerged applications, e.g. content delivery, social network updates, wireless sensor networks, just to mention few.

In this chapter we discuss the idea of the publish-subscribe paradigm and its advantages and disadvantages with respect to the traditional end-point centric communication, especially focusing on the control and management plane communication. We give an overview of the performance of the publish-subscribe pattern under critical conditions such as wireless mobile environments. The publish-subscribe paradigm has been studied very little in the context of wireless mobile environment. In this thesis we will alleviate this problem by studying performance in the mobile environments. This is extremely important as the mobile Internet use is becoming dominant. Furthermore, we examine the publish-subscribe behavior when used for some of the common traffic types, e.g. voice applications. The detailed evaluation of specific application types is beyond the scope of this chapter.

3.1 PRINCIPLES OF PUBLISH-SUBSCRIBE MODEL

A generic publish-subscribe communication system can be defined by a triple $< PUB, SUB, RZV >$, where each element in the triple represents a set of network entities. Depending on the functionality of the network entities the sets can be differently classified. The data sources are identified by the set $PUB =$

$P_1, P_2, ..., P_i$ of *publishers*, the data consumers form the set $SUB = S_1, S_2, ..., S_j$ of *subscribers*, whereas the dispatching system which performs the match between publications and subscriptions is defined by the set $RZV = R_1, R_2, ..., R_k$ of *rendezvous* entities. Figure 3.1 illustrates the publish-subscribe system and its main operational principle. The network entities can also have overlapping functionalities, e.g. being publishers and subscribers for different data items. However, only after the match between publishers' and subscribers' interests occurs on the rendezvous entity the actual data transfer is carried out. Otherwise, the direct connection among publishers and subscribers is not possible. Such a decoupling of network entities is a powerful tool for minimizing the issues such as addressing, synchronization and the reception of unwanted traffic.

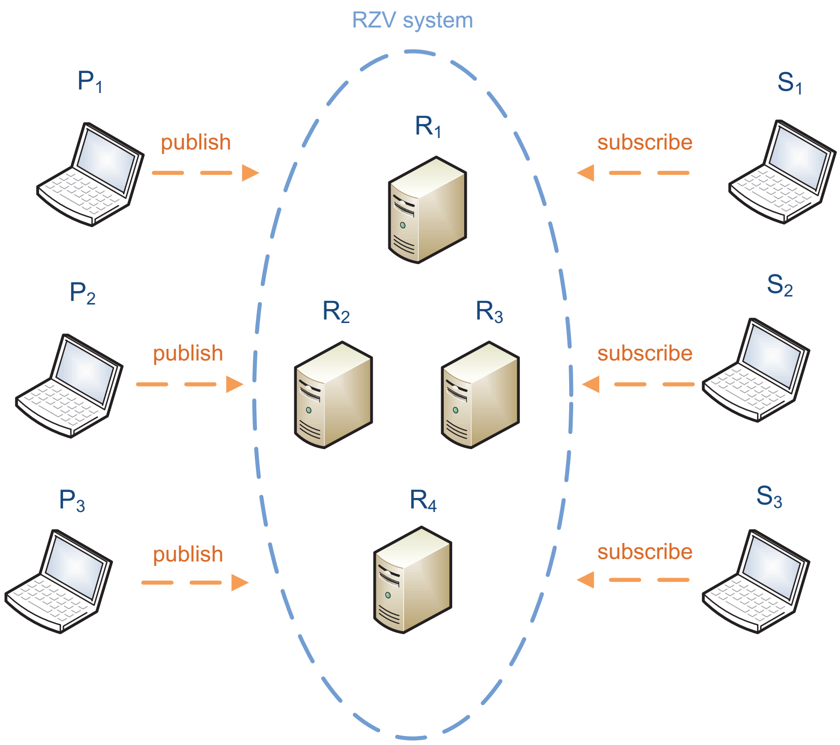

Figure 3.1: A generic public-subscribe system.

Different types of publish-subscribe systems can be identified according to the applied subscription model. In general, the subscription models differ in the level of their expressiveness, i.e. the accuracy of matching the user interests with available data. The choice of the subscription model significantly influences the realization of the publish-subscribe model, where the trade-off between the expressiveness and simplicity needs to be made.

One of the simplest subscription models is *topic-based*, where the subscriber expresses the interest in particular topic and is consequently being notified only about the data related to that particular topic. Such a model has been adopted in the first generation of publish-subscribe implementations [83, 84, 85]. The topic-based low level of expressiveness, which was caused by over simplicity of the model, could not satisfy the user requirements.

In order to provide more sophisticated and expressive methods to reflect user interests in detailed manner, the *content-based* subscription model has been proposed. Based on such model the subscribers are able to specify the conditions or constraints of the attribute values of the data or events that they are interested to receive. The efficiency of such a model highly depends on the subscription language used and the type of the attributes and events. Due to its high flexibility and efficiency the content-based model has been applied to the variety of architectures [86, 87, 88, 89, 90].

Following the object-oriented programming concepts the *type-based* model has been proposed in [91]. According to this approach the subscriptions belong to specific type which consists of different attributes that can be defined. Thus, the model resembles the combination of the topic-based and content-based principles, aiming at providing the balance between efficiency and simplicity.

3.2 PUBLISH-SUBSCRIBE MODEL IN WIRELESS NETWORKS

Due to the wide deployment and popularity of wireless networks, one of the most important requirements in development of new applications or communication models is their integration into wireless systems. Furthermore, the effects of wireless environment on the overall performance of a novel communication model represent an important evaluation aspect. Due to the unpredictable and highly variable nature of wireless medium, the exact insight on the actual performance of deployed model is very hard to achieve, especially over the large scale. However, publish-subscribe model focusing on asynchronous and loosely coupled interaction between communication points appears as an attractive approach for variety of wireless and mobile applications. Therefore, the deployment of publish-subscribe model in wireless networks can be expected to have a positive impact on the overall performance and functionality of the network.

One of the examples of successful integration of publish-subscribe paradigm into wireless domain is deployment of Wireless Sensor Networks middleware solutions that are based on publish-subscribe model [92, 93]. The middleware acts as a bridge between communication entities in distributed and heterogeneous environments. It can use various communication models for a sensed data collection and dissemination. Relying on unique identifiers assigned to sensor nodes and address-based forwarding of required data to the users interested in particular event is one of the possible data distribution approaches. Due to its end-point centric nature this solution is sensitive to the frequent topology change, which is the common occurrence in highly dynamic environments.

In order to overcome such issues, some of the recent WSN middleware solutions advocate the idea of equipping the sensor nodes with publish-subscribe communication model [94]. The basic principle is that sensor nodes can distribute the acquired data in the form of publications. The users (sinks) willing to receive this particular information announce their interest by simple subscriptions. As soon as the data becomes available the interested users are able to retrieve the required information. Taking advantage of asynchronous and event-based communication model, sensor nodes consume less energy and processing resources. This is of fundamental importance for systems and operating environment which have scarce power and computational budgets.

Frequent handovers in mobile environments pose a supplemental challenge for utilization of the publish-subscribe communication model. Particularly a problem of temporarily, or permanently lost connectivity is an issue that must be addressed with wireless publish-subscribe implementations. All messages generated during a handover phase should be delivered to the subscriber without a loss. Moreover, a signaling overhead as well as the network overload due to the unnecessary packet duplicates need to be avoided.

One of the rare, early works that has considered the mobility case, presented in [95], discusses potential of publish-subscribe model to be used as a promising solution for mobility issues. Results show that the performance of the publish-subscribe model depends on the large set of parameters, e.g. message size or the type of the handover scheme used. Therefore, due to the complexity of mobile use case, the integration of publish-subscribe communication requires additional research in order to gauge optimal parameter space and deployment scenarios. This thesis focuses on detailed evaluation of this problem, utilizing the publish-subscribe model not just as an overlay solution, but as a core communication paradigm present at all network levels. Extended and more detailed analysis on the performance of the publish-subscribe model in mobile environments, that has been done during this thesis work, will be discussed in Chapter 5.

3.3 PUBLISH-SUBSCRIBE MODEL AND MULTICAST COMMUNICATION

Multicast communication [96, 97, 98], conceptually similar to publish-subscribe model, has been identified as a way of reducing the bandwidth requirements for content delivery in wireless networks. It was envisioned as a network layer service providing the possibility for an arbitrary node to send data to, or receive the data from, any multicast group. Multicast communication is supported in the network layer, therefore, applying it in the application framework within current Internet architecture is often complicated. The establishment of the multicast delivery tree on a local level by means of IGMP [99] is required, and support for multicast communication at all involved network domains needs to be ensured.

A simple illustration of a network using multicast is shown in Figure 3.2. Two sources Src_1 and Src_2 send traffic to group G_1. Multiple receivers *Rcv* receive the data from the group G_1. In order to forward the packets correctly, each router needs to maintain the routing state for every group in which transfer of packets it is involved. In some cases, a router needs to maintain a forwarding state per source, in addition to the per group states. Storing and processing of such state information is one of the problems for multicasting, as it requires computing and memory resources that can become excessive if the number of sources with states is very high.

Figure 3.2 illustrates the case where all multicast packets of the group G_1 are forwarded to the *Router 1* whereas the multicast packets from the same group but with the source Src_1 are forwarded to *Router 2* and *Router 3*. However, the maintenance of such a large number of states at each router in the network can rapidly lead to severe scalability problems and inability to support a large number of multicast groups. A numerous multicast architectures and protocols have been designed with a common aim of improving the multicast efficiency and offering better scalability [100, 101, 102, 103, 104, 105, 106]. A considerable research effort has been especially focusing on reliability issues [107, 108, 109].

Additionally, application layer programming environments do not always provide the native support for multicast. All of these impediments have negatively influenced to the wide-spread deployment of the multicast communication, both in wireless and fixed networks, resulting in a rather marginal role of multicast in the present day communication in the Internet [110].

A widespread deployment of publish-subscribe communication either in the network layer, or as a part of the application-layer programming frameworks is envisioned to significantly change the situation. The difference between the unicast and multicast communication would not be directly visible to users or application developers, but the network would perform the optimization by choosing the most appropriate delivery mechanism to be used on lower layer. Such a broad acceptance and deployment of the publish-subscribe principles would bring the benefits by improving the efficiency and utilization of network resources. Moreover, the deployment of publish-subscribe framework into the future network would also automatically enable benefits of multicasting.

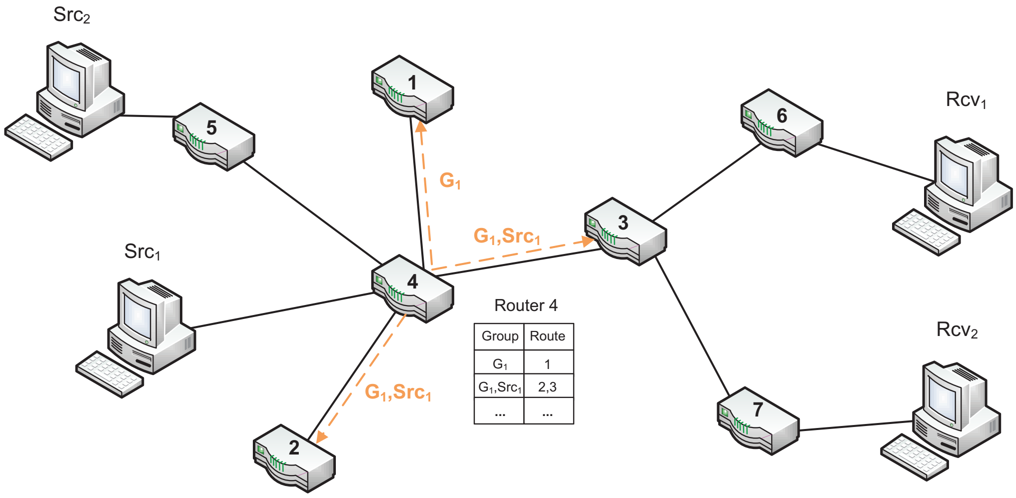

Figure 3.2: An illustration of multicast communication with one group G_1 and a set of senders and receivers.

3.4 PUBLISH-SUBSCRIBE MODEL IN CONTROL AND MANAGEMENT PLANE

In addition to the application layer, there is a great potential in employing the publish-subscribe communication directly in the control and management plane within wireless networks. Many of the existing radio resource management functions, such as measurements of link quality or signaling related to mobility or roaming are not naturally supported by unicast, end-to-end transport. The decoupling provided by publish-subscribe communications seems as a more appropriate approach. Thus, it is interesting and highly important to explore the potential benefits of developing publish-subscribe model in the control and management plane of the wireless networks and compare its efficiency with more traditional solutions, e.g. currently used within 3GPP. The work carried out within this thesis specially focuses on such novel idea of exploitation of publish-subscribe model in previously unexplored space of control and management plane functions.

In order to get a closer insight to the impact of the publish-subscribe model and evaluate its performance in the wireless networks, we have analyzed control traffic in the wireless dynamic environments. We focused on the simulation studies carried out using the Qualnet 4.5 network simulator [111]. Qualnet simulator is a fast and scalable evaluation tool for performance measurements of wireless, wired, and mixed-platform networks. We conducted our simulations in the context of IEEE 802.16 and IEEE 802.16e standards [112].

One of the targets of our work was to examine the scaling by performing tests with various network sizes, i.e. we studied networks having 80 to 200 nodes within a WiMAX cell in initial simulation setup. In order to better estimate the effect of a network size on the system performance, the results for a smaller networks of 20 nodes are provided, as well. We carried out the single cell experiments having an omni-directional antenna with transmission power of 15 dBm, available bandwidth of 20 MHz and applying 2048 sub-carriers version of OFDM. Multiple cell scenarios and handover issues in publish-subscribe networks will be discussed in details in Chapter 5. Two types of applications are considered: traditional end-point centric constant bit rate (EPC-CBR) and a publish-subscribe application types, both transferring the packets with the 1 Kbps data rate. Such a low data rate communications are commonly needed for control plane protocols, as well as for various low data rate applications such as RSS feeds.

The mobile nodes in initial setup are scattered around base station in cyclic manner and moving in the direction from the base station towards the edge of the coverage area with speed of 10 m/s, as illustrated in Figure 3.3. In order to estimate the impact of nodes' speed on the network performance we extended our analyses with additional use cases varying nodes' velocity.

3.4.1 *Publish-subscribe communication performance*

In the first scenario we investigate the performance of the publish-subscribe communication model where the base station is given the rendezvous functionality. Therefore, it keeps the track of existing publications and subscriptions and performs the match between them. The mobile nodes can play the role of publishers and subscribers, providing the content and registering to the data they are interested in. Upon the match between publications and subscriptions occurs, the rendezvous node initializes the mechanisms for the final data delivery. In our scenarios all mobile nodes are subscribed to the information coming from a single publisher. The publisher is connected with the base station over a 10 Mbps link and generates a control traffic.

While increasing the network size the publish-subscribe communication shows slight degradation of performance in the terms of received data on the subscriber, as illustrated in Figure 3.4. Further analysis shows slight increase in average end-to-end delay experienced by each subscriber due to the network growth, as shown in Figure 3.5. As the network size increases, the rendezvous node needs to handle a larger amount of subscription requests. Therefore, a slight performance degradation is expected. However, for the considered network size the control traffic does not experience a severe deterioration.

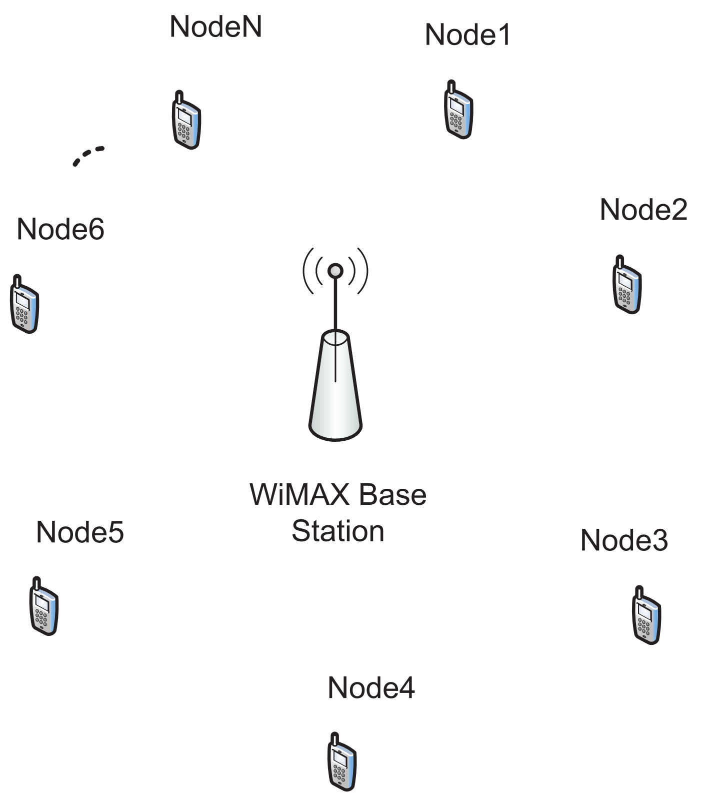

Figure 3.3: Initial simulation network setup.

3.4.2 *End-point centric communication performance*

In order to compare traditional end-point oriented communication scheme with publish-subscribe based traffic pattern, we deploy two scenarios of traditional unicast end-point centric constant bit rate (EPC-CBR) traffic. In the first experiments we create the similar scenario as in the publish-subscribe tests setup. While the same node remains the traffic source, due to the absence of the rendezvous entity the connection between sender and receiver is established directly. Thus, the traffic source manages a large set of separate unicast CBR flows depending on the network size.

The application data rate for our scenario remains quite low, and thus "unfolding" of the publish-subscribe flow resembling the multicast communication, into a number of unicast flows might be expected not to result in performance hit. However, our results indicate a significant increase in average end-to-end delay in this case compared to the publish-subscribe model. Handling multiple traffic flows separately generates an additional processing load on the traffic source, even in the low-bitrate control traffic scenario such as considered here. Moreover, in the context of received data, the traditional communication model performs considerably worse than publish-subscribe one.

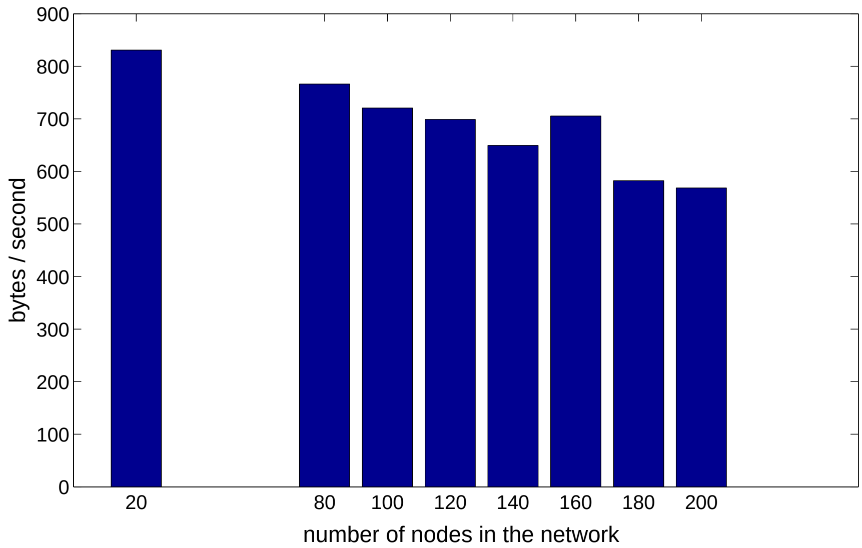

Figure 3.4: Publish-subscribe communication performance. Average throughput on each node.

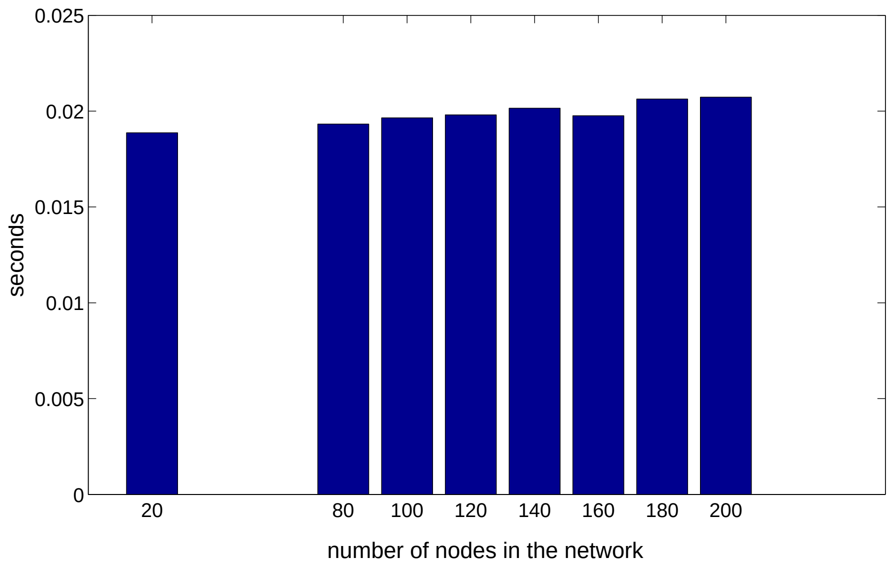

Figure 3.5: Publish-subscribe communication performance. Average end-to-end delay on subscribing nodes.

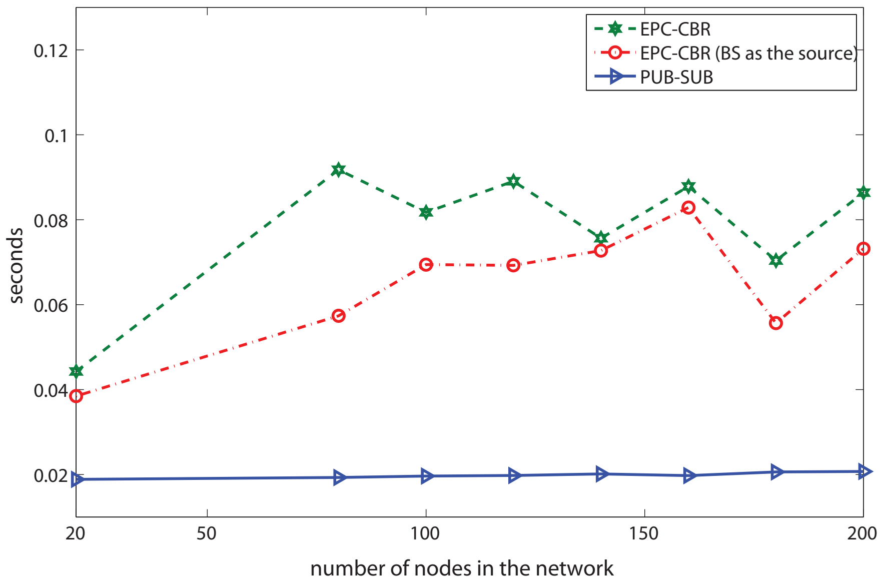

Figure 3.6: Publish-subscribe and end-point centric unicast communication, average end-to-end delay.

We repeated our experiments varying the data source placement in the network setup. Given the increased processing requirements in the case of "unfolded" publish-subscribe traffic, a slightly better performance for end-point centric communication is obtained when the base station acts as a traffic source. In the context of the average end-to-end delay experienced by subscribers the system performs better. The similar improvement is evident in terms of received information. However, regardless of the data source location the end-point centric communication type performs significantly worse with respect to the publish-subscribe model. As an illustration of obtained results we provide combined graphs for all executed experiments, see Figure 3.6 and Figure 3.7.

Similar to the results obtained from publish-subscribe experiments, a slight drop in performance is evident for end-point centric communication, as well, due to the increase of the network size. However, comparing the performance of the end-point centric and publish-subscribe communication under the same network environment, the publish-subscribe model achieves significantly better results. Moreover, a large-scale deployment of such a system, having a great number of users interested in receiving the same particular information appears as much simpler within the publish-subscribe concept. Any change in the structure or the number of subscribers would require only corresponding subscription/unsubscription to the data of interest. Thus, the scalability can be easily achieved. On the other hand, adding new receivers within traditional

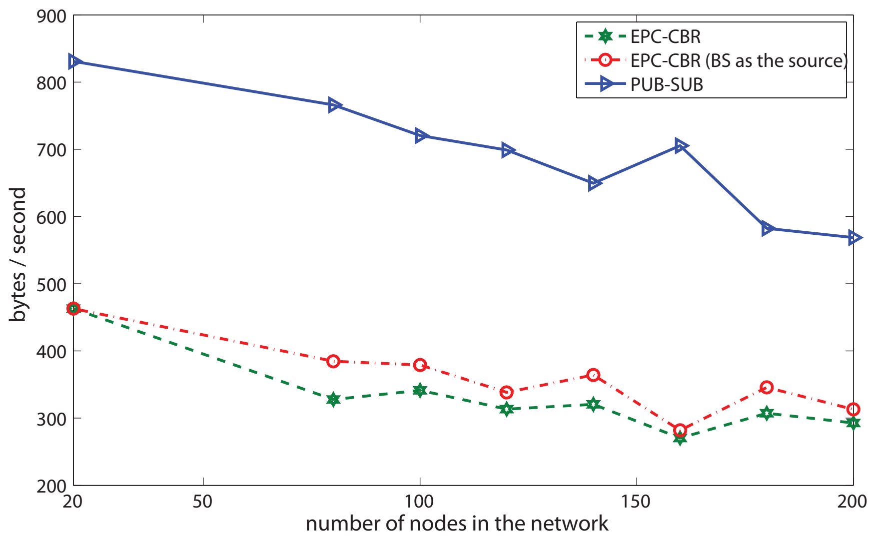

Figure 3.7: Publish-subscribe and end-point centric unicast communication, average throughput.

end-point centric system requires establishing of completely new connections, making the system less flexible and scalable than the publish-subscribe approach.

3.4.3 *The effect of mobility on the publish-subscribe model*

Analyzing the performance of publish-subscribe oriented communications under dynamic conditions represents an important aspect of overall evaluation. In order to obtain a better insight on the publish-subscribe model behavior in mobile environment we repeat our tests varying the velocity of nodes from 5 m/s to 15 m/s. Results illustrated in Figure 3.8 show performance degradation in terms of achieved throughput while the speed of mobile nodes is increasing.

The network dynamics significantly affects the overall network performance regardless of the communication pattern applied. Generally, the decrement in the amount of received packets, as well as increase of average end-to-end delay while raising the velocity is evident in both cases. Nevertheless, our analyses show significantly better performance of publish-subscribe traffic compared to the end-point centric approach within the same network dynamics. Due to the flexibility of the publish-subscribe model, in particular its ability to dynamically adapt to changed network topology and retrieve the data from the closest existing subscriber, it appears as more robust in mobile environments.

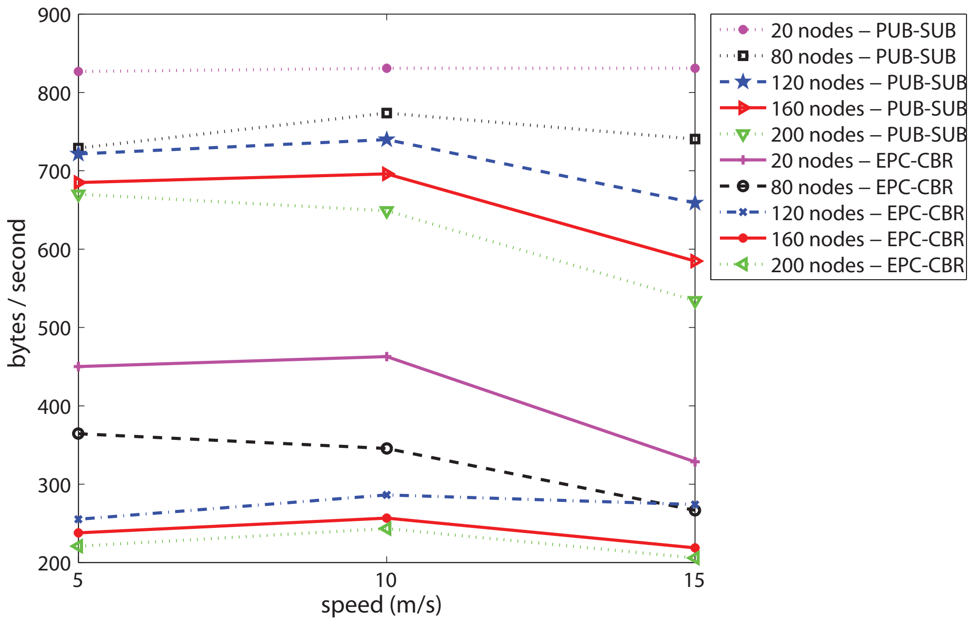

Figure 3.8: Publish-subscribe and end-point centric unicast communication. Achieved throughput with variable speed of mobile nodes.

3.5 Voice applications over publish-subscribe

Apart from the performance evaluation of the publish-subscribe model applied in the control and management plane, we aim at examining the applications characterized with the higher data rates, e.g. voice applications, and their implementation in the publish-subscribe context. Therefore, we performed supplemental experiments over smaller networks with respect to the network setup in the Section 3.4, i.e. ranging from 10 to 120 nodes. Furthermore, we increased the traffic data rate to the values corresponding the high-quality voice applications, i.e., 64 Kbps and 256 Kbps. The traffic source is communicating with the base station over 100 Mbps link, thus, minimizing the probability of a bottleneck creation on this path.

We restricted the network size to relatively small dimension, commonly found in practice, up to 120 nodes, concentrating mainly on the effect of the voice applications and the data rate increase on the overall performance. Positioning and mobility characteristics of nodes remain the same as in initial scenario, i.e. circular distribution around the base station moving towards the direction of cell edge with speed 10 m/s. As in previous experiments, we focused on average end-to-end delay and throughput in terms of received bytes/second as evaluation metrics.

Results obtained from multiple simulation runs confirm better performance of publish-subscribe traffic type in the case of higher data rates, as well. Improvement is evident in terms of obtained throughput while applying publish-subscribe based pattern compared to traditional end-point centric approach. Figure 3.9 and Figure 3.10 show the achieved throughput for 64 Kbps and 256 Kbps traffic respectively. The benefit of publish-subscribe over end-point centric model is mainly based on its inherent multicast nature and enforced anycast. Multiple users can be served by a single publisher. Additionally, the network can be configured in a way to always find ”the best” match between publisher and subscriber according to different criteria.

Average end-to-end delay remains lower in the case of publish-subscribe oriented traffic applied for the voice applications, thus, in situation of increased data rate, as shown in Figure 3.11 and Figure 3.12. The higher data rates do not compromise the benefits of publish-subscribe over end-point centric communication. Furthermore, carrying the data related to voice type of applications can easily fit into the publish-subscribe concept. The call setup as well as the transmission of voice packets can be performed by means of publications and subscriptions.

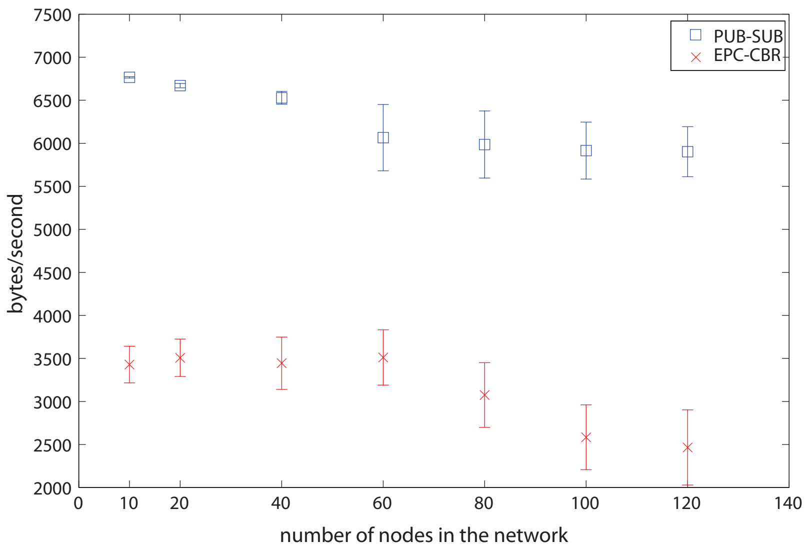

Figure 3.9: Publish-subscribe and end-point centric unicast communication - achieved throughput - mean value and standard error for 64 Kbps data rate.

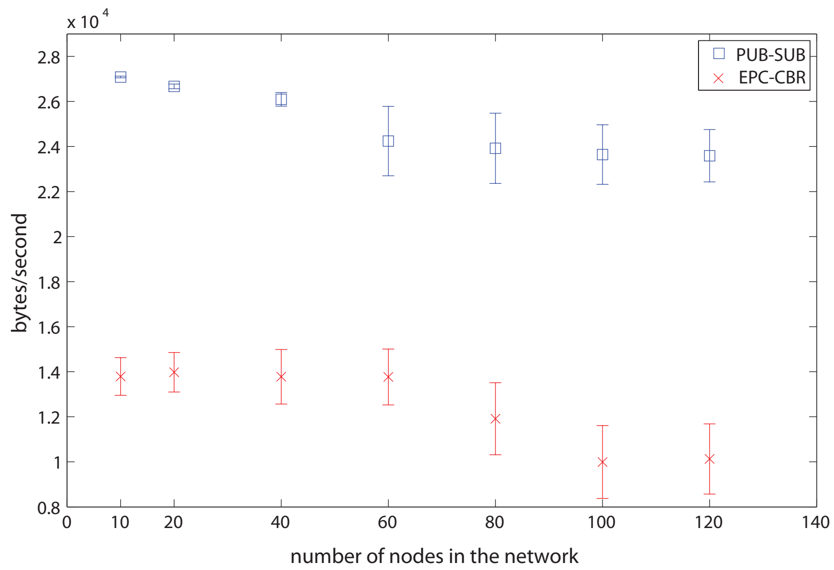

Figure 3.10: Publish-subscribe and end-point centric unicast communication. Achieved throughput - mean value and standard error for 256 Kbps data rate.

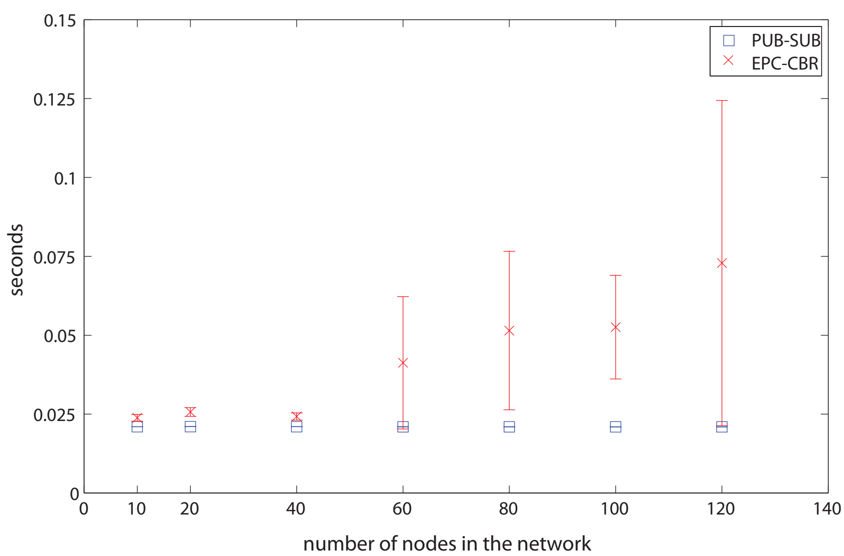

Figure 3.11: Publish-subscribe and end-point centric unicast communication. Average end-to-end delay - mean value and standard error for 64 Kbps data rate.

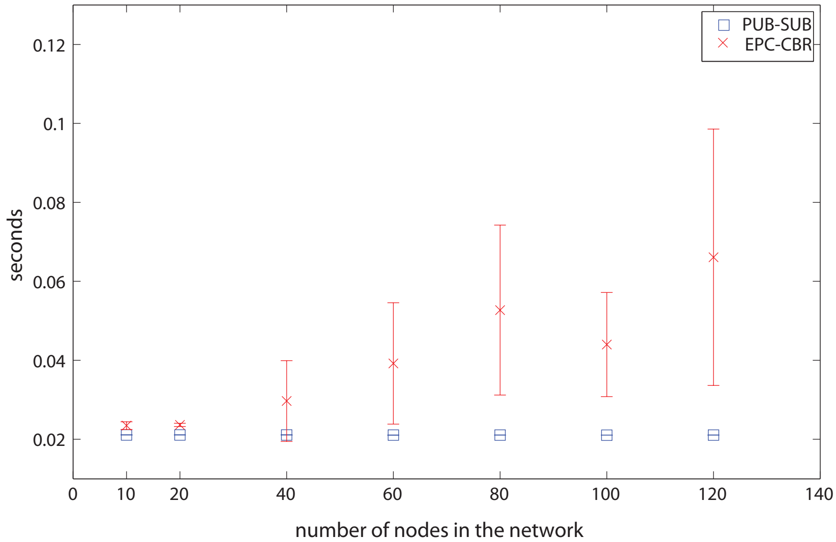

Figure 3.12: Publish-subscribe and end-point centric unicast communication. Average end-to-end delay - mean value and standard error for 256 Kbps data rate

3.6 SUMMARY ON PUBLISH-SUBSCRIBE MODEL EVALUATION

In this chapter we focused on quantitative analyses of a wireless publish-subscribe system. We investigated overall performance of publish-subscribe based communication in mobile environment and compared it with the traditional end-point centric communication model that would be a norm for the present day all-IP wireless networks. As the wireless and mobile Internet access is becoming crucial, our studies of publish-subscribe communication in mobile environment are particularly important. We especially focused on the control and management plane functions and performance of publish-subscribe model in this domain. Additionally, we investigated the behavior of the publish-subscribe communication while increasing the application data for voice services. Considerably better performance of publish-subscribe model is obvious in all examined cases. Hence, wide deployment of publish-subscribe framework to wireless communications shows great potential for enhancing overall network performance while saving network resources at the same time.

