Cooperation and Accounting in Multi-Hop Cellular Networks

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von
Attila Weyland
aus Deutschland

Leiter der Arbeit:
Prof. Dr. T. Braun
Institut für Informatik und angewandte Mathematik

Cooperation and Accounting in Multi-Hop Cellular Networks

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von
Attila Weyland
aus Deutschland

Leiter der Arbeit:
Prof. Dr. T. Braun
Institut für Informatik und angewandte Mathematik

Von der Philosophisch-naturwissenschaftlichen Fakultät angenommen.

Der Dekan:
Bern, den 10. November 2005
Prof. Dr. P. Messerli

Abstract

In the last decade a new paradigm in computer networks gained a lot of popularity, especially in the wireless network research community: mobile ad hoc networks. The main difference between this new and the existing conventional networking paradigm lies in the architecture of the network and the role of the different network components. In the conventional paradigm, a person with a computer connects to a pre-established network infrastructure, which consists of dedicated device to transport and deliver information among computers. In ad hoc networks, the pre-existing infrastructure is missing. Instead, the network is created on demand, with the help of each person and her/his computer. Thus, the computers of individuals take the role of the networking devices. The independence from any pre-installed network infrastructure allows these networks to be ad hoc deployable and to operate at low costs. However, the ad hoc communication paradigm also raises many questions, on if which is how to ensure the participation of the individual persons in the network. If the network participants do not cooperate by providing the network service to others, but only want to use the network selfishly, the network ceases to exist.

Because many open questions could not be satisfactory answered yet, mobile ad hoc networks are not widely used outside the research community. With multi-hop cellular networks, this situation might change. Multi-hop cellular networks combine the flexibility of mobile ad hoc networks and the reliability of infrastructure-based networks. Usually, they are comprised of an ad hoc as well as a conventional cellular network part. Therefore, these networks can provide access to persons or information located outside an ad hoc network. Because of the increased reachability and their low deployment costs, they appear to be a viable option for wireless Internet service providers. Still most of the open questions from ad hoc networks remain, however some can be addressed with the help of the provider.

In this thesis we argue that cooperation among network participants is a fundamental requirement of ad hoc networks and needs to be guaranteed in order to show the viability of multi-hop communications. Therefore, we propose a novel cooperation and accounting architecture for multi-hop cellular networks, which ensures cooperation among network participants. In particular, we specify a charging and rewarding mechanisms for network participants, which makes cooperation a rewarding alternative to selfishness. In order to retain as much as possible of the

flexibility of mobile ad hoc network, we use decentralized charging and rewarding on the computers of the participants. At the same time we want to keep the provider in control of the cash flow and therefore use the centralized exchange of rewards at dedicated terminals. To decrease the dependency on these terminals, we introduce resellers, which are selected network participants. Resellers are allowed to exchange the rewards of normal participants without a terminal. Further, we evaluate our mechanism via simulations as well as tests in a real environment using an implementation under Linux. Finally, we describe the possibilities of network planning and how our cooperation and accounting architecture can support the provider in this process.

Acknowledgements

This thesis is the outcome of about four years of research, which started in February 2002 at the Institute of Computer Science and Applied Mathematics of the University of Bern. The work presented here mainly was performed in the framework of a project financed by the Swiss National Science Foundation. Many people have contributed in various ways to this work and I want to take the opportunity and thank them.

First, I want to express my sincere gratitude to my advisor, Prof. Dr. Torsten Braun, who supported my work throughout this time. His ideas and encouragement were a great help for me. Prof. Dr. Torsten Braun also gave me the opportunity to present the work at various conferences and thereby receive valuable feedback from the research community, for which I thank him.

I also like to thank Prof. Dr. Burkhard Stiller who agreed to conduct the Koreferat of this work. Also, Prof. Dr. Oscar Nierstrasz who was willing to be the co-examinator of this work deserves many thanks.

Many thanks go to my colleagues at the institute and in our research group for providing a pleasant and friendly working environment with discussions on various topics. In particular, I thank Peppo Brambilla, Marc Brogle, Thomas Bernoulli, Marc Danzeisen, Marc Heissenbüttel, Dragan Milic, Matthias Scheidegger, and Markus Wälchli. Special thanks go to Marc-Alain Steinemann for the fruitful co-operation in the VITELS project. I am grateful to Ruy de Oliveira for his friendship as well as the many valuable discussions and hints. I am indebted to Florian Baumgartner for his friendship and support as well as the many interesting discussions we had. I also would like to thank Carolin Latze and Thomas Staub for their dedication and effort while performing their master's thesis with me. I am grateful to Ruth Bestgen, the secretary of our research group, for her help and friendship during all these years. I also like to thank our former group members, Roland Balmer as well as Günther and Silvia Stattenberger.

I am deeply grateful to my family and my friends, who supported me in many ways and always kept the connection - even over long distances. Finally, I thank my fiance Chika Hamagami for sharing with me the good and difficult times, especially for her encouragement and her patience while performing this work.

Contents

Chapter 1

Introduction

1.1 Overview

Multi-hop cellular networks are a promising network architecture, which give the mobile ad hoc communication paradigm the opportunity to become commercially viable and thus widely used. Ad hoc networks fundamentally differ from the existing conventional networking paradigm. In the conventional paradigm, a person with a computer connects to a pre-established network infrastructure. This existing infrastructure consists of dedicated devices, which perform specific networking tasks such as the transportation and delivery of information among computers. Thus, there is a clear distinction regarding the role in the network between normal computers operated by individuals at home or in a company and the network devices operated by provider. This separation is removed in ad hoc networks. Instead of each computer connecting to an existing network infrastructure, the computers directly interconnect to each other. Thus, the computers take the role and tasks of the network devices from the provider, who is not required anymore in such a scenario. The independence from any pre-installed network infrastructure allows these networks to be ad hoc deployable and to operate at low costs.

The ad hoc communication paradigm raises many questions for example regarding the overall performance and the security. In a conventional network, dedicated network devices maintained the network. In the ad hoc paradigm, the computers operated by persons have to maintain the network in addition to their already existing tasks. Considering that the ad hoc communication paradigm is especially appropriate for mobile communications, the disadvantages of wireless (radio) technology such as the limitation of the available bandwidth and the high amount of transmission errors as well as the consequences of mobility come to effect. All this leads to a reduced performance of ad hoc networks compared to conventional networks. The security of ad hoc networks is important, because the trust relations become much more complex. In the conventional paradigm, each person operating a computer has to trust the network infrastructure provider, for example an employee trusts his administrator to ensure the secure operation of the company's network. In the ad hoc paradigm, each participant has to inherently trust all other participants, because they are her potential network service provider. Another open question closely related to security is how to ensure the participation of the individual persons in the ad hoc network, i.e. how to make sure that they cooperate by providing the network service to others and not only want to use it selfishly.

Because many open questions could not be satisfactory answered yet, mobile ad hoc networks are not widely used outside the research community. With multi-hop cellular networks, this situation might change. Multi-hop cellular networks combine the flexibility of mobile ad hoc networks and the reliability of infrastructure-based networks. Usually, they are comprised of an ad hoc as well as a conventional cellular network part. Therefore, these networks can provide access to persons or information located outside an ad hoc network. Because of the increased reachability and their low deployment costs, they appear to be a viable option for wireless Internet service providers. Still most of the open questions already known from ad hoc networks remain, however some can be solved with the help of the provider.

One of these issues is the cooperation among the participants in the multi-hop cellular network. Without cooperation, the participants do not provide the network service to others and thereby cause the bigger ad hoc part of the network to stop functioning. Therefore, we identify cooperation as a fundamental requirement, which needs to be ensured to show the viability of multi-hop communications. Consequently, we study cooperation mechanisms for multi-hop cellular networks in this thesis. We are especially interested in an incentive-based cooperation assurance and favor a decentralized design to retain the flexibility and attractiveness of mobile ad hoc networks. We propose a novel cooperation and accounting architecture for multi-hop cellular networks, which consists of a charging and rewarding mechanisms for network participants, a reseller extension for selected participants and network management support for the provider.

In the remainder of this chapter, we motivate and present our work in the context of the current state of the art. We also summarize the main contributions and conclude with an outline of this thesis.

1.2 Motivation

Cooperation among network participants is vital for the correct operation of multi-hop cellular networks. Without the participants providing the network service to each other, connections are lost and the network ceases to exist. Cooperation among network participants can be ensured in two ways, either through the fear of punishment in case of selfishness or by the hope for rewards in case of cooperativeness. The first are also called detection-based approaches, because they rely on monitoring the behaviour of the participants and react accordingly in case of selfish behaviour. The latter are called motivation-based approaches, as they distribute rewards in case of cooperative behaviour. All detection-based schemes in the literature target mobile ad hoc networks, most of the motivation-based schemes target multi-hop cellular networks.

The correct identification of misbehavior and false accusations seems very difficult in detection-based approaches. Therefore, we favour motivation-based approaches, as they seem more suitable for the applications in civilian scenarios. The existing motivation-based approaches, mostly rely on a centralized authentication and/or accounting infrastructure, which reduce the flexibility and dynamics initially gained from the ad hoc communication paradigm. The only decentral motivation-based scheme has difficulties in terms of ensuring a long-term operation, because it can not guarantee the availability of virtual currency required for transmission over time. Therefore, we designed a partly decentralized cooperation framework, which retains as much of the flexibility from ad hoc networks as possible and also ensures a long-term operation.

1.3 Contributions

Cooperation among node is one of the fundamental requirements in multi-hop communications and its assurance a challenging task. Throughout this thesis, we have explored the area of cooperation in multi-hop cellular networks and researched several new ideas, which lead us to the design of a new cooperation and accounting architecture. With this architecture, a wireless network provider is able to stimulate the cooperation among his network participants via virtual charges and rewards. This ensures the correct functioning of the multi-hop cellular network and thereby saves the provider the costs of additional base stations.

We also give the provider complete control over the virtual cash flow of charges and rewards, so that he can regulate the amount of virtual money in the network. In particular, the provider has the possibility to dynamically adapt the costs for the customer and thus maximize his revenue. We achieve this control by requiring the provider or a representative for the acquisition of new virtual money as well as for the exchange of obtained rewards. We introduce service stations, which are similar to low-bandwidth, stationary terminals, where customers can engage in the accounting. In addition we allow dedicated nodes to act as resellers to make up for the immobility of the service stations. We also offer two possibilities of charges, i.e. globally fixed and dynamic hop count related charges. Further, we share the charges between the originator and recipient of a communication, i.e. each communication participant only pays for the connection to her gateway. This gives the provider a better possibility to map the expenses as well as revenue to his individual multi-hop cellular networks.

Our evaluations through simulations show that a low number of service stations and resellers provide the best performance, in the sense that few nodes are rarely short of virtual money and thus unable to cover the cost for transmission or reception of packets. In such a setting our architecture comes close to the performance of a network without any cooperation framework, i.e. where all nodes are assumed to cooperate out of free will. The results also indicate that when we set the fixed charges close to the average hop count, the nodes are able to transmit more packets. We also obtained promising results with the reduction of the granularity of the rewards, i.e. by sending them less often, and at the same time increasing their value.

From our evaluations of other cooperation schemes, we found that a self-perpetuating cycle of virtual money is very difficult to achieve and thus additional sources of income are required to ensure the constant availability of virtual money. This result justifies the presence of a central instance (such as a provider) - even in an ad hoc network.

This presence is also helpful for ensuring the security. We analyzed various attacks on our architecture and found fraudulent attacks to be not beneficial for the adversary. In addition, our security architecture supports the identification of suspected malicious attackers.

We also gained more insight from the implementation of our cooperation architecture under Linux. While we found the introduced delay to be acceptable for small number of hops, it showed us that the security functionality is very demanding. We noted the strong influence on the jitter by computers with different processors.

Last, we see promising possibilities for the management of multi-hop cellular network provided by our cooperation and accounting architecture.

1.4 Outline

This thesis is structured into the following chapters:

Chapter 2 gives an overview of multi-hop cellular networks. We show the development process of multi-hop cellular networks to their current state and describe the underlying wireless technologies and interconnecting protocols. We also present the possible benefits and application scenarios of multi-hop cellular networks. In this context, we identify important challenges and give a brief overview on the related work in this research area.

Chapter 3 presents the challenge of cooperation in multi-hop cellular networks. We motivate the necessity for cooperation in multi-hop networks and describe the possible approaches to cooperation. Further, we analyze the related work in this research area and compare the existing concepts and architectures according to their key characteristics.

Chapter 4 introduces our cooperation and accounting strategy for hybrid networks. We describe the architecture and explain the different operation phases. We also give a security analysis of CASHnet, both from a general view as well as related to the different operation modes. In addition, we present a resale extensions to CASHnet and show its support in the management of multi-hop networks.

Chapter 5 shows our evaluation of CASHnet in the network simulator ns-2. We describe the network simulator ns-2 as well as our implementation of CASHnet into it. We explain our simulation setup and evaluation criteria. In addition, we identify the key parameters in CASHnet and analyze their effect. Further, we discuss the simulation results of CASHnet and compare them to Nuglet.

Chapter 6 describes our implementation and evaluation of CASHnet under Linux. We present our developing environment and the used netfilter/iptables. Further, we explain our testbed and our evaluation criteria. Then, we analyze and discuss the results from our tests.

In Chapter 7 we summarize our main findings and conclude the thesis.

Chapter 2

Multi-hop Cellular Networks

2.1 Introduction

A multi-hop cellular network is an architecture for wireless communication. It results from the combination of two prominent network architectures: mobile ad hoc networks and infrastructure-based wireless networks. The combined flexibility of mobile, ad hoc deployable infrastructure and reliability of stationary infrastructure covers scenarios, where both architectures would fall short alone. For example, in areas with no pre-installed infrastructure, multi-hop cellular networks allow the provision of connectivity among users and to the Internet at low costs. It is also interesting for scenarios, where the connectivity demand is highly variable over time, such as at public gatherings for games, etc. Compared to an infrastructure-based wireless network, the benefits achieved by the combination include the extended coverage at reduced infrastructure costs, the dynamic adaptation of the network topology to current needs, an increased efficiency in the usage of the available frequencies and the availability of indicators for network planning. Multi-hop cellular networks are also called *hybrid wireless networks*.

In this chapter we discuss the concepts and mechanisms of multi-hop cellular networks. We start with a description of the development process to illustrate the origin of these networks. Then, we explain the elements of this architecture in a general way and list the possible benefits. We continue with a presentation of current and future wireless technologies as well as interconnecting protocols, which enable multi-hop cellular networks. We also show application scenarios and analyze important challenges in these networks. Last, we give an overview about related work in the area of multi-hop cellular networks and summarize this chapter.

2.2 Development Process

Multi-hop cellular networks were first proposed by Hsu and Lin [HL00] with the intention to benefit from the advantages of combining mobile ad hoc networks and infrastructure-based wireless networks. Both network architectures originate from the US military projects on packet radio networks. We summarize the development process in the following two sections.

2.2.1 Mobile Ad Hoc Networks

The first work on mobile ad hoc networks dates from the early 70s. The US military was in need of a communication infrastructure, which would not depend on pre-placed components and be easily movable. Radio communication was chosen to mobilize the network infrastructure. However, it also introduces limitations. Radio frequencies higher than 100 MHz do not propagate beyond the line of sight.

In the beginning of the 60s Baran [Bar64] and Davies [DBSW67] had independently discovered the packet switching paradigm, which introduced bandwidth sharing and store-and-forward routing. Inspired by the effectiveness of packet switching and its application to a mobile wireless environment, the Defense Advanced Research Projects Agency, DARPA launched the Packet Radio Network project in 1973 as described by Freebersyser and Leiner in [FL01]. It addressed the radio coverage limitation by using multi-hop store-and-forward routing techniques. Within the project the first mobile ad hoc network was created.

Several consecutive projects solved issues regarding the scalability, security, performance and energy efficiency of the network. With the appearance of commercial radio technologies (e.g. IEEE 802.11) in the mid 90s, the potential for mobile ad hoc networks outside the military domain attracted many researchers. Even a few commercial solutions have been developed, such as Motorola's Mobile Mesh Networks [Mot05] and SPANworks' MultiPeer [SPA05]. In the early 2000s the major interest in the wireless research community focussed on wireless sensor networks.