Apart from wireless scenarios that have been analyzed in this chapter, there are many potentially interesting network settings to be investigated in the future. Similar evaluation could be done for WiFi 802.11 standards while utilizing the higher data rates, as well. As the structure of IEEE 802.11 standard is relatively similar to IEEE 802.16 we expect that our results are fully applicable also to Wi-Fi scenarios. Furthermore, detailed evaluation of different application types and traffic models and their implementation over publish-subscribe could be an interesting target. Additionally, having a heterogeneous network scenarios employed or investigating the impact of the handover might provide more complete picture of publish-subscribe manners and its applicability in specific use cases. Some of the above mentioned evaluation challenges will be further addresses in the following chapters.

4

Topology management in information-centric networks

In this chapter we analyze the performance of publish-subscribe based intra domain topology management as an important part of the information-centric networks. The topology management represents one of the central entities in the content-centric networks. The role of the intra-domain topology management is generation of the local network topology, i.e. within one administrative domain. Such a procedure is initiated by the discovery of topology information, followed by the computation of the necessary network states, and updating of forwarding information on the involved nodes. Hence, the topology management is crucial for correct network functionality. In publish-subscribe networks it requires special attention and profound research in order to make the network efficient. Nevertheless, the topology management in information-centric networks has not yet been studied much. This thesis is one of the early works considering in details the information-centric topology management, and one of the very first also to study related implementation challenges and complexity.

In this chapter we describe the development of the topology management model relying entirely on information-centric concepts, where the exchange of topology information is done using the publish-subscribe communication pattern. Furthermore, we describe optimization techniques that can be utilized in the path computation in order to better allocate the network resources.

4.1 Principles of the information-centric topology management

The focus of our work is on building flexible and scalable intra-domain topology creation mechanism, which entirely relies on publish-subscribe type of communication in order to perform network discovery and optimization of data delivery paths. Our publish-subscribe based network discovery and topology creation is designed, by necessity, to be modular and easily extensible mechanism. Therefore, apart from the core module performing the topology creation, we developed a set of supplemental modules accountable for improvement of its functionality.

The components additional to the basic design give a better insight on overall network, thus, they provide more functionalities and opportunities to be used for performance optimization. Such components can be responsible for gathering the information about different parameters critical for the performance, e.g. application requirements, current link delays. Instead of investigating a general impact of applying QoS requirements into publish-subscribe systems targeting different architectural pieces, see earlier work in the literature, cf. in [113, 114] we follow the different approach. We aim at inspecting the influence of QoS requirements of applications on the inter-domain path computation functionality. We use combined application demands with available network resources as an input for path calculation algorithm.

We use the PURSUIT and PSIRP architectures [78, 77] as the reference architectures in which the routing, addressing and name resolution mechanisms are adapted to the data-centric, event-driven structure. Therefore, our underlying architecture follows the idea of unique identification of all information items traversing through the network. The data is identified by the dedicated *rendezvous identifier* (RId). Semantically related information items are grouped into scopes identified by the *scope identifier* (SId). Therefore, the complete identification of an information item is achieved by means of the RId/SId pair.

Furthermore, the architecture assumes the existence of autonomous systems that are seamlessly interconnected via set of adequate mechanisms, similar to the organization of the current Internet. Based on this assumption, we distinguish two operational units in our model: inter-domain and intra-domain topology management. Inter-domain topology function is responsible for topology formation on the domain level, i.e. between administrative domains. On the other hand, the role of the intra-domain topology management is similar to the functions performed by the interior gateway protocols in the present Internet.

After the tasks of the inter-domain topology function has been carried out, one or more domain level paths have been formed. Then, the separate instances of the intra-domain topology management function operating in the individual domains generate the corresponding intra-domain forwarding paths. The main guideline for the intra-domain path computation is correct delivery of data to the subscribers within the domains and between ingress and egress routers in the case of transit traffic. Apart from fulfilling the ordinary functional requirements, the individual intra-domain topology functions can be equipped with a set of additional mechanisms for performance optimization.

Our proposed model for intra-domain topology function resembles the existing link-state routing protocols (such as OSPF). However, as we rely on the native publish-subscribe communication, it will be able to take the advantage of the underlying network characteristics. A node interested in receiving information about the physical connectivity between arbitrary nodes and managing the network topology is called *topology manager*. Therefore, every administrative domain in the network is associated to the single *topology manager*.

An arbitrary node can learn its local connectivity by means similar to the ones used in OSPF protocol. The local *Hello messages* are distributed and gathered using the publications and subscriptions. Upon the *Hello messages* have been collected, the learned local connectivity can be represented as a neighbors list, i.e., *Link State Advertisement*, and published to the *topology manager*. Such an information can be enhanced by the information about the link quality and forwarding node capability obtained through additional mechanisms. The *topology manager* can attain the knowledge of the overall network topology by simple subscribing to the local connectivity information.

4.2 ARCHITECTURE DESIGN AND PROTOCOLS

Our intra-domain topology management is based on the instance of the information-centric architecture and corresponding prototype implementation, namely PSIRP architecture [78] (see Section 2.3.6 for more details). The [78] prototype provides the necessary modules for performing publish-subscribe operations, such as publishing the content within a scope and subscribing to the information of interest. The implementation of these concepts is the starting point for creating a publish-subscribe based systems and building dynamic mechanisms such as topology management. The PSIRP [78] implementation has been developed in a collaborative project, in which the author has been actively participating. Hence, this platform is used as a basis for design, implementation, and testing of our topology management.

We followed the approach by which all the communication is carried out within predetermined scopes. Such scopes serve as a data aggregating mechanisms. Therefore, semantically similar data is grouped in a common scope which facilitates the information search and access. Additional identification of particular piece of data is obtained through the rendezvous identifiers, which combined with relevant scope identifier uniquely appoints every information item.

Our topology management takes advantage of the scoping concepts by separating different types of topology relevant information to be carried out within different scopes. In other words, in order to facilitate acquiring of data needed for delivery path calculation we distinguished the data flows carrying particular information, e.g. *Hello messages* and *Link State Advertisement (LSA)* messages into corresponding scopes. Once the sufficient amount of the topology knowledge is obtained by means of publications and subscriptions to the relevant scopes, Dijkstra's algorithm is performed for the shortest paths calculation.

The main functionality of topology management is divided into two concurrent operations carried out on the separate modules, namely *connectivity helper* and *topology manager* entities. Such modules can simultaneously coexist on any node. The connectivity helper module runs on each node and is responsible for discovering of local connectivity information. It collects the data about node's existing neighbors and established connections. The topology manager module gathers such an information assembling it together in order to form a picture of the network topology within the domain of operation. The topology manager module is additionally responsible for computing the optimized data delivery paths and publishing that information towards the nodes involved in the final data forwarding.

The connectivity helper module publishes periodically the updates on the existence of the node in the form of *Hello message* within predefined Hello scope. Simultaneously, connectivity helper module on every node is constantly subscribed to the *Hello messages* coming from surrounding nodes. Based on received *Hello publications*, each node is discovering its neighborhood and parses such an information into the list of nodes in the range. Acquired information is equivalent to the *Link State Advertisement* which is necessary for building the overall picture of the topology and computation of shortest paths between the nodes.

After obtaining the list of neighboring nodes, each connectivity helper module publishes this information in the form of new *Link State Advertisement* message generated within predefined scope. The *Link State Advertisement* publications can be issued periodically or asynchronously after the recent list of attainable neighbors has been changed. In other words, new LSA messages are published every time the current topology changes, e.g., new node becomes available, or the old one, already existing in the list disappears (does not publish *Hello message* within predefined period). Topology manager module continuously subscribes to the link state information based on which the path computation is performed. An important difference of our topology management with respect to the current Internet and its IP protocol is that all nodes (not only the routers) need to report their link state since the topology management calculates the routing paths between arbitrary pair of nodes (and not only between the routers).

Optimization of the basic topology management functionality can be achieved through the set of helper functions, e.g., link state and application helper modules, targeting different network operational areas. By collecting the information about network states, link properties and specific application requirements topology management gains important knowledge on current network conditions. Incorporating this information into the path generation algorithm leads to more intelligent routing decisions increasing overall performance and saving network resources.

Therefore, each node can run link state helper module which administers the table of links together with related additional information currently available. Details about relevant link properties are provided by physical level helper functions. Such helper modules continuously monitor the link properties and distribute the available updates on the link states. Such an information serves as a valuable input for optimization of data delivery paths.

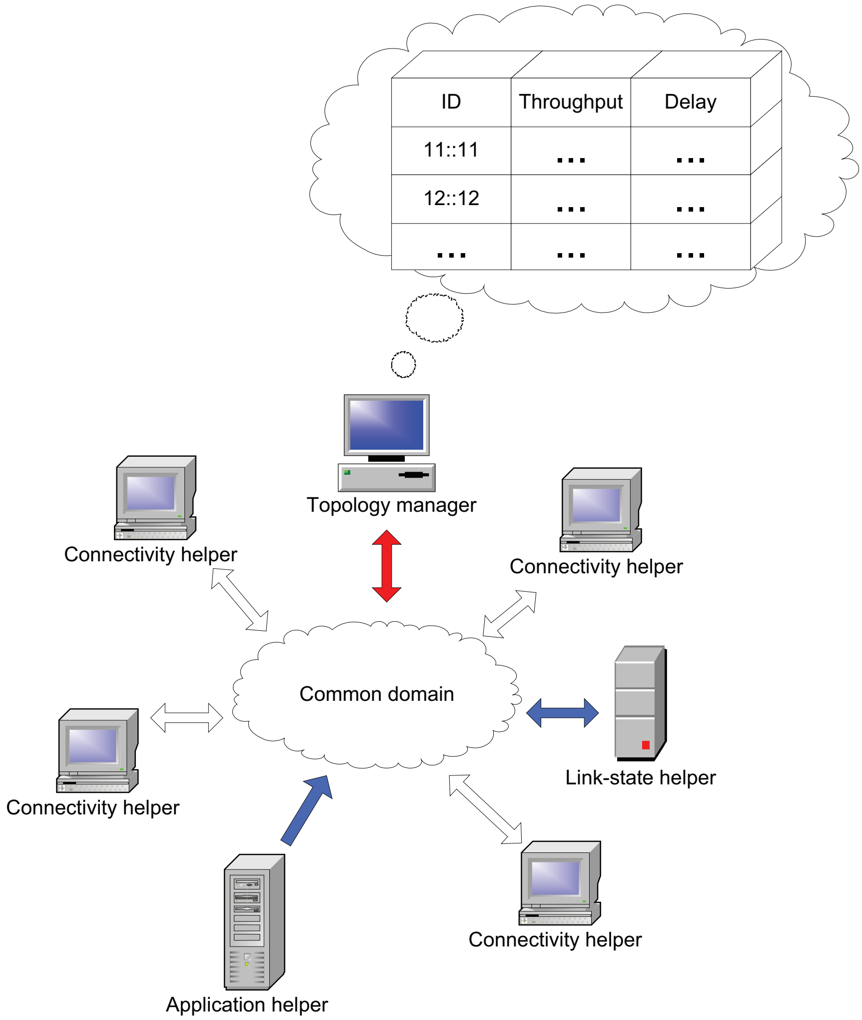

Figure 4.1: Topology management architecture. The connectivity helper and link state helper collect the network state information as an input for the topology manager, whereas the application helper provides additional information valuable for optimization of forwarding paths.

One of the simplest examples of a helper module supported by our topology management is a module responsible for obtaining the information about a link delay. The delay information is acquired using simple time stamps of publications. In order to analyze the impact of different application requirements on topology management functionality, we additionally implemented an application helper module. This module specifies the desired application conditions, e.g., maximum tolerable delay, and publishes this information in the network. Combining such information is crucial for improving shortest path computation. Figure 4.1 illustrates the topology management architecture and components.

The shortest paths generation is performed based on Dijkstra's algorithm [115, 116]. According to additional information obtained through the system of helper modules, topology manager assigns corresponding weights to existing links.

The exact appointed weight value can vary with respect to implementation details, but it always heavily relates to current application requirements and link conditions. An application requiring certain link properties specifies a threshold necessary for satisfying its requirements. This input is of high importance for the path calculation algorithm since it, together with current link conditions, directly influences the link weight assignment.

As a final outcome, the topology manager module performs a forwarding tree and zFilter [117] creation using simple shortest hop-count spanning tree formation. Resulting path is created in the form of the next hop nodes on the path from the publisher to the subscriber. Such a list is translated into the zFilter to be used by forwarding elements, inter-domain topology or other relevant functions.

The zFilter is equivalent to the Bloom filter (BF) structure [118]. It is created by performing Boolean OR operation over outgoing interfaces of nodes on the path from the publisher to the subscriber. As a final step in forming the zFilter the subscriber's identifier is added by means of logical OR operation. More details regarding the zFilter creation are provided in Section 4.3

4.3 CREATION OF FORWARDING PATHS

Our topology manager builds upon the Bloom filter inspired port forwarding mechanism (present in PSIRP network architecture) which is well-suited for flat identifiers. The Bloom filter is a mechanism for representing a set of n' blocks of data $Data = D_1, D_2, ..., D_{n'}$. One of the main benefits of such data structure is in fast and easy testing of membership of some particular piece of data D_i, where $i = 1...n'$, to a whole *Data*.

The Bloom filter data structure is created by mapping the individual data blocks D_i to a bit vector of the size k' by applying the set of m' hash functions over the data blocks, i.e., $H_1, H_2, ..., H_{m'}$ [119]. Every hash function applied over individual data block results in a number $r_j = H_j(D_i)$ that determines the position of bit 1 in the final bit vector representing the data block. The final Bloom filter corresponding to the whole data set, $Data = D_1, D_2, ..., D_{n'}$, is generated by applying bitwise OR operation over all bit vectors representing the individual data blocks D_i, where $i = 1...n'$.

Figure 4.2 illustrates the generation of Bloom filter for n' blocks of data $Data = D_1, D_2, ..., D_{n'}$. Four hash functions are applied over D_1 data block. The result of this operation sets the bit 1 in particular places (1., 3., 4. and 6.) in the bit vector. The final Bloom filter is generated by applying bitwise OR operation over all bit vectors corresponding to the data blocks D_i.

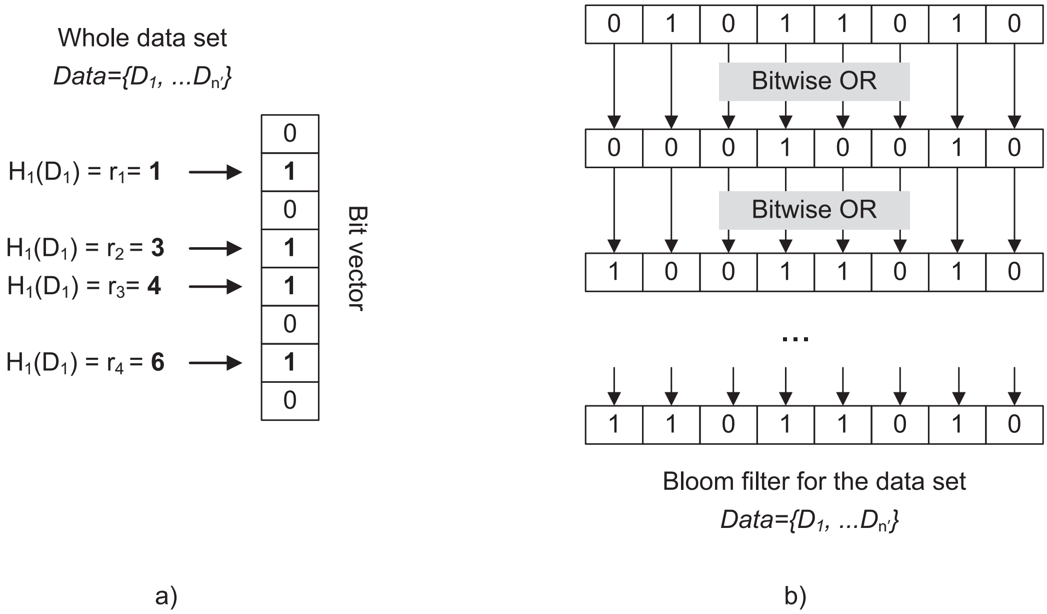

Figure 4.2: Bloom filter creation: a) The hash functions are applied on the data block in order to get the bit vector as a representation of data. b) The bit vectors of all data blocks are ORed together in order to form the final Bloom filter.

Belonging to particular Bloom filter is easily checked by applying bitwise AND operation between the Bloom filter and the bit vector corresponding to the searched data block. If the result of such an operation is not equivalent to the applied bit vector, the corresponding data block does not belong to the examined data set. Otherwise, the membership of the data block in the whole data set is confirmed.

Such a simple mechanism for determining the data membership is prone to variety of problems, especially the false positive errors during the checkup procedure. Due to such an error a membership of a data block in a Bloom filter can be confirmed, without the data block being contained in the examined data set. An illustration of a false positive problem is given in Figure 4.3. In order to improve the BF functionality different mechanism have been proposed [120, 121, 122, 123].

In essence, the main idea of the forwarding engine that we base our topology manager work on is to incorporate the Bloom filter-like forwarding state into the packets themselves. Following a source routing model the explicit definition of forwarding instructions is inserted in the packet structure. As an outcome of such a routing scheme, the forwarding nodes on the path from the publisher to the subscriber only need to check the membership of their local link identifiers in the source route in order to make the decision on the next hop forwarder. However, in such a routing model various problems might

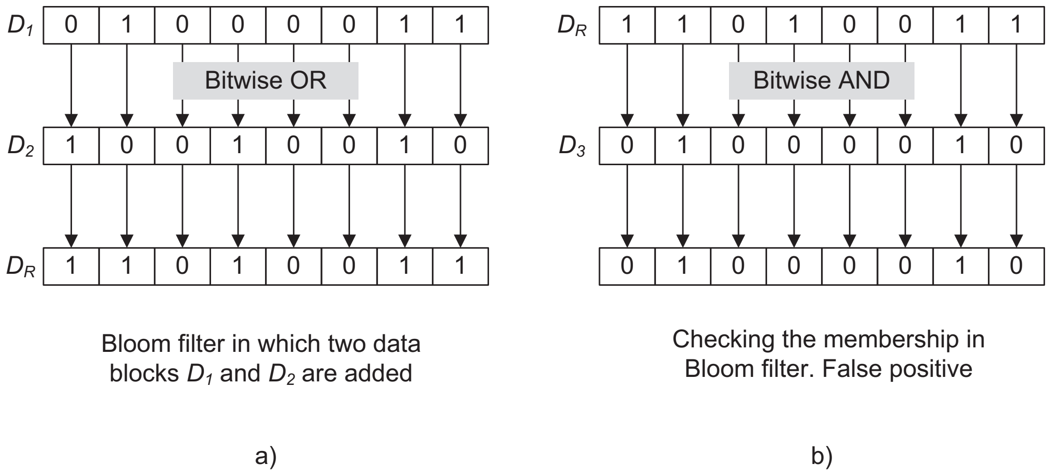

Figure 4.3: Determining the membership of an element in a Bloom filter. a) The resulting Bloom filter BF is created by adding two data blocks D_1 and D_2. b) Testing of a data block membership of a data block D_3 in the BF results in the positive outcome, although the data block D_3 is not contained in the BF.

emerge, e.g. forwarding efficiency, duplication of packets over unnecessary links. Furthermore, different parameters need to be taken into consideration for an optimal performance, e.g. network topology characteristics and amount of in-packet and in-network states.

Within PSIRP framework every forwarding element associates a Bloom filter per output link. Such a Bloom filter serves as an unique identification of the outgoing interface of the forwarder. The topology manager generates a Bloom filter-like forwarding source routing labels based on the output links of the forwarders on the path from the publisher to the subscriber. The lookup procedure carried out by the forwarders on the path returns always the inserted outgoing interface value, if the particular outgoing interface is originally contained in the forwarding label.

However, in some cases, due to the false positive problem, the lookup procedure causes additional unneeded forwarding operations by returning the outgoing interface which did not exist originally in the routing label. Such forwarding unnecessary overloads the network and might cause security problems. Thus, various mechanisms have been developed in order to diminish the false positive effect and forwarding over wrong links. Nevertheless, due to the content-centric model used, some of the redundant traffic can be used to cache the content in the network. Such a content can be easily used for facilitating the data retrieval in the case of the future subscriptions to that particular content.

The main requirement of such Bloom filter-based source routing is efficiency in terms of needed memory in forwarding elements, as well as efficiency in terms of processing, especially lookup operations. By using Bloom filters the source routes are kept compact, i.e. there is no increase in the route size as the number of hops between publisher and subscriber is increasing. Furthermore, the outgoing link identifiers are not explicitly revealed, as well as the number of hops and their exact sequence stay secret. Such an approach naturally minimizes the amount of unwanted traffic and ensures more control on receiver's site.

Figure 4.4 illustrates creation of forwarding label in the topology manager, as well as its processing by the forwarders on the path from the publisher to the subscriber. In order to encode delivery trees, the topology manager uses a set of statistically unique links on the path from the publisher to the subscribers and OR them together in order to create the Bloom filter-like identifier, i.e., zFilter [117]. Therefore, any forwarding tree represents a set of unidirectional links.

Upon the delivery path is calculated in the topology manager, it is incorporated in the packet header and sent out to the rest of the network. Each forwarding node in the network, after receiving a packet, can check if some of its outgoing interface IDs is belonging to the forwarding ID from the packet header. If the membership is confirmed, the packet is forwarded over the corresponding interface, otherwise it is dropped in order to avoid unnecessary traffic. The lookup procedure for determining the belonging of an outgoing interface in the forwarding ID in the packet header is done by simple Boolean AND operation. Thus, the processing requirements for performing such a check are minimized. Additionally, the forwarding operation is executed rapidly.

4.4 IMPLEMENTATION DETAILS OF PROTOTYPE PLATFORM

The core part of the topology management implementation is composed out of 8 Python modules with approximately 3000 code lines in total. We used the open source PSIRP prototype Blackhawk v0.3.0 [124] as an underlying framework for our implementation. In the common scenario of intra-domain publish-subscribe communication, connectivity helper modules are periodically advertising the information about their existence in the form of *Hello publications*. Such an announcement is carried out under predefined scope (Hello scope). *Hello message* contains the unique identifier of connectivity helper, together with the list of its interfaces. This information is obtained during the initial network configuration phase or throughout the attachment procedure afterwards, i.e. while the new node is joining the network.

At the same time, each connectivity helper is continuously subscribed to the same information coming from neighboring nodes (by subscribing to Hello scope). Hence, the connectivity helper collects the data about its neighborhood. After the connectivity helper acquires the knowledge about its surroun-

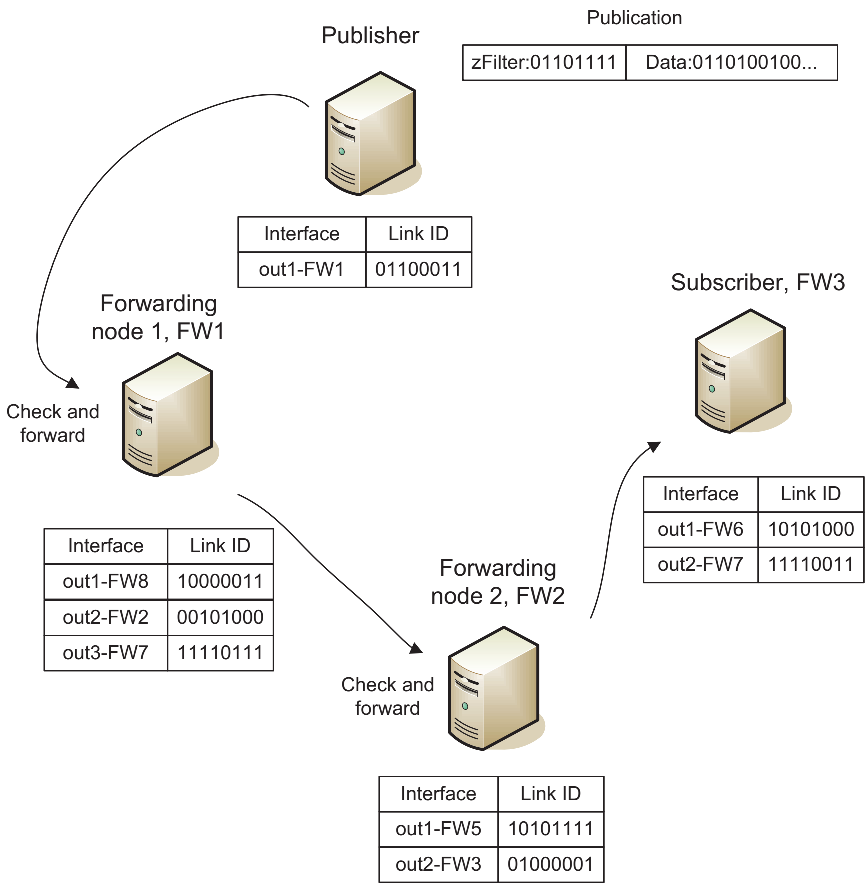

Figure 4.4: Creation of forwarding label in the topology manager, and its processing by the forwarders on the path from the publisher to the subscriber.

ding it distributes this information by publishing it under predefined scope (LSA scope). This publication contains the list of neighbors' IDs together with corresponding interface IDs. This data is crucial for topology manager module in order to perform the data delivery path calculation.

In order to receive topology information from all nodes in the network the topology manager module continuously subscribes to the LSA scope. The link state helper module distributes the information about current link conditions in the form of a publication under dedicated scope (Link scope). The publication is formed out of the link IDs followed by relevant link parameters and their present values. This information represents summarized data about network conditions, gathered from multiple physical level helper modules.

Physical level helpers are integrated with the connectivity helpers. Their main role is to continuously monitor the link properties, e.g. delay, and publish the updates within assigned scope (Phy scope). Concurrently, the application helper module informs the system about application requirements by publishing the message under predefined scope (App scope). The application requirements are represented by the list of relevant link parameters with corresponding thresholds. Therefore, the topology manager is able to constantly monitor the current network conditions and application requirements by subscribing to corresponding scopes (Link and App scope).

If no additional information is available, the data delivery paths are computed based on the simple shortest distance rule. However, the data obtained from relevant helper functions is of great importance for the path optimization process. Combined with specific application requirements the knowledge of link properties will dictate the assignment of link weights. This will improve path generation and ensure more efficient routing decisions. According to the additional information about network conditions and application requirements the topology manager applies corresponding algorithm for assigning link weights. Based on such input, the generated data delivery paths will be optimized for the given network conditions and running applications. Finally, obtained paths are translated into the form of zFilters generated by ORing all outgoing interfaces of the nodes on the path together with the node identifier of the subscriber.

The topology manager module also monitors the status of the matching interests on the rendezvous module. The rendezvous module keeps the record of existing publications and subscriptions in the network and performs the pairing between them. After the match between the interests of publisher and subscriber occurs the rendezvous system requests the creation of forwarding paths on topology manager module. The path calculation is triggered by publishing the rendezvous request. This publication contains the publisher ID and the list of subscriber IDs for which the match occurred.

As a final outcome, the topology manager module generates two types of forwarding identifiers, namely Forwarding ID (FId) and Metadata ID (MId). The FId represents the path from the topology manager module to the publi-

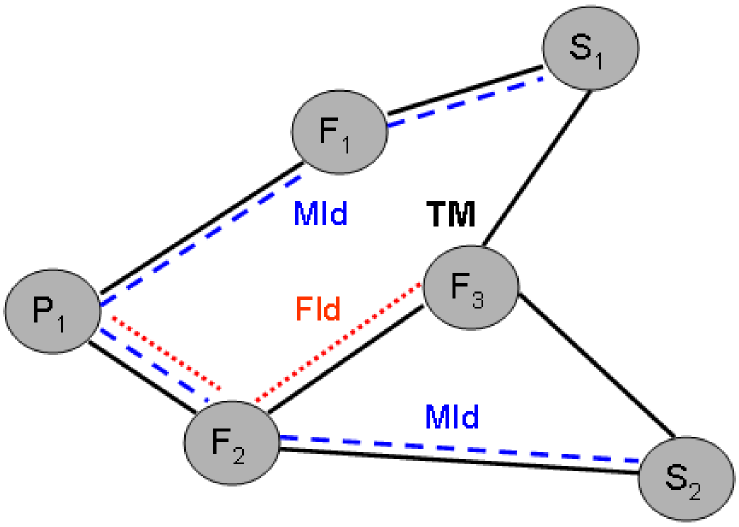

Figure 4.5: Forwarding paths creation. The topology manager creates two types of forwarding identifiers, i.e. Forwarding ID (FId) and Metadata ID (MId).

sher. This path is used for informing the publisher about the actual path over which the real data should be sent, i.e. the path from the publisher to the subscriber (MId). After receiving the MId path from the topology manager over the FId path, the publisher is able to direct requested data to the subscribers.

Figure 4.5 shows a simple example of topology management functionality. All the nodes in the network run topology connectivity helper module and can provide forwarding services for the packets. Apart from connectivity helper module, one node in the network runs topology manager module (denoted as F_3) and has capability of performing the optimal path calculation. Upon receiving the instruction from the rendezvous system about current publishers (P_1) and subscribers (S_1 and S_2) the topology manager module running in node F_3 calculates the set of forwarding identifiers. There is always one FId and MId generated corresponding to occurred publication-subscription match at the rendezvous. In the case of multiple subscribers the paths from the publisher to separate subscribers are merged in single MId by simple OR operation.

4.5 PERFORMANCE EVALUATION OF TOPOLOGY MANAGEMENT

Utilizing the publish-subscribe communication model in network discovery and topology creation procedure introduces certain delay that needs to be considered. In order to estimate such a delay we analyze the latency of messages essential for network discovery, e.g. *Hello messages*, by measuring the time difference between actual generation of publications and their reception at the user level after corresponding subscriptions. The measurements are performed in the testbed of machines running the PSIRP prototype, where experiments are repeated 5 times. Figure 4.6 shows the average delay of *Hello messages* [125]. For the larger network size the *Hello message* distribution experiences higher delay due to the network congestion. As more publishers and subscribers join into the network, the number of information items increases, as well. Under such conditions the rendezvous, topology and forwarding functions of the prototype need to handle more load and thus the slight performance drop occurs. However, the exact experienced delay heavily depends on the network conditions and link properties. Nevertheless, the message delay due to the network growth does not critically hinder the overall network operation.

Another interesting objective to investigate is the delay introduced by the computation process for generating different data delivery paths. Our topology management implementation requires creation of two types of data delivery paths, namely FId and MId identifiers. Given that the rendezvous request can contain the list of subscribers, if multiple sites expressed their interest in receiving a particular data item, the final MId will be obtained by ORing of MIds for individual subscribers. As the actual MId can be composed of the number of intermediate MIds for each of the subscribers, its calculation time is expected to be longer with respect to FId calculation.

Figure 4.7 illustrates obtained results of path calculation latency measurements. The showed delay includes also processing time of incoming rendezvous request. Computation of MId requires more time since more information needs to be processed. Prior obtaining the final MId identifier, the algorithm needs to wait for all related forwarding identifiers to subscribers as input parameters. Such path creation algorithm is inevitably connected with certain computational delays. Furthermore, increase in network size entails additional processing load, thus, slight performance degradation is expectable.

Our computation mechanism for forwarding states generation heavily relies on weighted Dijkstra's algorithm from igraph library. Appropriate link weights are added based on the input from physical level helper functions. Due to its important role in topology management functionality, we estimate the impact of Dijksta's algorithm to overall performance, measuring the delay it introduces.

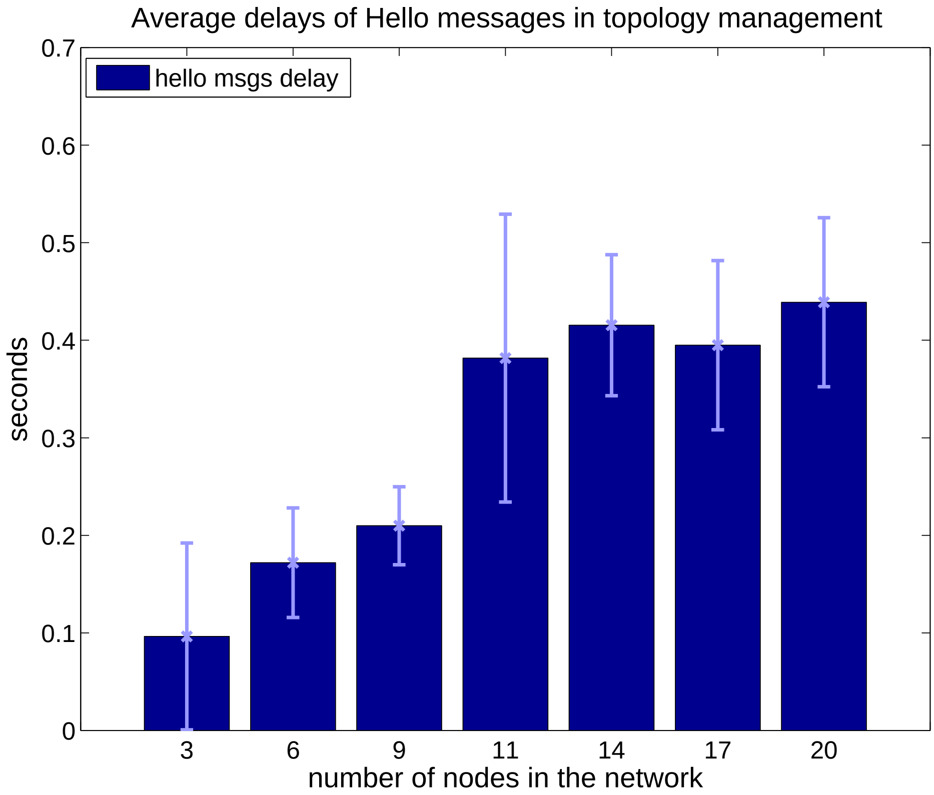

Figure 4.6: Average delay of *Hello messages* in topology management implementation.

In order to get a better insight on the influence of applying Dijkstra's algorithm in our topology management implementation we measure the time needed for execution of sole Dijkstra's algorithm over a larger network size and different link existence probability between nodes. The probability of link existence between the nodes reflects the connectivity of the network. We take into consideration a wide range of network connectivity cases in order to better capture the influence of this network characteristic on the performance of Dijkstra's algorithm. Obtained results, illustrated in Figure 4.8, are generic and demonstrate average delay introduced by execution of Dijkstra's algorithm in igraph, regardless of the implementation in which Dijkstra's algorithm is utilized. It is evident that for significantly larger networks, unrelated to the grade of connectivity the time needed for Dijkstra's algorithm processing increases. Having a larger graph of nodes requires more computational resources or, otherwise, introduces considerable calculation delay. However, the execution of Dijkstra's algorithm does not cause a significant burden to the

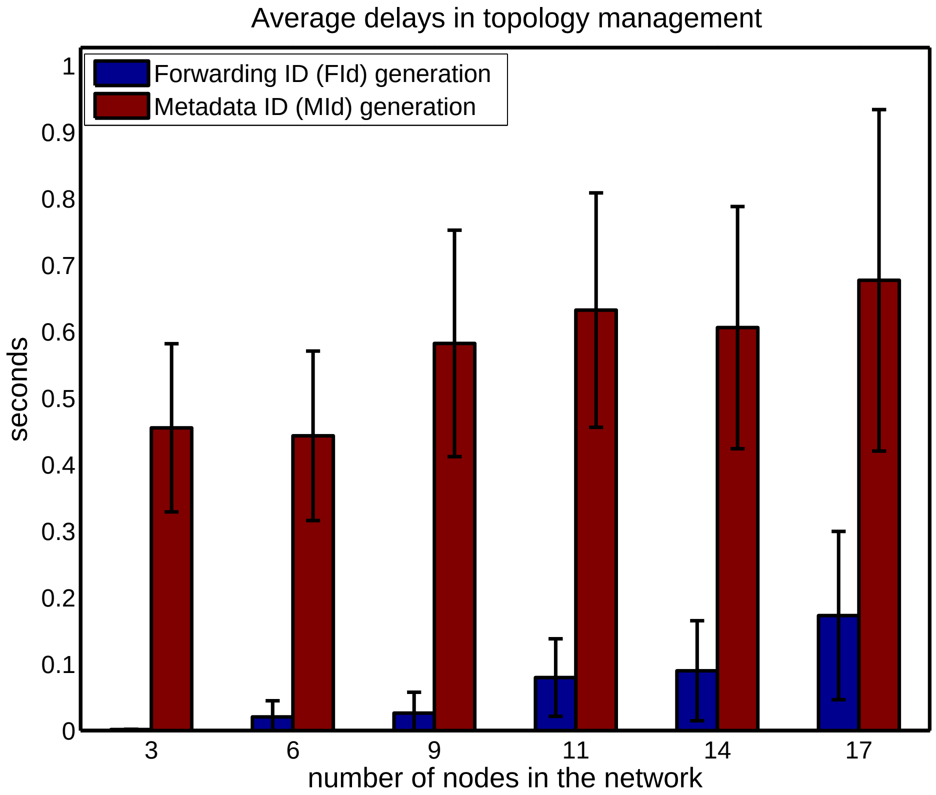

Figure 4.7: Average delay of calculation of forwarding paths, including processing time of rendezvous messages.

overall topology management functionality.

4.6 OPTIMIZATION TECHNIQUES FOR IMPROVING THE TOPOLOGY MANAGEMENT

Apart from a set of helper functions that have been discussed in this chapter there are numerous other optimization techniques that could be used to increase the efficiency of topology management. In the case of mobile nodes following some ordinary repetitive route, keeping a record on the positions of nodes can provide valuable information to topology manager module. Having the information of node's movement pattern and being able to predict its most likely future position can help generating more adequate routes and avoid their frequent re-computation. Therefore, the additional burden in the terms of frequent zFilter updates introduced by the mobility of nodes will potentially be minimized. This is particularly important and useful in the case of

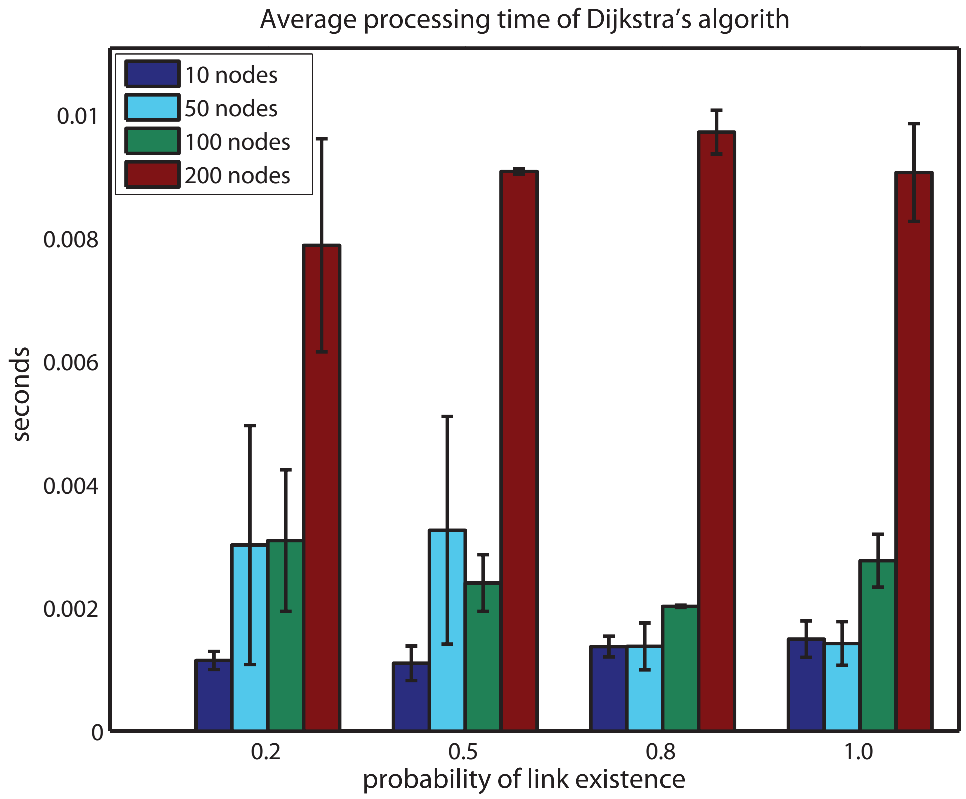

Figure 4.8: Average computing delay introduced by Dijkstra's algorithm.

highly mobile information networks and applications, such as the high-speed cellular networks.

Taking advantage of potential regularities in movement patterns will require implementation of additional helper functionalities, i.e. we need mobility helper functions. Such functions would collect the information about the current locations of nodes and estimate the future positions of a node based on the mobility information gathered over time. Such an estimation can be then distributed through predefined scope. Topology manager module being continuously subscribed to this scope will gain the knowledge of the next location of the node and incorporate it into the path generation.

The awareness of node's future positions might significantly improve the handover procedure. Especially in the case of rapidly moving mobile users the handover can be initiated well on time with high accuracy. The prediction of mobile movements has been considered also in the context of cellular networks to enhance handover efficiency, as well as many social networking applications are considering this approach. The detailed discussion about mobility helper functions in information-centric networks and their benefit for topology management operation will be presented in Chapter 5.

Furthermore, apart from application requirements playing a significant role in path generation, we can consider utilization of application policies in order to obtain optimal data delivery paths. Through the notion of policies, every application can define possible restrictions in data transfer that must be included during the path calculation. Applications embracing highly confidential data can place strict security demands which must be satisfied by traversing links. The applications can also specify the preferable or undesirable links, nodes or domains for they traffic flow. This information then serves as additional feed for path optimization.

Although potentially increasing complexity and computational needs of topology manager functionality, this is huge paradigm shift compared to the present Internet. This sort of publish-subscribe framework enables (content) policy based routing, where policies can be flexibly stated by end-users or end-subscribers. The current Internet systems have policy based routing functionality, but only in the form of external routing protocol (BGP) which is much less flexible and not directly usable by end-users. Ensuring higher granularity path selection or content based selection is not possible for end-users (publishers) by today's protocols, and special case would need handmade route selection tables, e.g. by use of MPLS.

Another important question is reliability of our topology management approach and its resilience to link failures. In the context of our zFilter source based routing, one of the possibilities for improving the network sustainability in the case of link failures is maintaining the list of backup routes. In our particular case, the model needs to generate a set of additional forwarding zFilters associated with each originally generated, optimized zFilter. In the worst case, the number of additional zFilters required for secure data transmission is equal to the number of traversing links.

In order to avoid large data overhead, the backup routes can be generated taking into account current link conditions, e.g., delay and throughput. In this case the backup zFilters would be generated to assure only "the weakest" links and spare the routing tree in the most critical points. Furthermore, in some cases, type of the link offers unambiguous information about its uncertainty. This can serve as valid input for optimization of rescue routes creation. For example, wireless links having higher failure probability can be prioritized in backup path generation over wired links.

Furthermore, different network coding mechanisms can be implemented as a helping functions of topology management. Such helper functions can be responsible for direct data dissemination as a replacement for source coding. On the other hand, the network coding can be used for improving the robustness of the actual data transmission by adding the redundancy to the transferred data. A detailed discussion on network coding helper functions, their applicability, implementation and benefits will be presented in Chapter 7.

4.7 CONCLUSIONS

In this chapter we have presented the intra-domain topology discovery and generation module, entirely relying on publish-subscribe communication model. We have developed the networking framework in which all nodes utilize publish-subscribe pattern for information exchange. Furthermore, the topology management follows the same approach in collecting network related information and building topology states. We illustrated average latency of applying publish-subscribe model in intra-domain topology creation. Obtained results are separated into two major categories: latency introduced by data dissemination, e.g. *Hello messages* delay, and the latency introduced by path computation. With respect to our implementation we presented two groups of forwarding identifiers, discussing the delays encompassed with their creation.

The work described in this chapter builds a background on the use of topology management in information-centric networks. We have made a fully functional reference implementation, which has been used to verify the design concept and to test performance of helper functionalities based on publish-subscribe communication. This work shows that the topology management helper function, as a crucial part of the system, can be implemented efficiently for publish-subscribe networks.

Our results demonstrated that publish-subscribe communication model can be used not only for content retrieval as already present in many Internet applications, but also for more fine grained network function such as topology generation. Different improvements to the current topology management implementation can be introduced as discusses in Section 4.6, leading to more efficient topology creation.

5

Mobility prediction in information-centric networks

In this chapter we examine the effect of mobility on information-centric network. We focus on the publish-subscribe based topology management and the computation of the data delivery paths as one of the critical operations affected by the mobility. We use IEEE 802.16 (WiMAX) as a reference architecture and enhance it with publish-subscribe features. As IEEE 802.16 and IEEE 802.16m are very similar architectures for the present day data oriented cellular networks, e.g. LTE and LTE-A, using this reference architecture has been done without losing generality to interpret these results also in the context of general cellular networks.

In order to optimize the network operation we enrich the topology management functionality of the publish-subscribe network with a set of mobility prediction mechanisms. We elaborate various factors that influence the performance of the prediction function. We address the critical aspects of applying the prediction function and discuss the impact of such aspects on overall system performance. Moreover, we examine the trade-off between the advantages offered by mobility prediction in the publish-subscribe context, compared to the signaling and complexity overhead.

5.1 The motivation

In order to address the mobility issues a variety of protocols have been introduced in the present IP-based Internet [36, 126, 127, 128, 129, 130, 131]. However, the tight connection between addresses and locations imposes the need of rather complex mechanisms in order to achieve satisfying performance of the system. Due to its data-centric characteristic the publish-subscribe communication pattern provides a firm basis for mitigation of mobility issues. Therefore, a special attention has been recently devoted to mobility support in information-centric networks.

Although appearing as a natural solution for majority of mobility problems, the publish-subscribe communication is encountering a variety of its own problems. Therefore, the issue of dynamic networks and frequent change of topology has to be addressed from various perspectives. One of the challenges is in finding the solutions for increased data delivery latency due to the frequent rebuilding of the delivery tree [132]. Furthermore, in the highly dynamic scenarios the presence of frequent handovers can substantially hinder the operation of an information-centric network [95]. As a possible remedy for mobility problems the caching of data can be introduced, i.e. the lost data due to the mobility can be retrieved from the closest cache [133]. Furthermore, the social relations between the users can be utilized for improving the routing under dynamic conditions [134].

Due to the mobility the data delivery paths might alter, even during the transmission phase, which consequently can cause significant packet loss. Given the structure of considered information-centric network the central entity responsible for the creation of data paths is topology management. Hence, optimization of data delivery paths generation in topology manager is a critical operation since it directly influences the efficiency of the entire system. The path from the publisher to the subscriber can be optimized with respect to a wide range of network parameters, thus it can affect the performance of the system in different ways.

In mobile environments the network topology changes frequently. The nodes come and leave and the links between them alter correspondingly. The topology manager needs to keep track of all modifications in the network configuration and to update the forwarding paths accordingly. This is not a trivial task, since the network changes can occur frequently and cause severe interruptions to the data transfer unless the topology manager comprises the mechanism to prevent it. Thus, the topology management in publish-subscribe architectures represents the key point for handling the mobility-related events.

Due to its highly important role in publish-subscribe architectures we implemented the topology management on top of PURSUIT network architecture and investigated the different optimization techniques for improving its functionality in mobile environments. We aim at utilization of inherent publish-subscribe data-centric characteristics, in particular the flexibility of providing the same data from different locations. The users are not restricted to data delivery from a single and unchanged network entity throughout all the communication, but the data can be fetched from any point in the network, currently the closest or the most suitable data source. This gives a freedom to the system to alter the data suppliers according to the current network state and the position of users. The possibility of such dynamic changes enables a better utilization of network resources and decrease of data latency.

Furthermore, we focused on employing the prediction techniques for the user movements. By collecting the information related to the user flows, it is possible to gain knowledge about the usual movement pattern and to use it for forwarding paths creation. Instead of ordinary re-computation of forwarding paths upon the topology changes due to the node's mobility, our objective is to secure the communication in such a dynamic environment by generating backup forwarding paths and bi-casting the data based on the prediction of node's next position. Moreover, we optimize the prediction algorithm by dynamically changing the prediction-relevant parameters according to the current network state.

5.2 SELECTION OF EVALUATION TOOL

We have utilized the ns-3 network simulator [135, 136] for development of our topology management model equipped with mobility prediction functionality. Ns-3 is an open source discrete-event network simulator designed mainly for networking research and education. It is publicly available, free software, which has encouraged a large portion of research community to use this tool. As the evolution of ns-2 simulator [137], it is widely adopted and known[1]. In the following we shall give a short overview of the ns-3 design, and discuss its applications within our work.

Ns-3 code is completely written in C++ with optional Python bindings and it uses WAF Python based building system [138]. For tracing, ns-3 provides pcap file support for every device in the simulation, which can be easily displayed using common visualization tools [139].

For the basic computing device abstraction, i.e. device that connects to a network, the ns-3 uses the general term *Node*. The Internet nodes are designed in a way that faithfully represents real computers including key network abstractions, e.g. sockets, network devices, multiple interfaces for nodes, using IP addresses. The *Node* is connected to the network through the *Channel*. The *Channel* in ns-3 can model simple point-to-point connection, but also more complex wireless or Ethernet channels.

As in realistic scenario, hardware components do not work properly without suitable device driver. The ns-3 provides similar abstraction in simulation space. Such an abstraction is identified as *NetDevice* and it acts at the same time as a software driver and simulated hardware. Thus, the ns-3 *Node* is able to communicate with other *Nodes* by using different *Channels* and installing proper *NetDevices*. One of the most important abstractions in ns-3 space are *Applications* which drive the execution of ns-3 simulation. Additionally, ns-3 provides *Helper* functionality in order to facilitate creating and connecting network components, i.e. *Nodes*, *NetDevices* and *Channels* in proper way. Figure 5.1 illustrates the basic ns-3 data flow model.

A wide area of different network features has been already supported by ns-3, e.g. Wi-Fi 802.11 models, IPv6 addressing, and bridging different LAN segments. Due to the large user community the ns-3 core is developing rapidly.

The focus of the ns-3 development is on providing a simulation environment, which should be as realistic as possible. In order to achieve this aim several emulation mechanisms have been implemented. The network simulation cradle allows real world TCP/IP stack to be used in simulations. Using

[1] In practice, ns-3 could be also called a re-designed version, since even the simulator core is mostly redesigned and reimplemented.

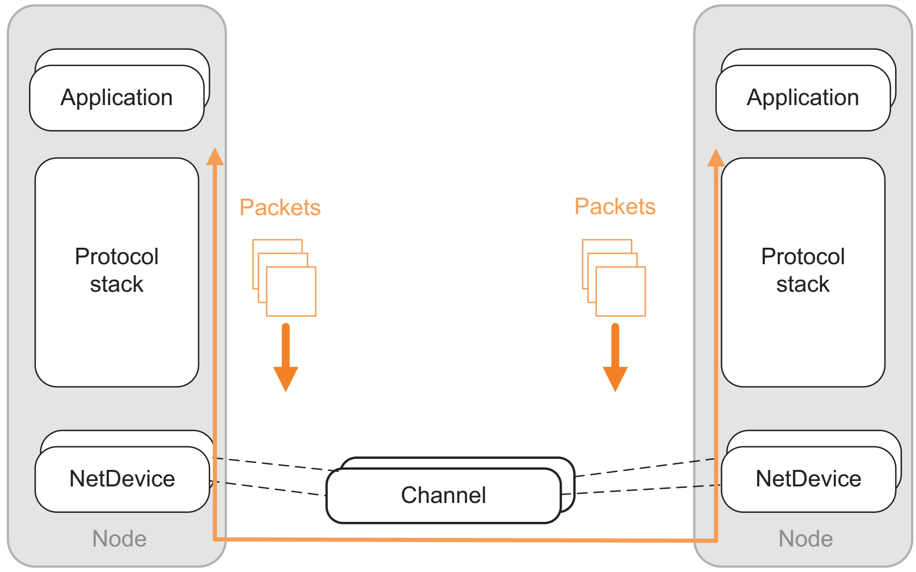

Figure 5.1: An illustration of the basic ns-3 data flow model.

the tap devices the real hosts are allowed to exchange the information via simulated network, whereas the network emulation devices allow simulated nodes to communicate over real links. This sort of ”hardware-in-the-loop”, or ”network-in-the-loop”, approach enables highly realistic simulations and validation of different concepts.

Ns-3 is easily extensible due to its object oriented nature. The desired extensions are generally done by defining entirely new classes, through inheritance or class aggregation. Defining the new protocols is facilitated through inheritance of default protocol classes and redefining their main methods. Therefore, ns-3 is as a good choice for evaluation of new architectures, which is the core aim of this thesis work. By enabling simple ways of making entirely new protocols, and not relying on existing TCP/IP implementation, the ns-3 allows us to easily test our developed models. Therefore, for the rest of the simulation work carried out within this thesis, we aim at utilizing the aforementioned benefits of ns-3 simulation tool.

5.3 MODEL DESCRIPTION

We have developed our topology management model equipped with mobility prediction functionality within the ns-3 simulation. In an initial network setup we deployed 256 base stations covering an area that represents metropolitan area and 256 mobile stations staying within their communication range. In fact, the main test case has been a real base station distribution that has been acquired from a major mobile service operator in the USA.

The base stations are characterized by their positions and the coverage area. Assuming the base station position to be x_B, y_B and the position of the mobile node x_M, y_M, the distance between the base station and the mobile station is $d = \sqrt{(x_B - x_M)^2 + (y_B - y_M)^2} = \sqrt{(\Delta x)^2 + (\Delta y)^2}$. Therefore, the probability of a node being in a coverage area of the base station is $P_r\{d \leq R\} = P_r\{\sqrt{(\Delta x)^2 + (\Delta y)^2} \leq R\}$, where the R represents the radius of the base station's coverage area. The mobile nodes are uniformly distributed over the coverage area of the base stations. Figure 5.2 illustrates the used network topology. Our network is composed of 256 base stations, numbered from 0 to 255. The data transfer rate along the base station connections, was set to 1 Gbps. In the case of wireless connections, the speed of communication is dependent on, e.g. the number of users in the channel or the propagation model. The minimum rate guaranteed for a WiMAX connection has been set to 10 Kbps.

In order to achieve the simulation setup that closely resembles real network scenarios, every fixed link between base stations encounters a transmission delay, which is initially set to 20 ms. The links between base stations can be seen as a logical links for which we describe higher delay as the first order approximation. This delay can be changed during the simulation, in order to simulate overloaded or congested zones within the network. The transmission delay over wireless links is assigned according to the applied propagation model and the distance between the mobile and the base station during the communication. In the initial simulation setup the COST231 [140] propagation model has been used.

The data transmission between mobile stations is conveyed in a publish-subscribe manner. Once the match between published and subscribed data occurs in the rendezvous, the topology management constructs the routing path from the publisher to the subscriber and signals it to the forwarding modules, which are thereby instructed how to transfer the data. The forwarding modules are capable of caching the content traversing the network. Therefore, the lost data can be easily obtained from the closest cache [81].

As discussed in Chapter 4 one of the main problems of Bloom filter-like forwarding paths is the possibility to have false positive errors. Such errors might lead to packet dissemination over incorrect links, i.e. data being received by unforeseen nodes, and thus generating unnecessary extra traffic load into the network. False positives can also produce undesired loops in the transmission path, causing an infinite transfer of a packet throughout the network unless there are safety mechanisms to stop this.

In order to diminish the possibility of false positive generation we aim at creating a set of redundant link IDs associated to each forwarding link as described in [117]. In addition to the regular link ID (LId), a number of link ID

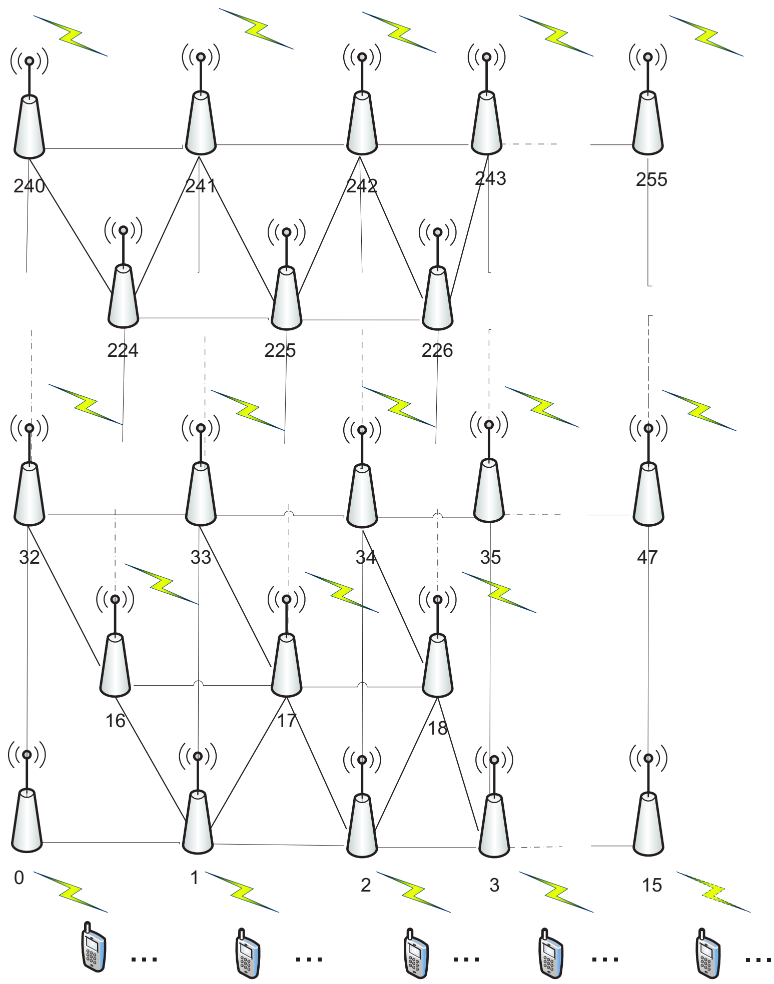

Figure 5.2: Illustration of the network topology.

tags (LIT) are assigned to the LId. A set of T LITs is calculated by applying hash functions over corresponding LId. Therefore, instead of each link being identified with an unique link ID, a set of T different LITs will be additionally associated. In the process of zFilter creation, such addition will enable generation of diverse candidate zFilters for the specified path. Every candidate zFilter will be stored in a forwarding table, each containing the LIT entries of the active link IDs. In the process of creating the zFilter the best performing candidate with respect to probability of false positive occurrence will be selected as a final zFilter. Additionally, other network parameters can be taken into consideration in selection of the best performing zFilter. Figure 5.3 illustrates the creation of T different LITs associated with one LId.

In order to study the dynamic scenarios, we have varied the speed and trajectories of nodes in our simulations. We have considered all the major velocity classes for our nodes, namely nodes with walking speed and vehicular nodes

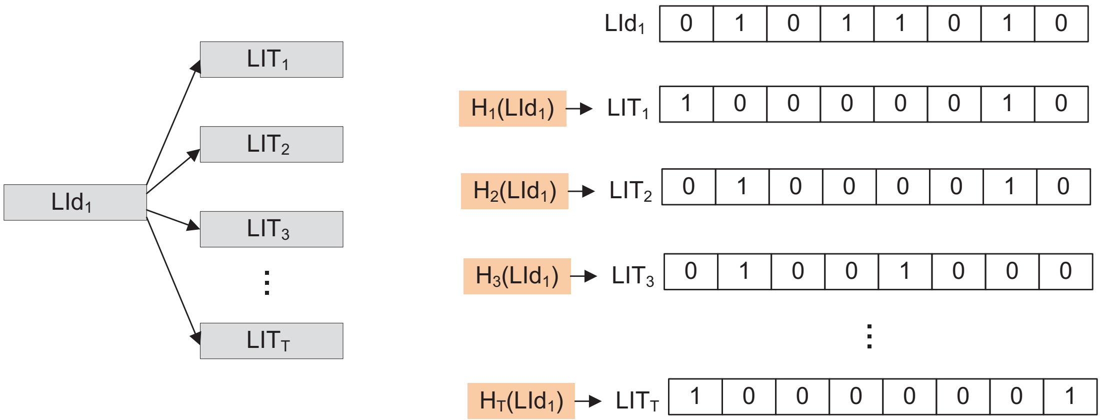

Figure 5.3: Generation of T diverse LITs ($LIT_1, .., LIT_T$) assigned to corresponding LId by using hash functions (H_i).

with both city and high-way speeds. The mobility patterns are generated randomly for a given area covered by the set of base stations, in a way that ensures handovers. The topology management keeps track of nodes' movement in the network by storing the nodes' location over time. Any regularity in node's movements patterns such as daily repetition of the trajectory for commuting users is used as an input for the mobility predictor module that estimates the possible future locations for nodes. Being equipped with such a movement database, the topology management is able to construct the backup path that indicates the routing path towards the node on a predicted location.

During the handover phase, i.e. while the node is leaving the coverage area of one base station and entering the new coverage area, the data is sent to both base stations by bi-casting. This minimizes the data loss due to the handover. The backup path creation and bi-casting is triggered upon the signal power level, P_{sig}, of the mobile node received by the current base station drops below a certain level, i.e. the prediction triggering threshold. The prediction triggering threshold is slightly higher than the actual handover threshold T_{hand}, thus, the signal dropping below this level indicates a possible transition into the coverage area of another base station.

We can select different handover triggering threshold according to the velocity of mobile nodes and the type of the application. The detailed information about the velocity of nodes and used applications is stored in the topology management database. Our aim is to use this data available to the topology management for setting the most suitable triggering thresholds. The benefit of this approach is that triggering thresholds can be changed by the topology manager or can be even learned by the system in self-organizing manner. Due to its awareness of the movement patterns that nodes exhibit and the requirements of existing applications with respect to the data transfer, the topology management appears as the most suitable candidate for instructing the system about the prediction triggering levels.

Based on the current network state in the simulation scenario the topology manager sets the initial prediction triggering threshold. In order to optimize its value, the topology management calibrates the prediction triggering threshold empirically, by monitoring the system performance. During this process the topology management acquires the knowledge about the most suitable prediction triggering level, T_{pred}, for the given scenario and stores this information in its database. The calculation of the most likely future location of the node is further on initiated according to such optimal triggering level. Algorithm 1 summarizes the overall mobility prediction procedure applied in our scenarios.

Algorithm 1 Applying the prediction in mobile scenarios

```
while T_hand < P_sig do
  if P_sig < T_pred then
    Start the prediction process. Calculate the most likely future location of
    the mobile node and bi-cast the data.
  else
    Operate without the triggering of prediction process.
  end if
end while
Start the handover process.
```

We have built the system in which it is possible to avoid unreliable triggering of data bi-casting if the topology management database does not give enough input for an accurate estimation of the future positions. The desired accuracy of the prediction algorithm can be set by the user or application. In such a way, a certain predefined probability of the next location needs to be satisfied prior the backup paths are created and bi-casting is initiated.

Apart from location prediction mechanisms, the topology management inherently utilizes the information-centric nature of the system, thus, instructing the forwarding elements to always deliver the data from the closest source. This implies changing the publisher from which the data is being retrieved, if during its movement a mobile node approaches another publisher of the same content. The topology manager thus changes the traffic flow in order to optimize the data delivery. We investigate the benefits of such content-oriented topology management equipped with prediction mechanisms for nodes' locations.

5.4 HANDOVER IMPLEMENTATION OBJECTIVES

When a mobile node departs from the coverage area of the base station it was connected to, it needs to immediately establish the connection with the next closest base station, in order to avoid the breakout of the communication in process. Such changing of the serving base station due to the node's mobility, i.e. handover is considered as one of the most challenging problems in wireless networks.

The ns-3 simulation tool does not provide any implemented mechanism for handling handover event in WiMAX standard [141]. Therefore, we implemented the handover procedure that follows WiMAX standard as a part of our implementation work and made it to work in the context of publish-subscribe architecture of this thesis.

The handover algorithm we applied is in general the following. Every base station periodically informs the topology manager about the received power level from the mobile stations in the coverage area. Upon the received power level from mobile station drops below a certain threshold, the involved mobile stations check the signal power level from other base stations in the range. According to the measurements of received signal from neighboring base stations the mobile station selects to which base station to connect next, executes the handover and informs the topology manager module about the performed changes.

Consequently, the topology manager performs the corresponding operations in order to update the topology states. As we are considering the data-driven network we are not very sensitive for small extra delay introduced by mobile station based handovers. Furthermore, extending the concept to the base station controlled based approach is quite trivial.

The original WiMAX model of ns-3 [142] does not allow a mobile station to receive information from or to be associated with more than one base station. Such a characteristic is critical in the handover procedure, since the mobile station continuously receives information during this process in order to make the optimal decision.

Therefore, in order to implement an efficient handover mechanism we have modified this feature in the ns-3 core. A physical layer of an arbitrary WiMAX device communicates with the physical layer of any other WiMAX device through a WiMAX channel. Such a channel is created and owned by a base station and it is shared between the mobile stations which are connected to it.

Since in ordinary cases the physical layer connects to only one channel from one base station, this feature has been changed by supporting multiple channels accessible from one physical layer in order to facilitate the handover procedure. However, one of the channels is always set as the main communication channel.

5.5 EVALUATION RESULTS

In the following, we examine the data-centric and prediction aspects of our model concurrently.

5.5.1 *The benefit of publish-subscribe data-centric nature*

The topology management regenerates the data delivery paths as the topology changes, in order to always get the content from the closest location. As long as the number of publishers of the same content remains on moderate level, there is evident benefit in having the same data available at multiple locations. However, for a large number of publishers the frequent path recalculation and the signaling overhead due to the change of the data source poses a considerable burden on the system. This can significantly diminish the benefits of the data-centric communication. Figure 5.4 illustrates the described impact of the number of publishers on the throughput of the system and the drawback of a large number of publishers.

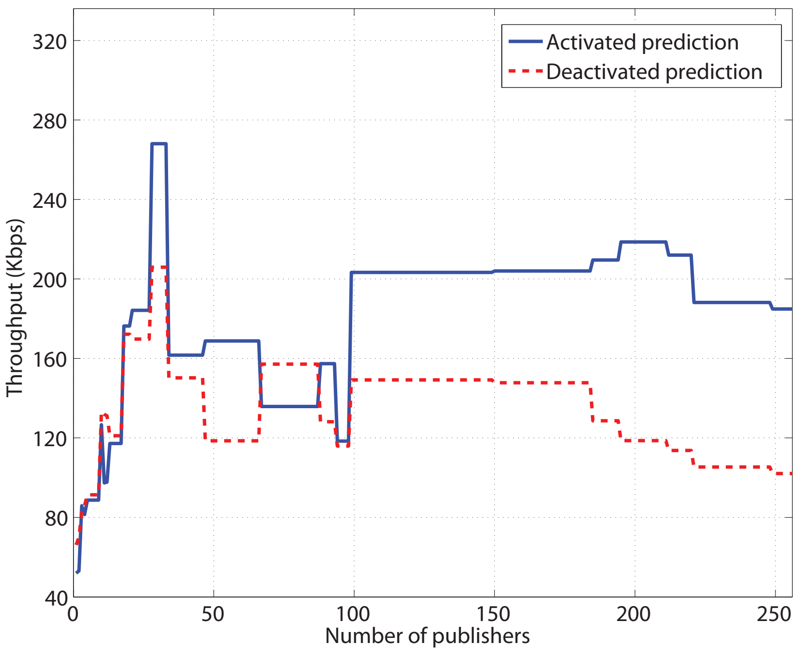

Figure 5.4: Impact of the number of publishers on the throughput of average user.

5.5.2 *The benefit of applied prediction algorithm*

With respect to the result shown in Figure 5.4, our further elaboration of topology management in a dynamic environment is focused on lower density of publishers (up to 10) which is also more likely case in practice. In order to understand the benefits of applied prediction model we performed a range of tests. It is notable that the advantage of the mobility prediction functions is strongly related to its reliability.

According to the collected user movements data, the prediction function can guess with variable probability the next position of users. By existence of rich record on movements or in the case of firmly settled regularities in the mobility patterns, e.g. daily commuting between certain places, the prediction utilization can give very accurate results on the locations. Thus, it will substantially contribute to performance improvement.

On the other hand, if there is an insufficient number of movement samples in the database or if the user does not follow any mobility pattern, the use of the prediction function can even have negative consequences, placing an additional load on the system by generating unneeded traffic.

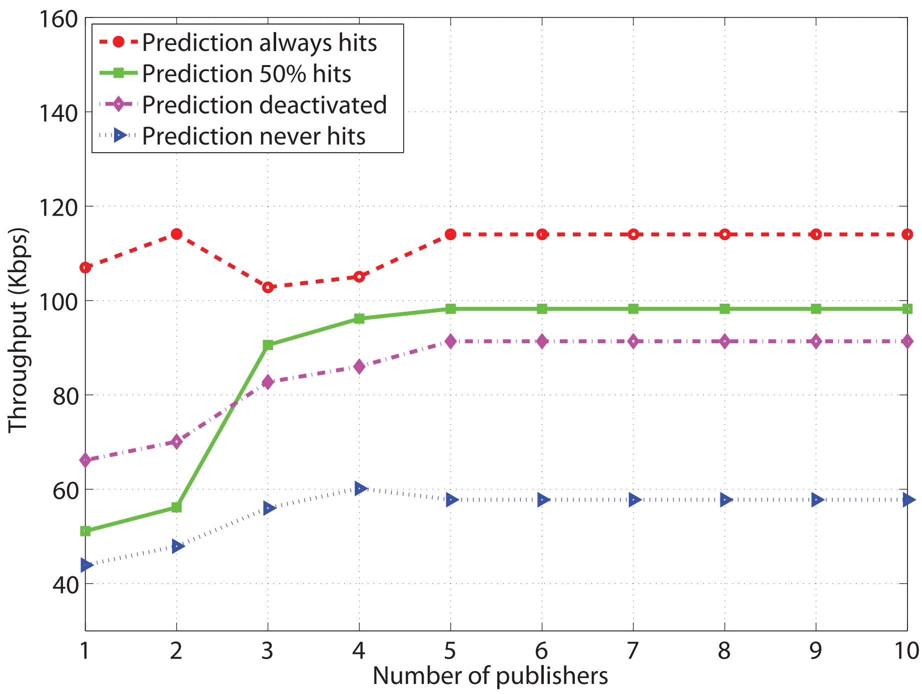

Figure 5.5: Increase in throughput due to the prediction. The speed of mobile nodes is 100 km/h, prediction algorithm is activated at T_{pred}=-89 dBm.

Figure 5.5, Figure 5.6 and Figure 5.7 illustrate the improvements in system performance with respect to throughput for different accuracy levels of the prediction function and different network setups. It can be observed that the use of prediction, even accounting for 50% of errors in data bi-casting, can lead to throughput improvement.

However, extremely frequent wrong guess of most likely future location can diminish the positive effect of prediction. This brings us to the conclusion that employing prediction is appropriate only under certain conditions and with a guaranteed minimal accuracy. However, the high prediction rate is indeed possible without any doubts.

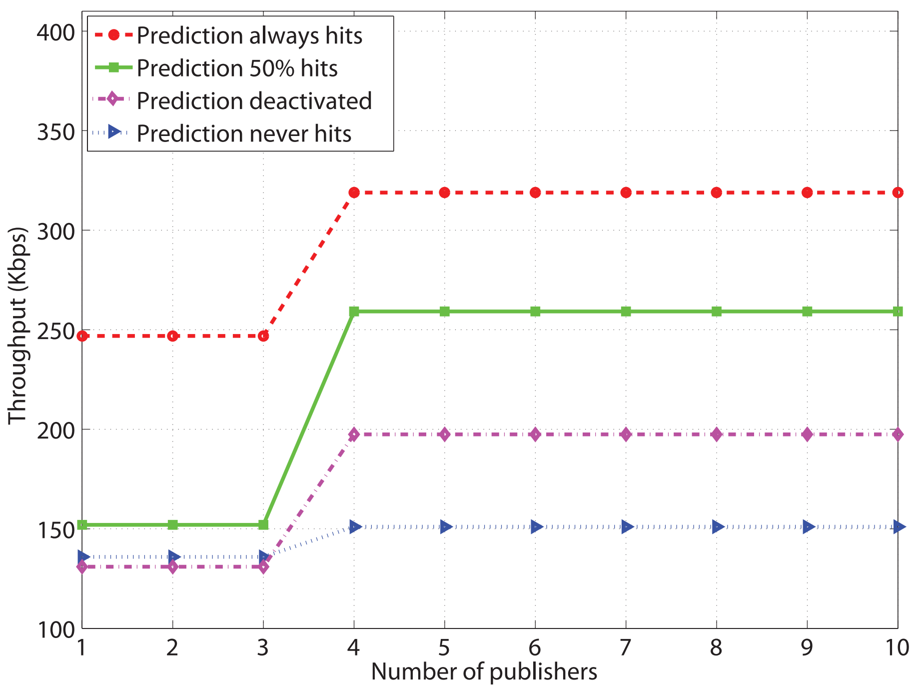

Figure 5.6: Increase in throughput due to the prediction. The speed of mobile nodes is 50 km/h, prediction algorithm is activated at T_{pred}=-87 dBm.