In their extensive survey Chlamtac et al. [CCL03] also give some detailed insight on the development process of mobile ad hoc networks. Mobile ad hoc networks are also called *infrastructureless networks*.

2.2.2 Infrastructure-based Wireless Networks

Beside the work on infrastructureless wireless networks, research was also conducted in the area of infrastructure-based radio networks. These networks consist of base stations covering a certain area called cells and mobile stations roaming between these cells. A prominent commercialization is the cellular communication system (e.g. ETSI GSM) introduced in the 80s. With the standardization of radio technology for computer communication in the mid 90s (e.g. IEEE 802.11) wireless communication in local area networks began to spread. Starting as an extension to the wired Ethernet, the so called wireless LAN technology addressed the increasing need for mobility and connectivity in the civilian sector. In the early 2000s, new standards defined wireless communication in metropolitan area as well as personal area networks.

2.3 Multi-hop Cellular Network Architecture

Hybrid wireless networks are composed of *base stations* and *mobile stations*. A base station is a stationary device which provides the interconnection between the wireless access network and the wired backbone network. A base station is also called *access point*. A mobile station is a device which moves around and is capable of connecting to a wireless network in range. A mobile station is also called *mobile node*. All wireless devices have a limited communication range.

For comparison, a simplified view of infrastructure-based, infrastructureless and the resulting hybrid wireless networks is given in Figure 2.1 - 2.3. The maximum communication range is indicated by a surrounding border in form of a circle or a hexagon. Each device can only communicate with devices inside its border. A wireless connection between two devices is indicated by a dashed line. A wired connection to a backbone network is indicated by solid line. We distinguish between stationary (gray) and mobile (white) coverage areas. The total coverage area of a network is composed of all circles or hexagons.

Figure 2.1 depicts an infrastructureless wireless network, which is also known as mobile ad hoc network, MANET. It solely consists of wireless (mobile) stations. In such a network, connections over multiple hops between mobile stations are necessary to allow building up the network as no other infrastructure besides the mobile stations themselves is available. The coverage area of the network is completely dynamic due to the movement of the mobile stations and mostly limited by the communication range of the mobile stations.

Figure 2.2a and Figure 2.2b illustrate two infrastructure-based wireless networks; a cellular network and a wireless LAN. Both architectures consist of base stations and mobile stations. In these networks only direct, single-hop connections between a mobile station and a base station are supported. The base stations themselves are interconnected via a (stationary) wired or wireless backbone network. All connections between the mobile stations have to pass via the base station and - if necessary - via the backbone network. The coverage area of such a wireless network is therefore mainly limited by the number and location of deployed base stations.

Figure 2.3 shows a hybrid network, which consists of base stations and mobile stations. In this network architecture, multi-hop connections are used to increase the existing coverage area provided by the base stations. The base stations are interconnected via a wired backbone network. A connection can either pass solely via mobile stations or via mobile stations and base stations. The coverage area of the base stations is fixed; the coverage area formed by the mobile stations is changing due to their movement.

The theoretical benefits resulting from the combination of infrastructureless and infrastructure-based wireless networks are numerous. However, in practice some of the benefits can not be used due to the limitations of the current predominant wireless technologies and communication protocols.

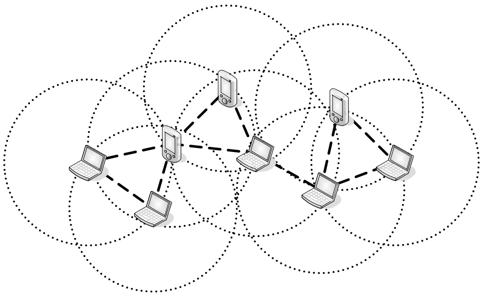
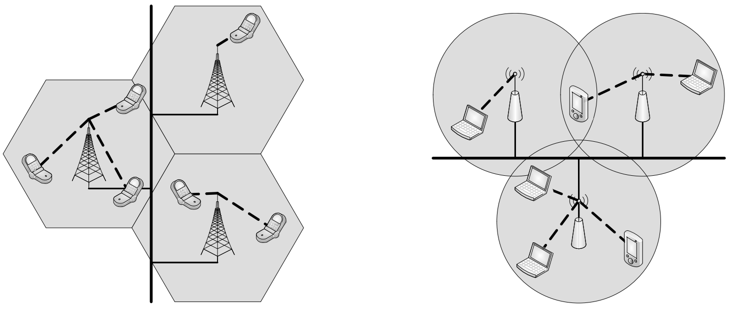


Figure 2.1: Infrastructureless wireless network



(a) Cellular network

(b) Wireless LAN

Figure 2.2: Infrastructure-based wireless networks

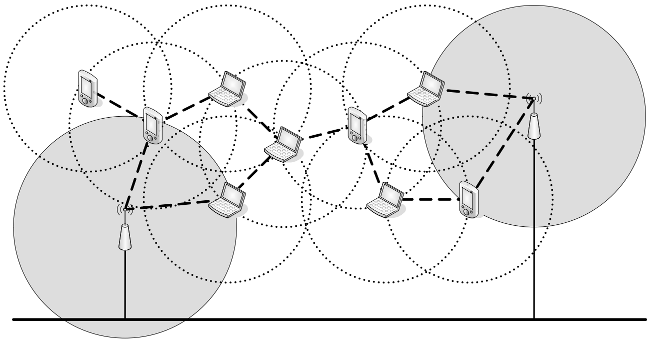


Figure 2.3: Hybrid wireless network

Extended coverage at reduced costs: The coverage of an infrastructure-based wireless network depends on the number of base stations. However, a wireless Internet service provider will deploy a base station only if the revenue covers the initial deployment costs in foreseeable time. Therefore, the base stations are mostly deployed in so called hot spots, where the number of potential users is expected to be constantly high. Typical examples are airports and train stations. The need of a certain number of users prevents the deployment of base stations in large scale. Thus potential customers outside hot spot areas can not be reached. With multi-hop networks, the coverage of the network can be extended without requiring pre-installed stationary infrastructure. Only a certain number of cooperative users is required in the vicinity of an existing access point.

Dynamic network topology: The deployment of several base stations to cover a hot spot area is rather expensive, especially the measurements and adjustments of signal propagation are time intensive tasks. Once the base stations have been set up, their location can not be changed easily. With multi-hop networks, the base stations is like an anchor point, around which a dynamic cloud of cooperative users accumulates. The user behaviour (movement, cooperativeness) directly influences the network topology.

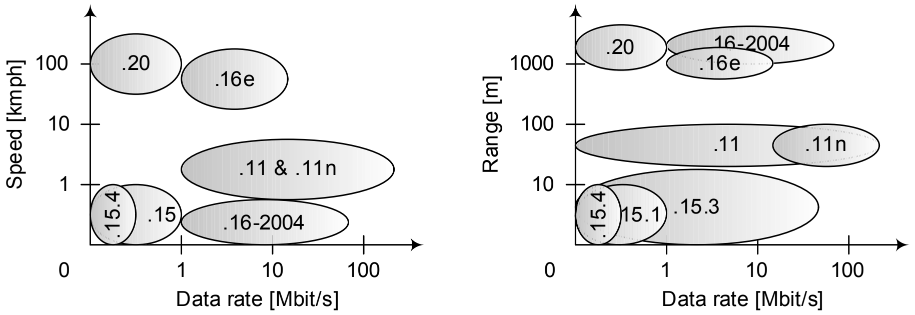
Spatial frequency reuse: As mentioned above, infrastructure-based networks only support single-hop connections within a cell. The communication between two mobile stations in the same cell has to go via the base station. Additionally, their communication uses frequencies which are then not available in the whole cell. In multi-hop networks, the spatial reuse of frequencies is possible due to shorter communication ranges. Directly communicating node pairs which reside in the base stations' cell and which do not interfere with each other can use the same frequency for their communication.

Network planning indicators: The cost of deploying base stations is quite high. To reduce the financial risk, only areas with an assured amount of users receive coverage. Even in popular locations, it may not be obvious where exactly the coverage is required and how lasting the demand is going to be. The wireless Internet provider can use multi-hop cellular networks to test an area for its connectivity demand and finally support the decision about and location of the deployment of a base station.

2.4 Wireless Technologies

Several wireless technologies with support for multi-hop networks already exist or are in the process of standardization. Each technology targets specific application scenarios and therefore differs in supported data rate, communication range and mobile station speed. The targeted network size ranges from body (worn by a person) and personal (close to a person) over local (up to 1 m^2) to metropolitan (city-wide) areas.

At the beginning of 2005, the predominant wireless technology in local area networks is wireless LAN based on the standard IEEE 802.11. However, IEEE 802.11 works best indoor (less than 100 m range) and with few users (less than 20 per base station). These restrictive characteristics exclude many important application scenarios. The increasing demand for ubiquitous broadband connectivity lead to the foundation of several new working groups IEEE 802.15 [I1505],



(a) Speed vs. data rate

(b) Range vs. data rate

Figure 2.4: Wireless technologies related

802.16 [I1605] and 802.20 [I2005] with the task to specify standards which enable wireless broadband communication in personal and metropolitan area networks.

We find it arguable whether devices in a body or personal area network require incentives for cooperation. In such networks the devices typically belong to one person, who should be the single authority for them and thereby render the need for stimulation of cooperation among these devices obsolete. Another point is that these devices are designed to perform simple tasks and only have limited resources. An additional cooperation framework would pose an unbearable burden for them. To date, we see the problem of cooperation in local area networks, where each communicating node is operated by an individual person. It will probably expand to metropolitan area networks with future wireless technologies. In Chapter 3 we explain our view on cooperation in detail.

Figure 2.4 relates several ratified and draft standards according to the specified data rate, mobile station speed, and communication range. Table 2.1 lists the exact values. Note, that these values represent the theoretical maxima and that in practice a reduction of 20% or more is to be expected. IEEE 802.11, 802.15, 802.16 and 802.20 complement each other as they address different application scenarios and make ubiquitous mobile broadband wireless access feasible. Figure 2.5 shows an example of a possible future interaction of the different technologies. IEEE 802.16 provides connectivity between corporation buildings or community houses, IEEE 802.16e is targeted at vehicles, e.g. bus and car. IEEE 802.11 and IEEE 802.15.3 will coexist depending on the application in the local and personal area, e.g. IEEE 802.11 allows web browsing and reading E-mail while IEEE 802.15.3 enables streaming video and voice. In the following paragraphs, the four base standards are briefly explained, with a focus on IEEE 802.11.

2.4.1 IEEE 802.11

With the ratification of the first standard called IEEE 802.11 [IEE99] in 1998, the usage of Wireless Local Area Network technology has continuously increased in the public, private and commercial sector. The IEEE 802.11 working group [I1105]

IEEE 802 Standards	.11		.15			.16		.20[2]
	.g[1]	.n[2]	.1[1]	.3[1]	.4[1]	-2004[1]	.e[2]	
Data rate [Mbps]	54	200	1	55	0.25	75	15	1
Range [m]	100	100	10	30	10	5-8 k	2-5 k	15 k
Speed [kmph]	3	3	-	-	-	none	130	250

Table 2.1: Wireless technologies in detail

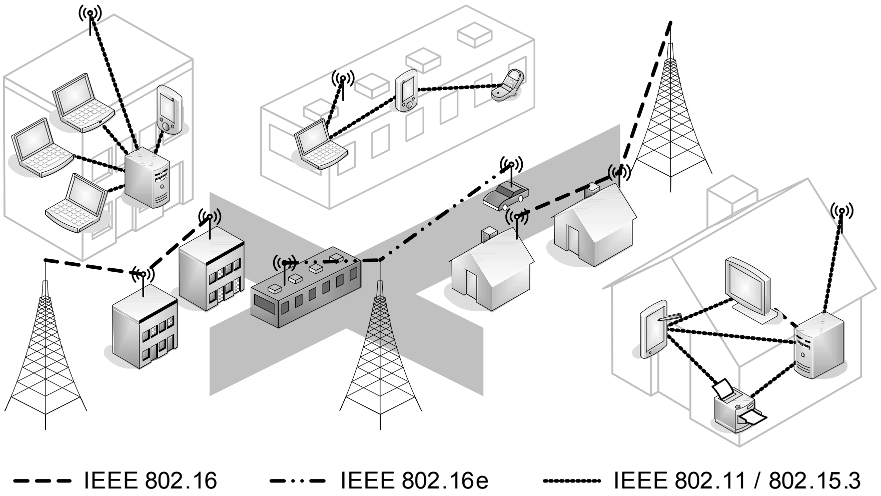


Figure 2.5: Ubiquitous Mobile Broadband Wireless Access

defined several amendments to the standard, adding new and optimizing existing functionality. Since the standard leaves many decisions to the implementors, the Wi-Fi alliance ensures the interoperability of devices from different vendors by certifying products compliant to a common functional set. The current Wireless LAN standards define a peak data rate of 54 Mbps (IEEE 802.11a/g) and operate in license-exempt frequency bands at 5/2.4 GHz. A future amendment (IEEE 802.11n) is expected to achieve data rates well above 100 Mbps [AFN04].

When using radio technology all nodes share the same medium, but due to the limited communication range not all nodes can hear each other. This can lead to nodes interfering with their transmission/reception of packets. Therefore, special mechanisms are required to ensure reliable communication. In the following, we focus on the Distributed Coordination Function, DCF which is used in infrastructureless networks. IEEE 802.11 DCF uses Carrier Sense Multiple Access with Collision Avoidance, CSMA/CA for access to the medium. CSMA/CA is a contention based scheme, where all nodes have to compete for the access.

A node senses the medium before it transmits its packet. If the medium is busy,

[1] Ratified Standard

[2] Draft Standard

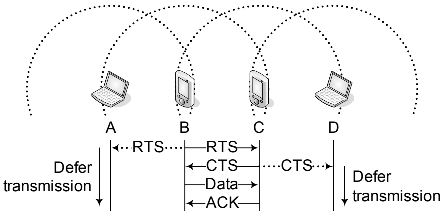


Figure 2.6: IEEE 802.11 4-way handshake

it defers for a time derived from the binary exponential backoff algorithm. If the medium is free for a short time period, the node is allowed to transmit. The recipient of the packet verifies its CRC and - if correct - sends and acknowledgement (ACK) packet to the sender. If the sender does not receive an acknowledgement, a transmission error has occurred. The sender then retransmits the packet until an acknowledgement is received or the maximum number of allowed retransmissions is reached.

To reduce the probability of two nodes colliding because they can not hear each other (hidden node problem) a virtual carrier sense mechanism is specified. It is called virtual because, instead of sensing the physical medium directly, it is performed at the MAC layer. Before transmitting its data packet, a node sends a short request to send (RTS) control packet, which includes the sender, recipient and duration of the following transmission. If the medium is free, the recipient replies with a clear to send (CTS) control packet including the same information. Now, all nodes within the communication range of sender or recipient know and respect the information contained in the RTS/CTS packets and wait before transmitting their own packets.

Figure 2.6 shows a sequence diagram of the 4-way handshake, which is necessary to transmit data from one node to another. Node B wants to transmit data to node C. Node B sends an RTS, node C replies with a CTS. Now node B transmits data and node C acknowledges its correct reception. Node A and D overhear the RTS/CTS respectively and defer their transmission for the indicated duration.

2.4.2 IEEE 802.15

In 1999 the IEEE 802.15 working group for wireless personal area networks was established. Its goal was to specify a standard, for small-sized, low-cost and low-power communication devices. In the same year Ericsson ratified Bluetooth, an industrial specification for wireless personal area networks. The working group ratified its first standard called 802.15.1 [IEE02] in 2002. It is an adaption of the Bluetooth 1.1 specification. IEEE 802.15.1 supports data rates up to 1 Mbps and a communication range up to 1 m. It operates in license-exempt frequency bands at 2.4 GHz. In Bluetooth, a master node manages up to seven slaves. This group is called a picconet. Up to ten picconets can coexist within communication range. The interconnection of picconets is called scatternet and requires a node which is at the same time master and slave in a different picconet respectively. The master node schedules the medium access via a polling and reservation scheme. The medium access is based on Time Division Multiple Access, TDMA. The up- and downlinks are separated via time division multiplexing.