5.5.3 *The optimization of the prediction triggering level*

Furthermore, we have investigated the impact of the prediction triggering level on the performance of the prediction function. Setting the power threshold for activating the prediction is a critical operation. This parameter affects the quantity of signaling traffic, hence, an unsuitable triggering level can drastically reduce the benefits of prediction by causing unnecessary traffic.

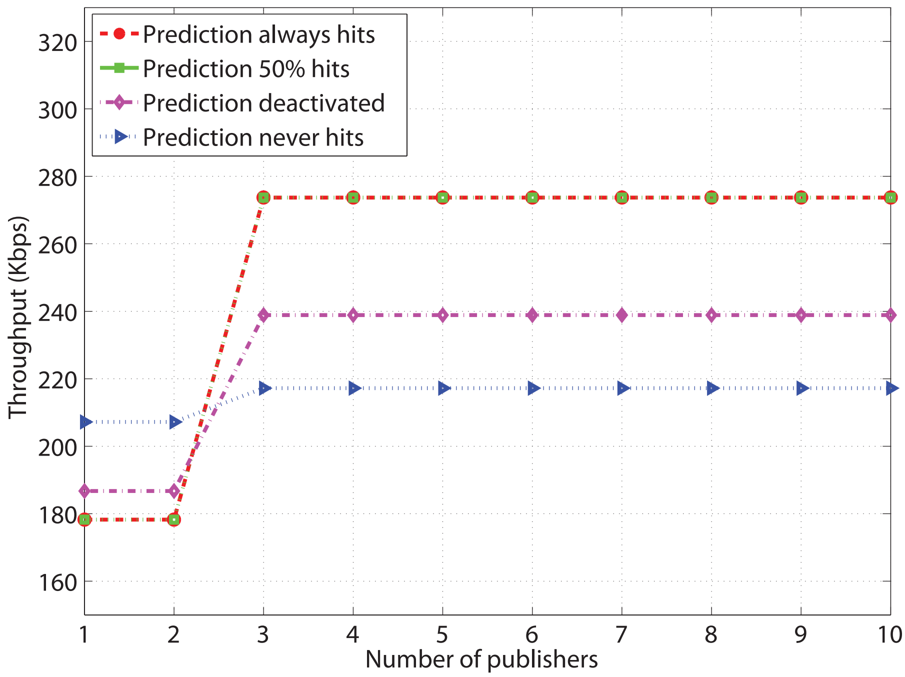

Figure 5.7: Increase in throughput due to the prediction. The speed of mobile nodes is 10 km/h, prediction algorithm is activated at T_{pred}=-85 dBm.

The optimal prediction activation level depends on the current topology and a large set of network parameters. In our scenarios the topology management helper has information about network structure, speed and position of mobile users and it is constantly monitoring the system performance. This allows us to dynamically optimize the triggering thresholds.

Figure 5.8 and Figure 5.9 illustrate the obtained throughput for different user speeds, for the case of an unoptimized prediction triggering threshold. It is evident that applying highly wrong predictions can have an extremely negative impact on the overall throughput. In fact, this can occur at such a level that the deactivation of prediction can lead to the most optimal system performance.

On the other hand, for the same network setting and the speed of mobile nodes the system shows much better performance using the optimized prediction triggering threshold as illustrated in Figure 5.5 and Figure 5.7.

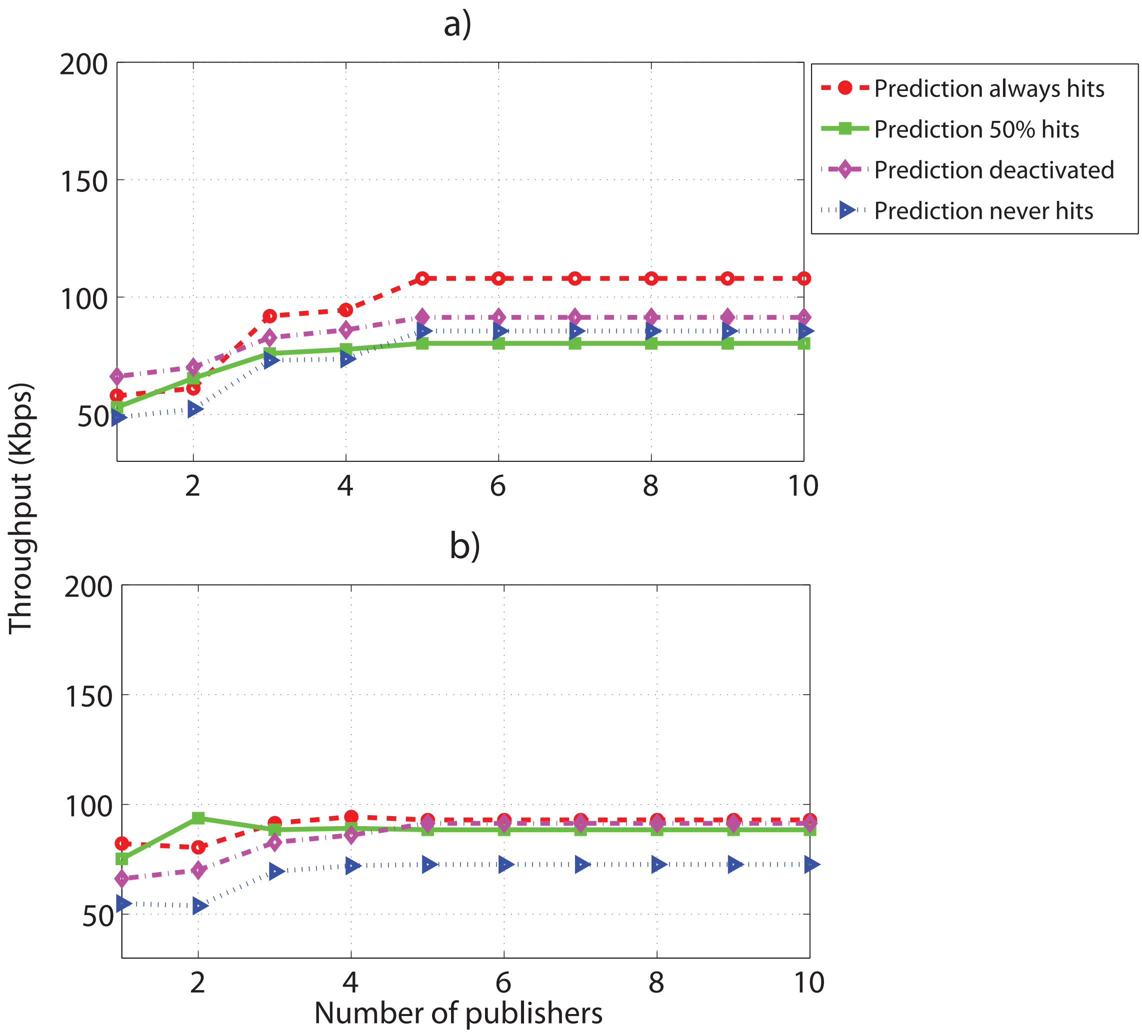

Figure 5.8: Impact of unoptimized prediction triggering threshold on the throughput. The speed is 100 km/h. a) The triggering threshold is -85 dBm b) The triggering threshold is -80 dBm

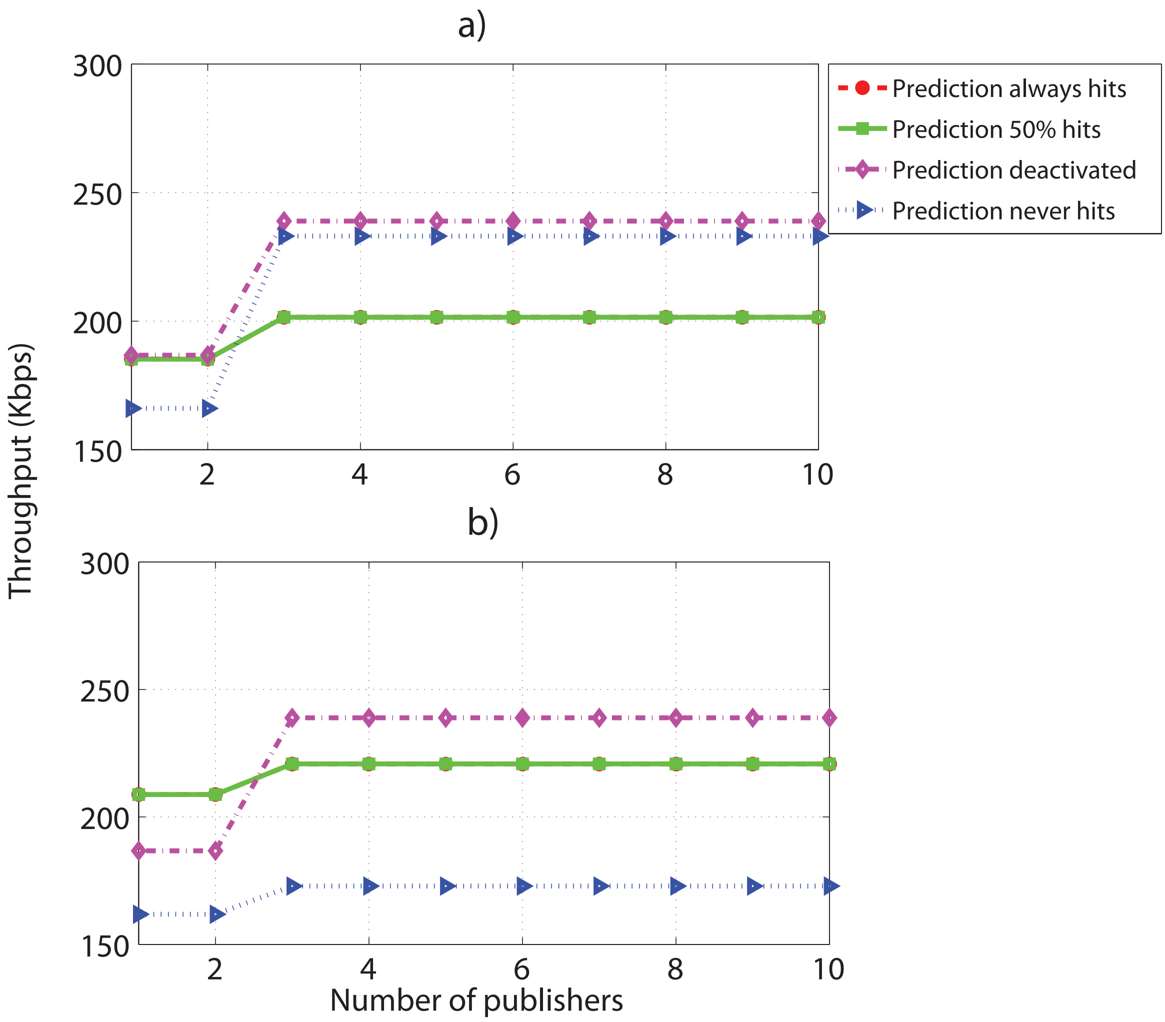

Figure 5.9: Impact of unoptimized prediction triggering threshold on the throughput. The speed is 10 km/h. a) The triggering threshold is -89 dBm b) The triggering threshold is -80 dBm

5.6 CONCLUSIONS

In this chapter we have presented a model of publish-subscribe based topology management enriched by mobility prediction functionality. We have simulated a dynamic IEEE 802.16 (WiMAX) network in which all nodes communicate using a publish-subscribe paradigm. The movement pattern of nodes is tracked throughout the time and used as an input to prediction function.

Having the information about most likely future locations of nodes, the topology management can compute the backup forwarding paths for mobile nodes. Such mechanism is particularly important for rapidly moving nodes, where the forwarding paths might change very frequently. Therefore, the system becomes more robust and more quickly responsive to sudden changes in the positions of nodes. This diminishes the packet loss due to handover, thus, improves the system performance.

Our evaluation work demonstrates the advantage of utilizing publish-subscribe characteristics in data delivery. The throughput gain is achieved due to the fact that the missing data is not necessarily obtained from the original source, but it can be fetched from any publisher, or cache close to the subscriber. The topology manager keeps track of all available data publishers and creates the shortest paths to the subscribers.

Furthermore, our evaluation results demonstrate that the publish-subscribe model equipped with intelligent location prediction can be a winning paradigm in mobile networks. The performance can be significantly increased in the cases where sufficient amount of network information is known by the topology management. However, for the optimal performance of such a system various parameters need to be adapted. The topology manager having the information about the network structure and node mobility appears as the most natural candidate for performing such adaptation.

In order to further improve the considered system different caching techniques can be applied. By dynamically detecting a highly popular content the forwarding nodes in an area distant from the publisher of such information could cache this content acting as temporary publishers and improving the traffic distribution in the network.

The topology manager having a complete overview of the network topology can choose the suitable caching point in order to decrease the distance from the requesting nodes and the provider of the content. The caching point will become a new publisher for the popular content. In such a way, the data latency can be considerably decreased, as well as the overall traffic in the network, since the data will be traversing over decreased number of links.

6

Topology discovery and network attachment

In this chapter we focus on the topology discovery and the procedure for joining the network in the information-centric context. We develop and analyze such a network attachment procedure in an information-centric network utilizing the publish-subscribe paradigm for the data exchange. We aim at using the publish-subscribe concept not only as a communication means, but we fully exploit its characteristics for native merging of fine-grained network operations such as topology management and network connectivity establishment. Such an integration adds not only to the simplicity of the network and efficiency of the topology information gathering, but includes the means for handling mobility issues.

We examine the performance characteristics of the proposed solution particularly focusing on complexity and introduced message overhead. The evaluation results obtained from the testbed experiments show the outstanding performance in terms of delay, while the signaling overhead remains at a very low level.

6.1 The motivation

Given an arbitrary node in the network the prerequisite for establishing the communication with the rest of the nodes is to be connected with at least one another node that can act as a bridge towards the rest of the network. The process of attaching a new node into the network commences by setting up such initial connectivity. Depending on the network and the protocol design the establishment of the primary connection with the network may be followed by a sequence of additional operations before the node completely joins the network. We refer to such a protocol for establishing a full network connection as a *network attachment*.

We consider the *network attachment* as a set of operations required for initializing and maintaining the connectivity. It consists of the discovery of a new node followed by corresponding modification of network states in order to address the changes in network configuration. The attachment procedure can embrace a variety of additional mechanisms, e.g. authorization and security

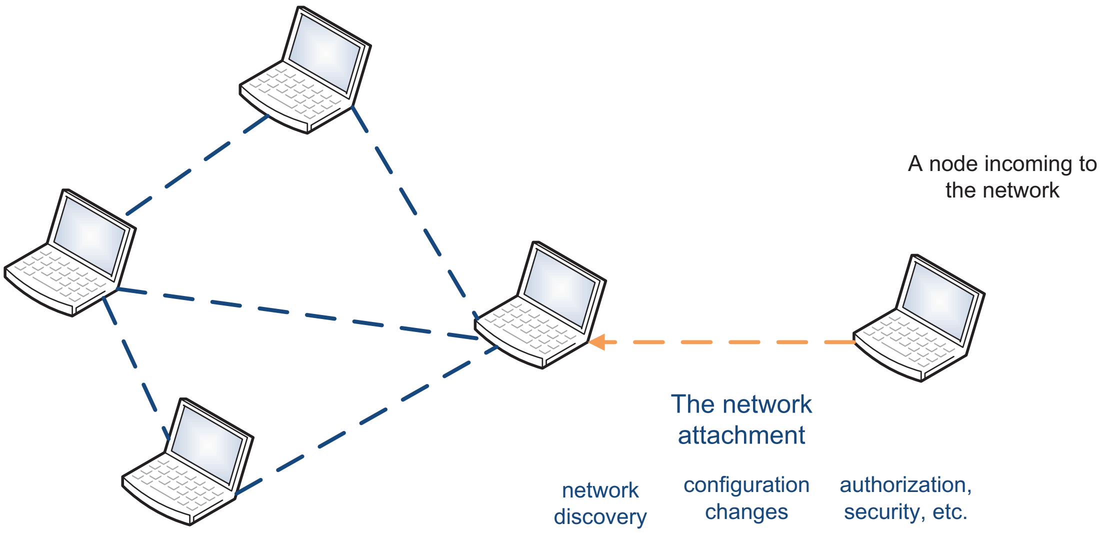

Figure 6.1: An illustration of the attachment procedure in an arbitrary network.

features which are left beyond the scope of this chapter. Figure 6.1 illustrates the attachment procedure in an arbitrary network.

The *network attachment* in the currently predominant IP-based networks is handled on the data link layer and the network layer, using the common send-receive pattern for communication bootstrapping. In this chapter we are considering the novel approach in attachment procedure specially designed for information-centric networks, entirely relying on data-centric type of communication. Instead of send-receive primitives the connection bootstrapping is based on the publish-subscribe paradigm [82].

In the present Internet the user interested in receiving some data initiates the communication process by somehow retrieving the IP address of the data source. The communication is not possible without IP address. The existing name server (DNS) infrastructure allows the translation of names (such as www.googla.com) to IP addresses through lookups. However, there is no capability to have a content based communications in the context of the Internet[1]. Having the target IP address allows the end user to directly contact the source in order to request the desired data. Therefore, the sender is given the power to drive the data transfer, leaving the receiver without direct control over the data transmission. As opposed to the standard sender-oriented communication, in the publish-subscribe context the end users only express their interest in receiving some data by subscribing to it. On the other hand, the users de-

[1]In a certain sense one could consider Google itself as a content search database that allows finding name-based links to a content that is then interpreted through DNS-lookup to a proper IP-address. Of course, in this case a user itself is a part of the rendezvous process.

clare their willingness to provide the data by publishing it. As an outcome, the end users are supplied with data from the most suitable source without knowing its identity or the address and only after explicitly expressing their interest in receiving that particular piece of information.

Due to the advantages offered by the content-centric, publish-subscribe type of communication, we study the possibility of building a flexible network attachment procedure. As in the previous chapters we use the PURSUIT architecture [77] as the underlying reference architecture and framework for our work (see Section 2.3.6 for more details).

Our target is to enrich such information-centric architecture with the flexible *network attachment* protocol in order to better address the possible change (fluctuations) in the topology. Having a *network attachment* which can provide the smooth integration of a new node in the network without affecting the usual network functionality improves the performance of the network in dynamic environments. Apart from the prime functionality of incorporating the new node into the network, the information-centric nature of underlying network offers a variety of benefits and optimization opportunities. Furthermore, utilizing the hierarchical naming structure present in the PURSUIT architecture gives us the possibility of integrating the topology management and *network attachment* functionalities.

The procedure of attaching a new node to the network is tightly related to the dissemination and collection of network knowledge. Merging it with the naming structure of the node allows instantaneous distribution of the updates on topology states. Such a novel approach, designing the attachment protocol as a helping functionality of topology management and incorporating the naming scheme into the topology data gathering and dissemination, fully utilizes the information-centric communication paradigm. As far as we are aware of, this is the first work on information-centric network attachment that takes the advantage of the naming scheme in order to facilitate the concurrent dissemination and collection of the network knowledge.

6.2 THE UNDERLYING NETWORK STRUCTURE

According to the PURSUIT architecture the information exchange is conveyed by means of publications and subscriptions to specific data item. The information is structured as an acyclic graph, in which the leaf vertices represent the individual information items. In order to facilitate data dissemination and searching mechanisms the notions of scopes and dissemination strategies are introduced. The scoping represents the grouping of related data items into a whole for easier data accessibility.

The scopes are defined in a way that every information item belongs to at least one scope. Different scopes are linked together depending on their semantic relations. The dissemination strategies dictate the exact implemen-

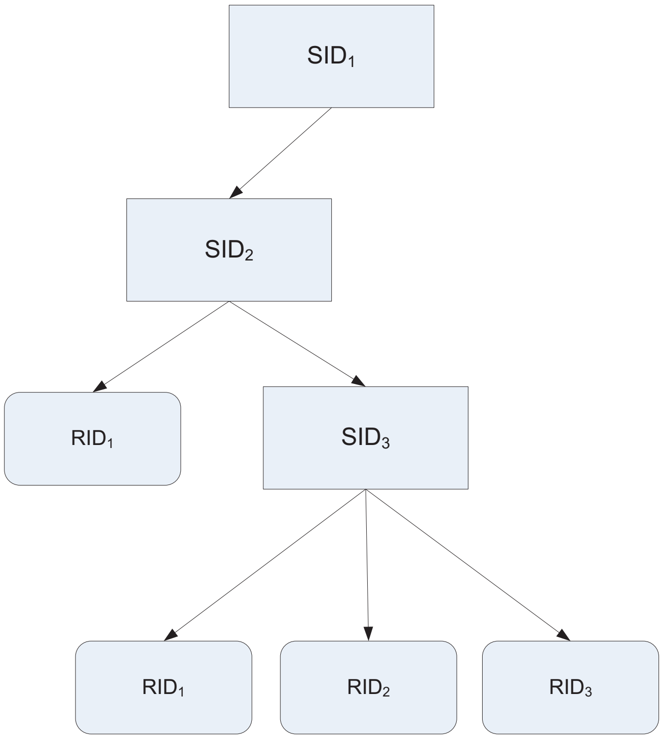

Figure 6.2: The PURSUIT hierarchical data structure.

tation of a scope by restricting its visibility to particular domains. Individual data items are denoted by the rendezvous ID (RId), which is unique in the scope to which the data item is assigned.

The scope IDs are created in the similar manner where the given scope ID (SId) is unique within the super-scope to which it is linked. Therefore, the RId together with the belonging scope IDs fully and uniquely identifies the data. Similarly, every scope is uniquely marked with its SId along with the identifiers of the scopes that it is semantically linked to. Such data structure resembles the hierarchical organization of scopes and belonging sub-scopes and information items. Figure 6.2 illustrate the described data structure. The information item with the rendezvous ID *RId3* is fully identified by the *SId1/SId2/-SId3/RId3* while the scope with the scope Id *SId2* is identified by *SId1/SId2*. The information items denoted by *SId1/SId2/RId1* and *SId1/SId2/SId3/RId1* refer to different data, the identifier *RId1* is unique for the scope it belongs to.

Topology management module represents one of the fundamental building blocks of the information-centric architecture. Its function is to find the optimal forwarding paths from publishers towards subscribers, with respect to current network conditions. The common topology discovery and delivery path formation mechanisms such as OSPF [47] can be easily implemented using publish-subscribe pattern. Furthermore, such implementation can be applied in information-centric architecture [125]. Therefore, the publish-subscribe paradigm is proven to be flexible enough for supporting such fine grained network operations such as topology topology discovery and management.

However, in order to fully utilize the strength of publish-subscribe information-centric model and to further demonstrate its flexibility in lower level network operations, we aim at merging the network attachment process with the topology discovery and intra-domain topology management. The hierarchical organization of the PURSUIT data structure provides the needed prerequisite for such integration.

Furthermore, grouping of data by means of scopes and dissemination strategies provides a powerful tool for flexible addition of new protocols. For example, required signaling can be conveyed in a specially dedicated dissemination strategy and/or scope without affecting the ordinary network functionality. Therefore, developing a new protocol on top of existing structure is carried out in a straightforward manner. Following such principles, our goal is to develop the attachment protocol by which the initialization of network connectivity will entail the instantaneous dissemination and gathering of network knowledge.

6.3 NETWORK ATTACHMENT PROTOCOL DESCRIPTION

For initial network attachment signaling and distribution of advertisements on nodes' presence we follow the principles similar to OSPF protocol. In order to announce its presence each node can publish the new scope message, having the scope ID assigned according to the node's ID. Such publications can be distributed using a separate strategy. By means of dissemination strategies the data flow can be highly adapted, since the applied strategy defines the way of realizing the main functions, i.e. rendezvous, topology, and forwarding.

The communication among the nodes prior they are attached into the network, i.e. the initial bootstrapping message exchange, can be enabled by a dedicated strategy, e.g. "broadcast" strategy. This type of dissemination strategy allows nodes to communicate only with the nodes within the transmission range, similar to broadcast type of communication. Furthermore, every new node into the network needs to be subscribed to this strategy in order to obtain needed information. During the network attachment phase, the node subscribed to the "broadcast" strategy will be able to receive the announcements of the neighborhood nodes. Thus, the joining node will gain the knowledge about its neighbors and existing links.

In order to distribute this information the node will publish its existence in the form of new scope publication, where the scope ID will be composed as neighborID/nodeID. Following such a principle, every generated scope, representing the new node in the network, carries not only the ID of the incoming node, but the information about interconnectivity and topology, as well.

In other words, the bootstrapping of connectivity is seamlessly combined with the topology knowledge collection and dissemination. The topology management can collect the information about incoming nodes by subscribing to the dedicated strategy. By these means the topology manager is constantly updated about the current network states.

Let us assume existence of only one node in the network, with the ID *AA*. In order to announce its presence in the network the node publishes the scope publication with the ID *AA*, using the "broadcast" strategy. Every new node, in order to attach to the network, needs to subscribe to this "broadcast" strategy. The new node with the ID *BB* will hear the message from the node *AA* and infer that there is the link between them. Hence, the new node will announce its presence by publishing a new scope under the ID which is the composition of the neighbor ID and its own. Thus, in this particular case the ID *AA/BB* would represent the new node (scope).

In this way such an announcement will contain not only the information about the node's ID, but the information about its place in the network with respect to other nodes, as well. Such announcing procedure repeats in the same way for each incoming node, regardless of the position in the network. The nodes can be connected with more than one neighbor at the same time, which would be announced by publishing the same scope under different paths.

Finally, the topology manager will receive all messages of the type: new scope ID *AA*, *AA/BB*, *AA/DD*, *AA/BB/CC*. Thus, based on the scope IDs it can deduce the information about links and nodes existing in the network and perform required changes. Announcements are normally periodic, so that the topology manager is dynamically updated in the case that the nodes join or leave the network. Figure 6.3 illustrates the described hierarchical name assignment in the attachment procedure.

6.3.1 Spatially and technology related naming hierarchy

Similar to the organization of the current Internet, the information-centric systems assume the network division into the operational units equivalent to administrative domains. Although the interconnections between different network domains can have a highly complex structure, the above described hierarchical data organization facilitates the establishment of semantic references among the network units. Moreover, a fine-grained partitioning within the single domain with respect to the technology used in the sub-domains can be easily delineated, as well.

In the context of information-centric networks every administrative domain is controlled by a single topology management entity. Such individual network unit can be identified by the scope ID. Furthermore, spatially smaller operational areas or technologically distinctive parts of a single domain (e.g. area covered by one access point, or the area applying one particular com-

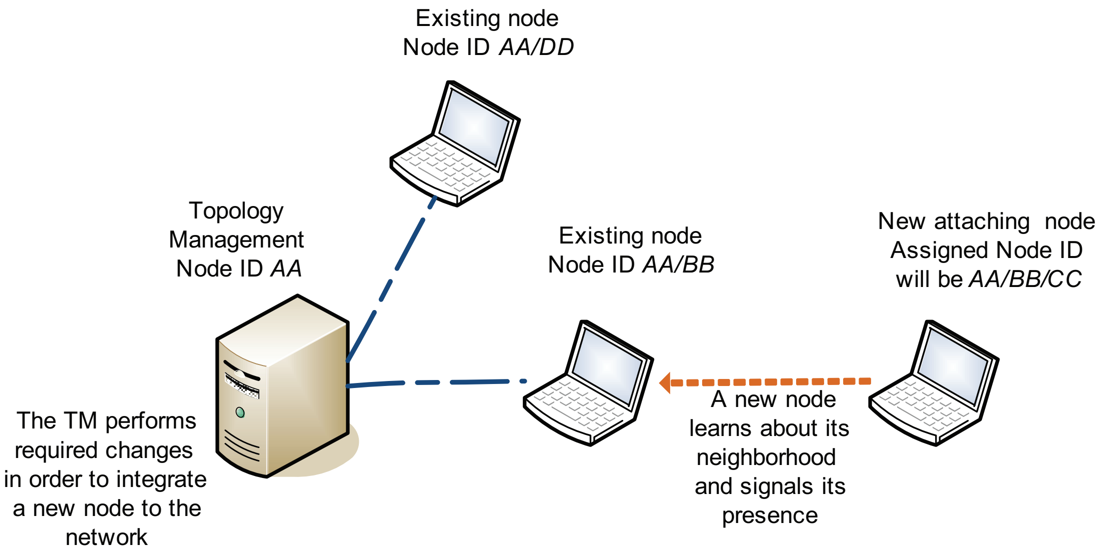

Figure 6.3: An illustration of the use of hierarchical naming for dynamic topology discovery.

munication standard) can be denoted by separate scope IDs derived from the parent domain scope ID. Such a generation of the scope IDs according to the belonging scope ID can be done by utilizing various algorithms.

The individual identifiers remain statistically unique while keeping the algorithmic relation with the rest of the identifiers of the belonging namespace. Any network entity having the knowledge of the applied algorithm is able to infer the existing relationship between the identifiers. Otherwise, the identifiers appear as random and it is hard to deduce existence of a relation between them. The most important characteristic of applying algorithmic IDs generation in naming the different parts of the single domain is in keeping the tight naming relations between the structural pieces of the domain.

Moreover, such mapping of distinctive spatial and technological units with hierarchical naming structure can be combined with aforementioned integrated data dissemination and gathering procedure. If during the attachment procedure for each of the joining nodes, the node ID is assigned according to its affiliation to particular spatial or technological segment of the domain, the announcement of node-existence in the network will also signal its belonging to the specific domain.

Therefore, the procedure of network discovery attains another dimension. Apart from the process of gathering the knowledge about existing connections, the network attachment can concurrently provide the information regarding the belonging area or technology and standard used. Figure 6.4 illustrates the spatially and technologically related naming hierarchy applied for assigning

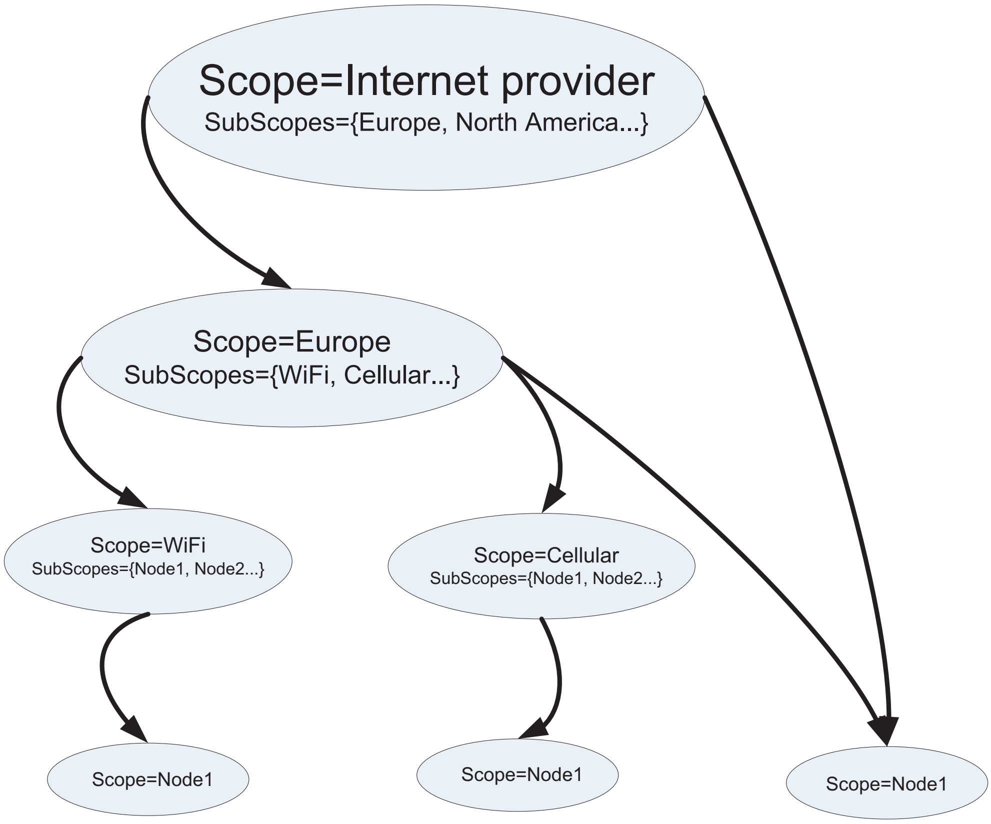

Figure 6.4: An illustration of the spatially and technologically related naming hierarchy for facilitating the topology data dissemination and gathering.

IDs of nodes. The identifiers of nodes can be derived from the hierarchy of belonging Internet provider, region or protocol used.

6.3.2 *Native support for mobility*

The frequent changes in network topology due to the mobility of nodes can cause a significant packet loss. The methods for handling the mobility issues can place a considerable burden on the overall network operation. However, the publish-subscribe communication pattern provides a firm basis for mitigation of mobility issues due to its data-centric characteristics. The desired content can be retrieved from various locations, thus, the connection loss due to mobility does not necessarily involve inability of fetching the desired data. Moreover, efficient mobility solutions can be incorporated within network attachment model described above. Given that the announcement messages prior attaching indirectly contain the information about the neighborhood, the modification in the location of the node due to mobility will be directly reflected in its announcements.

By changing the position, the node's (logical) neighborhood may change. The moving node can come into the proximity of other nodes to which it was not connected before. Due to our topology control design the simple subscribing to the dedicated strategy during the attachment phase on the new location will result in acquiring an updated neighborhood information. The following announcement of node's existence will incorporate the updated paths to the node.

Due to the periodic nature of such announcements the topology manager is able to notice any change in nodes position relative to the other nodes. After the node movement the topology manager will stop receiving announcements from the node on the old location, but the announcements from the same node on changed position will be signaled. Thus, the topology management can immediately perceive occurred modification in the location of the node and change the topology states accordingly. For the correct functionality of our attachment model the unique identification of the nodes is needed, since a node attached to a new position needs to be matched with its previous location.

6.4 IMPLEMENTATION OBJECTIVES

The hierarchical organization of the PURSUIT functional model provides the base for utilizing the naming structure for more efficient network discovery through the seamless integration of the topology management and network attachment operations. Having nodes' naming structure depicting also the interconnectivity in the network enables instantaneous gathering of network knowledge upon the initialization of the attachment process.

For this mode of operation the underlying architecture needs to offer the possibility of collecting the network information prior the attachment, e.g. utilizing the broadcast. Such approach allows the new-coming nodes to collect the knowledge about the current network state, based on which the attaching procedure, taking advantage of hierarchical naming, can be performed. In general, the initial information gathering can be performed in different ways depending on the underlying architecture and protocols used.

Our implementation efforts focused on building the attachment procedure utilizing the hierarchical naming structure and entirely relying on publish-subscribe communication pattern, regardless of the method used for initial information gathering. Our attachment procedure signals the existence of a new node willing to connect to the network, and handles the necessary operation for its smooth integration. Our implementation guideline is to keep the ordinary network operation unaffected by such procedure. Furthermore, we aim at keeping delay, complexity, and introduced signaling overhead at the lowest possible level.

We aim at building such network attachment module as a helping functionality of the existing PURSUIT intra-domain topology management [143]. All nodes in the network can signal to the topology manager the arrival of a new node in the network by publishing the messages in dedicated strategy and scope. The attachment module, as a helping module of the topology manager, supplied with the information about new nodes triggers the set of operations for integrating the new nodes into the network. The adequate modifications in the topology management, as well as in the particular forwarding nodes are required.

The topology management maintains the igraph [116] states as the graph representation of the network. Thus, upon the arrival of the new node, the igraph states need to be updated accordingly by adding the new vertices and edges. After the igraph states have been changed, the forwarding nodes being affected by this modification need to update their states correspondingly.

In the current PURSUIT implementation architecture, all nodes are determined by the set of the Click modular router elements [144]. The forwarding functionality of a node is defined in the forwarding Click element. This element maintains the forwarding states needed for proper data routing, as a counterpart of routing tables and corresponding entries. Adding the new node, and thus the new links, needs to be signaled to affected forwarding elements for updating their states accordingly.