Initially, the 802.15 working group focussed on Bluetooth only. However, Bluetooth is neither suitable for broadband applications nor for sensor/actor scenarios. Therefore, IEEE 802.15.3 [IEE03a] and 802.15.4 [IEE03b] respectively have been ratified in 2003. IEEE 802.15.3 allows a data rate up to 55 Mbps, a communication range up to 30 m and operates in the licence-exempt frequency band 2.4 GHz. Future amendments will support data rates up to 100 Mbps. The WiMedia alliance has been established to ensure interoperability of ultrawideband technologies. IEEE 802.15.4 supports a data rate up to 250 kbps, a communication range up to 10m and operates in license-exempt frequency bands at 915 MHz, 868 MHz and 2.4 GHz. In 2004, the ZigBee alliance defined a security, network and application extension to IEEE 802.15.4.

2.4.3 IEEE 802.16

Because no wireless standard was available for metropolitan area networks and with the intention to cover the last mile and standardize broadband wireless access, IEEE initiated a working group called 802.16 in 1999. The ratification of the most recent standard IEEE 802.16-2004 [IEE04] includes all previous amendments and further improvements. It provides a data rate of up to 75 Mbps with an expected communication range of 5-8 km. However, recent analysis of Ghosh et al. [GWAC05] show the performance to be well below these values. IEEE 802.16-2004 operates in licensed as well as licensed-exempt frequency bands between 2 and 11 GHz and does not support mobile stations (subscriber stations). A future amendment (IEEE 802.16e [IEE05]) will address connectivity for mobile stations supporting speeds up to 130 kmph and data rates up to 15 Mbps. The alliance for Worldwide Interoperability for Microwave Access, WiMax ensures the interoperability of devices based on IEEE 802.16. So far, IEEE 802.16 focusses on single-hop connections, permitting only direct connections between base stations and subscriber stations (point-to-multipoint). Beginning of 2005, the support for multi-hop connections over subscriber stations (point-to-point) is still under discussion.

The access mechanism to the medium is more sophisticated than in IEEE 802.11. A base station schedules the access for the subscriber stations, with a distinction between up- and downlink. Because multiple subscriber stations may try to access the base station simultaneously, the medium access mechanism on the uplink uses Demand Assignment Multiple Access-Time Division Multiple Access, DAMA-TDMA. On the downlink, only the base station initiates the access to the medium. Therefore, a Time Division Multiplexing, TDM mechanism is used.

The subscriber stations have to compete for a time slot on the uplink once. After successfully registering with the base station, a subscriber station gets assigned time slots on demand. The base station can dynamically enlarge or shorten the allocated time slots to reflect changes in the requirements of the subscriber stations.

2.4.4 IEEE 802.20

As IEEE 802.16 initially supported stationary devices (subscriber and base stations) only, IEEE established the 802.20 working group in 2002 to specify a standard for ubiquitous mobile broadband wireless access. IEEE 802.20 intends to support mobile station speeds up to 250 kmph, data rates up to 1 Mbps and a communication range up to 15 km. It operates in licensed frequency bands below 3.5 GHz. The work is still in preliminary stages and a first draft standard is expected in late 2006.

While IEEE 802.16e and 802.20 both aim at providing broadband wireless access to mobile devices, there are some important differences. IEEE 802.16e will be based on the already existing standard IEEE 802.16-2004, while IEEE 802.20 will be designed from scratch. Also, in contrast to IEEE 802.16e, which provides connectivity up to 130 kmph, IEEE 802.20 will support mobile stations moving at higher speed, e.g. fast cars and trains.

2.5 Interconnection of Mobile Stations

The interconnection of multiple hops to form a network requires addressing and routing mechanisms. In multi-hop cellular networks, the nodes have additional abilities compared to nodes in a single-hop network. Besides transmitting self-generated packets and receiving packets addressed to it, each node acts like a router, i.e. it receives and forwards packets addressed to other nodes. Therefore, each node needs to discover the address of the destination and decide how to forward the packet based on that information. The mobility of the nodes makes these challenging tasks.

On the one hand, a multi-hop cellular networks can just be seen as a mobile ad hoc network with one important destination being the base station, which provides the interconnection to the stationary backbone network, e.g. the Internet. Here the routing schemes from ad hoc networks only need to be extended with base station discovery mechanisms. This approach retains as much of the flexibility and dynamics of a mobile ad hoc network, but does not take advantage of the existing infrastructure. On the other hand, the base station can be made to an integral part in the routing scheme, thereby centralizing the network management. This can increase the reliability of the network. To date several proposals have been made, some of which we discuss in the related work in Section 2.7. As no standard has been agreed upon yet, it is common to use standardized mobile ad hoc routing protocols and extend them with the appropriate functionality. Therefore, we briefly introduce characteristics and classifications of mobile ad hoc network routing schemes. We also categorize well-known protocols according to these classifications and give an overview on the routing protocols AODV and DSR. We also describe the necessary extensions to AODV for global connectivity support, which we use in our evaluations.

2.5.1 Mobile Ad Hoc Network Routing

Routing in mobile ad hoc networks has to adopt to the specific properties of this architecture. Routes must be discovered over multiple hops from the source to the destination. A fast reaction to reflect changes in the topology is required. A minimal control message and processing overhead is necessary to allow the efficient usage of resources, such as network bandwidth and node battery power. The prevention of routing loops also supports resource efficiency. These requirements partly exclude each other (e.g. fast reaction and minimal control messages) and an enormous amount of research in optimizing these properties has been conducted in the last decade. The resulting number of routing protocols is so huge, that we focus on the initial proposals and classify them according to three characteristics:

- Information the routing decision is based on: *Topology-based* schemes are based on the network topology, which consists of the existing connections among nodes. They need to establish and maintain routes to destinations. *Position-based* schemes rely on the geographic positions of the source, its one-hop neighbor and the destination. They do not need to establish and maintain a route to the destination. Instead, they require a location service, which provides the geographical position of nodes. Such a service can be based on the global positioning system, GPS.

- Acquisition and maintenance of routing information: *Proactive* strategies maintain information about all the available paths in a network - independent of their current use. The maintenance of these paths consumes a considerable amount of bandwidth, especially when the network changes frequently due to node mobility. *Reactive* strategies try to address this communication overhead by reducing the number of maintained routes to the ones only currently in use. Here the drawbacks lie in the initial delay needed to discover a route before it is usable and the increased packet loss probability due to route changes during transmission. In scenarios with a high number of moving and communicating node pairs, reactive protocols may even perform worse than proactive ones. *Hybrid* strategies combine proactive and reactive routing techniques to better adapt to the current network situation.

- Support for hierarchy: *Hierarchical* routing places nodes into groups, also called clusters based on location or functionality of the nodes. The intention behind the hierarchy is to reduce the number of routing entries and the involved maintenance per node and make routing in mobile ad hoc networks more scalable. It requires hierarchical identifiers for nodes (e.g. IP

Routing Protocols	DSDV [PB94]	OLSR [CJL$^+$01]	AODV [PR99]	DSR [JMB01]	ZRP [HP01]	LAR [KV00]
Decision Acquisition	topology proactive	topology proactive	topology reactive	topology reactive	topology hybrid	position reactive

Table 2.2: Mobile ad hoc networks routing protocols

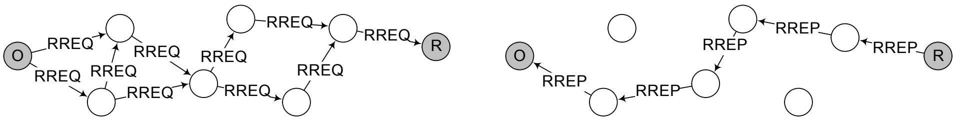
addresses), which might not be usable in mobile ad hoc networks (e.g. a node with an external IP address). *Flat* routing does not take hierarchical information into account and therefore each node needs to maintain a routing entry for its active destination.

Table 2.2 classifies a selection of established routing protocols according to the characteristics mentioned above. This selection does not include any hierarchical routing protocol, as the research to date has not yet lead to a widely accepted solution. Belding-Royer [BR04] gives an overview on routing in mobile ad hoc networks in general and analyzes some protocols in detail.

AODV

The Adhoc On-demand Distance Vector protocol, AODV [PR99] is a topology-based, reactive routing protocol. AODV combines the hop-by-hop routing, sequence numbers and beacons of DSDV with the basic route discovery and maintenance of DSR. The sequence numbers ensure loop freedom. AODV has reached the request for comments status, RFC3561 [PBRD03]. AODV has four message types: route request $RREQ$, route reply $RREP$, route error $RERR$ and route reply $REP - ACK$ acknowledgement. All AODV messages are transmitted via UDP. A routing table entry maintained by AODV contains the following information:

- Destination IP Address
- Destination Sequence Number
- Valid Destination Sequence Number flag
- Other state and routing flags (valid, invalid, repairable, being repaired)
- Network Interface
- Hop Count (number of hops needed to reach destination)
- Next Hop
- List of Precursors (nodes, which use this routing entry with the current node as next hop)
- Lifetime (expiration or deletion time of the route)



(a) Originator broadcasts route request

(b) Recipient unicasts route reply

Figure 2.7: AODV route discovery operation

Route Discovery AODV reactively creates routes upon request. If a node requires a route to a destination, it broadcasts a route request. The route request includes a hop count field set to zero, the incremented originator's sequence number and - if known - the destination sequence number. Each intermediate node which receives the $RREQ$ message increases the contained hop count field. The node also creates or updates the reverse path to the originator of the route request by setting the destination sequence number to the originator sequence number from the $RREQ$ message.

When the route request reaches the destination or another node with a fresh route entry to the destination, the respective node sends back a unicast route reply to the originator of the $RREQ$ message. The destination increments its own sequence number and puts it in the destination sequence number field of the $RREP$ message. The intermediate nodes use the reverse routes created earlier to forward the $RREP$ message. When the originator node receives the route reply, it is able to transmit data packets to the destination. Figure 2.7 illustrates the route discovery process in AODV.

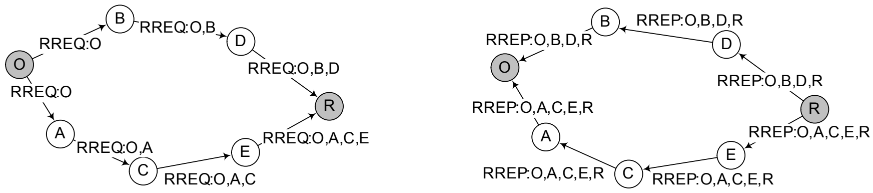
Route Maintenance The maintenance of routes in AODV relies on beacons and observation of the local connectivity status. Each node, which is part of an active route broadcasts connectivity information to its one-hop neighbors using a beacon message called $HELLO$. Such a $HELLO$ message is a route reply message with the TTL flag set to one. A node can monitor the link status to its active neighbors by observing the $HELLO$ messages or information from the link layer.

If a node does neither receive a $HELLO$ nor other messages from a neighbor in an active route it assumes the link to be broken. The node notifies those active one-hop neighbors called precursors, which use the node as next hop towards their destination by sending a route error message.

AODV only keeps routes as long as they are needed. In case a route is not active for a specified amount of time, the corresponding entry is removed from the routing table.

DSR

The Dynamic Source Routing protocol, DSR [JMB01] is a topology-based, reactive source routing protocol. It is expected to reach the request for comment status



(a) Originator broadcasts route request

(b) Recipient unicasts route reply

Figure 2.8: AODV route discovery operation

and is currently an Internet-Draft [JMH04]. The route discovery phase of DSR is similar to AODV. The main difference lies in the routing of data packets. In AODV data packets are forwarded hop by hop. In DSR the complete route information from the source to the destination is specified in advance, i.e. a source route, which includes every hop from the source to the destination. This source route is included in every data packet and the packet is routed accordingly. In order to obtain the complete route to the destination, the route request and reply messages accumulate the route information during their travel through the network.

Figure 2.8a shows the process of the route discovery. The originating node places its own IP address as well as the destination IP address into the route request $RREQ$ and broadcasts this message to its neighbors. When a neighbor node receives a $RREQ$ message, it updates its route to the source and appends its own IP address to the route request. As the $RREQ$ message travels through the network all traversed nodes are accumulated in the route request. Also, intermediate nodes, which receive a route request can update their route to all nodes passed so far until the source.

When a node receives a route request of which it is the destination or for which it has a route to the destination it generates a route reply $RREP$. The node includes the complete source route in the $RREP$ message. If the node is the destination, it uses the accumulated route from the $RREQ$. If it is not the destination, it combines the accumulated route with its existing route to the destination. Then, the node reverses the source route, adds it to the route request and unicasts it back along the same path to the source. The transmission of the route reply is illustrated in Figure 2.8b. Again, as intermediate nodes receive the $RREP$ message, they can update their route information to any node on the source route.

In case of a link break, the node upstream of the break sends a route error $RERR$ to the source. Because DSR allows to store multiple routes per destination in the route cache of each node, a $RERR$ message may not necessarily result in the transmission of a route request. Instead, alternative routes from the route cache may be available.

2.5.2 Global Connectivity Support for AODV

The routing protocols for mobile ad hoc networks consider them to be isolated networks with exclusive, interior addressing and routing. In multi-hop cellular networks, connectivity outside the mobile ad hoc network is required and thus addressing and routing becomes more demanding. The idea of global connectivity support for IPv4 mobile ad hoc networks has been initially proposed by Belding-Royer et al. [BRSP01]. This Internet-Draft is now expired, but Wakikawa et al. [WMP$^+$05] continue the work on global connectivity support for IPv6 mobile ad hoc networks.

Hamidian studied the support of Internet connectivity in mobile ad hoc networks and extended the AODV implementation in ns-2 accordingly as part of his master thesis [Ham03b]. We use his AODV extension called AODV+[Ham03a] in our simulations. He adds support for gateway discovery similar to the previously mentioned Internet Drafts.

Gateway Discovery

In case a node wants to communicate with a recipient located outside of the current mobile ad hoc network, it needs to find a gateway. This task is performed by the routing protocol - in our case AODV+[Ham03b], an extension of the AODV protocol - which runs on each node including the gateway. There are three methods to acquire information about a gateway: proactive, reactive and hybrid gateway discovery.

In the proactive gateway discovery the gateway periodically broadcasts a gateway advertisement $GWADV$ throughout the mobile ad hoc network. In order to limit the additional load introduced by this flooding, the advertisement interval needs to be sufficiently large. Figure 2.9 illustrates the propagation of a gateway advertisement throughout the mobile ad hoc network, where the gateway floods the network with a gateway advertisement.

The reactive gateway discovery is based on the route discovery messages from AODV $RREQ$ and $RREP$, which are each extended by an additional flag called Internet-Global Address Resolution Flag. An originator in need for a global connectivity broadcasts an extended route request $RREQ_I$. Only a gateway replies to this message with an extended route reply $RREP_I$, while all intermediate nodes towards the originator node add the path information to the gateway as default route. Figure 2.11 shows the operation of the route request and reply phase, where the originator broadcast an extended route request and the gateway answers with an extended route reply.

On the one hand proactive gateway discovery provides up to date routes, but introduces considerable communication overhead due to the flooding with gateway advertisements. On the over hand, reactive gateway discovery has no unnecessary overhead, but introduces delays because the route to the gateway must be acquired before any normal traffic can be transmitted. The hybrid gateway discovery scheme

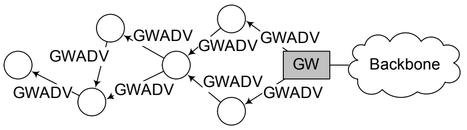


Figure 2.9: AODV+ proactive gateway discovery operation

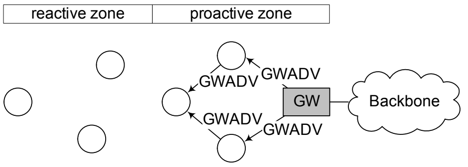
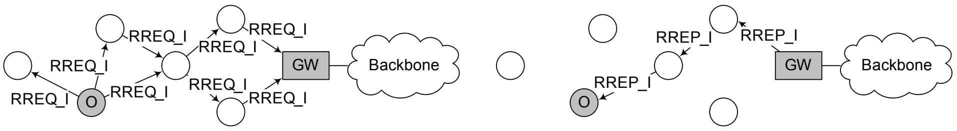


Figure 2.10: AODV+ hybrid gateway discovery operation



(a) Originator broadcasts a route request

(b) Gateway unicasts a route reply

Figure 2.11: AODV+ reactive gateway discovery operation

tries to combine the advantages of both, proactive and reactive methods. The mobile ad hoc network is separated into two zones from the gateway perspective. In the proactive zone, gateway advertisements are broadcasted. The range of the gateway advertisements and thus the range of the proactive zone is limited via the TTL field. The remaining part of the mobile ad hoc network, which is not reached by gateway advertisements is called the reactive zone. Here, the nodes have to reactively acquire path information to a gateway. Figure 2.10 depicts the operation of the hybrid gateway discovery, where the proactive zone has been restricted to a maximum distance of 2 hops from the gateway.