Being aware of all occurred changes that the joining of a new node caused, the attachment module generates the instruction messages for changing the forwarding states. These messages contain the information about the forwarding states that need to be added to the existing forwarding configuration, as an outcome of establishing new links between nodes. The messages are published to all involved nodes using the special strategies and scopes for distribution of network notifications, to which all the nodes in the network are constantly subscribed.

Once the involved node receives the instruction message and verifies that it is the designated receiver, it incorporates the updates of forwarding states into its configuration. Along with this update the network attachment procedure is completed; the new node is integrated in the network and can perform the data exchange by means of publications and subscriptions.

6.5 PERFORMANCE EVALUATION

In order to evaluate the performance of the developed network attachment module, we executed the set of experiments in the testbed environment. The nodes in the testbed are running the PURSUIT prototype implementation and are connected through a local area network. Our aim was to investigate the impact of attaching of a single node in terms of delay, complexity, and overhead. Therefore, we performed the experiments by attaching the node in different positions in the network with respect to the topology manager node [145].

We measured the delay due to the processing in igraph and in the Click forwarding elements. The processing in igraph is required for changing the network graph representation by adding the new vertices and edges. The modifications in the forwarding nodes are required due to the additions of the new entries in the forwarding configuration. These forwarding changes occur directly in the Click forwarding elements. Figure 6.5, Figure 6.6 and Figure 6.7 show the obtained results. Regardless of the position in which the new node is attaching, the processing requirements in igraph and forwarders remain the same. The delay introduced by performing necessary modifications has the approximately constant value regardless of the conditions under which the new node attaches.

Increasing the distance of the attaching point with respect to the topology manager has as a result only the increase of the messages transmission time, as expected. The messages transmission delay is dependent on the network setup and current state of the links and is not related to the attachment operation. Figure 6.5, Figure 6.6 and Figure 6.7 illustrate the delay of the overall attachment procedure, as well. This delay is primarily dependent on the message transmission delay, thus, the network conditions.

However, the processing delay caused by the network attachment stays nearly constant and negligibly low (few milliseconds) regardless of the network conditions and the location of the attaching node. The results are also independent of the network size. This is due to the nature of our implementation architecture. The increase in the network size can have as a consequence only the increase in the message transmission delay, but the actual processing time required by attachment process remains unchanged. Furthermore, being focused on intra-domain solutions, we limited our testbed size to be at maximum few dozens of nodes, and the results indicate that even larger intra-domains can be handled without problems.

Another important aspect in the evaluation of the network attachment module is an investigation of the overhead that the overall attachment procedure introduces. In order to accomplish the attachment process the exchange of only two messages is required. The first message signals the arrival of a new node, which serves as a trigger for starting the attachment procedure. After the adequate changes in the igraph have been made the second message is published by the network attachment towards involved nodes as an instruction for changing the forwarding states. Figure 6.8 shows the obtained results in terms of the message overhead. The message overhead, i.e. the amount of data generated due to the attachment, is a constant regardless of the position of the attaching node and it includes only two above described messages.

Investigating the robustness of our system is another evaluation target. We concentrate on the robustness against the larger attachment traffic. For this purpose we extended the initial testbed setup and created the testing environment in which a large number of nodes are willing to join the network at the same time. Additionally, we improved our initial protocol implementation by optimizing the igraph related processing, making it faster and, thus, better res-

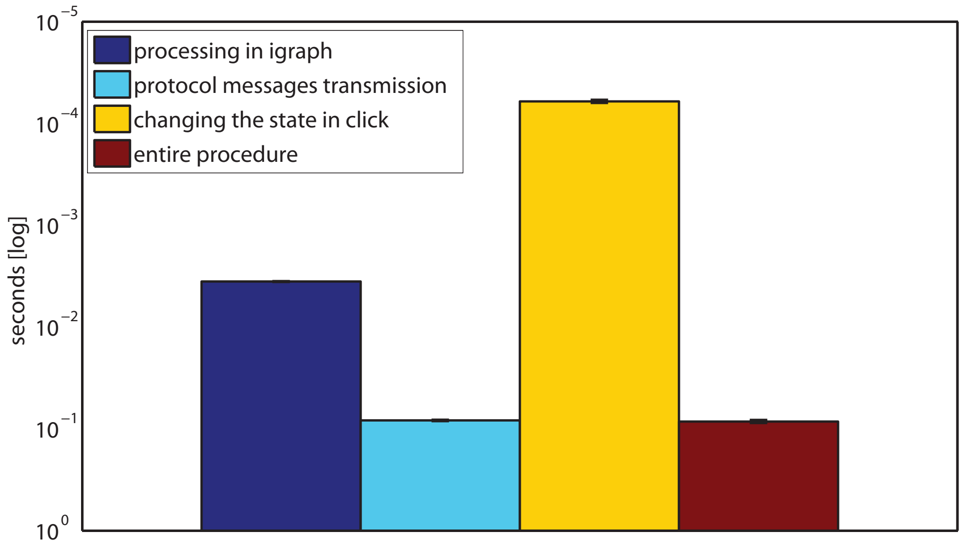

Figure 6.5: The delay introduced by the attachment procedure. A new node is attaching directly to the topology management node. *Note: Due to the low delay values the y-axis is given in logarithmic scale. The higher bars indicate lower delay values.*

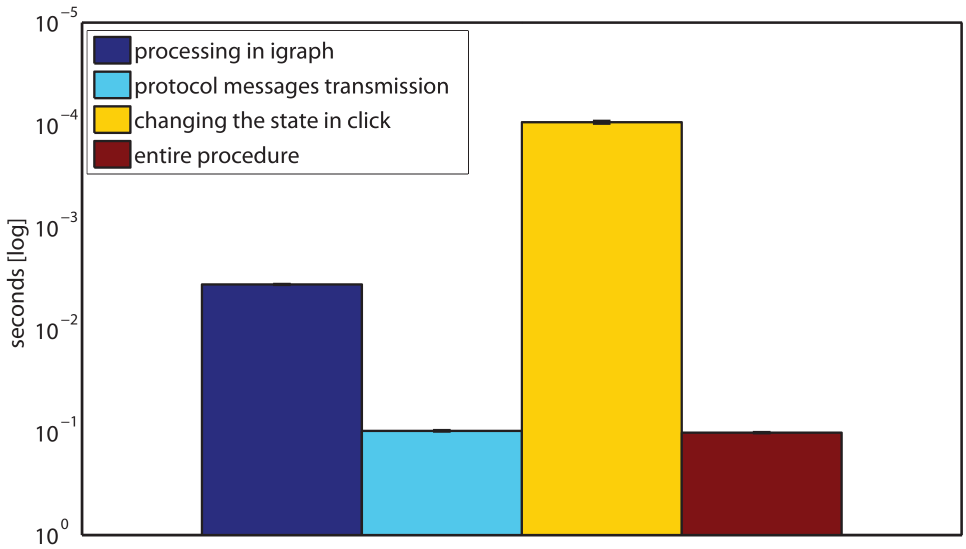

Figure 6.6: The delay introduced by the attachment procedure. A new node is attaching to the node which is positioned 1 hop away from the topology management node.

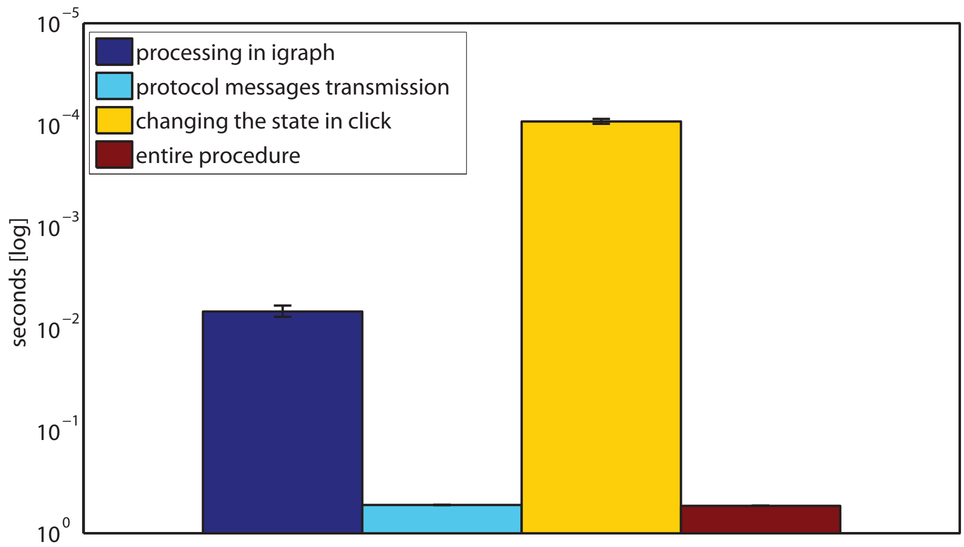

Figure 6.7: The delay introduced by the attachment procedure. A new node is attaching to the node which is positioned 2 hops away from the topology management node.

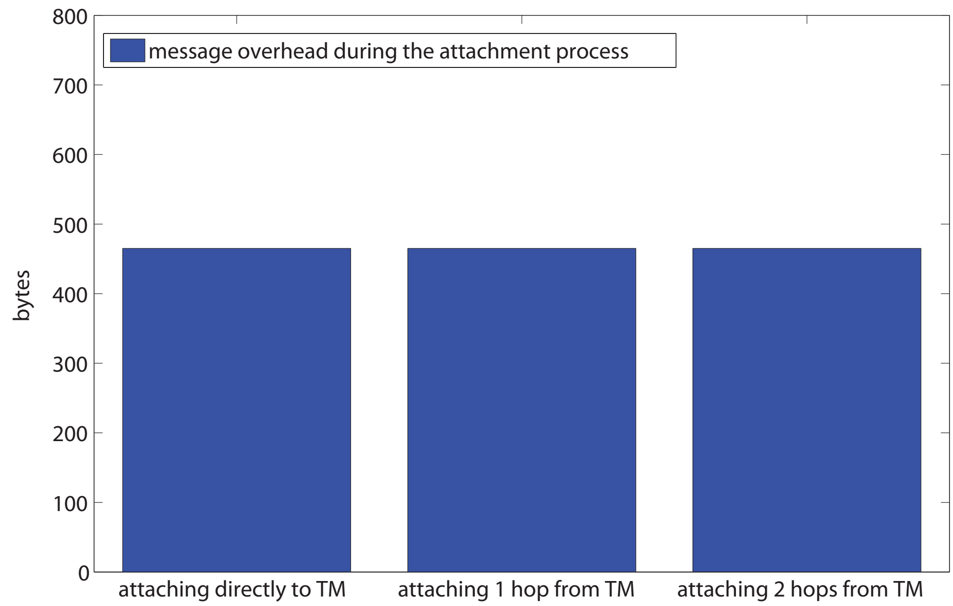

Figure 6.8: The message overhead due to the network attachment procedure. The new node is attaching on different locations in the network with respect to the topology manager.

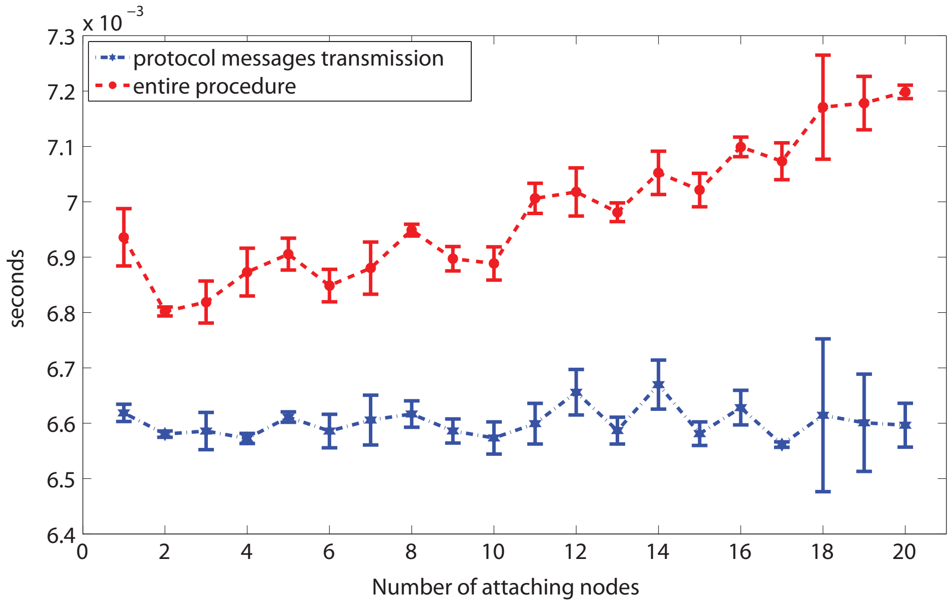

Figure 6.9: Attachment protocol delay with standard error when the number of attaching nodes is increasing.

ponsive to the increased number of simultaneous requests. In order to better test the robustness we augmented the burden on our testing network by making the new nodes always attaching to the same node that already exists in the topology, i.e. all new nodes have the same attachment point.

In such a situation of high attachment demand the attachment module in topology management, as well as the forwarding elements in involved nodes need to handle a large number of requests, therefore, the drop in the performance is expected. Our tests show the slight increase in the delay introduced by overall attachment procedure as more nodes are willing to join the network, as illustrated in Figure 6.9. Since the delay introduced by the transmission of protocol messages stays constant due to the unchanged network conditions, the increase in the delay is primarily caused by the protocol required processing.

In order to better understand the exact reasons of such performance we made more detailed tests with the focus on the processing parts of the attachment protocol. Such analysis have helped us understanding the bottlenecks of proposed protocol and eventual scalability issues. Figure 6.10 shows the obtained results. The delay due to the changes in the Click forwarding elements stays nearly constant regardless of the number of attaching requests and the node that performs the change (the attaching node or the attachment point).

On the other hand, having states processed in igraph requires additional time as more nodes are attaching to the network. The lookup procedure for

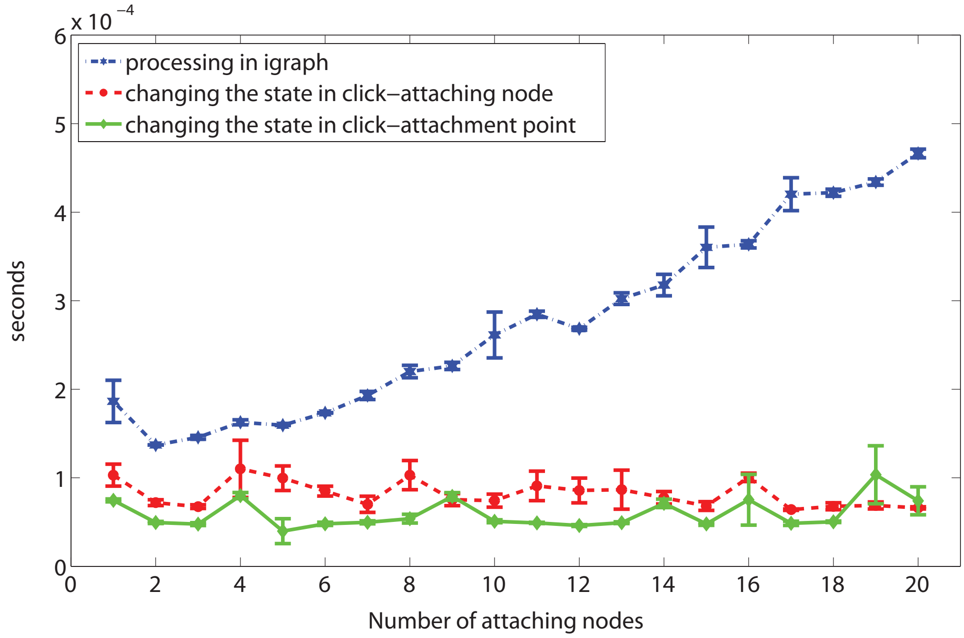

Figure 6.10: Attachment protocol delay (with standard error) introduced by different processing parts while the number of attaching nodes is increasing.

finding the unallocated identifiers to be assigned to new vertices and edges, as well as handling the graph representation of the topology, becomes more complex with the network enlargement. Therefore, despite the performed implementation optimization, a minor increase in igraph processing delay is expected. However, based on our results the additional time needed for processing in igraph does not pose a considerable burden to the system. The value of the additional delay remains low (less than a millisecond), keeping the time needed for entire procedure very modest and manageable even with resource constrained systems.

6.6 INCORPORATING SERVICE DISCOVERY

Service discovery is one of the fundamental parts that is required for information centric networking. The main role of the service discovery is in exploring the ability of network to provide certain services. Service discovery is a part of the resource discovery procedure, where main functionality is to find out available resources. A large portion of work has already been carried out within service discovery area, especially focusing on designing of new service discovery solutions. The taxonomy, main characteristics and performance of service discovery protocols have been recently studied in details in [146, 147, 148, 149].

As the number of wireless mobile devices has dramatically increased over recent years, the importance of service discovery protocols has been growing. The nature of wireless mobile networks characterized by high probability of disconnections and packet loss, restricts the functionality of devices operating in such environment. Moreover, mobile devices often have limited processing capabilities, memory and battery resources, making the dynamic wireless network even a bigger challenge. Thus, in order to optimize the performance of such dynamic system, a flexible, rapidly responsive, and scalable service discovery mechanism is necessary.

Due to its generality and flexibility, our proposed topology discovery methodology can be easily incorporated with most of the resource and service discovery strategies and protocols. The network entity responsible for managing available services and resources would only need to be subscribed to the dedicated strategy and scope in order to obtain the relevant information. Furthermore, by the means of dedicated strategy and scopes the service discovery procedure can be highly adapted and optimized.

The service discovery mechanism can be either structured or unstructured. In the structured service discovery approach the knowledge about available services is located only at certain nodes in the network. Such nodes can be further grouped either in centralized and decentralized way based on the implementation of the service information distribution and the data retrieval model. In the centralized model the responsible servers store the information about available services. Thus, the interested users are able to retrieve such information directly from the servers by means of subscriptions. The centralized approach for service discovery is already widely utilized by many systems [150, 151, 152, 153, 154].

In order to increase the robustness of the system with respect to the single point of failure the centralized placement of servers can be replaced by distributed organization. Following such approach the service information is disseminated over the set of responsible servers. The level of network dynamics plays a significant role in the efficiency of such systems since it directly influences the maintenance overhead.

On the other hand, unstructured approach encounters higher service discovery delays given the fact that there are no dedicated servers for storing the service information. However, such model is more flexible in highly dynamic environment, when the nodes are characterized by high mobility.

In order to combine the characteristics of the unstructured and structured model in the most efficient way the hybrid models can be employed. Such approach relies mainly on clustering [155], which brings the weak structure in unstructured model.

Depending on the environment and the system requirements different aforementioned approaches could be appropriate. Nevertheless, the implementation taking advantage of dissemination strategies and scopes can be applied to any selected approach without loss of efficiency. Therefore, our proposed model for service discovery fits well to a wide range of network settings and environments.

6.7 CONCLUSIONS

In this chapter we presented a novel method and design to enable network attachment in information-centric network. The developed framework fully utilizes the benefits offered by the content-centric communication paradigm by seamlessly integrating the network discovery with the topology creation process. Such a model can be used to facilitate the dissemination of spatial and technology specific information. The model incorporates the means for mitigation of mobility issues, as well. Furthermore, a large set of service and resource discovery protocols can be easily incorporated into the proposed topology discovery model. This combination allows efficient reuse of resources also for service discovery.

Our implementation of the attachment process shows excellent results in terms of introduced processing delay and signaling overhead. The performed experiments demonstrate that the publish-subscribe paradigm is suitable not only for the content retrieval but also for more integral network (management) operations, such as network discovery and topology formation. For further evaluation of proposed model different additional features of the underlying architecture and prototype implementation can be taken into consideration. This thesis is one of the first ones that is considering using publish-subscribe paradigm not only for the information dissemination, but has demonstrated that such communication model can be used also for signaling and management purposes.

7

Network coding in information-centric networks

Another important helper functionality in the context of information-centric networks is error control and correction in data transmission. In order to be able to recover partially corrupt messages at the subscriber's side without retransmissions, a publisher can add redundant data to its message (error correction code). Upon receiving the message containing redundant data, the subscriber is able to discover possible errors and to correct them. Depending on the transmission channel and strength of the forward error correction, this mechanism can allow discard of back channel or ARQ-mechanisms, which can be very useful in the context of content push through publish-subscribe networks.

Instead of considering classical error correction methods, we analyze in this chapter the performance of network coding in information-centric networks. We focus on two specific network coding schemes: XOR and random linear network coding. We have simulated the use of network coding with different topologies and traffic patterns in order to provide better understanding of network coding behavior and its possible bottlenecks. As a part of our performance evaluation we consider the latency introduced by coding and decoding operations.

In particular, we will indicate potential drawbacks and trade-offs of network coding when applied on specific topologies under specific circumstances by monitoring the differences between XOR and random linear network coding approaches. Finally, we apply network coding on the topology of existing research network Abilene [156] in order to evaluate network coding performance under more realistic conditions.

7.1 Scope of the network coding analysis

Network coding has received a lot of interest in networking community as it shows great potential to be a generic methodology for many purposes. It has been recently applied in a wide range of communication engineering domains, e.g. in information and complexity theory, cryptography, network optimization, and wireless communications [157, 158, 159, 160, 161]. Network coding has been envisioned as one possible solution to increase throughput and enable higher data rates as compared to conventional source coding. Additionally, its strong potential to recover from network failures has motivated research community to investigate more in this direction considering different network coding schemes.

Early research has been mostly focused on analytical examination. Results have demonstrated significant capacity gains, but most of the analyzed cases have been based on relatively simple topologies [162]. Extending these conclusions to larger networks has been done using mathematical abstractions.

Due to the broadcast nature of wireless links most of the recent attention has been directed into applying network coding to wireless networks. The seminal theoretical work has been followed by a more practical approach, that was done by Katti et al. [157], with a prototype implementation that has raised a number of interesting questions on implementation constrains.

While majority of existing work has been focused on analyzing wireless network coding using mathematical abstractions, our goal is to evaluate network coding behavior in medium-sized networks corresponding to one administrative domain and under more realistic circumstances that we expect to occur with the information-centric networks. We justify our selection of network size based on the assumption (see also [157]), that the network coding is not expected to bring significant benefits in the large-scale networks with high-traffic loads and high traffic speeds. It seems that the main benefits would be reaped in data distribution in more local area networks.

In this chapter we aim at getting clear insights on performance of network coding in intra-domain context and possibility of applying such a mechanisms within the publish-subscribe communication pattern. In order to get a detailed overview of network coding benefits we initially focus on the network coding mechanism in general, regardless of the type of the underlying network. Having such initial input will help us in evaluating the suitability of information-centric networks as a potential underlying platform for network coding.

Furthermore, we focus on two commonly considered network coding approaches, namely XOR and linear network coding, which appear as one of the most promising coding schemes. We evaluate performance of these approaches under different conditions using ns-3 network simulator [135]. As metrics we monitor a number of packets successfully decoded at the receiver side and latency that coding/decoding process introduces, as well as the overall dependency of performance on topology and traffic patterns.

7.2 GENERAL OVERVIEW ON NETWORK CODING

Following two simple examples provide the basic idea of scenarios and network coding models that we are investigating. These are given as illustrative examples to provide adequate background for our later discussion.

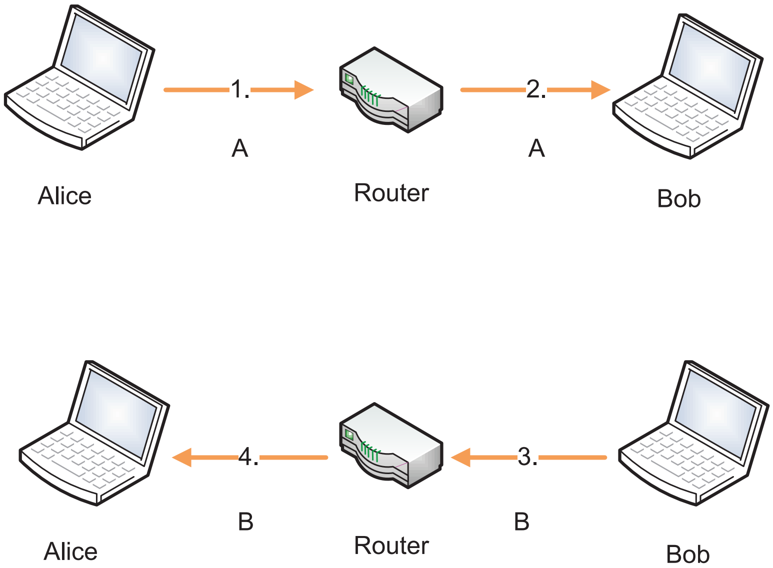

Figure 7.1: Exchange of packets without network coding.

7.2.1 XOR coding

Let us consider the classical scenario where two sides (Alice and Bob) want to exchange a pair of packets via a router. In such a scenario 4 transmissions are required as shown in Figure 7.1. First, Alice sends packet to the router, which forwards it to Bob. Then Bob sends packet to the router which forwards it to Alice.

Instead of this, XOR combination of packets is possible at the router. Both, Alice and Bob send their packets to the router, router applies XOR operation on them and broadcasts XORed version of the packet. After receiving the XORed packet, A$\oplus$B, both Bob and Alice are able to decode the packet sent from the other side by applying a simple XOR operation between the received and their own sent packet. This is illustrated in Figure 7.2. Moreover, encryption is achieved by the fact that it is impossible to reverse the operation (decode message) without knowing a value of one of two initial arguments.

In more complex scenario, such as one shown in Figure 7.3, a router will have different coding possibilities, among which it could choose the one that maximizes the number of packets delivered in one transmission. In this particular example, router having packets $p_1 p_2 p_3 p_4 p_5 p_6$, etc., will combine them before sending in a way which optimizes the decoding at receivers. For the given example in Figure 7.3 all receivers will be able to decode missing packet, if the router combines p_1, p_2, p_3, and p_4 together.

This leads to simple conclusion: router will maximize coding gain by making n packets combinations if all recipients already have $n-1$ packets of the same combination. In order to optimize XOR network coding gain in our im-

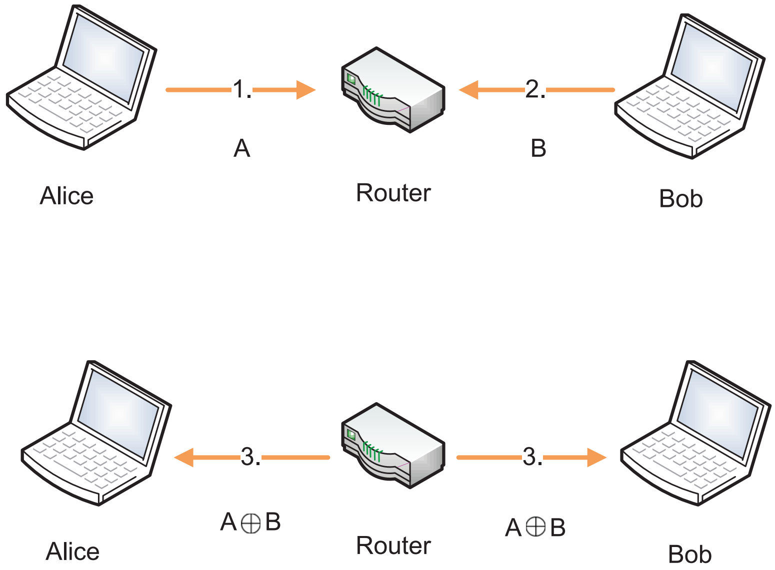

Figure 7.2: Exchange of packets using network coding.

plementation, we will be following this rule, while taking other objectives into consideration.

7.2.2 Random linear network coding

Random linear network coding approach is, in general, similar to XOR coding with the difference that the XOR operation is replaced with linear combination of packets. Matrix multiplication in which coefficients of linear combination are taken from a certain finite field is applied. This provides more flexibility in how the packets can be combined. Successful reception of information does not depend on receiving a particular data packet but on receiving sufficient number of independent packets.

Let $M^1, M^2 ... M^n$ denote the original packets generated by several sources, also called native packets [163]. The encoded packet would be a linear combination of $M^1, M^2 ... M^n$ with associated set of coefficients $g_1, g_2, ... g_n$ from a finite field F which implies that it has a form of $X = \sum_{i=1}^{n} g_i M^i$. In other words two vectors exist: first, $g_1, g_2, ... g_n$ -encoded vector, which is used at the receiver side to decode the message, and second, $X = \sum_{i=1}^{n} g_i M^i$ - information vector [164], which represents the linear combination of encoded packets. Furthermore, encoding can be performed recursively, with already encoded packets, as shown in Figure 7.4.

In order to retrieve original message decoder has to solve the system of m equations $X^j = \sum_{i=1}^{n} g_i^j M^i$ using Gaussian elimination algorithm, where unknowns are M^i. This system with m equations has n unknowns, thus having $m \geq n$ is prerequisite for decoding. Fulfilling this requirement is not a guarantee

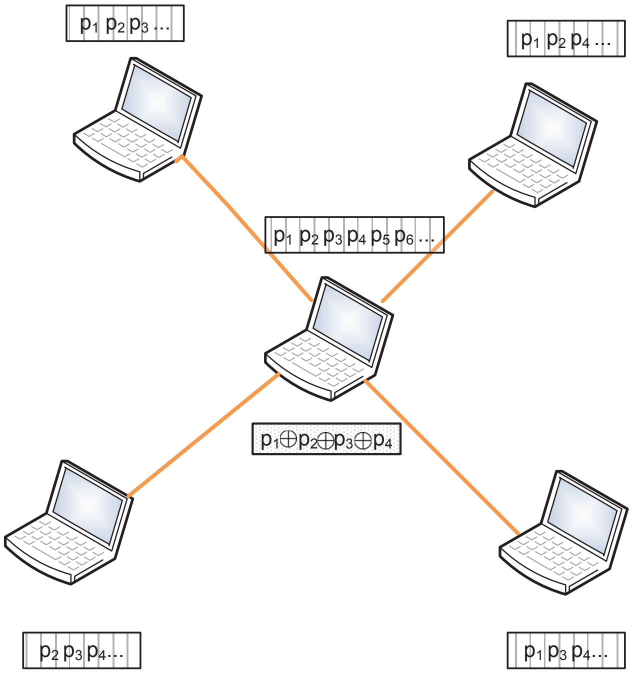

Figure 7.3: Maximizing coding gain by making combinations of n packets if all receivers already have $n-1$ packets of the same combination.

that the message will be decoded since some of the linear combinations might be linearly dependent.

7.3 IMPLEMENTATION OBJECTIVES AND EVALUATION SCENARIOS

In our test implementation, the network coding/decoding process is inserted on the top of the network layer of ns-3 simulator. This is an optimal placement due to the availability of information about packet source and destination that IP header contains, which is necessary for XOR coding/decoding. We are exploiting helper functionalities of ns-3 to build various testing topologies with desired characteristics.

Initially, we assume that the simulated network nodes have buffers of infinite size for storing packets. Every node keeps track of received and sent packets, based on which it gains knowledge about packets distribution in the neighborhood. Hence, during coding and decoding every node relies on local information about overall packet distribution, without contacting other nodes.

The knowledge about packet distribution in the neighborhood is of high importance for performance optimization, since every packet combination will not lead to successful data recovery and network coding gain. We aim at maxi-

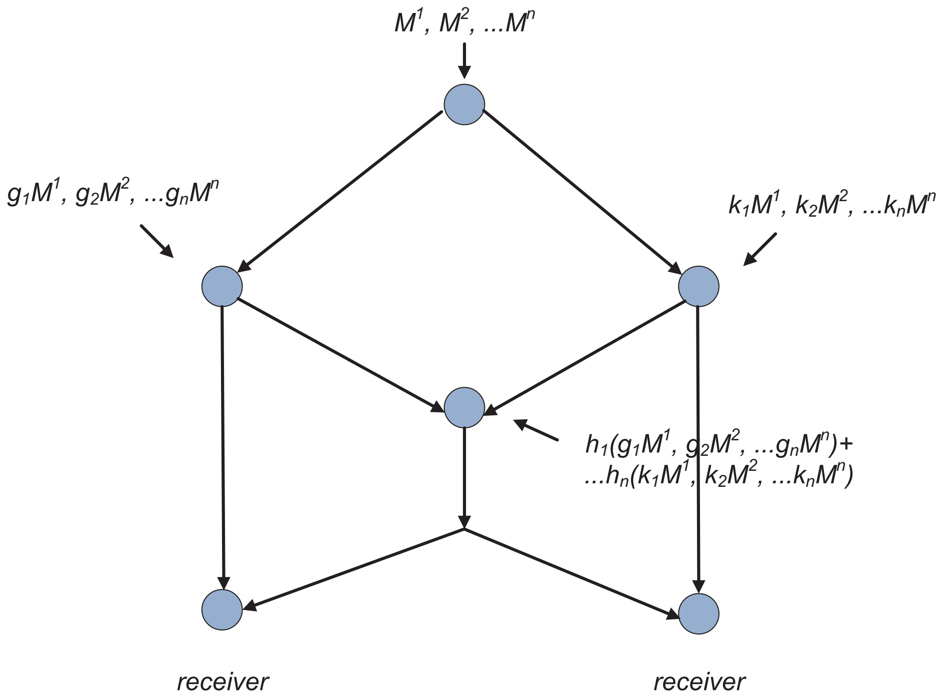

Figure 7.4: An example of linear network coding, where $M^1, M^2 ... M^n$ are source packets multicast to the receivers, and coefficients g_i, k_i and h_i are randomly chosen elements of a finite field.

mizing this gain by following the rule illustrated in Figure 7.3, only if the possibility of applying such rule exists. Therefore, each packet has to be created in a way that every neighbor after receiving packet combination and performing XOR operation with packets it posses, retrieves a missing packet.

This is possible only if node, receiving n packets combination, already owns $n-1$ packets of received combination. In other words, node having M neighbors, in order to make beneficial broadcast will make M packets combinations, ensuring that every neighbor already has $M-1$ native packets of that combination.

Before combining the packets to be sent, the node performs look up in its received packets buffer finding all packets with the same identifier which were received from $M-1$ different sources. Such packets, having the same ID, but which have been received from $M-1$ different sources denote one possible packet to be put into the final packet combination for broadcast. At the end, the node makes all possible combinations out of packets that have $M-1$ sources, and broadcasts it.

Information about combined packets is stored in metadata of packet combination to facilitate decoding process. Upon receiving the packet, the node performs lookup in the buffer one more time to find combination of packets that has to be XORed with received combination in order to retrieve a missing packet. Validity of resulting packet is checked performing the checksum over the packet payload [165].

Our implementation of XOR coding/decoding at the sender and receiver sides is seen as auxiliary process to common sending and receiving procedures. Thus, after receiving the packet, node first checks if incoming packet represents a packet combination. If this is not the case it performs ordinary processing of packet like in the case that no coding/decoding mechanisms are implemented.

On the other hand, while sending a packet, the node first checks if there is possibility to send a packet combination based on information it has about the packet distribution in its neighborhood. If such opportunity does not exist it sends packets following the regular pattern. In other words, applying the network coding in our network model is opportunistic, because it performs coding operation when such possibility arises.

With respect to the implementation, random linear network coding appears as simpler due to the fact that node does not need to have overall knowledge of its surroundings, as the packets are combined probabilistically. Our implementation of this model relies on ns-3 random number generators for creating coefficients and selecting packets to be combined.

A node generates output packet as a linear combination of randomly chosen packets from the buffer and random coefficients $g_1, g_2, ...g_n$. The structure of packet combination is recorded in the form of vector of coefficients and packet IDs and stored in the metadata of packets. Based on this information decoding side is aware of each packet composition and performs Gaussian elimination after sufficient number of combination packets of the same structure is received (n packets with linearly independent vector of coefficients). Overhead introduced by such additional information is negligible compared to the packet size.

Our network coding scenario resembles constant bit rate communication between nodes. In our experiments the capacity of a point to point links is initially set to 4.5 Mbps with the delay of 5 ms. Network topology varies based on the number of nodes involved ranging from 25 to 121 nodes and the type of connectivity (grid or butterfly networks).

We additionally exploit the case of randomness in building network topology by applying certain probability of link existence between two nodes. Finally, we observe network coding performance applied to realistic networks by studying the performance in the Abilene topology. Basic traffic model relies on the OnOff Application of ns-3, which switches between transmission and idle states according to the predefined On and Off intervals. During the On interval, CBR traffic with defined data rate of 448 Kbps and packet size of 210 bytes is generated, while Off interval denotes no traffic. Similar to the randomness applied in building different topologies we create random traffic patterns by varying transmission probability of nodes in the network.