When a node learns about a gateway, it adds the path information as default route to its routing table. Upon transmission of a packet, the node looks up a route for the packet recipient. If the node does not find an entry for the recipient in the routing table, the node transmits a normal route request. If no route reply arrives within a specified interval, the node assumes the recipient to be located outside the current mobile ad hoc network and transmits the packet to the gateway using the default route.

2.6 Application Areas

Multi-hop cellular networks have numerous areas of application. Generally speaking, they can be used when the deployment of a complete stationary network infrastructure is infeasible due to the costs, the environment or the application. With the benefits listed in Section 2.3 in mind we see the following possibilities for multi-hop cellular networks.

2.6.1 Wireless Sensor Networks

Wireless sensor networks represent the next step towards extensive environment monitoring. Due to the recent advances in miniaturization, it is possible to build small devices capable of sensing and processing information. Wireless sensor networks consist of stationary sensor nodes, which act as routers. The flow of data is usually directed towards a sink, where the monitored information is processed. Wireless sensor networks require self-organizing capabilities and are similar to ad hoc and hybrid wireless networks. Example applications include disaster recovery, healthcare and home automation. In their survey, Akyildiz et al. [ASSC02] show open issues in sensor networks.

2.6.2 Wireless Mesh Networks

Wireless mesh networks represent the next step towards ubiquitous connectivity, with the intention to provide networking across different wireless technologies such as IEEE 802.11, 802.15 and 802.16. They consist of stationary mesh routers and stationary or mobile mesh clients. The clients themselves can also act as routers. Thus, wireless mesh networks are closely related to mobile ad hoc networks, in particular to hybrid wireless networks. Example applications include the interconnection of communities, neighborhoods and enterprises. Akyildiz et al. [AWW05] give an extensive survey on wireless mesh networks. An overview of mesh networks is presented by Bruno et al. [BCG05].

A special case of mesh networks are relays. A relay network consists of mobile stations which are connected to base stations directly (single-hop) or via a relay station (multi-hop). These relay stations only store and forward data, they do not generate packets themselves nor do they provide a connection to a wired backbone. They are deployed and operated by the provider. Example applications include coverage extensions to previously dead spots. Pabst et al. [PWS$^+$04] give an overview on the technical aspects of deploying relays.

2.7 Related Work on Multi-hop Cellular Networks

Considerable work has been done in the area of multi-hop cellular networks. The proposals study the theoretical and practical improvements achievable through the usage of multi-hop cellular networks. The results are rather encouraging, as the multi-hop architecture enables better reuse of frequencies while the available backbone network reduces traffic load. However, node mobility and high node density pose problems to IEEE 802.11 with its contention-based medium access control mechanism. The research on IEEE 802.11 MAC layer improvement tries to address this weakness. In the following sections we distinguish between proposals on new architectures and studies on the theoretical capacity of multi-hop cellular networks.

2.7.1 Multi-hop Cellular Network Architectures

As mentioned before, Hsu and Lin [HL00] first defined multi-hop cellular networks as a new architecture. The authors investigate the general principles of using multi-hop paths to base stations. They compare single-hop and multi-hop cellular networks in terms of mean hop count, hop-by-hop throughput, end-to-end throughput, and mean number of channels (i.e. simultaneous transmissions) under different traffic localities and transmission ranges. Their numerical evaluation shows that multi-hop cellular networks can perform better than single-hop networks in terms of throughput per node. Lin et al. [LHO$^+$00] continue their pioneering work and present a prototype implementation of their scheme based on IEEE 802.11. The define a bridging protocol, which enables routing and roaming between the base stations. The authors perform some tests with up to three hops to demonstrate the feasibility of their scheme.

Ananthapadmanabha et al. [AMM01] extend the architecture proposed in the pioneering work before. They concentrate on single-hop and multi-hop cellular networks based on IEEE 802.11. The authors specify a routing protocol for multi-hop cellular network. They evaluate their work via simulations and show that their routing algorithm increases the end-to-end throughput in the multi-hop network compared to the single-hop network. Wu et al. [WQDT01] introduce an ad hoc relaying architecture to cellular networks to reduce congestion caused by unbalanced traffic. So called ad hoc relay stations are placed within the cells and dynamically transfer traffic from one cell to another. The authors compare their architecture with traditional cellular systems in terms of call blocking/dropping probability, throughput and signaling overhead using analysis and simulations. They find that with a limited number of ad hoc relaying stations in a congested cellular network the overall systems performance can be improved.

Hsieh and Sivakumar [HS02] study the benefits of using an ad hoc network model in cellular wireless packet data networks. They use a communication model based on IEEE 802.11, DSR and TCP. They analyze their model with simulations in terms of throughput, energy consumption, mobility and fairness. Their results indicate, that although multi-hop communication allows for better spatial reuse, it does not automatically lead to increased throughput. To the contrary, the authors find that compared to single-hop connections within a cell, the throughput decreases for their communication model. They attribute this to the weaknesses in the used protocols. Dousse et al. [DTH02] consider a large-scale wireless network with low density, where they deploy fixed and interconnected base stations to increase connectivity. They evaluate their work using analysis and simulations based on poisson distribution and real population data by comparing pure ad hoc and hybrid networks in terms of node connectivity. The authors conclude that for nearly one-dimensional node densities the deployment of base stations increases connectivity, whereas for normal two-dimensional node density the connectivity does not increase.

Luo et al. [LRS$^+$03] describe an architecture to increase cell throughput of third-generation (3G) wireless data networks. To do so, each mobile station is equipped with a 3G and an IEEE 802.11 interface. Packets with destinations on poor quality 3G channels are routed via the IEEE 802.11 radio other multiple hops. They also introduce a new ad hoc routing protocol to reflect the availability of the 3G network. The authors evaluate the performance of their scheme through simulations. They find that the aggregate throughput on downlink (from base station to mobile station) can be increased up to 60%. Lee et al. [LBB04] present a multi-hop architecture based on IEEE 802.11, which is interoperable with existing single-hop wireless LANs. To evaluate their proposal, they perform measurements in real-life scenarios with up to three hops towards the base station using a single channel. They also perform simulations for two hop scenarios using more channels. The authors conclude, that nodes with multi-hop extensions as well as nodes without these extensions benefit from their architecture, because the throughput per node is increased.

2.7.2 Capacity of Multi-hop Cellular Networks

The per user throughput of a network is a strong indicator for the overall network performance. The aggregated throughput is also referred to as capacity. Gupta and Kumar first studied the capacity for mobile ad hoc networks in [GK00] and found the per user throughput to be $\Theta(W/\sqrt{n \log n})$, where W is the data rate for each wireless node and n is the number of nodes in the network.

Based on the pioneering work in mobile ad hoc networks, Liu et al. [LLT03] investigate the aggregate throughput capacity of hybrid wireless networks. The authors consider a model with base stations placed on a regular grid and randomly distributed mobile stations. They use two different routing strategies and study the scaling behaviour of such networks. Their analysis shows that for n nodes and m base stations to achieve a considerable capacity gain the number of base stations should be at least $\sqrt{n}$. Kozat and Tassiulas [KT03] extend the model proposed in [LLT03] by distributing both, base and mobile stations randomly. They show that a per source node capacity of $\Theta(W/\log n)$ can be achieved. Zemlianov and de Veciana [ZdV05] build on work from [LLT03] and [KT03] to specify the per user throughput. They also use a model where mobile and base stations are randomly distributed. However, they allow base stations to adjust their transmission range. The authors show that the number of deployed base stations m must exceed $\sqrt{n/\log n}$ for the users to effectively share the spatially distributed infrastructure.

2.8 Challenges of Multi-hop Cellular Networks

The multi-hop cellular network architecture not only brings benefits, but also introduces challenges. The allowance of connections to pass via multiple, mobile and individual hops requires special care. Challenges arise in the area of interaction among existing protocols, mobility management, power consumption as well as security and cooperation. Most of these challenges are known from mobile ad hoc networks, where a majority of research has been conducted in the area of routing.

2.8.1 Weaknesses of Existing Technologies and Protocols

Some limitations arise directly from the multi-hop paradigm applied to current technology and protocols. The prevalent wireless technology for mobile ad hoc networks is IEEE 802.11, with nodes having only one wireless interface. The base stations in a multi-hop cellular network are unable to coordinate the access to the medium for all mobile stations, since only a certain number of mobile stations is within communication range. When using IEEE 802.11 DCF, the base station competes like any other mobile station for access to the medium. Also, the permanent unscheduled access to the medium leads to an increased collision probability, especially under high mobile station density.

For connection-oriented communication, the standard protocols in the Internet, TCP/IP are usually taken. However, TCP is unable to handle the effects of mobility, such as packet loss, correctly. Being designed for the (nearly) loss free cable medium, it interprets packet loss as sign of congestion and therefore starts its congestion control mechanism. This decreases throughput even more.

Based on these preferences most evaluations find that the throughput of multi-hop cellular networks decreases exponentially with the number of hops. A good solution requires coordination among the different layers, which had been initially introduced to support better abstraction and separation of tasks and functionalities in the protocol stack. Anastasi et al. [ACG04] give a detailed overview on the issues of mobile ad hoc networks based on IEEE 802.11.

Most computer communication protocols are designed according to a layered communication model, e.g. the ISO/OSI model [ISO94]. Such a model helps to abstract and separate the functionality and the information flow among the layers. However, this model assumes stationary communication partners and is therefore not applicable to a mobile environment. In a mobile environment more information exchange is required, e.g. to notify higher layers about changes at lower layers. One approach in this direction is to design protocols with cross-layer communication in mind. Conti et al. [CMTG04] describe an approach to cross-layering, where protocols from different layers commonly share network status information, yet operate independently on their respective layer.

2.8.2 Mobility Management

The mobility of the nodes in multi-hop cellular networks raises two issues. One is how to locate a node in such a network. The other is how to keep the location information up to date. Therefore, a node requires a unique identification and some means to propagate and retrieve location information of nodes. A centralized solution is available with Mobile IP [PRP02]. The presence of a base station is a clear advantage over mobile ad hoc networks. However, the multi-hop connections over mobile nodes limit the scalability of a centralized mobility management scheme. Mauve et al. [MWH01] describe some location service schemes in the context of position-based routing in mobile ad hoc networks. Mobility management in multi-hop cellular networks has not received many attention, but with the increasing popularity of wireless mesh networks, this is expected to change.

2.8.3 Power Consumption

A multi-hop cellular network attributes base station functionality (forwarding) to mobile stations and thereby also increases the power consumptions on the mobile stations. Because mobile stations run on battery power, their resources are limited. Although the multi-hop communication paradigm decreases the energy consumption for a single transmission due to the reduced communication distance compared to infrastructure-based networks, the overall number of transmissions increases due to the forwarding of other nodes' packets. Another issue is that nodes close to the base station will likely have to forward more packets towards and from the base station and thus experience a faster battery depletion. Numerous proposals exist for energy efficiency ranging from protocols to overall system design. Jones et al. [JSAC01] wrote an extensive survey about this research area.

2.8.4 Security

The vulnerabilities in multi-hop cellular networks are numerous. The wireless medium allows for passive attacks, e.g. sniffing of information. This information can then be used by an adversary to perform an active attack. Due to the multi-hop communication, an intermediate node can drop packets instead of forwarding them. An adversary can also attack the management protocols (routing, cooperation) of the multi-hop cellular network, either provoking a disruption or a malfunction of the provided services.

A major source for the security problems lies in the lack of a reliable authentication of nodes. Although, base stations are available in multi-hop cellular networks, many nodes do not have a direct (single-hop) connection to them. In a communication session, it is thus necessary to authenticate all nodes on the path within the multi-hop cellular network. However, with increasing node mobility the establishment and maintenance of security sessions between nodes and the base station does not scale. Depending on the scenario the security issues in multi-hop cellular networks are closely related to the ones in mobile ad hoc networks. The report from Buttyán and Hubaux [BH03a] shows approaches in this research area. Sanzgiri et al. [SLD$^+$05] propose a protocol called authenticated routing for ad hoc networks, which is based on public key cryptography and allows secure routing in managed and open environments, where not all participants need to be authenticated in order to participate.

2.8.5 Cooperation

Behind each node in a civilian (public or commercial) multi-hop cellular network stands an individual. There are several reasons for a node to deny cooperation and refrain from forwarding other nodes' packets. Forwarding packets occupies transmission time, which the node can not use for transmitting its own packets. Transmitting packets consumes battery power, which is an exhaustible resource on mobile devices. However, with uncooperative nodes communication over multiple hops becomes impossible as no packets get forwarded and the multi-hop cellular network ceases to exist. Therefore, cooperation is one of the key factors in civilian multi-hop cellular networks.

The solution is to effectuate the cooperation of nodes either by punishing non-cooperative behaviour or by rewarding cooperative behaviour. In Chapter 3 we discuss the challenge of cooperation and accounting in multi-hop cellular networks in more detail.

2.9 Conclusion

Multi-hop cellular networks have some distinguishing characteristics which make them attractive for commercial or public wireless Internet service providers as well as customers. The provider can benefit from low deployment costs, flexible adaptation to the demand in connectivity, better spatial reuse of frequencies as well as new network planning indicators. In addition, the user or customer can receive connectivity outside hot spot areas.

Nonetheless, multi-hop cellular networks - as mobile ad hoc networks - still have open issues, which hinder their wide-spread use. The wireless research community tries to address the challenges in all areas. While the focus on mobile ad hoc networks has turned in favor to wireless sensor networks, the upcoming applications in ubiquitous broadband wireless access lead to increasing research in the area of wireless mesh networks. Both architectures, wireless sensor and wireless mesh networks, include communication over multiple hops and the combination of infrastructureless and infrastructure-based networks, which relates them closely to multi-hop cellular networks.

The improvements in radio technology, such as adaptive antennas as well as the cross-layering paradigm for protocol design are very promising and will most likely weaken some of the issues and make multi-hop wireless communication more attractive. A majority of research so far focussed on physical and network issues. But with the civilian use of multi-hop cellular networks, the individuality of nodes brings up the question of cooperation among nodes. In our work we focus on this challenge of cooperation in multi-hop cellular networks and we explain our view on the subject in the next chapter.

Chapter 3

Cooperation in Multi-Hop Networks

3.1 Introduction

Cooperation in multi-hop networks is another term for describing the participation of nodes in the packet forwarding process. It is important because without cooperation among nodes, the network can not function. If all nodes only transmit their own packets, but never forward packets from other nodes, the nodes stay largely disconnected - except when the destination is a direct neighbor. Depending on the application scenario, cooperation among nodes can not be taken for granted. The research community in the wireless network area has studied this challenge for 5 years now. Buttyán and Hubaux [BH00] as well as Marti et al. [MGLB00] were the first to present cooperation work and several approaches and concepts followed. However, their assumptions and requirements on architectures and security render most of them suitable only for specific scenarios.

In this chapter we describe the challenge of cooperation in multi-hop networks in detail. We begin with a motivation as to why cooperation is necessary and why it has to be ensured. Then, we illustrate the two possible approaches to effectuate cooperation. We present the related work in detail and give a comparison based on the key characteristics of the presented schemes and architectures. We conclude with a summary on the current state of the cooperation in multi-hop networks.

3.2 Motivation

Cooperation among nodes is vital in multi-hop networks. Without nodes forwarding other nodes packets, communication over multiple hops is impossible and the nodes remain disconnected. Thus, a constant contribution from all participants of a multi-hop network is necessary to keep the nodes connected and thereby the network operational.

Considering the military origin of multi-hop networks, cooperation among nodes is not an issue in the corresponding application scenarios. This is true for all scenarios, where nodes are under control of a single authority and the multi-hop network is established for the purpose of the application. Example scenarios include military operations and disaster recovery.

In scenarios without any single authority, cooperation among nodes is not obvious. A single authority prescribes the behaviour for all nodes respecting this authority. Thus, the single authority can ensure cooperation. When each user of a node is her own authority, she can decide by herself what to do. This individual freedom of each user leads to selfishness. Helping other users by forwarding their packets results in the consumption of the own node's limited resources, such as processing and transmission time as well as battery power. Regarding the resource consumption, it is better for a node owner to be uncooperative, because he can save the resources for his own transmissions. When applying this attitude to all nodes in a multi-hop network, no forwarding takes place and communication over multiple hops becomes impossible. Although a common goal in connectivity among the nodes might exist, the necessity of cooperation to achieve that goal is difficult to comprehend by individual users. Especially, when the communication partner is located outside the current multi-hop cellular network, e.g. in the Internet, the benefit of helping neighbors is not apparent.

Therefore, the cooperation in non-single authority application scenarios must be effectuated by additional measures. The challenge of achieving cooperation in multi-hop networks lies in the management of cross-layer information flows and the coordination of actions on different layers. Cooperation clearly requires cross-layer protocol design and is tightly connected to security.

3.3 Cooperation Approaches

Cooperation in multi-hop networks can be looked at from two sides, the network and the user/node perspective.

From the network perspective, the nodes have to cooperate because they act as the backbone infrastructure. If they do not cooperate, the communication over multiple nodes becomes impossible. Thus, any uncooperative node harms the network and poses a threat to the network's correct functioning. Often, an uncooperative node is considered as a security threat, because it reduces the number of available communication paths and thereby the overall connectivity in the network. The consequence is, that cooperation must be enforced by all possible means.

In the cooperation enforcement schemes, uncooperative nodes get punished so severely, that they have no other choice but to cooperate. The underlying assumption is that all nodes are always able to cooperate. So, uncooperativeness is just a sign of bad behaviour and must be corrected using appropriate measures.

However, this assumption ignores situations, where a node may not be able to cooperate at all, even if it wants to. This includes nodes running on very low

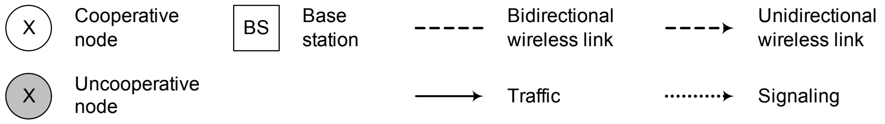


Figure 3.1: Notation for the operation figures of the cooperation schemes

battery power, nodes located at border areas with few packets to forward or nodes with a full buffer. A node might be located at a congestive point in the network and it might not be able to process all packets in time, thus the queue fills up and packets get dropped. Another problem arises in the determination of the cooperativeness of a node. In enforcement approaches it is common to perform some kind of neighborhood watch, that means each node is monitored and evaluated by its neighbors. Therefore, the enforcement approaches are also called *detection-based* schemes. The surveillance results are then used to optimize the operation of the network.

From the user perspective, cooperation is costly, because it consumes resources such as processing and transmission time as well as battery power. It is not obvious for a user, to allow her node to forward other users' packets. To make up for this loss in resources caused by cooperation, the nodes should obtain some kind of reward. Thus, cooperation must be encouraged by giving an incentive to the user.

Encouraging cooperation in multi-hop networks is based on the assumption, that nodes may be reluctant or unable to cooperate. Reasons for uncooperative behaviour include the avoidance of additional costs imposed on a user/node or the inability caused by the state of the node or the network, e.g. congestion. To make up for the additional costs of cooperation, the user should be compensated. This compensation should be high enough to overcome the user's reluctance and make cooperation attractive. Also, in case of the inability to cooperate, nodes do not get punished.

Due to the usage of incentives to encourage cooperation, an additional valuable good is introduced into the architecture. Therefore, the encouragement approaches are also called *motivation-based* schemes. Besides the connectivity, the chosen incentives must be protected from misuse. This requires security measures beyond simple trust relations.

In the following three sections, we describe the related work. We start with the enforcement approaches and continue with the encouragement approaches in chronological order (publication date). Figure 3.1 shows the symbols we used to describe the important operation phases of each scheme. Last, we summarize the work of cooperation principles based on game theory and the position papers in an extra section.

3.4 Related Work on Cooperation by Enforcement

From 2000 to 2002, the majority of the publications on cooperation approached the matter by imposing such severe punishments on selfish nodes, that they have no other choice than to cooperate, if they want to use the network. Usually, the cooperativeness of each node is observed by its neighbors and - in case of selfishness - punished by (partial) exclusion from the network. In their survey on ad hoc networks, Chlamtac et al. [CCL03] dedicate a section to the cooperation area, with a focus on detection-based mechanisms. Buchegger and Le Boudec [BL05] discuss and compare several detection-based cooperation schemes in a recent publication.

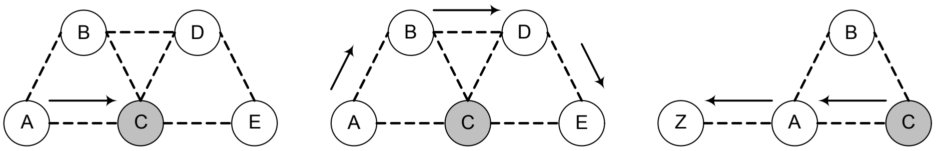
3.4.1 Mitigating Routing Misbehavior

Marti et al. [MGLB00] are the first to introduce detection-based routing protocol enhancements for mobile ad hoc networks. They add observation and circumvention techniques to a routing protocol to avoid uncooperative nodes by choosing another path. In this scheme, cooperation is neither enforced nor encouraged. The authors evaluate their scheme via simulations and find that in the presence of 40% selfish nodes, the overall network throughput increases by 17% compared to a network without a detection-based routing protocol.

Assumptions: The authors assume that neighbor nodes can overhear the communication of each other, i.e. that each node's network interface card operates in promiscuous mode. The authors also require the source routing protocol DSR (see Chapter 2, Section 2.5.1 on page 19).

Operation: Each node runs two programs: a watchdog to identify misbehaving nodes and a path rater to support the routing protocol in avoiding these nodes. The watchdog keeps a copy of every transmitted packet and compares every overheard transmission to it until the packet matches or a timeout is reached. In case of a timeout, a failure counter for the neighbor node that should have forwarded the packet is increased. The path rater also rates nodes, where the initial value is neutral (0.5) and misbehaving nodes are rated negatively based on the averaged failure counters on the nodes. The path rater tries to find the most reliable route by choosing the one with the highest sum of node ratings. Thus, an increasing failure counter leads to the circumvention of the misbehaving node, however the node is still able to transmit its own packets. Figure 3.2 illustrates the avoidance caused by uncooperative behaviour.

Discussion: The scheme has some critical issues. The authors do not consider the problem of node identification and trust among nodes. Thus, false accusations are easily possible. Also, there is neither a disadvantage for an uncooperative node nor an advantage for a cooperative one. This stimulates uncooperative behaviour, as the node has an advantages, when it does not forward other nodes' packets and thereby saves its own resources.



(a) C drops A's packets destined for E

(b) A avoids C and uses another route to reach E

(c) C can still send its own packets to Z via A

Figure 3.2: Mitigating Routing Misbehavior operation

3.4.2 CONFIDANT

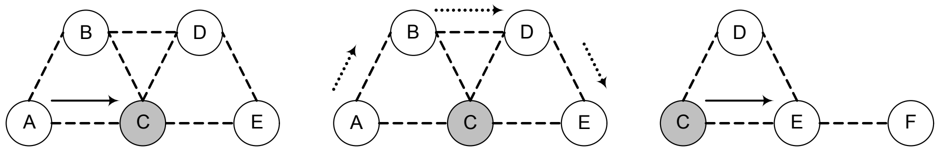
Buchegger and Le Boudec [BL02] propose a concept, which enforces cooperation among nodes in mobile ad hoc networks. They add observation, detection and reaction mechanisms to a routing protocol to exclude uncooperative nodes from the network. The security architecture is based on a distributed trust manager running on each node. The authors evaluate their concept via simulations and find that it can support up to 60% selfish nodes and still perform like a network without selfish nodes.

Assumptions: The authors assume that each node is authenticated and that no identities can be forged, i.e. some tamper resistant hardware is used. The authors also require the source routing protocol DSR.

Operation: Each node runs a monitor program, which observes all one-hop neighbors. Upon detection of a potential misbehavior event, the information is given to the node's reputation system, which decides about its significance. If the event is classified as misbehavior, the rating of the originator node is updated accordingly. When a node's rating exceeds a certain threshold, all paths containing the accused node are deleted from the route cache and an alert message is sent to potentially interested nodes, e.g. the source of the route. The alert message is processed by each node along its path and the credibility of its sender is evaluated with the help of the trust manager. An increasing bad rating leads to the isolation from the network, i.e. the misbehaving node is neither given any packets to forward nor are its own packets forwarded. Figure 3.3 depicts the exclusion due to uncooperative behaviour.

The scheme addresses misbehavior by the threat of punishment, that is the exclusion from the network. The reputation system maintains a list of all nodes and their negative ratings. The reputation system rates all events according to the characteristics of the source, e.g. location and trust. The trust manager maintains a list of nodes and indicates their trust level. The incoming alert messages are rated according to the trust of their generator. The trust manager also has a list of trusted nodes, which a node can alert about detected misbehaving nodes.

Discussion: The idea to use the social attribute of reputation for forwarding decisions is tempting. However, the weaknesses in this concept clearly lie in the trust manager and the reputation system. How to attribute and measure trust in a mobile ad hoc network with individual nodes is not an easy task. Another problem is the



(a) C drops A's packets destined for E

(b) A propagates C's misbehavior to B, D and E

(c) A, B, D and E do not forward C's packets anymore

Figure 3.3: CONFIDANT operation

correct detection of misbehavior by the reputation system. In a mobile ad hoc network the reason for not forwarding a packet can have other causes than selfishness, e.g. a link break. In addition, false accusations from colluding nodes seem possible. Although the authentication of nodes is assumed, it does not prevent them from changing their identities to circumvent the punishment. Also, the reintegration of nodes, which become cooperative after they have been uncooperative, is left open. Buchegger and Le Boudec [BL03] extended the protocol to weaken the effect of the alert messages. To result in a bad reputation, the alert messages have to be recent and sent by a considerable number of friends.

3.4.3 CORE

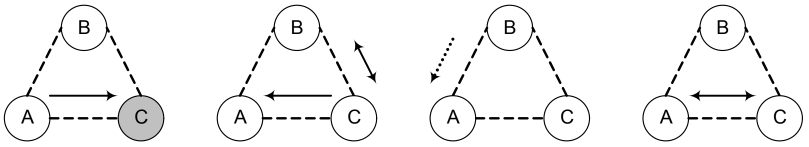
Michiardi and Molva [MM02] study mechanisms to enforce node cooperation in mobile ad hoc networks. They use a watchdog and a reputation mechanism to keep track of the node's cooperativeness. Uncooperative behaviour is punished locally, cooperative behaviour is propagated globally.

Assumptions: The authors assume that each node's network interface card operates in promiscuous mode, so that neighbors can overhear their communication. The authors also base their model on the source routing protocol DSR.

Each node has a watchdog and a reputation table. The authors distinguish between different types of reputation (subjective, indirect and functional) to reflect the information source, which has been used to calculated the reputation. Nodes are seen as service requesters and providers.

Operation: When a node requests a service from another node, it activates the watchdog and specifies the expected result. The watchdog monitors the provider node and evaluates the result. If the node provides the service as expected, a positive report is propagated through the network (indirect). If not, a negative rating is entered in the local reputation table (subjective), which results in a denial of service to the previously misbehaving node. Negative ratings are only performed locally at the monitoring node and not propagated throughout the network. A misbehaving node has the possibility to repent by providing service to other nodes, by which it has not been rated negative yet. Figure 3.4 shows the effect of uncooperative and cooperative behaviour.

Discussion: The exclusive distribution of positive information protects against misuse. A weakness of the model seems to be the high computation and com-



(a) Failed request from A to C, A excludes C

(b) A denies C's request, successful exchange between B and C

(c) B propagates C's behaviour, A includes C

(d) Successful request between C and A

Figure 3.4: CORE operation

munication overhead, as each successful request results in the adjustment of the reputation table and in propagating the success. Each unsuccessful request also results in an adjustment. How to identify nodes and to trust the propagated messages in such an environment is also not obvious.

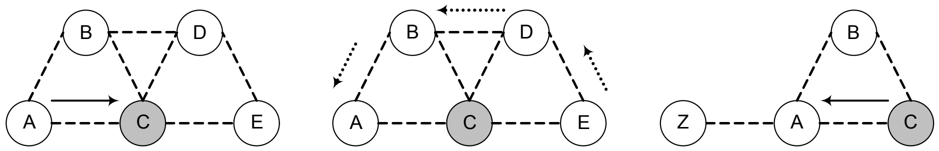
3.4.4 Context Aware Detection of Selfish Nodes

Paul und Westhoff [PW02] specify a cooperation enforcement framework for the source routing protocol DSR in mobile ad hoc networks. They introduce neighborhood watch, accusation messages and context-aware inference to exclude uncooperative nodes from the network.

Assumptions: The authors assume previously established pairwise shared secrets between all nodes participating in the accusation and inference process. They assume the network interface card of each node to operate in promiscuous mode and that the identity of a node can not change, i.e. its IP and MAC addresses are protected by tamper resistant hardware. They require the source routing protocol DSR to be used.

Operation: The protocol particularly secures the route discovery and maintenance phases. When a source transmits a route request, it adds a hash on its own identity, the identity of the destination and the shared secret. Each intermediate node computes a new hash over the old hash and its identifier. The destination recomputes the hash and compares it to the received one. To detect attacks on the route discovery process, the neighbor nodes monitor the progress of the route request messages and compare them to previous route requests. Upon detection of a misbehaving node, the accusing node sends the observed change, its own identity and the identity of the accused attacker to the source of the route request. If the number of accusations exceeds a threshold the accused node is blacklisted on the source node. If only one accusation is received, the source node blacklists the accuser. Figure 3.5 illustrates the protocols functioning.

Discussion: The reporting to the source, minimizes the communication overhead and the threshold for accusations reduces the probability of false exclusion. The assumption of previously existing trust relations and pairwise shared secrets between participating nodes is difficult to realize in practice. The possibility of be-



(a) C drops A's packets destined for E

(b) B, D and E propagate C' behaviour to A

(c) A does not forward C's packets destined for Z

Figure 3.5: Context Aware Detection of Selfish Nodes operation

ing punished if the own accusation is the only one arriving at the source makes reporting less attractive. No possibility for the reintegration of uncooperative nodes, which become cooperative again, is given.

3.4.5 Self-Organized Network Layer Security

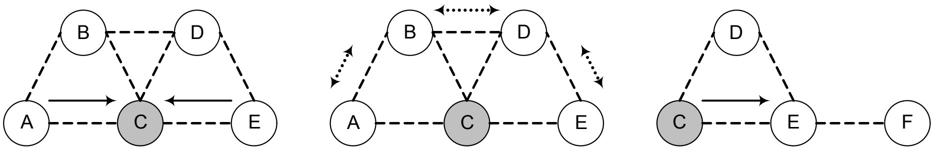
Yang et al. [YML02] propose a network layer security architecture, which also enforces cooperation among nodes in mobile ad hoc network. They introduce a token for each node, which allows the node to take part in the network. The nodes monitor each other and misbehaving nodes can not renew their token, resulting in their exclusion from the network. The security architecture is based on decentralized public-key cryptography, with a single public-/private-key pair.

Assumptions: The authors assume limited collaboration among attackers. The authors also require each node to be uniquely identifiable and each node's network interface card to support the promiscuous mode.

In an initial phase, each node receives a token signed with the private key. The token is bound to the node and expires over time. If the token expires, the node must obtain a new one with the help of its neighbors. The private key is shared by a polynomial among the nodes and only with a certain amount of neighbors, a valid token can be created. Each node also maintains a token revocation list, where the tokens of misbehaving nodes can be listed.

Operation: When a node wants to take part in the network it has to present its token. If the token has not expired and is not blacklisted, the node is allowed to participate, e.g. forward other's or transmit own packets. Each node monitors its neighbors. Upon detection of misbehavior, each node broadcasts alert messages in three phases. Exceeding the thresholds on all three phases leads to the revocation of the accused node's token. A node with a revoked token is permanently excluded from the network. Figure 3.6 describes the main phases of the scheme.