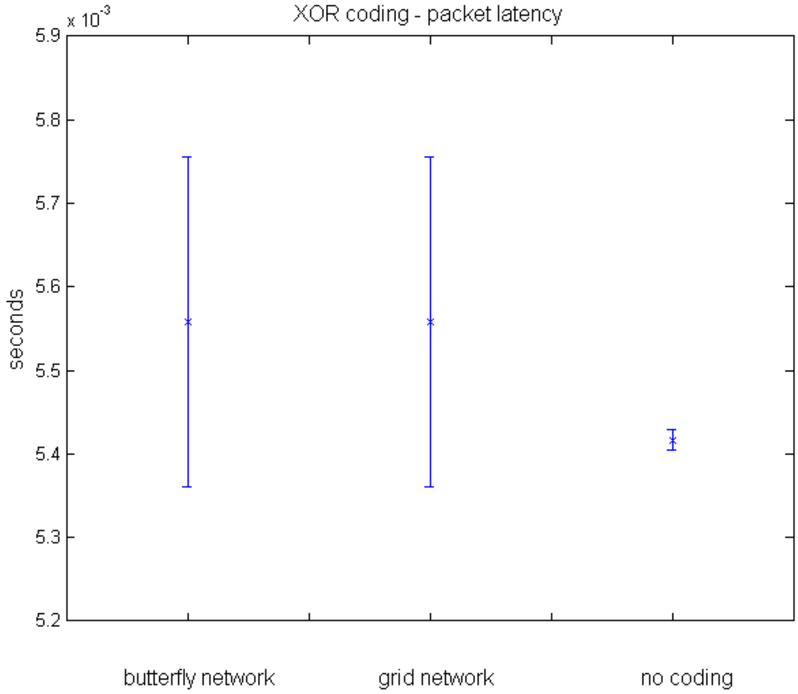

Figure 7.5: Mean latency and standard deviation for XOR coding (butterfly and grid networks) and no coding.

7.4 EVALUATION RESULTS

Large part of our obtained results are based on measuring network coding impact in terms of packets that are additionally decoded at the receiver side compared to the case without coding. We are varying data rate, packet delay, traffic patterns, as well as the underlying topology, and monitor impact of these parameters on the overall performance. Different traffic patterns are obtained by altering transmission probability for every node in the network. For underlying topology we choose butterfly and grid networks with the number of nodes ranging from 25 to 121, extending our observations with the random topology case generated by varying connection probability between nodes. Results show that benefit of both types of network coding highly dependents on a large set of parameters.

One of important concerns of network coding implementation is the mean packet latency introduced by the coding and decoding. The lookup procedure significantly adds to the overall network coding delay. Increasing the size of packet buffers enlarges the amount of data that each node has to be aware of and to process during the lookup procedure, leading to higher decoding delays.

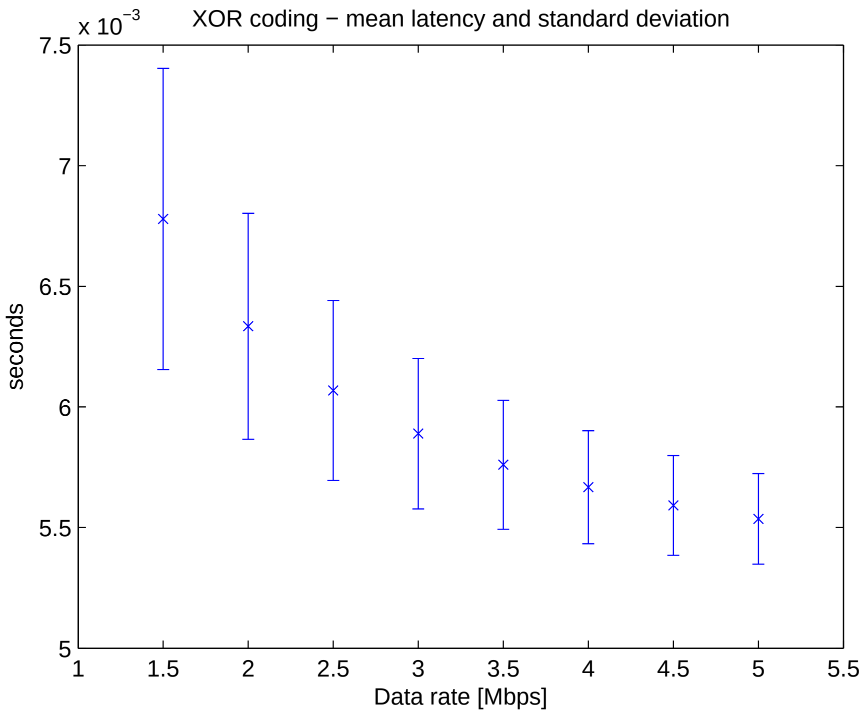

Figure 7.6: The mean latency and standard deviation for XOR coding over different data rates.

For XOR coding, having a relatively small number of packets P in the buffer that originated from $M-1$ sources, combination of M over P becomes very large (due to $P!/M!(M-P)!$), and the task of finding combination which results in benefits for all neighbors, time consuming process. On the other hand, if the size of packet buffer is not sufficiently large the node will not have enough information based on which it can perform beneficial coding.

Our results show that for network settings we observe latency caused by network coding is negligible for most of the applications. The difference in packet latency when XOR coding is applied over 36 nodes butterfly and grid networks compared with the case of no coding is illustrated in Figure 7.5.

Results show that latency is not strongly dependent on the underlying topology, but primarily varies based on the channel parameters and the data rate. The considered topologies are well connected, thus the nodes are able to send and receive sufficient amount of packets for successful decoding. For the fixed channel delay, latency added by XOR coding decreases with increasing the data rate of transmission as shown in Figure 7.6.

Evaluation of linear network coding latency and comparison with XOR coding case is done by repeating the tests under the same network settings as in the XOR case and fixing the data rate to 4.5 Mbps. Additionally, we monitor

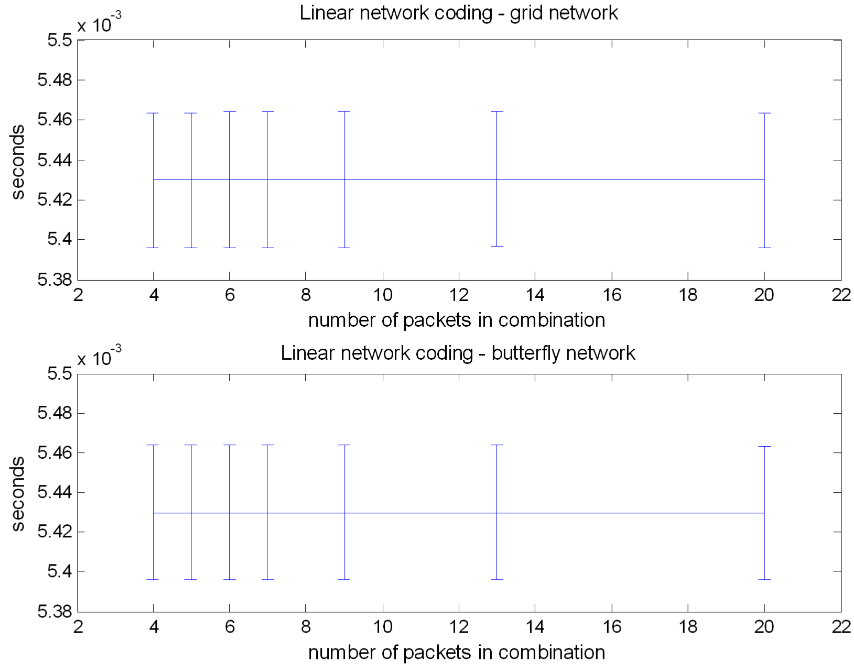

Figure 7.7: The mean latency and standard deviation for linear network coding (grid and butterfly networks).

the impact of packet combination size on coding/decoding delay. Our results show that in terms of latency linear network coding performs better compared to XOR coding for the same channel parameters.

Lower delay occurred when we applied linear coding for data rate 4.5 Mbps and had a fixed channel delay as shown in Figure 7.7. Moreover, Figure 7.7 demonstrates influence of packet combination size on latency. For relatively small packet combinations, e.g. containing less than 20 packets, Gaussian elimination does not introduce any noticeable additional delay due to the increased number of native packets combined. On the other hand, number of packets combined and underlying topology have impact on the performance of linear network coding in terms of decoding gain. Variation of packet number in combination leads to different decoding outcome for considered topologies as illustrated in Figure 7.8.

Increasing the number of packets in the packet combination might lead to performance degradation due to the higher complexity and longer time needed for a sufficient number of coded packets to be received. In both cases, XOR and linear network coding results depend on packets distribution and the number of packets each node stores in its buffer. We assume initially that nodes have unlimited buffer capacity for collecting packets. Thus, every received packet is stored in the buffer and potentially influences decoding outcome.

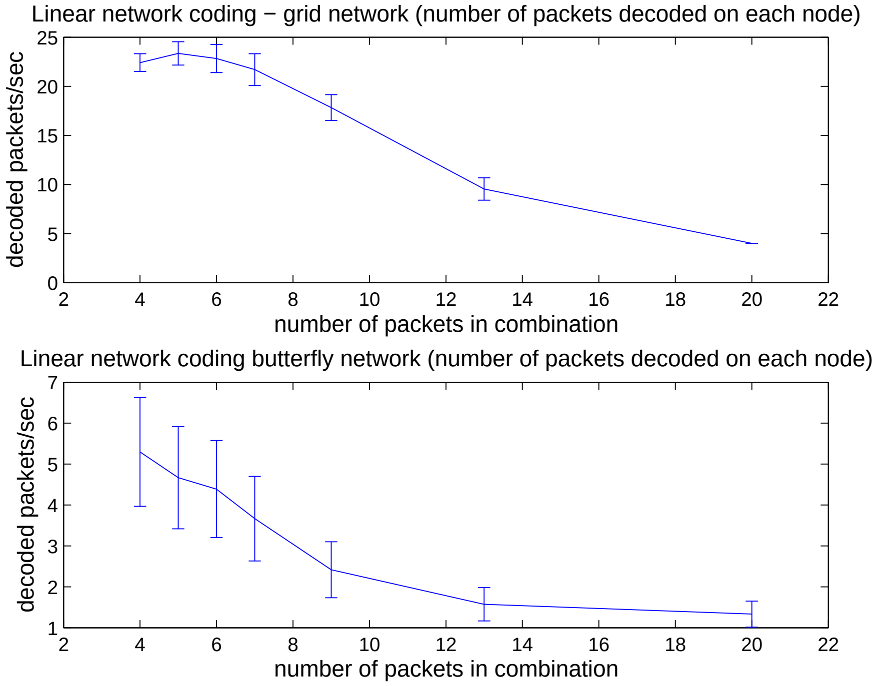

Figure 7.8: The mean number and standard deviation of packets decoded on each node in the case of linear network coding (grid and butterfly networks).

Based on packet buffer size at the time of coding/decoding decision network coding gain varies. In order to better address packet diversity and buffer assets' influence on decoding outcome we compare two network cases sharing the same network parameters set but differentiating in packet resources of nodes participating in communication. For such experiments we keep the buffer size of participating nodes unlimited.

Figure 7.9 and Figure 7.10 show the difference in decoding gain due to the buffer properties. Both, Figure 7.9 and Figure 7.10 illustrate the grid networks containing 25, 36, 49, and 64 nodes, while the node buffers of network from Figure 7.10 contain 30% less packets than the network in Figure 7.9. Nodes having 30% more packets stored in buffers are able to decode much less than 30% additional packets, thus the number of decoded packets on each node is not linear function of buffer size. Nevertheless, having larger packet buffers gives more combining opportunities to nodes and ensures higher network coding gain. The shape of the decoding curve varies based on the buffer size of the node.

Another interesting issue is to understand the relation between buffer size and decoding gain. As the first step in our evaluation work we limited buffers to particular size. In the case of receiving a packet after the buffer has reached

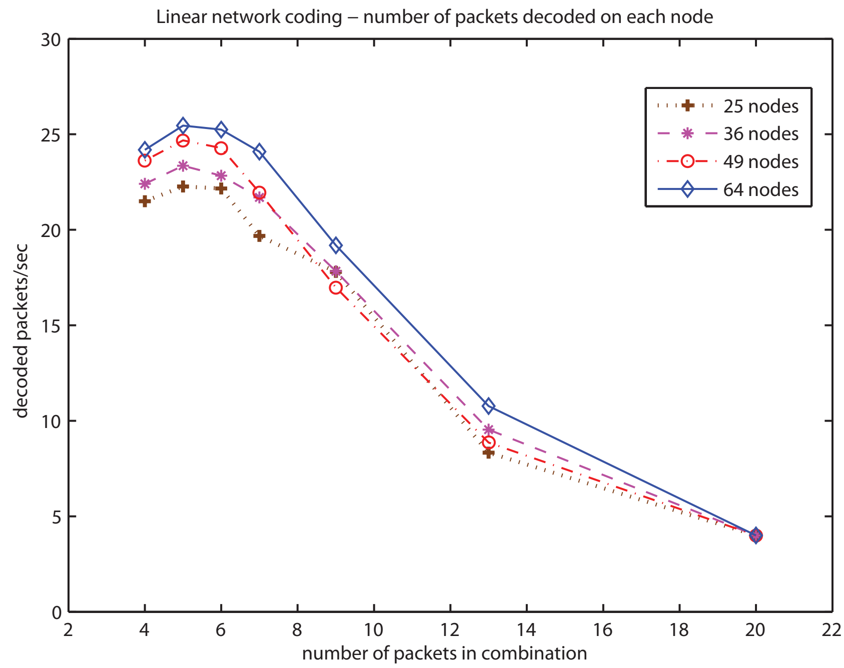

Figure 7.9: The influence of network size and buffer size on number of packets decoded in the case of linear network coding.

its limit, the node deletes the first enqueued packet in the buffer in order to make space for incoming one. We start our evaluation in monitoring the influence of buffer limit to decoding gain. We examine five different buffer sizes with 100, 150, 200, 350, and 500 packets.

Results show slight increase in the number of decoded packets due to the increase of buffer size. Applied to the same packet combination size a larger packet buffer does not bring significant decoding gain. Smaller packet combinations perform better like in the case of infinite buffers (cf. Figure 7.11). This raises the question on the utilization of large buffers and justification of applying them in network coding scenarios. While introducing additional processing costs, infinite buffers do not provide fundamental enhancement of network coding performance.

However, there is a significant difference in decoding gain while limiting lookup area at the receiver's side. If even less than 5% of available packets in the buffer are not considered during the decoding process the decoding outcome is heavily changed. Neglecting more than 10% of packets during the decoding lookup phase deteriorates network coding performance dramatically. Hence, the decoding gain does not justify implementation of network coding as illustrated in Figure 7.12. We observe that a relatively small number of packets exist that heavily influence the decoding gain. This observation have

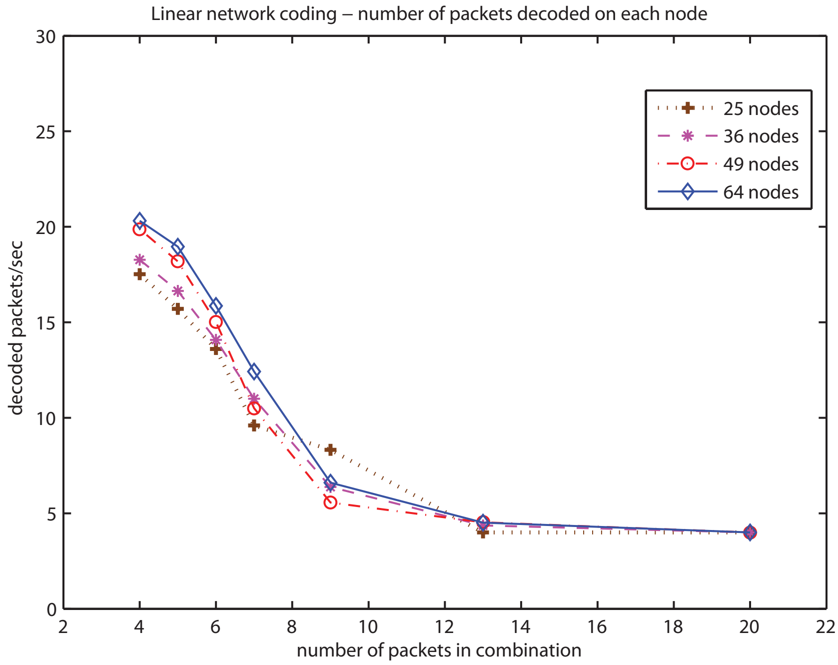

Figure 7.10: The influence of network size and buffer size on number of packets decoded in the case of linear network coding (30% less packets in buffers compared to the network setting from Figure 7.9).

potentially a large impact for considering the introduction of mechanisms for fast packet lookups with the possibility of false negatives. In any case one should note that disregarding even few packets during decoding lookup due to false negatives will cause poor decoding results.

The size of the network is another crucial parameter influencing the decoding results. Increasing the number of nodes involved in communication is linked with the increase of the number of decoded packets, for the same traffic and topology parameters applied. Figure 7.13 illustrates the impact of network size on number of packets decoded on each node for grid networks built upon 36, 64, 81, 100, and 121 nodes.

7.4.1 Random traffic and topology settings

Changing the traffic pattern so that all nodes are not active, influences overall performance of network coding in terms of decoding gain while packet latency remains the same. Determining the certain probability for each node to broadcast the data defines its participation in packet transmission, as well as in the network coding process itself. In our implementation we examine relation between given broadcast probability and network coding performance.

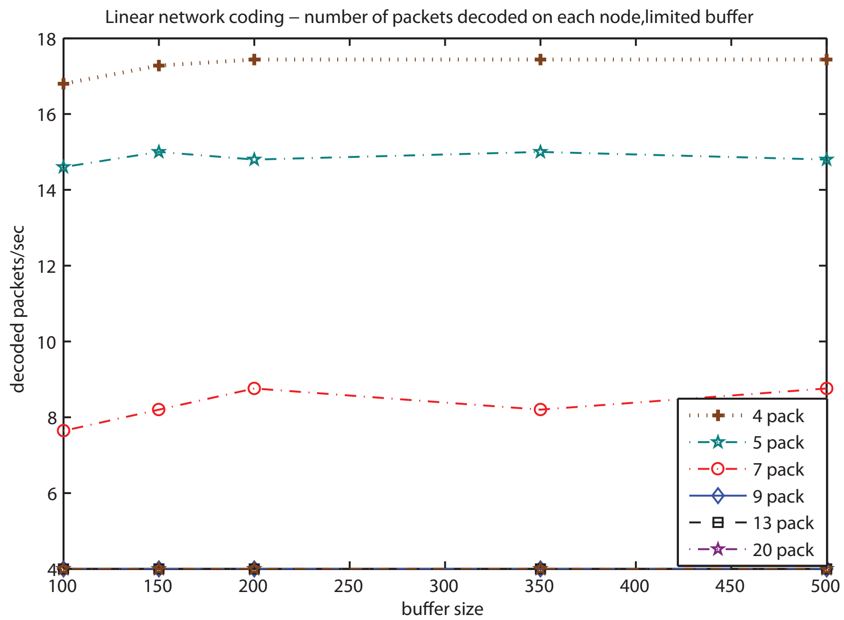

Figure 7.11: The influence of the buffer size and combination size on the number of decoded packets in the case of linear network coding.

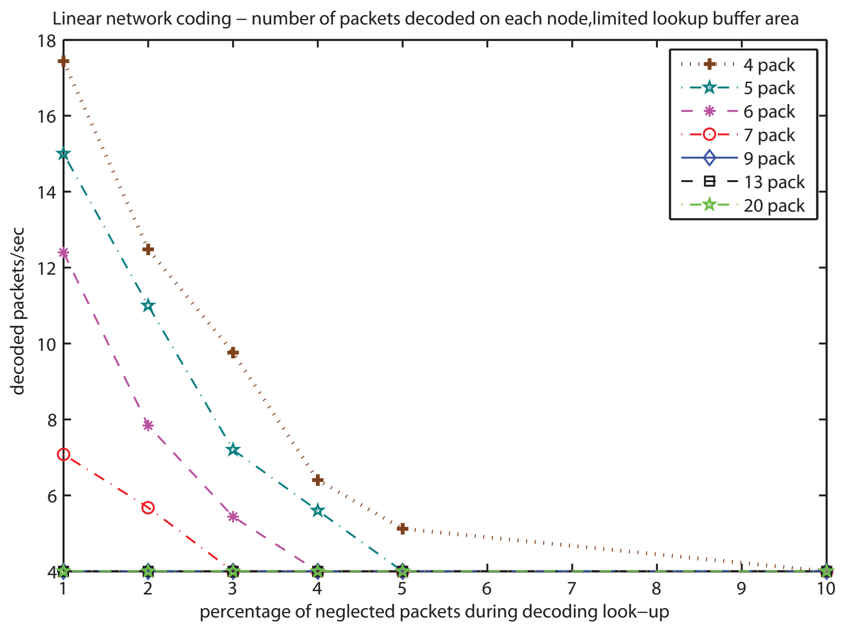

Figure 7.12: The influence of limiting decoding look-up area in buffers on the number of decoded packets in the case of linear network coding.

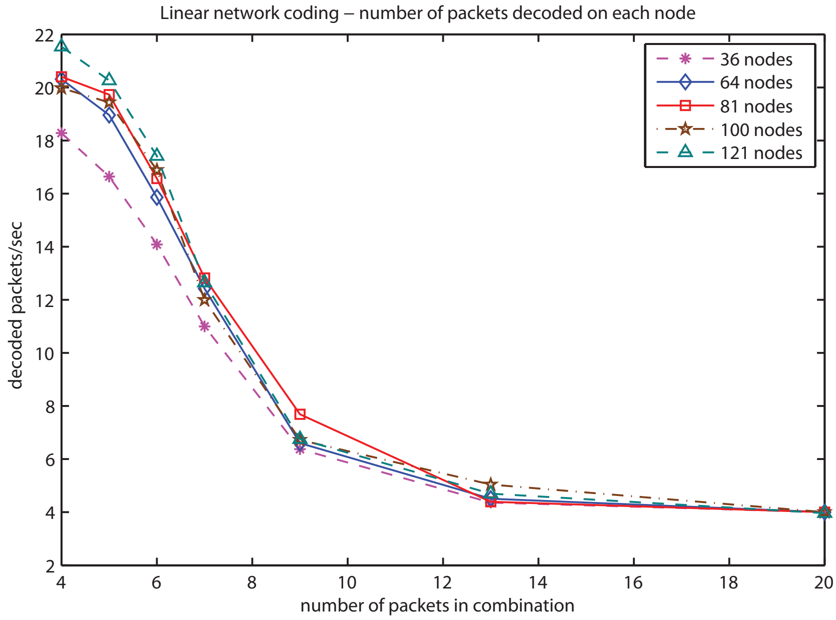

Figure 7.13: The influence of network size on the number of decoded packets in the case of linear network coding.

Figure 7.14 shows results for 90% broadcast probability (in each node). We observe that the mean number of decoded packets is proportional to this predefined probability (approximately 90% of network coding benefit compared to the case where all nodes broadcast).

The standard deviation for the number of packets decoded is high due to the large variations in the number of sent and received packets on the same node as illustrated in Figure 7.14. The high positive and negative peeks of standard deviation represent two extreme cases that may happen in random traffic scenarios. High positive values are reached when node is not sending packets, since it is always able to receive and decode, thus the difference between number of received and sent packets is exactly the number of received and decoded packets. On the other hand, the negative extreme is obtained when the node is sending packets but it is not able to receive anything from its neighbors since they are not involved in communication.

Setting the broadcast probability to be very low, e.g. 10% in our implementation, leads to negligibly low network coding benefit. In such a situation there are not enough packets traversing the network in order to achieve coding benefit.

We generate random topologies by defining probability of link existence between nodes and monitor network coding behavior under these circumstances. Results are similar to the case of random traffic pattern in the sense

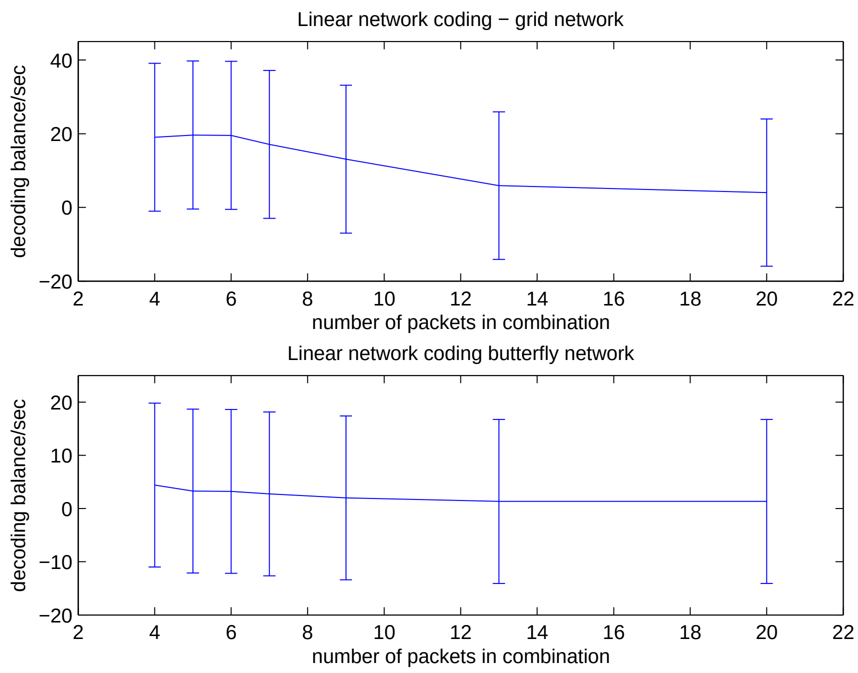

Figure 7.14: Random traffic pattern, mean number and standard deviation of decoded packets (node broadcasts with 90% probability).

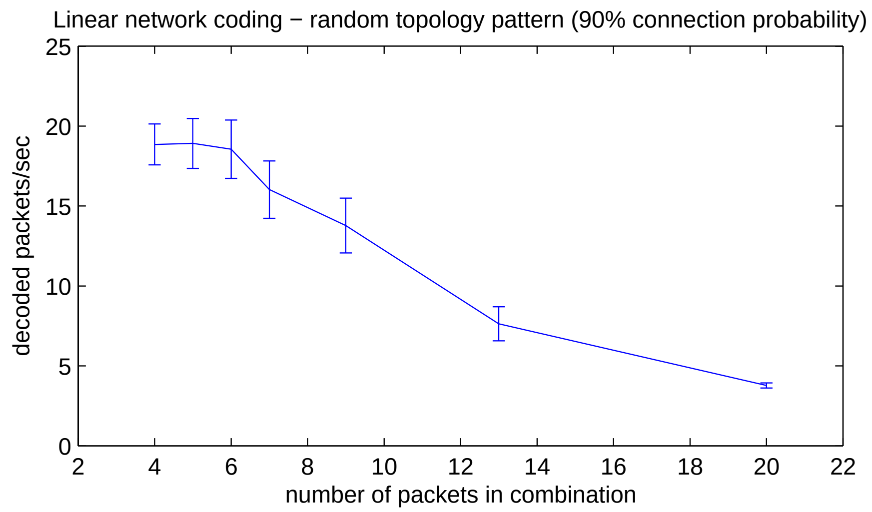

Figure 7.15: Random topology pattern, mean number and standard deviation of decoded packets (90% probability of link existence between two nodes).

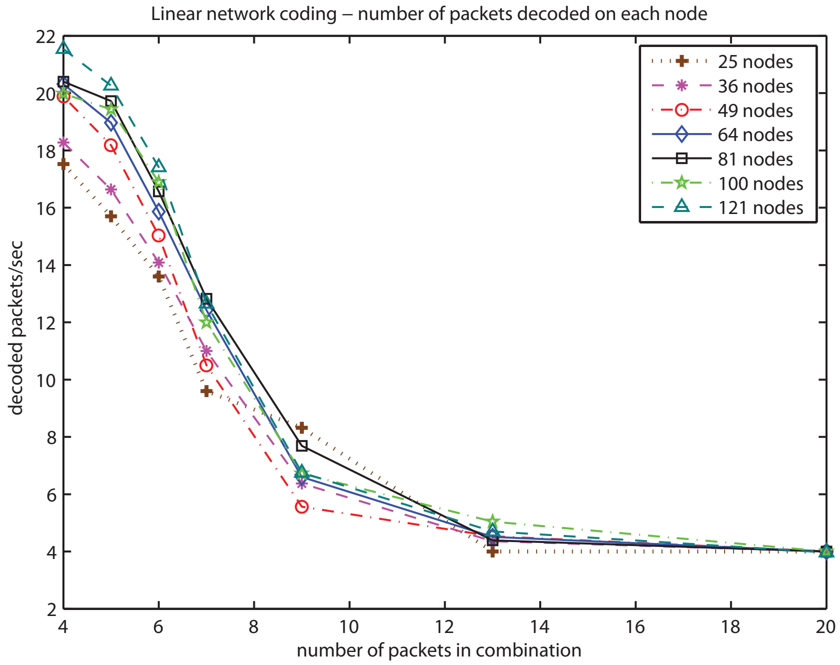

Figure 7.16: Grid networks built upon 25, 36, 49, 64, 81, 100 and 121 nodes with 100% connection probability between nodes.

that network coding gain is proportional to the given link existence probability. The gain degrades dramatically and can be disregarded for low link connectivity probabilities due to the network disconnections. Opposed to the random traffic pattern case, the standard deviation of the number of packets decoded is relatively small since there is no possibility for node to send neither to receive to/from no existing links as shown in Figure 7.15.

In order to better understand the dependency of decoding benefit due to network connectivity, we extended out analysis to larger networks and probed the "critical" connection probability between nodes. We chose 55% link existence probability to avoid cases of obvious network disconnections due to low probability, i.e. we keep the network still connected as one single graph. One should notice that the decoding gain is not anymore proportional to connection probability as it was in the case of high connection probability.

Results show that decoding curve has the same shape as for the networks with the higher link existence probability. However, the network coding gain is much lower due to network disconnections and nodes lacking sufficient information to perform better coding/decoding, as shown in Figure 7.16 and Figure 7.17.

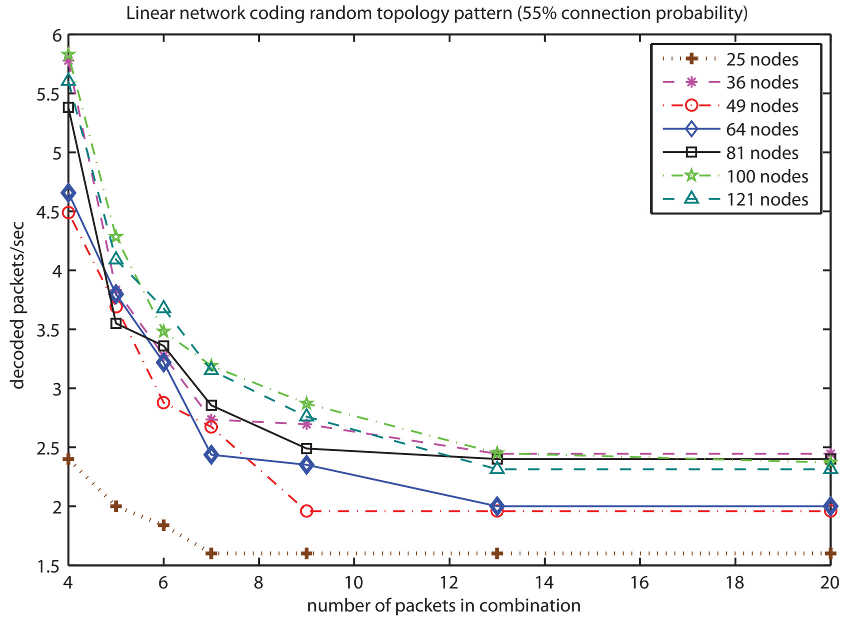

Figure 7.17: Grid networks built upon 25, 36, 49, 64, 81, 100, and 121 nodes, with 55% connection probability between nodes.

7.4.2 Abilene network use case

As a final extension of our work we study the behavior of network coding applied to the realistic topologies. We use high speed research network Abilene [156, 166] as a test case. Abilene is 10 Gbps connected network, mainly serving for educational and research purposes. It represents a backbone network with the goal of supporting advanced network applications and evaluation of their performance before integration with widely used systems. Configuration files of Abilene routers are publicly available for research purposes. Thus, the exact topology settings of this network can be used for simulation and testing purposes.

Our topology consists of 12 main nodes with link capacities and connections as described in Abilene backbone specifications. Network is shown in Figure 7.18, and we will apply network coding in all communications in this backbone network. Our aim is to use the same traffic generation parameters as applied in topologies that we have already tested. Thus, the evaluation goal is to find out the impact of Abilene topology on network coding performance. Results show network coding benefit followed by large deviations in number of packets additionally decoded as shown in Figure 7.19. Relatively poor decoding gain is related to the type of traffic used.

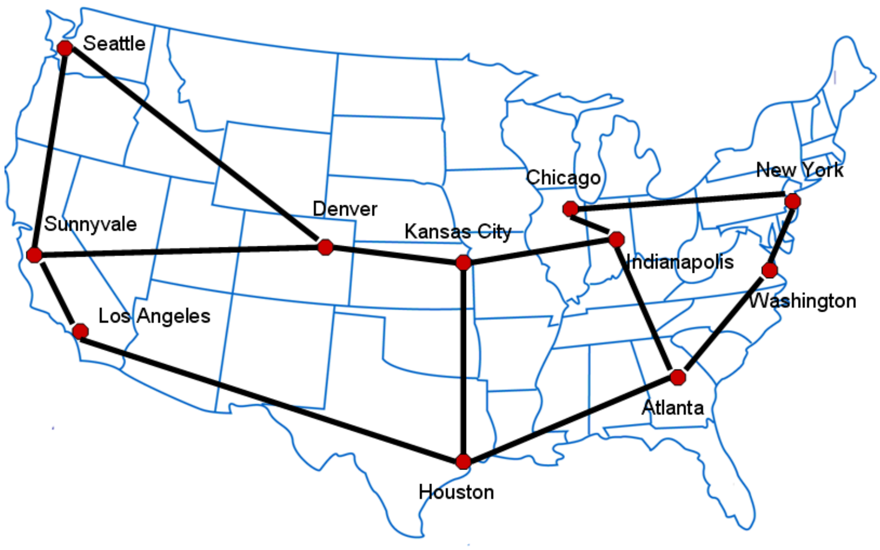

Figure 7.18: Abilene backbone topology.

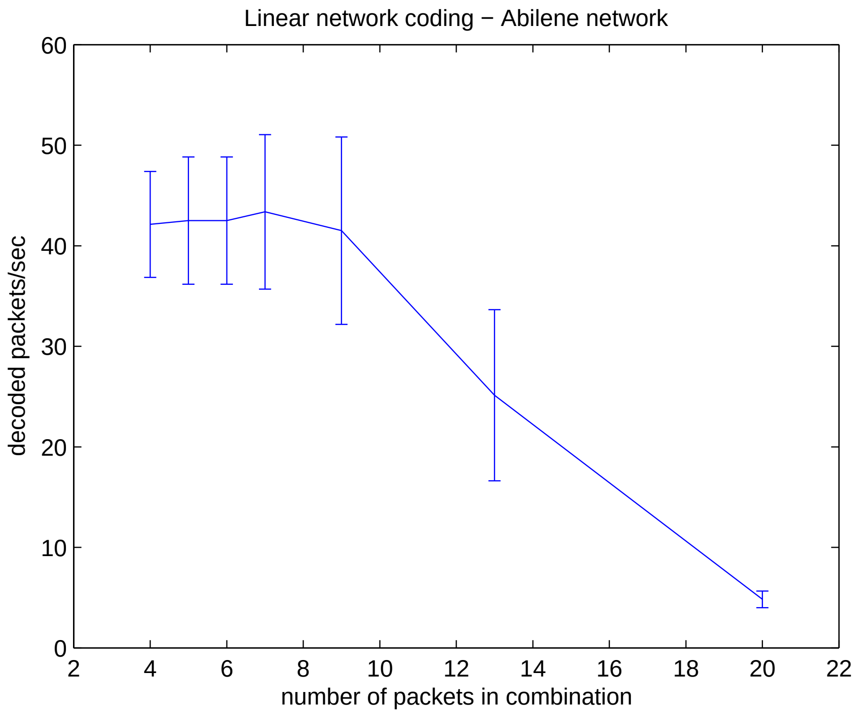

Figure 7.19: Abilene topology. The mean number and standard deviation of packets decoded on each router in the case of linear network coding.

7.5 NETWORK CODING IN INFORMATION-CENTRIC NETWORKS

After the above general evaluation of network coding mechanisms we now focus on considering network coding in information-centric networks. As in the earlier cases we consider intra-domain based information-centric networks. This is well-justified, since as the general results have shown the network coding gains will be diminishing with the increasing size of the network. Under intra-network conditions we expect that the network coding does not experience major deterioration in performance when applied to publish-subscribe architecture. We study in this section if this assumption is valid.