Discussion: The token renewal from neighbors functions like some kind of social group pressure, which works in case of pre-existing social links among the group members. The scheme's 3-phased broadcasting of alert messages places a considerable communication overhead on the network. The requirement for a certain number of neighbors to be available when issuing a new token seems vulnerable to misuse and difficult to achieve in practise.



(a) C drops A's and E's packets (b) A and E propagate C's behaviour (c) A, B, D and E do not forward C's packets anymore

Figure 3.6: Self-Organized Network Layer Security operation

3.4.6 CineMA

Frank et al. [FMP04] describe a cooperation enforcement scheme for mobile ad hoc networks. They introduce a watchdog module to monitor the neighbor nodes, a reputation system to rate their cooperativeness and an interface queue to punish detected selfish nodes. Trusted nodes form a group in which reputation information is exchanged and by which punishment is executed, i.e. reducing the throughput of the accused node according to its selfishness. The authors evaluate their scheme using simulations. They find that for a detection rate of 80% and a transmission range of 250 m 8 enhanced nodes per km^2 are required. For a transmission range of 150 m 17 nodes per km^2 are necessary.

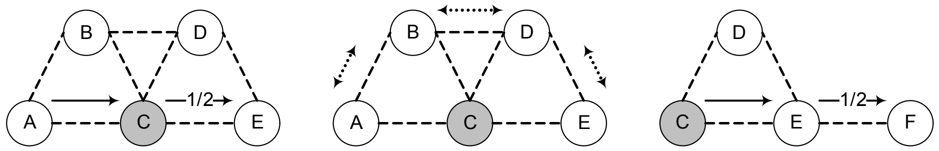
Assumptions: The authors assume that nodes initially form a group and establish a common secret to protect future communication among the group members. They require each group member's network interface card to operate in promiscuous mode and a source routing protocol, e.g. DSR.

Operation: A group node maintains two lists for each neighbor node, one for incoming, the other for forwarded packets. The cooperation level of a node is determined according to the ratio of incoming and forwarded packets. An aggregated version of the lists is exchanged among group members. Each group member has an interface queue, which drops packets of selfish nodes according to their level of cooperation. Figure 3.7 depicts the main operational phases of the scheme.

Discussion: The constant monitoring of a nodes performance allows the reintegration of previously uncooperative nodes, which become cooperative again, in the network. The grouping concepts allows the coexistence with nodes not operating under the scheme. Grouping in real life requires some kind of previously existing social links, e.g. students in the same semester. However, in a lot of real life situation this can not be assumed. The authors also do not address how the group communication is to be secured nor how malicious group members are to be treated.

3.5 Related Work on Cooperation by Encouragement

Beginning of 2003, the majority of publications in the area of cooperation focussed on motivation-based mechanisms. In these architectures, the cooperativeness of



(a) C drops half of A's packets destined to E

(b) A, B, D and E exchange cooperation level of C

(c) E forwards only half of C's packets destined to F

Figure 3.7: CineMA operation

each node is traced with the help of reports and each node is remunerated according to its contribution by some virtual currency.

3.5.1 Nuglet

Buttyán and Hubaux [BH00] present a scheme to ensure cooperation among nodes in mobile had hoc networks. They introduce a virtual currency called Nuglet, which is used to charge for the transmission of packets and to reward the forwarding process. The security infrastructure is based on public-key cryptography, with additional symmetric-key sessions between each communicating pair of neighbors. They give an analysis of the implementation.

Assumptions: The authors assume a tamper resistant security device, such as a smart card, that manages the nuglets account and stores the cryptographic keys. This smart card is loaded with an initial amount of nuglets. Other assumptions are an existing public-key infrastructure, a slow changing neighborhood, the reliable estimation of the initial amount of nuglets in the packet purse model and the reliable determination of the resell price in the packet trade model.

Operation: The authors describe two charging mechanisms, the packet purse model and the packet trade model. In the packet purse model, the originator of the packet places nuglets inside the packet. The amount is based on the estimated hop count to the destination of the packet. Each intermediate node towards the destination takes some nuglets from the packet it forwards. If an intermediate node finds that the amount of nuglets inside a packet does not cover the node's cost of forwarding, it drops the packet. In the packet trade model, the packet is resold among nodes until it reaches the destination. A node receives the packet from the originator for free and sells it to another interested node. This node tries to resell the packet (at a higher price to reflect its cost of forwarding) to another node. At some point the packet is sold to the destination, which pays the total cost.

The scheme also implements an exclusion mechanisms for nodes, which continuously do not forward packets. In a so called fine counter the number of packets, which have been sent to a neighbor node and not forwarded by this node, is registered. Forwarded packets are acknowledged by the security module of each node. The fine counter is advertized to the respective node, which gets the chance to reduce the counter. After the counter exceeds a certain threshold, no more packets are sent to that node.

Discussion: The introduction of a virtual currency allows fine-grained control over the cooperation among nodes. The drawbacks in this scheme lie in the charging mechanisms. In the packet purse model, the correct estimation of the amount of nuglets required for a packet to reach the destination and not being dropped along the way seems rather difficult. An underestimation wastes resources of all involved nodes (battery power) and the network in general (bandwidth) as the packet will not reach the destination and has to be retransmitted. An overestimation lets nodes run out of nuglets quickly, as the overestimated amount of nuglets is lost. As the overall amount of nuglets in the network decreases, the number of packets being (successfully) transmitted also decreases, which leads to non-functional network. In the packet trade model, the originator of a packet is not charged, but the destination pays the total costs from all the resales. Because the nuglet account balance of the destination is not considered when the packet is generated, the network can become overloaded quickly.

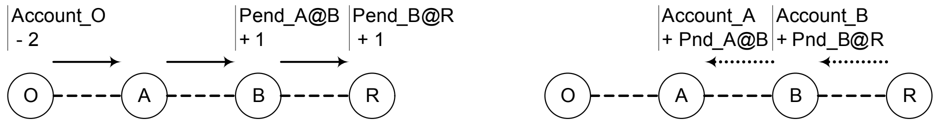
Buttyán and Hubaux [BH03b] proposed a revision of their previous scheme, with a new charging mechanism. They evaluate it using simulations and find that the amount of virtual currency in the network is related to the cooperativeness of the nodes. Instead of sending nuglets along with each packet, each originating node is charged with the estimated number of intermediate nodes to the packet destination. If a node can not afford the transmission, the packet is dropped. The rewarding is now done by the neighbors of a node. A node keeps a pending Nuglet counter for each neighbor node, with which it has established a symmetric-key session. When a node receives a forwarded packet, it increases the pending Nuglet counter of the forwarding neighbor node. The distribution of the pending nuglets is done periodically, via a specific synchronization protocol based on a timer. Nodes, which are not reachable at the time of synchronization lose their pending nuglets. Figure 3.8 shows the charging and rewarding during the transmission of a packet and the synchronization phase.

Discussion: The problems in this proposal lie in the additional network traffic caused by the synchronization protocol and the correct coordination of the synchronization phase itself.

The issues we see in both, the initial and the revised scheme, is that a node can be excluded from the network without the node itself being at fault. A node might not get enough packets to forward from its neighbors, so that it will not earn enough nuglets to transmit its own packets. Also, the complete scheme must run on the smart card. Despite the usage of a virtual currency to stimulate cooperation, both mechanisms really enforce cooperation as there is no alternative for the node. If nodes do not cooperate for whatever reason, they are excluded from the network.

3.5.2 A Micro-Payment Scheme Encouraging Collaboration

Jakobsson et al. [JHB03] are one of the first to encourage cooperation in (asymmetric) multi-hop cellular networks via rewards. The authors use payment tokens to be sent along with the self-generated packets and an accounting center, which is con-



(a) Transmission: counter on node O is decreased by number of intermediate nodes, pending counters are increased by one

(b) Synchronization: counters of node A and B are increased by their respective pending counters

Figure 3.8: Nuglet operation

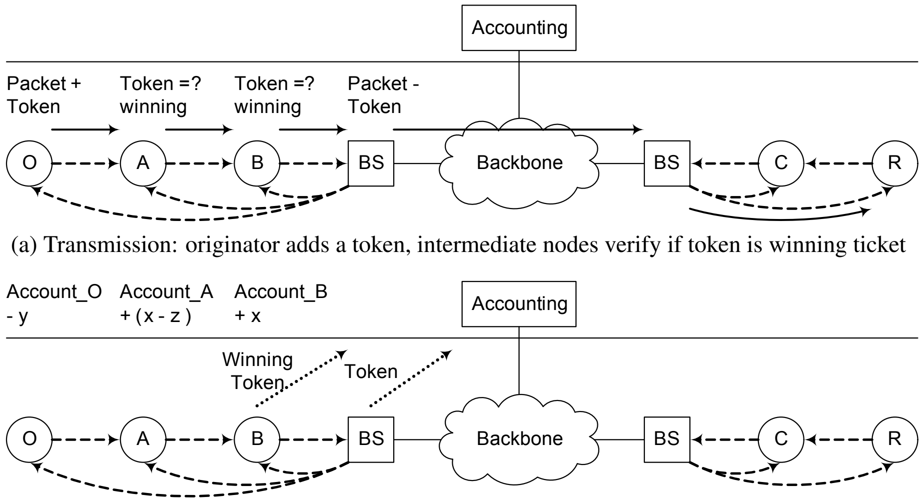
nected to the base station. Rewards are granted similar to a lottery, which is based on the payment tokens as proposed by Micali and Rivest [MR02]. The security architecture is based on secret-key cryptography. The authors give a case-by-case analysis of their security scheme.

Assumptions: The authors assume an asymmetric multi-hop cellular network, that is the path from the node to the base station (up-link) can contain multiple hops, but the path from the base station to the node (down-link) is always single-hop.

Operation: The originator node adds a payment token to its self-generated packet. A token is considered as a lottery ticket for intermediate nodes and their two neighbors on the up- and downwards direction of the path towards the base station. The intermediate node can claim its reward from a base station, which forwards the request to the accounting center. Before claiming a reward, the node itself can apply a function to the payment token, to see if the token is a winning ticket. The base station sends a fraction of the arriving payment tokens to the accounting center. The accounting center processes payment tokens from originators and reward claims from forwarders statistically. Originators are charged on a usage-based fee and forwarders and their two one-hop neighbors on the path towards the base station get rewarded if no cheating behaviour can be detected. The reward is inversely proportional to the frequency of winning tickets to ensure an appropriate remuneration. Figure 3.9 illustrates the transmission and the rewarding phase.

In their concept, the authors address cheating by making it financially unattractive to the nodes. The rewarding of neighbors stimulates the forwarding of payment tokens, which did not win for the current node. However, the main security focus lies in the detection of misbehaving nodes with the help of statistical analysis of tokens from originators, claims from forwarders and user location reports from the base station.

Discussion: The scheme's management traffic is reduced by only selecting a fraction of the payment tokens to be winning. The availability of the single-hop downlink can probably not be guaranteed easily - although it is beneficial, since it eliminates the need for rewarding intermediate nodes on the downlink connections. The detection of cheating behaviour heavily relies on the statistical analysis of tokens and claims, which are generated inside the network. The reliable detection



(a) Transmission: originator adds a token, intermediate nodes verify if token is winning ticket

(b) Rewarding: token and winning tickets are sent to accounting center, which distributes charges and rewards

Figure 3.9: A Micro-Payment Scheme Encouraging Collaboration operation

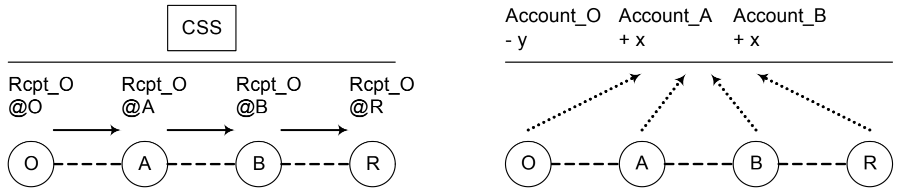
of colluding fraudulent attacks (e.g. generating and passing tickets of intercepted packets towards the base station) seems difficult. The initial request for cooperation sent to a neighbor node before the actual data packet gets transmitted, introduces an additional overhead.

3.5.3 Sprite

Zhong et al. [ZCY03] make one of the first proposals, which uses rewards to encourage cooperation among nodes in mobile ad hoc networks. The authors introduce a virtual currency called Credits and a centralized account management via a Credit Clearance Service for all nodes. Also, here the transmission of self-generated packets is charged and the forwarding of other nodes' packets is remunerated. The security architecture is based on public-key cryptography. The authors use game theory to give a formal model and analysis. They also implement a prototype of their proposal and their evaluation find that the introduced overhead is low.

Assumptions: The authors assume the periodical availability of a fast connection to the Credit Clearance Service for reporting the receipts, e.g. via using another wireless technology. As there is no tamper resistant device, the scheme requires a scalable public-key infrastructure, in particular, scalable certificate management. Also, the sender has to know the full and correct path to the destination, e.g. a secure source routing protocol is assumed.

Operation: When a node transmits a packet, it looses Credits to the network and when it forwards packets, it gains Credits. For each transmission, the Credit Clearance Service balances the accounts of all involved nodes, according to their



(a) Transmission: each node keeps a report for the transmitted packet

(b) Reporting: each node transmits its reports to CSS, which distributes charges and rewards to the respective central accounts

Figure 3.10: SPRITE operation

role in the transmission, e.g. the originator is charged and the forwarders get rewarded. The nodes have the possibility to buy additional Credits from the Credit Clearance Service. Figure 3.10 demonstrates the report generation during a packet transmission and the reporting to and charging at the Credit Clearance Service.

To correctly balance the accounts, the Credit Clearance Service needs to keep track of each transmissions in the mobile ad hoc network. For this purpose, a node generates and keeps a receipt of each forwarded message. Each node periodically transmits the collected receipts to the Credit Clearance Service, which determines the charges and rewards based on all reported receipts.

To prevent nodes from cheating, the authors introduce some security measures in conjunction with the accounting. Sprite only supports sender-based payment to avoid DoS on the receiver. The Credit Clearance Service uses different rewards for cooperative and selfish nodes. For example, the last node towards the destination which received the message but did not forward it, obtains less credit than previous nodes. To prevent colluding attacks with false receipts, the amount charged from the originator and rewarded to the intermediate nodes depends on the successful delivery of a message.

Discussion: The central accounting allows a global view of the nodes involvement in each transaction. The possibility of filling up its own account using real money gives the freedom of choice to the node. The problems of Sprite lie in the centralized accounting and authentication as well as the local collection of receipts on each node. The centralization of the accounting and the authentication run contrary to the scalability in an mobile ad hoc network. Also, the scheme consumes a lot of resources on the nodes, which have to store receipts until they find a connection to the Credit Clearance Service.

3.5.4 Node Cooperation in Hybrid Ad Hoc Networks

Ben Salem et al. [BBHJ03, BBHJ05] also propose a scheme for cooperation encouragement in multi-hop cellular networks. They add traces to the packets to identify the originator, the intermediate nodes and the recipient. The accounting is done by the operator, which maintains the accounts of all nodes. The security architecture is based on secret-key cryptography. The authors give a case-by-case security analysis of their scheme and calculate the communication and computation overhead introduced by their protocol.

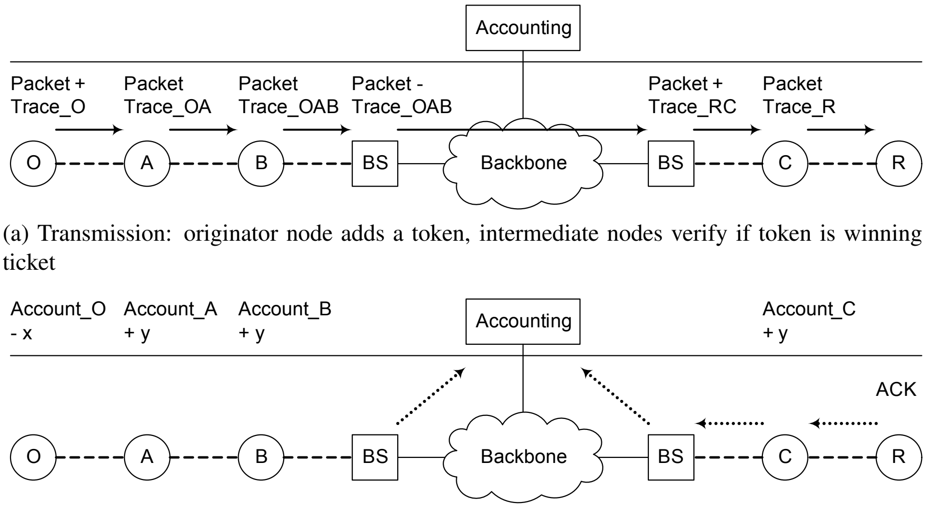
Assumptions: The authors assume all nodes to be registered at the single operator as well as a shared long-term secret key to be established between each node and the operator. All traffic in the multi-hop cellular network is required to pass via the base station. A routing protocol, which delivers the full route between originator and recipient is assumed. The scheme also assumes reduced node mobility for longer lifetime of the route.