We aim at applying the network coding mechanism on opportunistic basis, i.e. only when the network conditions necessary for obtaining performance gain are fulfilled. Therefore, implementing the network coding as a helping functionality of information-centric network appears an attractive solution. According to the input collected from the set of helper functions presented within this thesis the network coding helper function will determine the current network state. Based on such input the conclusion upon the use of network coding for data distribution can be made. Furthermore, the choice in terms of the most suitable network coding techniques can be made. This proposed method is a novel and original approach, as far as we are aware of, for network coding in the information-centric networking.

Such mode of operation can significantly improve the network performance by applying the network coding when appropriate conditions exist. On the other hand, this will avoid unnecessary signaling overhead by avoiding the network coding use in the cases where the network coding is not suitable. However, since the network coding introduces additional delay when applied for data distribution, such technique may not be suitable for real-time, high-performance services. We expect it to be better suited for more delay-tolerant data transfers, and this policy can be enforced by publish-subscribe hierarchy if necessary.

7.6 ADDITIONAL CODING MECHANISMS

Apart from network coding, many source coding techniques have been developed for transforming messages in the form that improves the probability of reception and successful decoding. Therefore, another important coding-based helper function in information-centric networks can be responsible for error control and correction in data transmission. By means of dedicated helper functions it is much easier to have different channel coding and error control mechanism in the network that can be activated at will. This is much more flexible than fixed FEC and similar mechanisms that are implemented in traditional networks. Usually such mechanisms can not be changed and can be used only if all the connecting entities agree beforehand to have those capabilities. They can be also implemented as a part of application protocol in which case they are often inefficient and work only in end-to-end basis and only for particular flows.

In order to be able to recover partially corrupt messages at the subscriber's side without retransmissions, publisher can add redundant data to its message, i.e. error correction code. After receiving the message containing redundant data, the subscriber is able to discover possible errors and to successfully correct them. Following such a principle it is possible to omit the use of back channel, since the probability of correct data transport without retransmissions increases. Let us illustrate this with a more precise example in the following.

One of the first coding techniques applied is Reed-Solomon (N_{rs}, K_{rs}) block coding [167, 168, 169]. The main idea behind this technique is that encoder takes K_{rs} information symbols of m_{rs} bits each and adds $N_{rs} - K_{rs}$ parity symbols to make N_{rs} symbols codeword. Based on such added redundancy, the subscriber is able to decode complete messages even if some of the symbols are lost. In general case, a decoder can correct up to $(N_{rs} - K_{rs})/2$ error symbols in received codeword.

The most used Reed-Solomon code is RS$(255, 223)$. It consists of 255 code words of 8 bits, where the number of parity symbols is 32, which implies that it can correct up to 16 erroneous symbols. Errors can occur at the single bit in the symbol, or at all 8 bits of it. Both of mentioned situations will be considered as one error symbol. This means that algorithm can correct up to $m_{rs} * (N_{rs} - K_{rs})/2$ erroneous bits (or in this example 8*16 bits). It can correctly decode a symbol with the same success, either the error was caused by one bit being corrupted or all bits in the symbol being corrupted. This gives a Reed-Solomon code great burst-noise decoding capability, making it especially appropriate for transmissions over wireless links.

In the information-centric space the input from different helper functions can be utilized for the estimation of the network state and selection of the most appropriate coding parameters. In such a way the coding functionality can be optimized, while minimizing the unnecessary redundancy.

Another promising coding technique is based on fountain codes [170, 171]. In contrast to traditional transmission and coding techniques which chop the message to be transmitted into parts, send each part of it separately, and wait for acknowledgment from destination, fountain techniques send randomly all parts of the message which is slightly extended beforehand by adding redundant data. Fountain codes are rateless since the limitation in number of encoded packets generated from the source message does not exist, and can be changed dynamically. Source can send as many encoded packets as is needed for receiver to successfully recover the data.

According to the network conditions different number of encoded packets may be sufficient for correct data reception and decoding. Such an information can be obtained from information-centric lower-level helper functions. Nevertheless, static determination of FEC (Forward Error Correction) mechanism properties, e.g. code size to be used may degrade its performance due to inaccuracy in estimation of underlying channel state. Furthermore, possible variations of environment characteristics can additionally hinder the FEC operation. For example, changes in BER over time may cause inefficient bandwidth usage due to the fixed FEC schemes applied.

Therefore, continuous subscription to lower-level information-centric helper functions and receiving the constant updates on current network states can significantly improve the error correction mechanism or even completely bypass the aforementioned problem. In order to optimize the implementation of error correction mechanism in information-centric networks we envision it as a separate helper functionality. Applying the FEC helper function might represent the ideal way to collect the data necessary for decision on appropriate FEC mechanism to be used.

Due to the modular nature of helper functions, there are many possibilities for implementing the FEC helper functions. In a general case, a FEC helper function can subscribe to data on which FEC coding needs to be applied. Furthermore, based on current environment information it collects from responsible helper functions, it can perform suitable FEC scheme with the most appropriate parameters. Such an operation might be especially beneficial near wireless edges of the network to provide additional protection over lossy segments.

7.7 TRANSCODING HELPER FUNCTIONALITY

Over last years we have experienced a fast growth of the number of data formats used in communication, followed by the proliferation of various electronic devices that support them. Therefore, due to the need of delivering large amount of data in different formats, applicability of various coding mechanisms became highly important. Multiple coding and re-coding schemes are performed on one data format to transform it into the structure that can be transmitted over specific media or interpreted on specific device.

This adaptation process, called transcoding, is essential for future continued evolution of various data formats, e.g. video data format. The performed changes can be with respect to various parameters, e.g. bit rate, resolution and format of existing video content in order to adapt it to displaying on another device.

In order to transform the required data in appropriate format, the overall picture of the network characteristics is needed. For example, in transmitting video over heterogeneous network over the links with different characteristics and capacities selection of appropriate coding according the source properties, only, does not always result in optimal performance. Often, the source performs coding with respect to the capacity of "the weakest" link on the route, sacrificing quality of video. Besides problem of channel characteristics, the different end user devices introduce constrains due to the lack of capabilities required for high quality video decoding and displaying. Therefore, the helper functions can be applied as a mechanism for collecting the information required for transcoding.

Dedicated helper function can monitor capacity and characteristics of links all over the path, as well as characteristics of end user devices. According to information it obtains it can make decision about appropriate transcoding mechanism. Furthermore, as the number of coding/decoding standards is constantly increasing (MPEG-2 [172], MPEG-4 [173], H.261 [174], etc.), the need for general mechanism which performs the translation between them becomes more evident. Due to the modularity and easy integration of helper functions, using this mechanism as a general solution for standards translation seems like the most flexible approach.

7.8 EVALUATION FRAMEWORK FOR INFORMATION-CENTRIC NETWORKS

The insight about performance of the network coding applied in information-centric network can be obtained by using different simulation tools, as it has been presented within this thesis. However, the simulation results frequently do not satisfy the reliability requirements. Thus in order to verify the simulation results and to understand better particular implementation challenges actual testbed and test network environment are commonly used.

Due to the increased hardware costs and setup complexity accompanied with such an evaluation approach, the large scale experiments are difficult to perform directly in the testbed. Therefore, a combined approach, taking advantage of testbed reliability, on one side, and the possibility for easily building of large-scale experiments in a simulation environment, on the other side, appears as optimal solution. These approaches are sometimes referred as ”network-in-the-loop” or ”hardware-in-the-loop” approaches in the simulation and testing community. Such approaches can be utilized not just for helper functions testing, but for detailed, large-scale evaluation of many other techniques in information-centric domain.

7.8.1 Testbed environment

As mentioned earlier a large part of our experimentation work is conducted in the testbed environment. As a part of the PURSUIT project we had at our disposal the international testbed with dozens of computers for testing the concepts of this thesis. The common PURSUIT testbed interconnects not only the project partners across the Europe, but is extended towards additional sites in USA (MIT) and Japan (NICT). About 40 machines in total are connected in

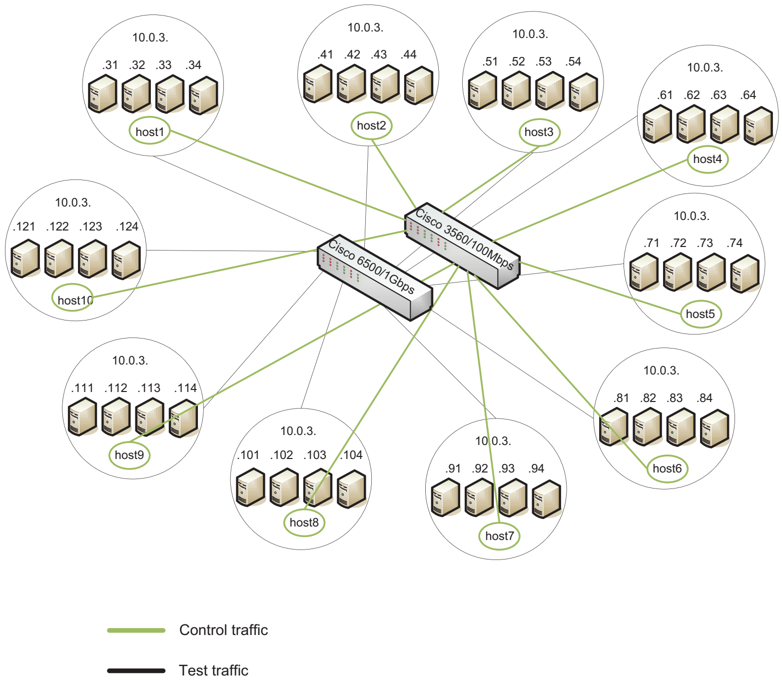

Figure 7.20: The architecture of the internal high speed testbed.

the virtual private network and run the PURSUIT prototype implementation that can be used for our experiments.

Apart from the common PURSUIT testbed we have deployed a high speed internal testbed for experimentation on information-centric concepts. Avoiding the use of the virtual private network in our internal testbed makes the data transfer more efficient. The testbed consists of a collection of ten powerful servers equipped with multi-core CPUs, 8 GB RAM memory, 320 GB disk space, and four gigabit Ethernet network interfaces each. The connection between the servers is established through a high performance switch of Cisco 6000-series allowing for different network topologies to be set up for the experiments. For increasing the number of nodes available for the experiments the virtualization techniques are used. In initial setup each of the servers is hosting four virtual machines running the PURSUIT prototype implementation, giving the total number of 40 machines available for testing purposes. The number of virtual machines on each server can be further easily increased. Individual virtual machines can have the role of publishers and subscribers, or nodes associated with the network infrastructure such as forwarding nodes, rendezvous points or topology management nodes. Figure 7.20 illustrates the architecture of our internal high speed test network.

In order to further extend our testing platform and to more closely examine the performance of information-centric network equipped with different helping mechanisms, such as network coding, we aim at improving the evaluation platform by extending the commonly used simulation tools and integrating them with our testbed. We focus especially on adaptation of ns-3 simulator to be used within information-centric concept. Furthermore, one of our main objectives is to keep the reality of experiments on high level.

7.8.2 Integration of simulation and testbed experiments

In order to enable the large-scale experiments by integrating the simulation and testbed environments we have incorporated the information-centric network implementation directly into ns-3 simulations. For this purpose we use the PURSUIT project [77] prototype implementation called Blackadder [143], which provides a full implementation of PURSUIT publish-subscribe architecture.

The implementation of the Blackadder prototype is based on the Click modular router framework [175]. The Click module facilitates building of user-defined flexible packet processing configurations. Therefore, it represents the powerful platform for building information-centric networks. The routing in the Click is done by using the combination of several packet processing entities called *elements*, where each of the *elements* performs predetermined operation in the packet processing chain. Based on such scheme the routing in information-centric space is done, thus every Blackadder router is defined by the Click graph containing the corresponding Click *elements*.

The integration of Blackadder with ns-3 is performed based on the Click support in ns-3 simulator. Such ns-3 feature allows addition of the Click elements directly into the ns-3 simulations. The simulated nodes route the packets according to the protocols defined by the set of Click *elements*. Figure 7.21 illustrates the protocol stack of ns-3 node with integrated Click packet processing. For designing the routing protocols the default Click *elements* or the user defined ones can be utilized. This characteristic of ns3-Click feature is our main objective in incorporating the Blackadder Click elements in ns-3 environment.

7.8.3 Implementation objectives for building the integrated environment

In order to preserve the reality and reliability of the experiments we aim at using the core elements of Blackadder prototype for defining the ns-3 simulated nodes. Such an approach makes the behavior of simulated nodes identical to the real Blackadder nodes deployed in the testbed. Therefore, the simulated nodes have the ability of performing the ordinary Blackadder operations, e.g. publishing and subscribing to certain information item, while the data is transmitted over simulated links. This provides us the possibility of performance evaluation of information-centric networks under conditions nearly

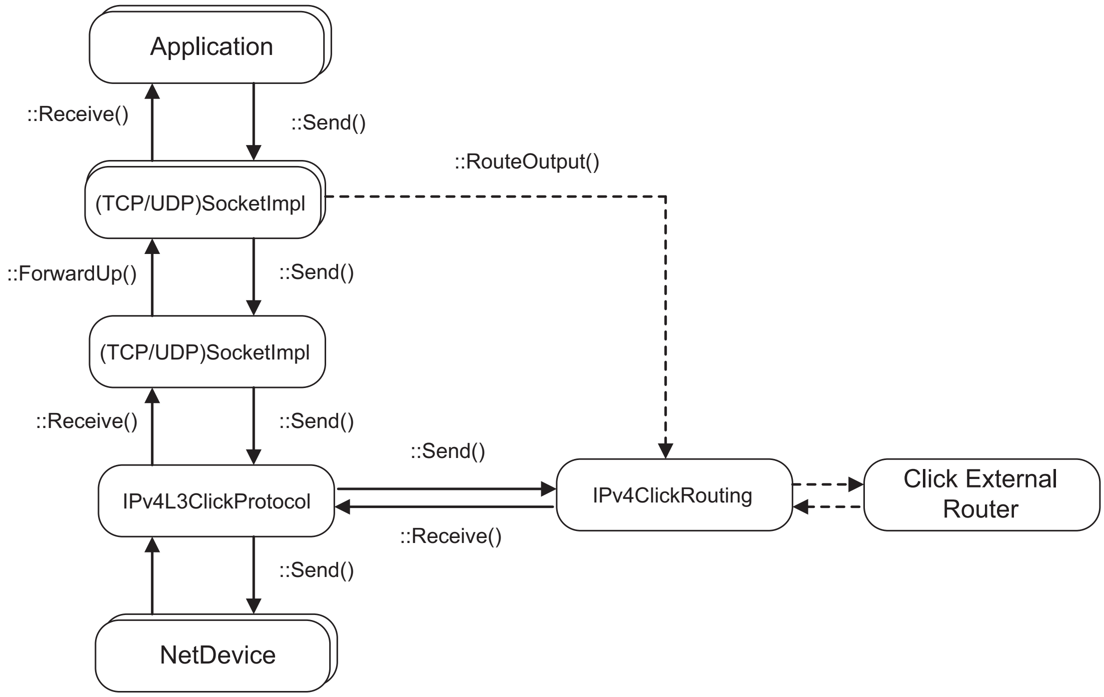

Figure 7.21: Path of the protocol stack in ns-3-Click. At Layer 3 the packet is passed to the Click and back via Ipv4L3ClickProtocol and Ipv4ClickRouting.

identical to the testbed environment, without increasing the hardware cost or the complexity of the experiments setup.

The communication between the Blackadder applications and the core in the prototype implementation is based on Netlink sockets. Due to the lack of direct support for Netlink sockets in ns-3 we aim at emulating the Netlink functionality in order to establish the connection between the Blackadder applications and the ns-3 simulator. However, such an approach experiences certain shortcomings due to the tight connection between existing ns-3-Click integration and IP protocol. This produces unnecessary overheads during the exchange of events between the ns-3 applications and the Blackadder core.

Nevertheless, building the Blackadder support in the ns-3 as a stand-alone module seems as more promising solution. Thus we have built different modules to adapt packet flows created by the Blackadder API so that those can be used in the ns-3 and routed according to the defined Click graph. The main functionality is provided by two modules, namely *BlackadderAPI* and *BlackadderClick*. The *BlackadderAPI* class provides the PURSUIT service model for the applications installed on simulated nodes. Each application thereby uses a distinct instance of this class. The *BlackadderClick* class is the main coordinator for packets distribution. It communicates with the *BlackadderAPI* instances of

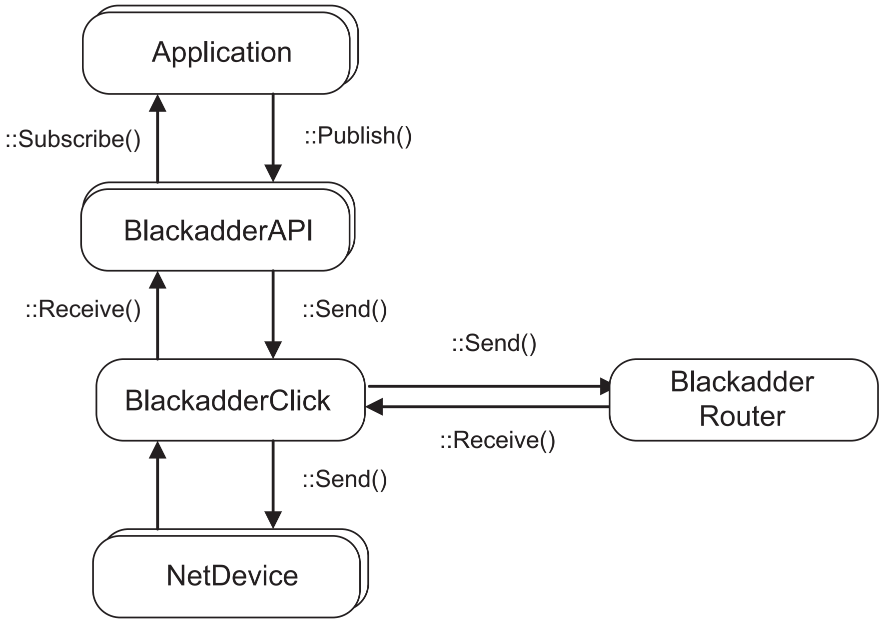

Figure 7.22: Path of the protocol stack in ns-3-Blackadder. The *BlackadderClick* module adapts the ns-3-Blackadder application packet flow to be routed by the Blackadder Click graph.

the various applications, as well as it receives and sends packets via the network devices of the node. Only one instance of *BlackadderClick* is created per node. Figure 7.22 illustrates the protocol stack of ns-3 node with integrated Blackadder packet processing. The ns-3 applications use the *BlackadderAPI* for publishing and subscribing operations. Such packet flow is adapted by the *BlackadderClick* module in order to be recognized by the Blackadder Click graph which defines node's routing functionality. The integration of simulation and our available testbed gives us the possibility to simulate hundreds and even few thousands of nodes which allows us to study scalability and more realistic performance within mid-size intra-domain systems.

7.8.4 Reliability of integrated environment

In order to evaluate the reliability of integrated Blackadder and ns-3 environment we performed a variety of experiments. We applied the same testing conditions on the same scenarios in both cases, ns-3 simulations and the testbed. The examined network has one publisher and two subscribers which use simple application for retrieving the data. We have carried out the experiments in wired, as well as wireless (ad-hoc) networks. For initial experiments three different scenarios are taken into consideration: having a single publisher and a single subscriber ($P_1 \rightarrow S_1$), a single publisher and two subscribers ($P_1 \rightarrow S_1, S_2$), and a pair of a publishers and the subscribers ($P_1 \rightarrow S_1, P_2 \rightarrow S_2$).

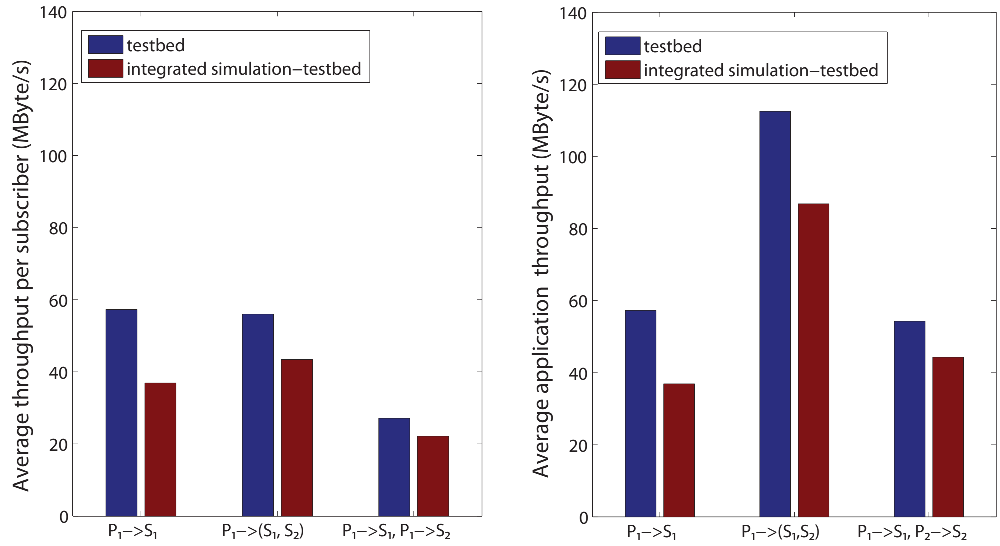

Figure 7.23: Comparison between real testbed and integrated simulation-testbed experiments in wired network.

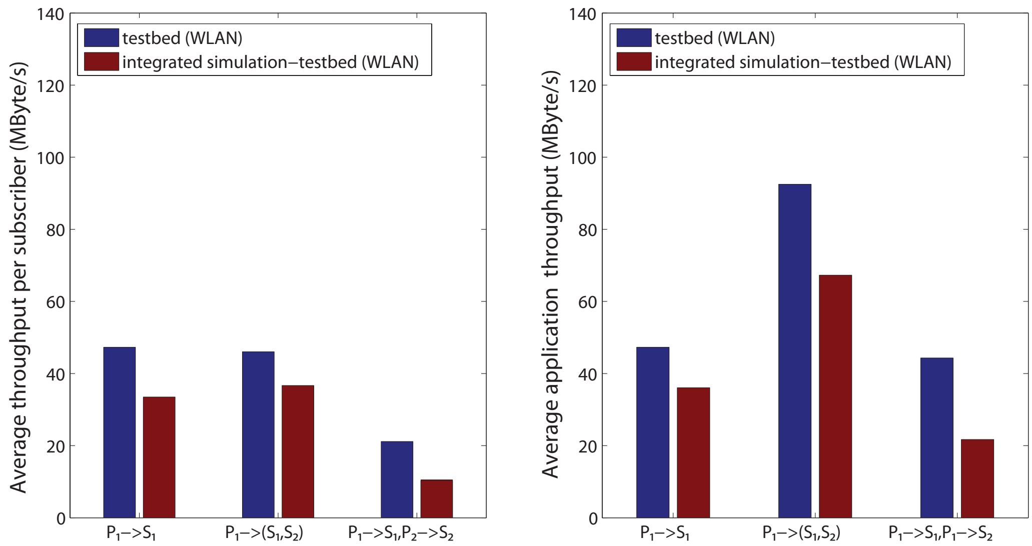

Figure 7.24: Comparison between real testbed and integrated simulation-testbed experiments in wireless network.

The output from our first experiments shows the promising results in the context of reliability of integrated simulation-testbed environment. Such results are very similar to those obtained from the realistic tests, as illustrated in Figure 7.23 and in Figure 7.24. Thus, the reliability of integrated simulation-testbed environment is on extremely high level, comparable with the real networks. Furthermore, such integration gives us the possibility for easy extension of the network size, therefore, performing the large-scale experiments without additional hardware requirements and added complexity.

These initial experiments and confirmation of good reality and reliability properties of integrated simulation and testbed environments give us the motivation for performing larger-scale experiments in our future work. Building such networks is significantly facilitated due to the nature of simulation-testbed integrated environment.

7.9 CONCLUSIONS AND FURTHER DISCUSSIONS

In this chapter we have presented the main idea behind network coding mechanism while focusing our evaluation particularly on XOR and linear network coding. Our results showed that combining numerous data units so that the designated destination is able to recover them, while not escalating the amount of data transmissions, is not a straightforward task. Performance gain of applying network coding is highly dependent on various input parameters in terms of channel properties, traffic patterns, and network topology. In addition, possible latency and complexity issues must be taken into account carefully when using network coding in different scenarios.

Apart from parameters and topologies that have been analyzed in this chapter, there are many potentially interesting network settings to be investigated in the future. One interesting extension would be to consider specifically the role of wireless channel errors for the performance gains. Additionally, network coding should be analyzed under different traffic conditions, and using larger network topologies in order to get more complete picture of network coding applicability.

The novel part of this work has been the discussion and analysis of the network coding in information-centric networks. We have shown the potential benefits of network coding when applied as a helper functionality to allow an opportunistic use of it under different network conditions. However, the introduced transmission delay may be too high for real-time, low-latency applications.

Finally, we have described our simulation-cum-testbed framework and implementation that has been done as a part of this thesis work to test performance of information-centric networks. By combining the simulation and testbed environment we are able to preserve the high level of reality and reliability. This approach has also allowed us to verify the simulation results against the actual hardware measurements in the case of a small number of nodes. The development of the testbed has been done in the context of joint European PURSUIT project, and provides quite unique environment for the further research on information-centric networks.

8

Conclusions

An obvious shift in user demands has resulted to the situation where increasing number of services and protocols are departing from the strict end-point centric communication. One of the key shifts has been that the information retrieval and sharing has become more dominant communication paradigm instead of end-to-end unicast flow based communications. This has caused the model mismatch between the data-oriented nature of the communication patterns and the end-point centric nature of the Internet architecture as it is implemented today. As an outcome of such a mismatch the network resources have been used in suboptimal way, and indeed many complex overlay structures are deployed at the application layer to provide better information centric frameworks.

This thesis work has focused on enhancing the information-awareness throughout the network starting from the lowest network layers. Such an approach will provide the means for a smooth integration of additional network mechanisms and will hopefully facilitate the wide-scale deployment of information-centric concepts in the future.

The wide-scale implementation of information-centric networks and the complete replacement of the current Internet still appears as an ambitious aim. However, a number of significant projects in this direction have already been initiated, resulting to new information-centric architectures offering a variety of solutions to current Internet problems. Such architectural proposals are being carefully evaluated with respect to different objectives. One should also note that many information-centric approaches, like those developed as a part of this thesis work, can be deployed as overlays or special purpose networks without requiring completely overhaul, or replacement, of the Internet.

One of the biggest challenges of information-centric networks are in flexibility and the complexity of implementations. Information-centric networks use a similar approach as the Internet with its protocol stack and layering that divides implementation blocks to functionally separate entities and protocols. As we have shown in this thesis information-centric network supports also a new paradigms for this that allows more flexibility than a traditional strict protocol stack layering , e.g. through scopes and helper functions (modules). The main role of such network mechanisms is to improve the network resource utilization and to enhance the end-user experience. Hence, intrinsic support of the envisioned mechanisms within the network core is preferred, instead of deploying those as an overlay implementation or a supplementary tool at the edges nodes.

In this thesis we have explored the space of granular network design by proposing the collection of helper functions as a set of network mechanisms that contribute towards building and optimization of information-centric networks. The main focus of the research community has been on the data plane and basic mechanisms of the information-centric networks. We have selected to focus more on applying such principles directly in the control and management plane of the network. First, this is important since also control and data plane structures are required to deploy operational and efficient publish-subscribe networks. Second, we believe that such approach adds to the flexibility of all the network while diminishing the required deployment costs. In this thesis we have shown that the information-centric network principles can be used in natural and efficient way to establish and implement required control and management plane functionalities. We also investigated the potential problems of employing the helper functions in terms of introduced delay and signaling overhead.

8.1 SUMMARY OF RESULTS

We started this thesis by examining the benefits of the publish-subscribe paradigm in general. Our simulation and analysis demonstrate that publish-subscribe architectures have clear benefits over traditional end-point centric communication. One of our own contributions is to show that this holds also for wireless and dynamic environments. Such findings encouraged our further investigation with the publish-subscribe model.

The first major goal in this work was investigation, and subsequent implementation of a helper function to enable an intra-domain topology manager for publish-subscribe based networks. Such module collects network states related data, and use it as an input for constructing the corresponding topology representations. This work served as a basis for the generation of data delivery paths from the publisher to the subscriber. A set of ancillary mechanisms have been additionally designed in order to improve the core operation of topology management. In the final stage we developed a concept, where the module collects not only link information, but also information on individual application requirements. Such information is fed into the topology manager and subsequent optimization based on this information enables better utilization of network resources while satisfying application needs. The evaluation carried out in the testbed environment demonstrates the high potential of such model to enhance network performance and to bring quality of service (QoS) context into networks in more natural way than in the present day Internet.

Furthermore, we have continued the development of helper functions by addressing the issue of mobility in the context of information-centric networks.

We designed a mobility prediction helper function with the main role of enabling the handover procedure using directly publish-subscribe infrastructure itself for it. We showed that by recording the user locations over time the mobility helper function is able to acquire a good prediction data and find out regularities in the movement patterns of users. Such information can be exploited to guess the most likely future position of nodes, which will enable a faster and more robust handover process.

Our simulation results show that by setting the appropriate network parameters such a mobility prediction function yields significant benefits in overall network performance. Through the lower level helper functions the significant amount of network knowledge can be easily collected. Hence, the decision on the suitable network parameters in order to obtain prediction benefits is facilitated.

One of the key results in the thesis has been the development of an efficient and native network attachment mechanism and procedure for information-centric networks. The developed network attachment procedure exploits the flexible naming scheme of the underlying information-centric network to enable efficient attachment that re-uses topology discovery information. The analysis of the prototype which was tested in the testbed demonstrated excellent performance of the proposed mechanism in terms of introduced delay and message overhead.

Final part of this thesis work considered the utilization of network coding mechanism for enhancing the operation of information-centric network. Although network coding has been studied extensively in the literature it has received relatively little attention in the context of information-centric networking. Our work focused on two types of network coding methods, namely XOR and linear network coding.

We analyzed their performance under different network settings in the intra-domain use to establish the base-line for our analysis for information-centric operations. Our evaluation indicates very good performance of network coding when applied under favorable circumstances. However, we emphasize that network coding gain can be insignificant is some other cases. We discussed an information-centric approach, where network coding can be enabled at-will when favorable conditions exists. Further research is required, however, to understand these issues more carefully, e.g. to understand better when the network coding can provide a significant gain.

Furthermore, we have proposed new models for wide-scale evaluation of building blocks of an information-centric network and the architecture as a whole. In this context we have demonstrated that the integration of simulation and testbed environments is easily achievable in the context of information-centric networking. The developed flexible approach facilitates large-scale experiments for the future research.

8.2 CONTRIBUTIONS IN THE INFORMATION-CENTRIC RESEARCH AREA

As mentioned most of the existing results and prototypes in the information-centric space advocate employing the data-oriented concepts as the overlay or as the mechanism for easier application implementation. Therefore, one of the main questions that this thesis contributes to, is whether a data-oriented communication paradigm can serve as an appropriate model for deployment in the lower layers of the network stack. Our work shows that due to the high level of flexibility of data-oriented communication a rich set of helper functionalities can be provided also in the control and management plane, which can significantly improve the network performance. Such an approach yields a powerful design space beyond traditional overlay techniques. It should be noted that these helper functions can be used either for native information-centric networks, or as helper functions in overlay networks.

Designing a full *implementation* architecture for information-centric networking is beyond the scope of this thesis. Instead, we decided to make a deeper study, and relate prototype implementations, on selected functionalities. Thus we chose an existing architectural framework as our reference architecture for which we have then provided specific and demanding contributions. That said due to our experimentation one of our main contribution is in demonstrating the applicability and the benefits of the publish-subscribe communication pattern in all spheres of network operation, and particularly we have shown the benefits that can be gained at the control and management plane.

Apart from the helper functions proposed within this thesis, other techniques and companion data structures may be developed in order to optimize the network operation. Potential new solutions may encounter different design trade-offs, while utilizing alternative techniques for integration with the information-centric space. We note, although we are not able to prove, that the data-centric protocol concepts seem to be more resilient to the integration problems originated from the design mismatch than a traditional protocol stack approach of the Internet.

We believe, and certainly hope, that our work on publish-subscribe based helper functions becomes subject of not only the further investigation on potential useful helper functions, but that it will also motivate the development and enhancement of similar modular solutions that rely on publish-subscribe mechanisms. The approach we have followed opens a wide avenue for further research in the area of native support of information-centric principles.

8.3 OPEN ISSUES AND FUTURE DIRECTIONS

Apart from the helper functionalities that have been already developed within this thesis a large number of additional mechanisms can be, and should be deployed to make information-centric model more efficient and attractive approach, e.g. network management, policy management, rate adaptation, error control, and transcoding functionalities. We have shortly discussed the main reasons and opportunities in applying such mechanisms. Due to the flexibility of underlying architecture the development of new helper functionalities, and their addition even after the deployment of the networks, appear as straightforward tasks.

More importantly large-scale experiments should be performed to understand the scalability issues that are inevitable in realistic commercial networks. We believe that one can use our proposed evaluation model to get a closer insight on information-centric network performance on the Internet scale. The designed hybrid software-hardware simulation approach could be a springboard towards large-scale experimentation. The network behavior on the interdomain level requires a detailed study in order to provide insights on global deployment and sustainability of information-centric networks.

Considering the broader information-centric perspective, there is still a huge number of open questions accompanied with architectural proposals, ranging from the routing, naming, and addressing issues. Furthermore, the development of suitable business models accompanied with the information-centric networks of the future, as well as its social impact are interesting questions that would require also interdisciplinary research.

A significant portion of work in design, implementation and deployment of information-centric networks is already being carried out within a wide range of dedicated projects and individual research efforts. However, main concerns remain in the development of name-based routing schemes that can easily scale up to the Internet scale. Further concerns include the forwarding mechanisms and transport functions, as well as caching techniques and their relation with congestion and error control. Finally, the security consideration poses one of the main challenges to the information-centric network design. Our future work will be strongly focused on detailed analysis of such problems and developing of mechanisms for their resolution.

We believe that some of the aforementioned issues can be solved or diminished by utilizing the various helper functionalities on different levels of protocol stack. The exact implementation of such helper functions requires deep consideration of the issue and careful analysis of possible solution.

8.4 FINAL REMARKS

The evolution of the Internet is a highly interesting area of networking research and an important topic in general. The Internet has became a prominent part of our everyday lives, playing a crucial role in our society. Infrastructure provided by the Internet today represent the main anchor of global business and the communication. Therefore, one of the biggest requirements that the Internet needs to satisfy is reliability. Despite the numerous claims that the Internet operates reasonably well several issues threaten the future prosperity of the Internet.

The work on the remedies for flaws of the Internet is an ongoing struggle and information-centric networking is only one of the possible solutions. The process towards the new improved architecture for the Internet will take a long time. However, not just that the intellectual and practical challenges of this sort of research are formidable and very interesting, but potentially, of course, a long term impact of this research can be huge.

The work conducted within the scope of this thesis represents a small part of this greater effort, and we have focused on understanding some of the more specific problems in details and developing the mechanisms and proof-of-concept prototypes for architecture improvements. As such, although these results represent incremental steps towards the future architecture solutions, we would dare to argue that those steps are both necessary and useful.

A

NOTATIONS

In the following we list the notations used in this thesis in an alphabetical order explaining shortly each variable. Notations used in figures to fix the ideas with examples are not a source of ambiguity for the reader and are therefore not listed. We list capital letters first.

Table A.1: Notation in capital Latin letters used throughout this thesis.

Symbol	Meaning
D_i	Data blocks
F	Finite field
G_i	Multicast group
H_i	Hash functions
K_{rs}	Number of information symbols in block coding codeword
M^i	Packets in the linear network coding combination
N_{rs}	Number of symbols in block coding codeword
P	Number of packets in network coding combination
P_i	Publisher in the set of publishers
P_{sig}	Signal power level of the mobile node received by the current base station
PUB	Set of publishers
R	Radius of the coverage area of a base station
R_i	Rendezvous nodes in the set of rendezvous entities
Rcv	Receiver from the multicast group
RZV	Set of rendezvous entities
S_i	Subscriber in the set of subscribers
Src_1	Sender to the multicast group
Src_2	Sender to the multicast group
SUB	Set of subscribers
T	Number of hash functions for LIT creation
T_{hand}	Handover threshold
T_{pred}	Mobility prediction triggering threshold
X	Linear combination in linear network coding

Table A.2: Notation in small Latin letters used throughout this thesis.