Operation: The scheme considers both, up and down-link connections to and from the base stations. The originator located in one multi-hop cellular network establishes an authenticated path to the destination located in another multi-hop cellular network. For this, each node on the path authenticates with its base station using its long-term secret key and obtains a symmetric session key. The originator node creates a message authentication code over the packet using its session key. Further, the node encrypts the packet using a stream cipher with its session key as input. Each intermediate node towards the base station computes and stores a receipt of each received packet. Before retransmission, the intermediate node encrypts the packet using its own session key.

At the up-link base station, all session keys for the nodes on the upstream route of the current packet are retrieved. The base station than recomputes all the stream ciphers to decrypt the original packet and to verify the message authentication code. If the verification is successful, the originator's account is reduced and the intermediate nodes get rewarded. The packet is now transmitted to the down-link base station. Here the base station applies the stream cipher for each intermediate node in advance and sends the packet to the destination, each intermediate node uses its session key to decrypt the packet until it reaches the destination. The destination acknowledges the reception of the packet to the down-link base station. The base station then distributes the rewards. Figure 3.11 shows the transmission and the rewarding phase of the scheme.

The authors address possible attacks with the help of the base stations, which maintain all session information. Also, a deposit on the account of a destination is charged and only refunded if an acknowledgements is received by the base station.

Discussion: The exclusive use of symmetric cryptography reduces the computational overhead considerably. Although the redirection of all traffic via the base station helps to better cope with attacks, it leads to inefficient routes for all communications within the same multi-hop cellular network. Also, the signaling communication with the base station is high, especially the transmission of receipts in case of missing rewards increases the overhead. The requirement on low node mobility compensates the cost of the session establishment but limits the possible application scenarios.



(a) Transmission: originator node adds a token, intermediate nodes verify if token is winning ticket

(b) Rewarding: token and winning tickets are sent to accounting center, which distributes charges and rewards

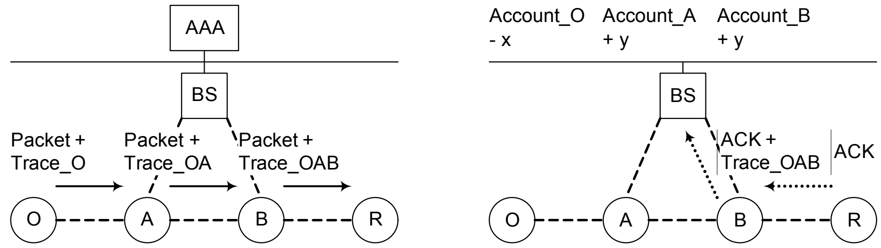
Figure 3.11: Node Cooperation in Hybrid Ad Hoc Networks operation

3.5.5 Charging Support for Ad Hoc Stub Networks

Lamparter et al. [LPW03] describe an architecture to encourage cooperation in multi-hop cellular networks, which they call ad hoc stub networks. They use traces of the originator, the intermediate nodes and the recipient to be sent along with each packet, and which are reported to the base station. The base station decides about the charges and the rewards and transmits the result to an accounting center which maintains the accounts of all nodes. The security architecture is based on public-key and secret-key cryptography. The authors give a formal validation of their proposed charging protocol as well as a case-by-case security analysis of their scheme.

Assumptions: The authors assume an initial mutual authentication between each node and the base station of the multi-hop cellular network and that a symmetric session key has been established. They propose to use a central authentication, authorization and accounting (AAA) infrastructure for this task. Also, their scheme requires a source routing protocol.

Operation: The originator digitally sings the complete path information to the recipient and initializes a keyed hash chain with its session key. This information is sent along with the packet. Each intermediate node verifies the signature and computes a new hash value using the hash value from the packet and its session key. The destination node generates a receipt of the received amount of data, digitally signs it and sends it to the last intermediate node. This node informs the base station about the receipt and the involved forwarding nodes. The base station verifies this information and calculates the charges and rewards. If the keyed hash chain is invalid no intermediate node gets rewarded. Figure 3.12 depicts the transmission



(a) Transmission: originator node adds signature and trace, intermediate nodes leave their traces

(b) Rewarding: recipient acknowledges packet to last intermediate node, which sends a report to AAA via the base station, AAA calculates charges

Figure 3.12: Charging Support for Ad Hoc Stub Networks operation

and the rewarding phase of the scheme.

The authors address the attacks by making it financially unattractive to cheat the system. The base station is expected to detect replay attempts for obtaining more money and the intermediate nodes to prevent forwarding of unauthorized packets.

Discussion: The decentralized and secure tracing of intermediate nodes is more scalable than a centralized solution and reduces the probability of misuse. Problems may arise due to the centralized authentication and accounting. The assumed reliable establishment of a symmetric session between each node and the base station in a multi-hop cellular network is not an easy task, in particular considering the security and scalability. The constant reporting of the last intermediate hops to the base station also introduce additional traffic in the network. Also, the required source routing does not scale well under high node mobility.

3.6 Related Work on Cooperation Principles

Besides the detection and motivation-based cooperation mechanisms, formal models of cooperation based on game theory provide some principles of cooperation in multi-hop networks. Further, position papers on the value of cooperation approaches in general appeared.

3.6.1 Cooperation from the Game Theory Perspective

The cooperation among nodes in multi-hop networks can also be seen as a non-cooperative game, in which the nodes represent the players and their actions are to forward or not to forward other node's packets. Thus, game theory can be used to derive optimal strategies under certain conditions (typically energy constraints) emerging from the Nash equilibrium [Nas50].

Urpi et al. [UBG03] develop a general model which formally describes the characteristics of mobile ad hoc networks. They analyze different cooperation enforcement mechanism from the literature and propose a simple strategy resulting in an equilibrium. This indicates that in their model, cooperation is possible out of a node's self-interest. Srinivasan et al. [SNCR03] obtain similar results. They use an algorithm based on the generous tit-for-tat (GTFT) strategy. GTFT has been the winning strategy to solve the iterated prisoner's dilemma in a tournament initiated by Axelrod [Axe85]. In GTFT each player mimics the action of the other player in the previous game and in addition is also slightly generous. In the case of packet forwarding, a node would occasionally also forward packets from selfish nodes.

Wrona and Mähönen [WM04] propose a dynamic game theoretic model of cooperation based on evolutionary game theory. In this model the network is comprised of selfish nodes and learning nodes, which can dynamically adjust their strategies to maximize their payoff. The authors show that if an ad hoc network implements a reputation mechanism, the majority of the nodes in the network will be cooperative.

Félegyházi et al. [FHB05] investigate whether cooperation can exist in wireless ad hoc networks without incentive mechanisms. They propose a model based on game theory and graph theory to investigate equilibrium conditions for packet forwarding strategies. Their model is the first to considers the network topology. They find that in theory conditions for cooperation out of self-interest exist, but their simulation show that in practice these conditions are almost never satisfied and there will always be nodes which need an incentive to cooperate.

3.6.2 Position Papers on Cooperation

Huang et al. [HCW04] state that incentives for cooperation in mobile ad hoc networks may not be necessary at all, because of the way new technologies and architectures have been adapted in the past. They argue that cooperation schemes only add complexity and thereby hinder the deployment of mobile ad hoc networks. The authors propose to leave out cooperation in the early deployment stages, because only (friendly) early adopters will use the network anyway. At a later stage, when mobile ad hoc networks enter the mainstream market, incentive-based cooperation systems should be deployed. However, these systems should be adaptive to the application and not generic.

The authors' view leaves out some important aspects of the development over the last decades regarding new technologies. A good example is the mainstream adoption of the Internet, which was possible mainly due to its open architecture. However, the initial open design left out security, which would add complexity. Although security was and still is added to the protocols, Internet users suffer from the initial decision to abstract away security issues, e.g. authentication of communication participants. Junk E-mail and distributed denial-of-service attacks are unpopular examples. What we can learn from the development in the Internet is, that it is better to consider security from the beginning than to add it at a later stage (when it may be too late). Cooperation as security are cross-layer issues and should be considered from the start. The author's argument that file-sharing peer-to-peer networks can exist without incentives, omits that the main incentive in file-sharing networks is the mostly copyrighted content, which the user can access for only the flat-rate connectivity cost. The ”charts” of online shared media files from BigChampagne [Big05] illustrate the motivation by content very well. This company measures parts of the peer-to-peer network traffic of numerous file sharing applications. Also, most peer-to-peer networks consist of stationary nodes, which do not have the resource limitations of nodes in mobile ad hoc networks, such as the wireless bandwidth and the battery power supply.

Lamparter et al. [LPW05] investigate the value of cooperation approaches for multi-hop ad hoc networks using statistics. They let each node decide for each packet whether it forwards or drops the packet. The authors assume the forwarding/dropping ratio to be uniformly distributed over all nodes in the network. They analyze to what extent the presence of a cooperation approach (either detection or motivation-based) increases the path reliability and thereby the overall network throughput. The authors find that cooperation approaches have the highest effect on performance in network with a low average hop count (4-5) between originator and destination.

3.7 Comparison of Cooperation Schemes

In Table 3.1 on page 48 we summarize all presented cooperation schemes according to their main characteristics and differences. We distinguish between detection-based approaches, where the nodes monitor their neighborhood and approaches which rely on motivation.

The detection-based schemes are listed in Table 3.1a. They were all designed for mobile ad hoc networks, use a decentralized architecture and require network interface cards to operate in promiscuous mode. All detection-based enforcement approaches also assume some pre-existing trust relations between nodes, which exchange reputation information and unchangeable identities of all nodes, e.g. tamper resistant hardware to ensure the effectiveness of the punishment. We list the general approach to cooperation and by what measures it is ensured. We also show the target of the measures for effectuating cooperation, the possibility of reintegration in the network as well as the type and destination of the propagated information.

The motivation-based schemes are listed in Table 3.1b. All of them introduce a virtual currency and most of them offer the possibility of spending real money to adjust the balance on the virtual account. We list the general approach to cooperation and by what measures it is ensured. We specify the accounting architecture and how a node can proof its cooperativeness. We also indicate, how nodes are authenticated and between what parties a symmetric session is established. Further, we give the number of supported hops on the up- and downlink to the base station, the target network of the scheme and the requirements on the routing protocol. We also indicate the supported node mobility, which we derive from the usage of sessions and the employed routing protocol.

Schemes	Marti [MGLB00]	CONFIDANT [BL02, BL03]	CORE [MM02]	Yang [PW02]	Paul [YML02]	Frank [FMP04]
Cooperation	Neutral	Enforcement	Enforcement	Enforcement	Enforcement	Enforcement
Measures	Avoidance	Exclusion	Exclusion	Exclusion	Exclusion	Throughput reduction
Measure target	Local	Global	Local	Local	Global	Global
Reintegration	-	Impossible	Possible	Impossible	Impossible	Possible
Propagated info	-	Misbehavior	Cooperativeness	Misbehavior	Misbehavior	Misbehavior
Info target	-	Source, friends	Whole network	Source	Whole network	Group members
Routing	DSR	DSR	DSR	DSR	AODV	DSR

(a) Detection-based cooperation schemes

Schemes	Nuglet [BH00, BH03b]	SPRITE [JHB03]	Jakobsson [ZCY03]	Ben Salem [BBHJ03, BBHJ05]	Lamparter [LPW03]
Cooperation	Hybrid	Encouragement	Encouragement	Encouragement	Encouragement
Measures	Reward/Exclusion	Reward	Reward	Reward	Reward
Accounting	Decentralized	Centralized	Centralized	Centralized	Centralized
Cooperation proof	Neighbor session	Payment token	Packet receipt	Hash chain trace	Hash chain trace
Node authentication	Certificate	Shared secret	Certificate	Shared secret	Certificate
Sessions between	Node pairs	-	-	Node and BS	Node and BS
BS up-/down-link # hops	-	Multi/Single	-	Multi/Multi	Multi/Multi
Target network	MANET	MCN	MANET	MCN	MCN
Routing requirements	# intermediate hops	-	source routing	source routing	source routing
Node mobility support	Medium - low	High	Medium	Low	Medium

(b) Motivation-based cooperation schemes

Table 3.1: Cooperation schemes overview

Detection-based approaches built on the user's fear of being punished. The biggest challenges in detection-based cooperation schemes are the secure trust management, reliable node identification and event detection. To date, these issues have not been solved satisfactorily. All schemes are designed for mobile ad hoc networks, while the availability of a provider in a multi-hop cellular networks could at least improve the trust management. Still, the reports on node reputation are vulnerable to misuse and increase the signaling overhead.

Motivation-based approaches built on the user's interest in financial or other type of gain. The challenges in motivation-based schemes lie in the reliable proof of node cooperation and in the protection from misuse of the scheme to increase the reward. Another issue is trade-off between scalability and computational complexity of the security mechanisms.

3.8 Conclusion

Cooperation among nodes in civilian multi-hop networks must be ensured. While the first work on cooperation was detection-based and targeted at mobile ad hoc networks, most of the current work focuses on motivation-based approaches in multi-hop cellular networks. These approaches all have a centralized accounting infrastructure, which means that all signaling traffic (e.g. proof of cooperation and proof of originating) must pass via the base station. This design decision places additional load to the already busy links towards and from the base stations, as the main reason for using multi-hop cellular networks is the network behind the base station, e.g. the Internet. This so called funneling effect is known from wireless sensor networks, where many nodes send packets to few sinks and thereby cause congestion at links close to the sinks, as they share the same wireless medium. Depending on the granularity of the reporting (e.g. per packet), the competition of signaling and data traffic will lead to congestion in multi-hop cellular networks too. Also, the security mechanisms required for centralized accounting only allow low node mobility or are difficult to realize. Thus, existing schemes do not support node mobility well.

We believe, that centralized accounting reduces the advantages and attractiveness of a multi-hop cellular networks, such as the reduced support for node mobility. Therefore, we propose a new original cooperation and accounting strategy for multi-hop cellular networks, which separates the accounting into decentralized and centralized tasks. In the next chapter we present our scheme in more detail.

Chapter 4

CASHnet - A Cooperation and Accounting Strategy in Hybrid Wireless Networks

4.1 Introduction

CASHnet, our cooperation and accounting strategy in hybrid wireless networks, provides a framework for the commercial application of multi-hop cellular networks. CASHnet makes cooperation a rewarding alternative to selfishness, by introducing gratifications for forwarding other nodes' packets. CASHnet also applies costs to the transmission of self-generated packets, which additionally stimulates cooperation in order to cover the expenses. Further, CASHnet allows cost sharing between sender and receiver located in different multi-hop cellular networks, which enables the provider to clearly separate expenses and revenue of each multi-hop cellular network. CASHnet uses highly decentralized accounting mechanisms, and still leaves the provider in control of the cash flow, by the usage of service stations. CASHnet's security mechanisms are based-on public-key cryptography and the sensitive data is managed on smart cards. Last, CASHnet supports the provider in the network planning process, by indicating potential hot and cold spots.

In the remainder of this chapter we explain the motivation for our a work, followed by detailed descriptions of the architecture and operation of the CASHnet framework. We analyze our scheme against possible security attacks. Further, we describe our resale extension to CASHnet and its network management possibilities. Finally, we summarize the characteristics and benefits of our CASHnet scheme.

4.2 Motivation

In the previous two chapters we have presented civilian application scenarios for multi-hop cellular networks and motivated their need for cooperation among the participants of the network. We also showed that cooperation can be effectuated in two ways: enforcement via fear of punishment or encouragement via hope for rewards. We believe that in civilian application scenarios encouragement is the more appropriate measure to ensure cooperation. Especially, when we consider scenarios where no or only few pre-established social links and trust exist among network participants, e.g. the individual customers in todays economic societies.