Symbol	Meaning
d	Distance between a base station and a mobile station
g_i	Coefficients of a finite field
h_i	Coefficients of a finite field
k_i	Coefficients of a finite field
k'	Size of a bit vector
m	Number of equations to solve linear network coding combination
m'	Number of hash functions
m_{rs}	Number of bits in symbols of block coding codeword
n	Network coding packet combination size
n'	Number of data blocks
p_i	Packets present at a router
x_B	x coordinate of a base station
x_M	x coordinate of a mobile station
y_B	y coordinate of a base station
y_M	y coordinate of a mobile station

B

ABBREVIATIONS

In the following we list the acronyms used throughout this thesis in the alphabetic order. We always list capital letters first.

3G Third Generation Mobile Telecommunications

AKA Authentication and Key Agreement Protocol

ARPA Advanced Research Projects Agency

B One byte, 8 bits

BF Bloom Filter

BER Bit Error Rate

CBR Constant Bit Rate

CCN Content-Centric Networking

CDN Content Delivery Network

CHAP Challenge Handshake Authentication Protocol

CIDR Classless Inter Domain Routing

DARPA Defense Advanced Research Projects Agency

DNS Domain Name System

DONA Data-Oriented Network Architecture

EAP Extensive Authentication Protocol

EPC-CBR End-Point-Centric Constant Bit Rate

FEC Forward Error Correction

FId Forwarding Identifier

Gbps Gigabit per second

HTTP Hypertext Transfer Protocol

I3 Internet Indirection Infrastructure

ICMP Internet Control Message Protocol

ICN Information-Centric Network

ID Identifier

IP Internet Protocol

IPsec Internet Protocol Security

IPv4 Internet Protocol version 4

IPv6 Internet Protocol version 6

Kbps Kilobit per second

LAN Local Area Network

LId Link Identifier

LIT Link Identifier Tag

LSA Link State Advertisment

Mbps Megabit per second

MId Metadata Identifier

MPEG-2 Moving Picture Experts Group Standard

MPEG-4 Moving Picture Experts Group Standard

MPLS Multiprotocol Label Switching

ms milliseconds

NAT Network Address Translation

NDN Named data Networking

OFDM Orthogonal Frequency Division Multiplexing

OSPF Open Shortest Path First

P2P Peer-to-Peer

PSIRP Publish-Subscribe Internet Routing Paradigm

PURSUIT Publish-Subscribe Internet Technology

RH Resolution Handlers

RId Rendezvous Identifier

ROFL Routing Over Flat Labels

RS Reed Solomon

RSS Really Simple Syndication

SId Scope Identifier

SNMP Simple Network Management Protocol

TCP Transport Control Protocol

TLS Transport Layer Security

UMTS Universal Mobile Telecommunications System

URL Uniform Resource Locator

VPN Virtual Private Network

WSN Wireless Sensor Network

WWW World Wide Web

XOR Exclusive OR operation

Bibliography

[1] "Miniwatts Marketing Group, Internet World Stats: Internet Usage Statistics. Online document." http://wwww.internetworldstats.com/stats.htm/, visited on July 2012.

[2] "The 32-bit AS number report. Online document." http://www.potaroo.net/tools/asn32/, visited on July 2012.

[3] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the internet impasse through virtualization," in *Computer*, vol. 38, no. 4, April 2005, pp. 34–41.

[4] J. H. Saltzer, D. P. Reed, and D. Clark, "End-to-end arguments in system design," in *ACM Transactions on Computer Systems (TOCS)*, vol. 2, no. 4, November 1984, pp. 277–288.

[5] J. Mirkovic, and P. Reiher, "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms," in *Sigcomm Computer Communication Review*, vol. 34, no. 2, 2004, pp. 39–53.

[6] D. Le, X. Fu, and D. Hogrefe, "A review of mobility support paradigms for the Internet," in *IEEE Communications Surveys and Tutorials*, vol. 8, no. 1, 2006, pp. 38–51.

[7] V. Jacobson, M. Mosko, D. Smetters, and J. J.Garcia-Luna-Aceves, "Content-centric networking," in *Whitepaper describing future assurable global networks. Response to DARPA RFI SN07-12*, 2007.

[8] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking Named Content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies,ACM CoNEXT*, Rome, Italy, December 2009.

[9] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker and I. Stoica, "A Data-Oriented (and Beyond) Network Architecture," in *Proceedings of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication SIGCOMM 2007*, New York, NY, USA, 2007, pp. 181–192.

[10] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, I. Stoica, and S.Shenker, "ROFL: Routing on Flat Labels," in *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, October 2006, pp. 363–374.

[11] I. Stoica, D. Adkins, S. Zhuang, S. Shenker and S. Surana, "Internet indirection infrastructure," in *Proceedings of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication SIGCOMM 2002*, vol. 32, October 2002, pp. 73–86.

[12] M. Gritter and D. R. Cheriton, "An Architecture for Content Routing Support in the Internet," in *Proceedings of the 3rd conference on USENIX Symposium on Internet Technologies and Systems*, Berkeley, CA, USA, 2001, pp. 37–48.

[13] H. Shen, *Content-Based Publish/Subscribe Systems*. Handbook of Peer-to-Peer Networking, Springer Science+Business Media, LLC, 2010.

[14] K. P. Birman, "The process group approach to reliable distributed computing," in *Communications of the ACM*, vol. 12, no. 36, December 1993, pp. 36–53.

[15] G. Fitzpatrick, T. Mansfield, S. Kaplan, D. Arnold, T. Phelps, and B. Segall, "Augmenting the workaday world with Elvin," in *Proceedings of the of the Sixth European Conference on Computer Support Cooperative Work (ECSCW-99)*, Netherlands, September 1999, pp. 431–450.

[16] G. Banavar, T. Chandra, B. Mukherjee, J. Nagarajarao, R. E. Strom, and D. C. Sturman, "An eficient multicast protocol for content-based publish-subscribe systems," in *Proceedings of the 19th IEEE International Conference on Distributed Computing Systems*, 1999, pp. 262–272.

[17] G. Bricconi, E. D. Nitto, A. Fuggetta, and E. Tracanella, "Analyzing the behavior of event dispatching systems through simulation," in *Proceedings of the 7th IEEE International Conference on High Performance Computing*, 2000, pp. 131–140.

[18] P. Pietzuch and J. Bacon, "A distributed event-based middleware architecture," in *Proceedings of the 1st International Workshop on Distributed Event-Based Systems (DEBS'02)*, Vienna, Austria, July 2002, pp. 611–618.

[19] L. Fiege, M. Mezini, G. Mühl, and A. P. Buchmann, "Engineering event-based systems with scopes," in *Proceedings of the European Conference on Object-Oriented Programming (ECOOP)*, vol. 2374, Malaga, Spain, June 2002, pp. 309–333.

[20] R. H. Zakon, "Hobbes' Internet Timeline 10.2," http://www.zakon.org/robert/internet/timeline/, visited on June 2012.

[21] J. B. Postel, C. A. Sunshine, and D. Cohen, "The ARPA internet protocol," in *Computer Networks*, vol. 5, no. 4, July 1981, pp. 261–271.

[22] D. Clark, "The design philosophy of the DARPA Internet protocols," in *ACM SIGCOMM Computer Communication Review, Symposium proceedings on Communications architectures and protocols SIGCOMM 1988*, vol. 18, no. 4, August 1988, pp. 106–114.

[23] "Cisco Visual Networking Index: Forecast and Methodology, 20112016. Online document." http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf, visited on July 2012.

[24] W. R. Stevens, and G. R. Wright, *TCP/IP Illustrated: Vol. 1: The Protocols*. Addison-Wesley Professional, 1994.

[25] J. Postel, "Internet Protocol, DARPA Internet program protocol specification," RFC 791, September 1981.

[26] J. Mogul, "Internet subnets," RFC 917, October 1984.

[27] J. Mogul, and J. Postel, "Internet Standard Subnetting Procedure," RFC 950, August 1985.

[28] V. Fuller and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan," RFC 4632, August 2006.

[29] P. Mockapetris, "Domain names - concepts and facilities," RFC 1034, November 1987.

[30] E. Rosen, A. Viswanathan and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031, January 2001.

[31] P. Srisuresh and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations," RFC 2663, August 1999.

[32] R. Hinden, and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture," RFC 3513, April 2003.

[33] Y. Rekhter, T. Li and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271, January 2006.

[34] D. Wallner, E. Harder and R. Agee, "Key Management for Multicast: Issues and Architectures," RFC 2627, June 1999.

[35] S.E. Deering, and D.R. Cheriton, "Multicast routing in datagram internetworks and extended LANs," in *ACM Transactions on Computer Systems (TOCS)*, vol. 8, no. 2, 1990, pp. 85–111.

[36] C. Perkins, "IP Mobility Support for IPv4," RFC 3344, August 2002.

[37] B. Gleeson, et al., "A Framework for IP Based Virtual Private Networks," RFC 2764, February 2000.

[38] S. Kent and K. Seo, "Security Architecture for the Internet Protocol," RFC 4301, December 2005.

[39] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246, August 2008.

[40] N. Bartolini, E. Casalicchio and S. Tucci, "A Walk through Content Delivery Networks," in *MASCOTS Tutorials*, 2003, pp. 1–25.

[41] B. Cohen, "Incentives build robustness in BitTorrent," in *Workshop on the Economics of Peer-to-Peer Systems*, Berkeley, CA, USA, June 2003, pp. 116–121.

[42] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications," in *Proceedings of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication SIGCOMM 2001*, New York, USA, October 2001, pp. 149–160.

[43] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. Kubiatowicz, "Tapestry: A Resilient Global-scale Overlay for Service Deployment," in *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, January 2004, pp. 41–53.

[44] A. Rowstron, and P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," in *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, London, UK, 2001, pp. 329–350.

[45] "Global Environment for Network Innovations Project," http://www.geni.net/, visited on December 2012.

[46] "Future Internet Research and Experimentation Project," http://cordis.europa.eu/fp7/ict/fire/, visited on December 2012.

[47] J. Moy, "OSPF Version 2," RFC 2328, 1998.

[48] "SNMP Protocol Overview," RFC 1157.,RFC 1902., RFC 1903., RFC 1904., RFC 1905., RFC 1906., RFC 1907, 1996.

[49] A. Conta, S. Deering and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification," RFC 4443, March 2006.

[50] B. Aboba, D. Simon and P. Eronen, "Extensible Authentication Protocol (EAP) Key Management Framework," RFC 5247, August 2008.

[51] T. Jeffree et al, "IEEE Standard 802.1X-2004: Port-based network access control," http://www.ieee802.org/1/pages/802.1x-2004.html, visited on December 2012.

[52] A. Niemi, J. Arkko, V. Torvinen, "Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)," RFC 3310, September 2002.

[53] W.Simpson, "The point-to-point protocol (PPP)," RFC 1661, 1994.

[54] W. Simpson, "PPP Challenge Handshake Authentication Protocol (CHAP)," RFC 1996, 1994.

[55] J. Arkko, P. Eronen, and H. Tschofenig, "Quick NAP - secure and efficient network access protocol," in *Proceedings of the 6th International Workshop on Applications and Services in Wireless Networks (ASWN 2006)*, Berlin, Germany, May 2006, pp. 163–170.

[56] "Akamai content delivery network," http://www.akamai.com/, visited on August 2012.

[57] "The eMule Protocol Specification," Technical Report TR-2005-03, January 2000.

[58] A. K. Pathan and R. Buyya, "A taxonomy and survey of content delivery networks," Technical report, GRIDS-TR-2007-4, The University of Melbourne, Australia, 2007.

[59] "Limelight content delivery network," http://www.limelight.com/, visited on May 2012.

[60] "Velocix content delivery network," http://www.velocix.com/, visited on May 2012.

[61] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," in *Computer Networks*, vol. 38, no. 4, 2002, pp. 393–422.

[62] P. Hebden and A. Pearce, "Data-centric routing using bloom filters in wireless sensor networks," in *Fourth International Conference on Intelligent Sensing and Information Processing (ICISIP-06)*, Bangalore, India, 2006, pp. 72–78.

[63] D. Atkins, and R. Austein, "Threat Analysis of the Domain Name System (DNS)," RFC 3833, August 2004.

[64] C. L. Schuba, I. V. Krsul, M. G. Kuhn, E. H. Spafford, A. Sundaram, and D. Zambodi, "Analysis of a Denial of Service Attack on TCP," in *In Proceedings of the 1997 IEEE Symposium on Security and Privacy*, Oakland, USA, May 1997, pp. 208–223.

[65] P. Syverson, "A Taxonomy of replay attacks," in *In Proceedings of the 7th IEEE Computer Security Foundation Workshop (CSFW 94)*, Franconia, USA, June 1994, pp. 187–191.

[66] P. Watson, "Slipping in the Window: TCP Reset Attacks," in *In Proceedings of CanSecWest (white paper)*, 2004.

[67] V. Ramasubramanian, E. G. Sirer, "The design and implementation of a next generation name service for the Internet," in *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, October 2004, pp. 331–342.

[68] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S.Shenker, "Holding the Internet Accountable," in *Proceedings of the 6th ACM Workshop on Hot Topics in Networks (Hotnets-VI)*, November 2007, pp. 7–12.

[69] D. Clark, B. Lehr, S. Bauer, P. Faratin, R. Sami, and J. Wroclawski, "Overlay Networks and the Future of the Internet," in *Communication and Strategies*, vol. 63, 2006, pp. 109–129.

[70] A. Feldmann, "Internet clean-slate design: what and why?" in *SIGCOMM Computer Communications Review*, vol. 37, no. 3, 2007, pp. 59–64.

[71] R. Guerin, "Does the Networked Generation Need a Next Generation Network?" Keynote presentation at the 5th Euro-NGI conference on Next Generation Internet networks (NGI), Aveiro, Portugal, July 2009, available at: http://www.seas.upenn.edu/ guerin/presentations/NGI-2009.pdf.

[72] K. Lai, M. Feldman, I. Stoica, and J. Chuang, "Incentives for Cooperation in Peer-to-Peer Networks," in *Proceedings of Workshop on Economics of Peer-to-Peer Systems*, Berkley, USA, June 2003.

[73] P. Golle, K. L. Brown, I. Mironov, and M. Lillibridge, "Incentives for Sharing in Peer-to-Peer Networks," in *Proceedings of the 2nd International Workshop on Electronic Commerce*, Heidelberg, Germany, 2001.

[74] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications," in *Proceedings of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication SIGCOMM 2001*, San Diego, USA, 2001, pp. 149–160.

[75] "Named Data Networking (NDN) Project," http://www.named-data.net/, visited on May 2012.

[76] "Content-Centric Networking (CCNx) Project," http://www.ccnx.org/, visited on August 2012.

[77] "EU FP7 PURSUIT project," http://www.fp7-pursuit.eu/, visited on September 2012.

[78] "PSIRP-Publish Subscribe Internet Routing Paradigm," http://www.psirp.org/ , visited on May 2012.

[79] D. Trossen, M. Särrelä, K. Sollins, "Arguments for an Information-Centric Internetworking Architecture," in *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 2, April 2010, pp. 26–33.

[80] A. Zahemszky, B. Gajic, C. Rothenberg et al., "Experimentally-driven research in Publish/Subscribe Information-centric Internetworking," in *Proceedings of Tridentcom 2010*, Berlin, Germany, 2010.

[81] S. Arianfar, P. Nikander, and J. Ott, "Packet-level caching for information-centric networking," in *Finnish ICT-SHOK Future Internet Project, Technical Report*, 2010.

[82] P.Th. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The Many Faces of Publish/Subscribe," in *ACM Computing Surveys*, vol. 35, June 2003, pp. 114–131.

[83] P. Gore, R. Cytron, D. Schmidt, and C. O'Ryan, "Designing and Optimizing a Scalable CORBA Notification Service," in *Proceedings of the 2001 ACM SIGPLAN workshop on Optimization of middleware and distributed systems*. New York, NY, USA: ACM, 2001, pp. 196–204.

[84] G. Krishnamurthy Att, R. E. Gruber, B. Krishnamurthy, and E. Panagos, "CORBA Notification Service: Design Challenges and Scalable Solutions," in *In 17th International Conference on Data Engineering*, 2001, pp. 13–20.

[85] S. Q. Zhuang, B. Y. Zhao, A. D. Joseph, R. Katz, and J. D. Kubiatowicz, "Bayeux: an architecture for scalable and fault-tolerant wide-area data dissemination," in *Proceedings of the 11th international workshop on Network and operating systems support for digital audio and video*. New York, NY, USA: ACM, 2001, pp. 11–20.

[86] R. E. Strom, G. Banavar, T. D. Chandra, M. Kaplan, K. Miller, B. Mukherjee, D. C. Sturman and M. Ward, "Gryphon: An Information Flow Based Approach to Message Brokering," in *Proceedings of the International Symposium on Software Reliability Engineering*, 1998.

[87] G. Cugola, E. Di Nitto, and A. Fuggetta, "Exploiting an event-based infrastructure to develop complex distributed systems," in *Proceedings of the 10th International Conference on Software Engineering (ICSE 98)*, April 1998, pp. 261–270.

[88] ——, "The JEDI event-based infrastructure and its application to the development of the OPSS WFMS," in *IEEE Transactions on Software Engineering*, vol. 27, no. 9, September 2001, pp. 827–850.

[89] R. E. Gruber, B. Krishnamurthy, and E. Panagosf, "The architecture of the READY event notification service," in *Proceedings of The 19th IEEE International Conference on Distributed Computing Systems, Workshop on Electronic Commerce and Web-based Applications/Middleware*, Austin, Texas, USA, 1999, pp. 108–113.

[90] P. Pietzuch and J. Bacon, "Hermes: a distributed event-based middleware architecture," in *Proceedings of the 22nd International Workshop on Distributed Event-Based Systems (DEBS02)*, Vienna, Austria, July 2002, pp. 611–618.

[91] P.Th. Eugster, R. Guerraoui, and Ch.H. Damm, "On objects and events," in *Proceedings of the 16th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications (OOPSLA '01*, vol. 36, no. 11. New York, NY, USA: ACM, 2001, pp. 254–269.

[92] K. Römer, O. Kasten, and F. Mattern, "Middleware challenges for wireless sensor networks," in *ACM Mobile Computing and Communication Review*, vol. 6, no. 4, 2002, pp. 59–61.

[93] K. Henricksen, R. Robinson, "A survey of middleware for sensor networks: state-of-the-art and future directions," in *Proceedings of the international workshop on Middleware for sensor networks, MidSens '06*, Melbourne, Australia, November 2006, pp. 60–65.

[94] E. Souto, G. Guimaraes, G. Vasconcelos, M. Vieira, N. Rosa, C. Ferraz, "A Message-Oriented Middleware for Sensor Networks," in *Proceedings of the 2nd workshop on Middleware for pervasive and ad-hoc computing*, Toronto, Ontario, Canada, October 2004.

[95] U. Farooq, E. W. Parsons, and S. Majumdar, "Performance of Publish/Subscribe Middleware in Mobile Wireless Networks," in *ACM SIGSOFT Software Engineering Notes*, vol. 29, no. 1, January 2004.

[96] S.E. Deering, "Multicast routing in a datagram internetwork," in *Technical Report STAN-CS-92-1415, Stanford University*, 1991.

[97] S.E. Deering and D.R. Cheriton, "Multicast routing in datagram internetworks and extended LANs," in *ACM Transactions on Computer Systems (TOCS)*, vol. 8, no. 2, 1990, pp. 85–110.

[98] C. Diot, W. Dabbous, and J. Crowcroft, "Multipoint Communication: A Survey of Protocols, Functions, and Mechanisms," in *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, 1997, pp. 277–290.

[99] B. Fenner, H. He, B. Haberman, H. Sandick, "Internet Group Management Protocol (IGMP)/Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")," RFC 4605, August 2006.

[100] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM)," RFC 4601, August 2006.

[101] N. Bhaskar, A. Gall, J. Lingard, and S. Venaas, "Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)," RFC 5059, January 2008.

[102] D. Kim, D. Meyer, H. Kilmer, and D. Farinacci, "Anycast Rendevous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)," RFC 3446, January 2003.

[103] S. Bhattacharyya, "An Overview of Source-Specific Multicast (SSM)," RFC 3569, July 2003.

[104] I. Abraham, D. Malkhi, and D. Ratajczak, "Compact multicast routing," in *In DISC09: Proceedings of the 23rd international conference on Distributed computing*. Berlin, Heidelberg, Germany: Springer-Verlag, September 2009, pp. 364–378.

[105] R. Boivie, N. Feldman, Y. Imai, W. Livens, and D. Ooms, "Explicit Multicast (Xcast) Concepts and Options," RFC 5058, November 2007.

[106] D.N. Yang, and W. Liao, "Protocol Design for Scalable and Adaptive Multicast for Group Communications," in *IEEE International Conference on Network Protocols ICNP*, November 2008, pp. 33–42.

[107] B. Whetten, T . Montgomery, and S. Kaplan, "A High Performance Totally Ordered Multicast Protocol, Theory and Practice in Distributed Systems," in *Lecture Notes in Computer Science*, vol. 938, 1995, pp. 33–57.

[108] H.W. Holbrook, S.K. Singhal, and D.R. Cheriton, " Log-Based Receiver-Reliable Multicast for Distributed Interactive Simulation," in *Proceedings of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication SIGCOMM 1995*, vol. 25, no. 4, October 1995, pp. 328–341.

[109] S. Floyd, V. Jacobson, S. McCanne, C. G. Liu, and L. Zhang, "Multicast Framework for Light-Weight Sessions and Application Level Framing," in *Proceedings of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, SIGCOMM 1995*, vol. 25, no. 4, October 1995, pp. 342–356.

[110] C. Diot, B.N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the ip multicast service," in *IEEE Network Magazine, special issue on Multicasting*, vol. 14, no. 1, February 2000, pp. 78–88.

[111] "Qualnet simulator," http://www.scalable-networks.com/, visited on August 2012.

[112] Institute of Electrical and Electronics Engineers, "IEEE Standard for Local and Metropolitan Area NetworksPart 16: Air Interface for Fixed and Mobile Broadband Wireless Systems Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands, and Corrigendum 1, IEEE 802.16e-2005 and 802.16/COR1," February 2006.

[113] B. Zieba, M. van Sinderen, and M. Wegdam, "Quality-constrained routing in publish/subscribe systems," in *In Proceeding of the 3rd international workshop on Middleware for pervasive and ad-hoc computing*, Grenoble, France, November 2005.

[114] F. Araujo, and L. Rodrigues, "Quality of Service in Indirect Communication Systems," in *Fourth European Research Seminar on Advances in Distributed Systems (ERSADS)*, Bertinoro, Italy, May 2001.

[115] E. W Dijkstra , "A Note on Two Problems in Connection with Graphs," in *Numerische Mathematik 1*, 1959, pp. 269–271.

[116] G. Csardi and T. Nepusz, "The igraph software package for complex network research," in *InterJournal*, vol. Complex Systems, 2006, p. 1695. [Online]. Available: http://igraph.sf.net

[117] P. Jokela, A. Zahemszky, C. Esteve, S. Arianfar and P. Nikander, "LISPIN: Line Speed Publish/Subscribe Inter-Networking," in *Proceedings of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication SIGCOMM 2009*, Barcelona, Spain, 2009.

[118] B. H. Bloom, "Space-time Trade-offs in Hash Coding with Allowable Errors," in *Communications of the ACM*, vol. 13, no. 7, July 1970, pp. 422–426.

[119] D. Eastlake and P. Jones, "Secure Hash Algorithm 1 (SHA1)," IETF RFC 3174, visited on August 2012. [Online]. Available: http://www.faqs.org/rfcs/rfc3174.html

[120] C. E. Rothenberg, C. A. B. Macapuna, F. L. Verdi, and M. F. Magathaes, "The Deletable Bloom Filter: a New Member of the Bloom Family," in *IEEE Communication Letters*, vol. 14, no. 6, May 2010, pp. 557–559.

[121] Y. Lu, B. Prabhakar, and F. Bonomi, "Bloom Filters: Design Innovations and Novel Applications," in *Proceedings of the Forty-Third Annual Allerton Conference*, Monticello, Illinois, USA, September 2005.

[122] F. Chang, W. Feng, and K. Li, "Approximate Caches for Packet Classification," in *Proc. of the International Conference on Computer and Communications (INFOCOM)*, Hongkong, China, March 2004, pp. 2196–2207.

[123] M. Yoon, "Aging Bloom Filter with Two Active Buffers for Dynamic Sets," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 1, January 2010, pp. 134–138.

[124] "Blackhawk: A Publish/Subscribe System for FreeBSD," http://wiki.hiit.fi/display/psirpcode/Home, visited on August 2012.

[125] B. Gajic, J. Riihijärvi, and P. Mähönen, "Intra-Domain Topology Manager for Publish-Subscribe Networks," in *18th International Conference on Telecommunications ICT 2011*, Ayia Napa, Cyprus, May 2011.

[126] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," RFC 3775, June 2004.

[127] H. Soliman, C. Castelluccia, K. E. Malki, and L. Bellie, "Hierarchical Mobile IPv6 Mobility Management (HMIPv6)," RFC 5380, October 2008.

[128] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, "Host Identity Protocol (HIP)," RFC 5201, June 2006.

[129] P. Nikander, T. Henderson, C. Vogt, and J. Arkko, "End-Host Mobility and Multihoming with the Host Identity Protocol," RFC 5206, April 2008.

[130] A. T. Campbell, J. Gomez, and A. G. Valko, "An Overview of Cellular IP," in *IEEE Wireless Communications and Networking Conference*, 1999, pp. 606–610.

[131] P. Reinbold, and O. Bonaventure, "IP micro-mobility protocols," in *IEEE Communications Surveys and Tutorials*, vol. 5, 2003, pp. 40–57.

[132] V. Muthusamy, M. Petrovic, D. Gao, and H. Jacobsen, "Publisher mobility in distributed publish/subscribe systems," in *In 25th IEEE International Conference on Distributed Computing Systems - Workshops*, 2005, pp. 421–427.

[133] R. Mardeni, T. Siva Priya, "Mobility support through caching in content-based publish/subscribe networks," in *International Conference on Computer Communication and Management CSIT 2011*, Singapure, 2011.

[134] P. Costa, C. Mascolo, M. Musolesi, and G.P. Picco, "Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks," in *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, June 2008, pp. 748–760.

[135] T. R. Henderson, M. Lacage, G. F. Riley, "Network Simulations with the ns-3 Simulator," in *Proceedings of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication SIGCOMM 2008*, Seattle,Washington,USA, August 2008, Demo abstract.

[136] G. F. Riley, and T. R. Henderson, "The ns-3 Network Simulator," in *Computer Science-Modeling and Tools for Network Simulation*, vol. 2, no. 10, 2010, pp. 15–34.

[137] T. Issariyakul, and E. Hossain, *Introduction to Network Simulator NS2*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[138] T. Nagy, "The WAF Book," visited on August 2012. [Online]. Available: http://docs.waf.googlecode.com/git/book_16/single.html

[139] G. Carneiro, P. Fortuna, and M. Ricardo, "Flowmonitor: a network monitoring framework for the network simulator 3 (ns-3)," in *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, ser. VALUETOOLS '09. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, pp. 1–10.

[140] V. Sourlas, G. S. Paschos, P. Flegkas, and L. Tassiulas, "Performance of Path Loss Model in 4G Wimax Wireless Communication System in 2390 MHz," in *In 5th International Workshop on Content Delivery Networks (CDN 2010) in conjuction with 10th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2010)*, Melbourne, Australia, May 2010.

[141] J. Rakesh, W. Vishal, and U. Dalal, "A Survey if Mobile WiMAX IEEE 802.16m Standard," in *International Journal of Computer Science and Information Security*, vol. 8, 2010.

[142] J. Farooq and T. Turletti, "An IEEE 802.16 WiMAX module for the NS-3 simulator," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques, Simutools '09*, no. 8, Rome, Italy, 2009, pp. 1–8.

[143] "The prototype implementation of PURSUIT architecture," https://github.com/georgeparisis/blackadder, visited on September 2012.

[144] "The Click Modular Router Project," http://read.cs.ucla.edu/click/, visited on August 2012.

[145] B. Gajic, J. Riihijärvi and P. Mähönen, "On Flexible Topology Formation in Publish-Subscribe Networks," in *International Conference on Communications (IEEE ICC), International Workshop on Network of the Future*, Otawa, Canada, June 2012.

[146] W.K. Edwards, "Discovery systems in ubiquitous computing," in *IEEE Pervasive Computing*, vol. 5, no. 2, April 2006, pp. 70–77.

[147] F. Zhu, M. W. Mutka, and L. M. Ni, "Service Discovery in pervasive computing environments," in *IEEE Pervasive Computing*, vol. 4, no. 4, October 2005, pp. 81–90.

[148] K. Vanthournout, G. Deconinck, and R. Belmans, "A taxonomy for resource discovery," in *Personal Ubiquitous Computing*, vol. 9, no. 2, March 2005, pp. 81–90.

[149] E. Meshkova, J. Riihijärvi, M. Petrova, and Petri Mähönen, "A taxonomy for resource discovery," in *The International Journal of Computer and Telecommunications Networking*, vol. 52, no. 11, August 2008.

[150] E. Guttman, C. Perkins, J. Veizades, and M. Day, "Service Location Protocol, Version 2," RFC 2165, June 1997.

[151] J. Waldo, "The Jini architecture for network-centric computing," in *Communications of the ACM*, vol. 42, no. 7, July 1999, pp. 76–82.

[152] K. Arnold, R. Scheifler, J. Waldo, B. O'Sullivan, and A. Wollrath, *Jini Specification*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.

[153] B. Miller, T. Nixon, C. Tai, and M. Wood, "Home Networking with Universal Plug and Play," in *IEEE Communications Magazine*, vol. 39, no. 12, December 2001, pp. 104–109.

[154] P. Bhagwat, "Bluetooth: technology for short range wireless applications," in *IEEE Internet Computing*, vol. 5, no. 3, May 2001, pp. 96–103.

[155] H. Unger, and M. Wulff, "Cluster building in p2p-community networks," in *Proceedings of the 14th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS)*, Cambridge, USA, November 2002, pp. 685–690.

[156] "The Abilene Research Network," http://www.cs.utexas.edu/~yzhang/research/AbileneTM, visited on June 2012.

[157] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, J. Crowcroft, "XOR in The Air: Practical Wireless Network Coding," in *IEEE/ACM Transactions on Networking*, vol. 16, June 2008, pp. 497–510.

[158] S. Katti, D. Katabi, W. Hu, H. Rahul, M. Médard, "The Importance of Being Opportunistic: Practical Network Coding for Wireless Environments," in *Proceedings of 43rd Annual Allerton Conference on Communication, Control, and Computing*, 2005.

[159] S. K. T. H. D. K. S. Jaggi, M. Langberg and M. Médard, "Resilient network coding in the presence of byzantine adversaries," in *26th IEEE International Conference on Computer Communications INFOCOM 2007*, Anchorage, Alaska, USA, May 2007, pp. 616–624.

[160] W. H. Q. Dong, J. Wu and J. Crowcroft, "Practical network coding in wireless networks," in *Proceedings of the 13th Annual International Conference on Mobile Computing and Networking, MOBICOM 2007*, Montréal, Québec, Canada, September 2007, pp. 306–309.

[161] J. C. G. O. W. Hu, M. Castro and A. I. T. Rowstron, "Network coding with traffic engineering," in *Proceedings of the 2006 ACM Conference on Emerging Network Experiment and Technology, CoNEXT 2006*, Lisboa, Portugal, December 2006, p. 40.

[162] R. Ahlswede, N. Cai, S.-Y.R. Li and R.W. Yeung, "Network Information Flow," in *IEEE-IT*, vol. 46, no. 4, July 2000, pp. 1204–1216.

[163] P. A. Chou, Y. Wu, K. Jain, "Practical Network Coding," in *Proceedings of 41st Annual Allerton Conference on Communication, Control, and Computing*, October 2003.

[164] C. Fragouli, J. Le Boudec, J. Widmer, "Network Coding: An Instant Primer," in *ACM SIGCOMM Computer Communication Review*, vol. 36, January 2006, pp. 63–68.

[165] R. Braden, D. Borman, C. Partridge, "Computing the Internet Checksum," RFC 1071, September 1988. [Online]. Available: http://www.ietf.org/rfc/rfc1071.txt

[166] R. Sherwood, A. Bender, N. Spring, "Discarte: a disjunctive Internet cartographer," in *ACM SIGCOMM Computer Communication Review*, vol. 38, October 2008, pp. 303–314.

[167] I. S. Reed, and G. Solomon, "Polynomial Codes over Certain Finite Fields," in *SIAM Jurnal of Applied Mathematics*, vol. 8, no. 2, 1960, pp. 300–304.

[168] K. Cheun, and W. E. Stark, “Optimal Selection of Reed-Solomon Code Rate and the Number of Frequency Slots in Asynchronous FHSS-MA Networks,” in *IEEE Transaction on Communications*, vol. 41, no. 2, February 1993, pp. 307–311.

[169] S. B. Wicker, V. K. Bhargava, *Reed-Solomon Codes and their Applications*. John Wiley and Sons, 1999.

[170] M. Mitzenmacher, “Digital Fountains: Survey and Look Forward,” in *IEEE Information Theory Workshop*, October 2004, pp. 271–276.

[171] J. W. Byers, M. Luby, M. Mitzenmacher, A. Rege, “A Digital Fountain Approach to Reliable Distribution of Bulk Data,” in *ACM SIGCOMM Computer Communication Review*, vol. 48, no. 4, October 1998, pp. 56–67.

[172] B. G. Haskell, A. Puri, A. N. Netravali, “Digital Video: An Introduction to Mpeg-2,” in *Digital Multimedia Standards Series*, USA, 2003.

[173] I. E. G. Richardson, “H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia,” in *John Wiley and Sons, Ltd*, 2003.

[174] D. Brinthaupt, L. Letham, V. Maheshwari, J. Othmer, R. Spiwak, B. Edwards, C. Terman, N. Weste, “A video decoder for H.261 video teleconferencing and MPEG stored interactive video applications,” in *IEEE International Solid-State Circuits Conference, ISSCC, Digest of Technical Papers*, February 1993, pp. 34–35.

[175] P. L. Suresh, and R. Merz, “ns-3-click: click modular router integration for ns-3,” in *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, ser. SIMUTools ’11, Barcelona, Spain, 2011, pp. 423–430.

CURRICULUM VITAE

Personal Information

Name	Borislava Gajic
Date of birth	06 December 1983
Place of birth	Brcko, Bosnia and Herzegovina
Nationality	Serbian

Education

2008 – 2013	Dr.-Ing studies **RWTH Aachen University**	Germany
2002 – 2007	Dipl.-Ing in Telecommunications **University of Novi Sad, Faculty of Technical Sciences**	Serbia

Experience

04.2008 – 01.2014	Research assistant **iNETS, RWTH Aachen University**	Germany
10.2007 – 04.2008	Network administrator **Elektroprivreda, state power supply company**	Bosnia and Herzegovina
03.2007 – 07.2007	Trainee, research in Wireless Sensor Networks area **SAP Labs, Research Department**	France

Languages

Serbian	mother tongue
English	fluent
Italian	fluent
German	proficient

LIST OF PUBLICATIONS

JOURNAL PAPERS RELEVANT TO THE THESIS TOPIC:

B. Gajic, J. Riihijärvi, and P. Mähönen, "Performance Evaluation of Network Coding: Effects of Topology and Network Traffic for Linear and XOR Coding". *Journal of Communications, vol. 4, No. 11, pp. 885-893, November 2009.*

CONFERENCE PAPERS RELEVANT TO THE THESIS TOPIC:

B. Gajic, J. Riihijärvi, and P. Mähönen, "Performance Evaluation of Network Coding in Middle-Sized Networks", in *Proc. of the International Wireless Communications and Networking Conference (IEEE WCNC)*. Sydney, Australia, April 2010.

A. Zahemszky, B. Gajic, C. Rothenberg et al.,"Experimentally-driven research in Publish/Subscribe Information-centric Internetworking", in *Proc. of the International Conference on Testbeds and Research for the Development of Networks and Communities (TridentCom)*. Berlin, Germany, May 2010.

B. Gajic, J. Riihijärvi, and P. Mähönen, "Evaluation of Publish-Subscribe Based Communication over WiMAX Network", in *Proc. of International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT-T 2010)*. Moscow, Russia, October 2010.

B. Gajic, J. Riihijärvi, and P. Mähönen, "Intra-Domain Topology Manager for Publish-Subscribe Networks", in *Proc. of the International Conference on Telecommunications (ICT 2011).* Ayia Napa, Cyprus, May 2011.

B. Gajic, J. Riihijärvi, and P. Mähönen, "On Flexible Topology Formation in Publish-Subscribe Networks", in *Proc. of the International Conference on Communications (IEEE ICC), International Workshop on Network of the Future*. Ottawa,

Canada, June 2012.

B. Gajic, J. Palenzuela, J. Riihijärvi, and P. Mähönen, "Mobility-aware Topology Manager for Publish-Subscribe Networks", in *Proc. of the International Conference on New Technologies, Mobility and Security (IFIP NTMS)*. Istanbul, Turkey, May 2012.