The first motivation-based cooperation scheme for multi-hop wireless networks was published by Buttyán and Hubaux [BH00]. It is called Nuglet and targeted at mobile ad hoc networks. Nuglet was the main inspiration for our work on cooperation in multi-hop cellular networks. Our analysis of the scheme in Section 3.5.1 on page 38 shows that Nuglet allows for only one way to earn the right for transmission by forwarding packets of other nodes. We suspected, that the limitation to a single source of income may easily lead to exclusions of nodes from the network, without the nodes being uncooperative. This may occur in areas, where nodes have a low number of transmitting or forwarding neighbors. If a node does not receive packets to forward, it can not earn nuglets and thus it can not afford to transmit its own packets. The strict binding between charging and rewarding forces a node to find packets to forward, which is an additional burden for the successful civilian application of wireless networks.

Therefore, we designed a cooperation-based scheme, which gives the node more freedom and reduces the probability of becoming excluded from the network. Also, we focussed on the multi-hop cellular networks architecture first proposed by Hsu and Lin [HL00], which provides a valid cause for a cooperation encouragement scheme in wireless multi-hop network operated by a provider. In Section 2.6 on page 22 and Section 2.8 on page 25 we explain our view on the application scenarios and challenges of multi-hop cellular networks. We designed a promising framework for cooperation encouragement [WB04a] followed by a detailed description of the CASHnet architecture in [WB04b].

By the time other researchers also had proposed motivation-based cooperation mechanisms in multi-hop cellular networks, such as Jakobsson et al. [JHB03], Zhong et al. [ZCY03], Ben Salem et al. [BBHJ03] and Lamparter et al. [LPW03]. These schemes all use centralized accounting and thus require the respective signaling traffic to pass via the base station or via another wireless network technology. This puts additional load on the already stressed links to the base station, as many services (e.g. Internet services) will be provided from outside the multi-hop cellular network. Also, all schemes require source routing, excluding scenarios with higher node mobility.

To evaluate our scheme, we implemented CASHnet in the network simulator and published the results in [WSB04]. We added the Nuglet scheme to the simulator in order to compare it with CASHnet. We found that CASHnet performs better with two or more service stations deployed in the network and showed these results in [WSB05]. The results also helped us to further improve CASHnet. Chapter 5 describes our evaluation process. To better validate our CASHnet scheme, we implemented a prototype under Linux, which we present in Chapter 6. To the best of our knowledge, we are the first to evaluate motivation-based cooperation schemes for multi-hop cellular networks via simulation scenarios using todays most common wireless technologies and protocols. In related work the evaluation is either restricted to theoretical security and performance analysis or leaves out underlying wireless technologies.

4.3 Conception

We developed our CASHnet concept as follows. First, we introduced rewards to the packet forwarding service in order to stimulate the cooperation in multi-hop cellular networks. That means, each node is rewarded if it forwards packets from other nodes. We have chosen a virtual currency as reward called *Helper Credits*, which need to have some value in order to be attractive for the node owner, e.g. the right for transmission of self-generated packets. The transformation of rewards into a valuable good is only possible with the help of the provider. This restriction keeps the provider in control of the cash flow as we show later.

From the network provider perspective we introduce expenses in the form of rewards, which can be used to generate more traffic by the nodes, so we also have to take care for revenue. The cost for the rewards could be covered by the provider of the wireless and backbone infrastructure, by the packet originator or shared among both. We decided to generate revenue where the expenses of rewarding occur. Therefore, we introduced another virtual currency called *Traffic Credits*, which is required for the transmission of a self-generated packet. In case the recipient of the packet is located in another multi-hop cellular network than the originator, the recipient of the packet is also charged. This allows cost sharing between the originator and the recipient of a packet. The separation of the charging and rewarding between originator's and recipient's multi-hop cellular network allows the provider to have better control over the actual network costs and thereby optimize network management and future planning.

Another design decision was where and how to perform accounting. In current multi-hop cellular networks the wireless medium is shared and scarce, i.e. the number of available frequency bands is limited. The signaling traffic of an accounting mechanism competes with the normal data traffic. Considering node mobility and communication over multiple hops, each signaling message imposes a considerable burden on the overall network throughput. Pure centralized accounting has scalability issues and does not support node mobility well. Fully decentralized accounting supports node mobility well, but leaves the provider without control over the cash flow.

To retain the properties of multi-hop wireless networks, we created a hybrid

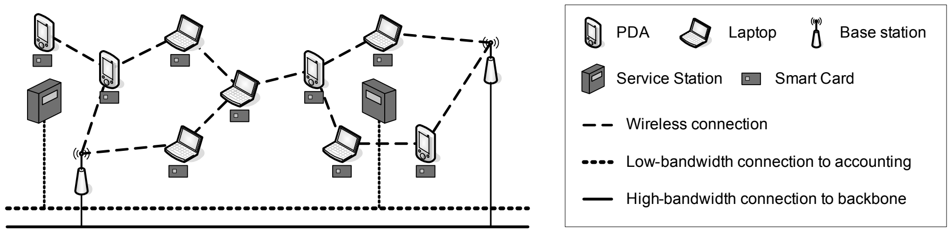

PDA
Laptop
Base station
Service Station
Smart Card
Wireless connection
Low-bandwidth connection to accounting
High-bandwidth connection to backbone

Figure 4.1: CASHnet example scenario

solution, consisting of decentralized metering and charging on the node, decentralized rewarding among nodes and centralized refill and reward exchange at service stations. Thus the provider controls the cash flow, as he can decide about the exchange rate for the rewarding and the costs of the refill. To the best of our knowledge, we are the first to propose such a hybrid accounting scheme in multi-hop networks. We give the provider control over the accounting procedure and reduce the signaling load on the base stations.

4.4 Architecture

Our architecture for the CASHnet concept contains both decentralized and centralized parts. A standard multi-hop cellular network consists of nodes and base stations, which we also call *gateways*. The nodes are usually mobile, the gateways are stationary devices. A node can be a laptop or a personal digital assistant. A gateway can be any kind of base station, e.g. an access point. The gateway provides the interconnection to the provider's backbone network, via which other networks can be reached, e.g. the Internet or other multi-hop cellular networks. For CASHnet, we extend each mobile device with a smart card, which we use to manage all critical information, e.g. the node's identity, cryptographic keys and credit accounts. For the refill and exchange of credits, we introduce service stations. A service station is similar to a low-cost terminal for loading prepaid cards and has a secure, low-bandwidth connection to the provider, which is used for authentication and payment operations. Figure 4.1 depicts an example scenario for CASHnet. It shows a multi-hop cellular network with several mobile nodes equipped with smart cards, two interconnected service stations and two interconnected base stations. The service stations and the base stations are connected to their respective backbone network.

To protect the charging and rewarding process we use digital signatures. Each packet is digitally signed by its originator and the currently forwarding node. Each intermediate node and the recipient verify the signatures. Thus, we can ensure that only packets from CASHnet nodes are transmitted via the network. In addition, we do not charge or reward traffic within the same multi-hop cellular network, because the provider does not have enough control inside the network to effectively prevent misuse. Only traffic passing the gateway can be monitored by the provider. If we do not charge ad-hoc only traffic, this traffic might compete with the traffic leaving the current multi-hop cellular network and thereby degrade the quality of the service. On the one hand, an ad hoc network can always be established in the area of the multi-hop cellular network without the help of a provider. On the other hand, the smart cards offer an additional authentication service of the network participants. We believe that the main attractiveness of a multi-hop cellular network lies in the connection to other networks, such as the Internet and the usage of its services. Further, the provider can cover the expenses of the smart card with a monthly subscription fee. The circumvention of higher hop count related charges by sending packets as ad hoc only traffic does not create a loss for the provider, because no rewards are distributed for the forwarding nodes. In Section 4.6.2 we explain this tunneling attack in more detail.

We require a tamper resistant device, such as a smart card and a corresponding reader in each node. A smart card is a plastic card with embedded memory as well as a microprocessor, which provides basic functions to manipulate information on the card. In addition the smart cards have a cryptographic coprocessor, which embeds cryptographic functions. Many recent laptops come equipped with a smart card reader and all recent devices have an USB interface to connect a small-sized smart card reader, such as the e-gate from Axalto [Axa05]. This device provides a protected environment, where all the critical data is stored. It holds the node's unique identity, the node's public/private key pair, the certificate from the provider, which securely binds the node's identity to the public key as well as the provider's public key. This information can only be changed by the provider at the service station.

Also on the smart card are two accounts - one for each virtual currency, i.e. traffic and helper credits. We assume the interaction between the CASHnet program and the smart card to be secure, e.g. the CASHnet program can safely read from and write to the smart card. In addition, CASHnet has a secured memory area for the critical data structures, e.g. the authenticated nodes and expected reward lists. We also assume, that before a self-generated packet leaves a node, its traffic credits account is charged and the packet's payload is digitally signed. We also require that when a packet arrives at its destination, its traffic credits account is charged. This can be achieved by securely binding the transmission and reception of packets at the network interface card level with the counter on the smart cards. For example, the firmware on the wireless network card would only work if the smart card is present and operational.

Further, we assume the availability of a routing algorithm, which provides the hop count to the gateway (e.g. AODV or DSR) for the dynamic transceiving cost operation mode, where cost is related to the route length. For example, the current AODV implementations supporting gateways provide this information. The fixed transceiving cost operation mode requires no hop count information, as the charges are equal for all nodes in the network. We also assume that a node can determine whether the recipient is inside its current multi-hop cellular network or

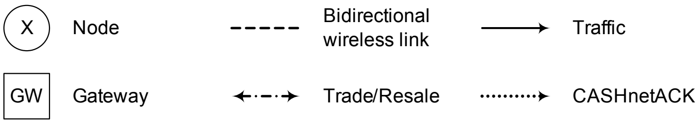


Figure 4.2: Notation for the operation figures of the CASHnet schemes

not. This could be achieved via appropriate addressing mechanisms. We note, that unlike most other motivation-based cooperation schemes, we do not require source routing.

Due to the usage of digital signatures, which are based on public key cryptography, we require sufficient processing power and memory on the node. Laptops and modern PDAs fulfill these requirements.

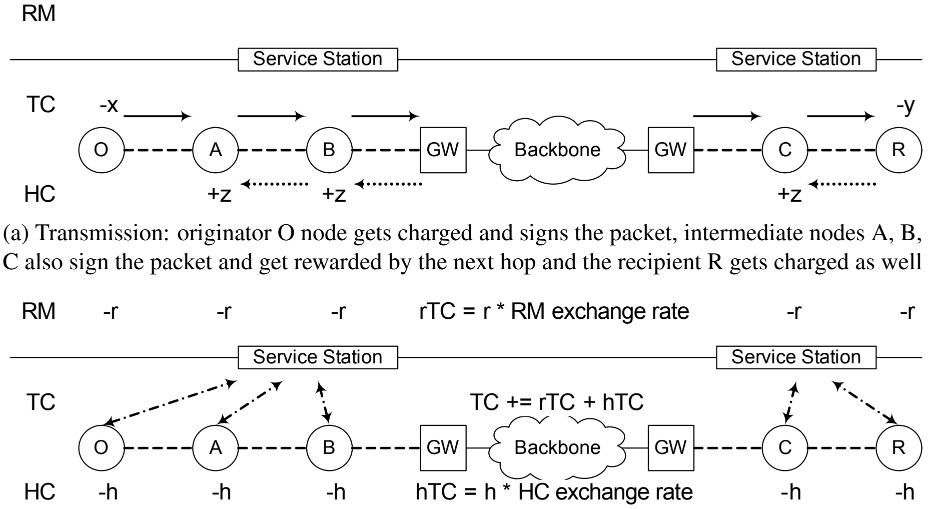
4.5 Operation

Our CASHnet charging and rewarding mechanism works as follows: Every time a node wants to transmit a self-generated packet or receive a packet addressed to it, the node has to pay with traffic credits. The amount is either related to the current distance in hop counts to the gateway or a globally fixed price. Every time a node forwards a packet, it gets helper credits. Traffic credits can be bought for real money or traded for helper credits at service stations. In the following 6 sections we first give an overview on the possible operation modes and then describe each operation phase in detail.

4.5.1 Overview

Figure 4.3 shows the basic operation phases of CASHnet in a scenario, where originator and recipient are located in different multi-hop cellular networks using the notation shown in Figure 4.2. This is the most complete scenario, which also includes cases, where only one of the two communication participants is located in a CASHnet-enabled multi-hop cellular network. Figure 4.3a illustrates the transmission process, where the originator and the recipient are charged and the intermediate forwarding nodes are rewarded. The traffic and helper credits accounts on each node get debited or credited according to the node's role in the transmission process. Figure 4.3b depicts the refill process, in which a node connects to a service station and fills up its traffic credits account by exchanging the helper credits on its smart card or paying with real money. With real money we describe a valid currency in the country of the multi-hop cellular network.

The typical course of action for a customer, who wants to participate in a CASHnet-enabled multi-hop cellular network, consists of five steps: preparation, authentication, transmission/reception & charging, forwarding & rewarding as well as refill. The first and the last step are performed at the service station, where the



(a) Transmission: originator O node gets charged and signs the packet, intermediate nodes A, B, C also sign the packet and get rewarded by the next hop and the recipient R gets charged as well

(b) Refill: nodes connect to a service station to refill their traffic credits, TC account by exchanging real money, RM or helper credits, HC

Figure 4.3: CASHnet operation

customer inserts her smart card. Figure 4.4 illustrates step two to four in a message sequence chart in our example scenario. The numbered gray markers refer to example positions of the actions from the following list.

1. Preparation: The customer obtains the smart card from the provider and loads the traffic credits accounts at the service station.

2. Authentication: Preliminary to the normal communication with a recipient, the originator O sends a certificate advertisement $CADV_O$ to the recipient R. Thereby all intermediate nodes (A, B and C) and the recipient obtain the authentication information of the originator. The recipient in turn replies with a certificate reply $CREP_R$ addressed to O. Now all intermediate nodes obtain the authentication information of the recipient.

3. Transmission / Reception & Charging: Before the transmission of a self-generated packet, the originator's traffic credits account is charged and the packet is digitally signed. Upon reception of a packet destined to the current node, the recipient's traffic credits account is also charged.

4. Forwarding & Rewarding: At the reception of a packet, the node rewards the previous forwarding node in case it was not the originator or a gateway by sending a digitally signed acknowledgement ACK immediately or after receiving several forwarded packets. Receiving an ACK increases the node's helper credits account. The node also removes the digital signature of the previous node and adds its own before forwarding the packet. In addition,

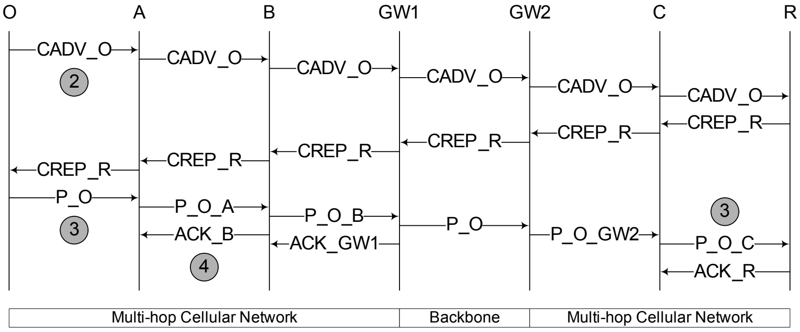


Figure 4.4: CASHnet operation with both originator and recipient in a MCN

the node keeps the digital signature of each forwarded packet in order to validate the $ACKs$.

5. Refill: After some time, the customer goes to a service station in order to refill her traffic credits account by exchanging available helper credits and/or buying traffic credits for real money.

Figure 4.5 describes the payload of the exchanged messages, when originator O sends a packet to the recipient R. The messages towards the recipient have a white background, the messages towards the originator have a gray background. As it can be seen, the originator O adds her identity ID_O as well as a nonce N_O to avoid replay attacks. She then digitally signs the payload. This new originator payload Pld_O is digitally signed between each intermediate hop until it reaches the recipient R. Each hop removes the signature from the previous hop (except the originator's). On the backbone of the provider the originator payload is not singed, because there is no need to identify the forwarding nodes, i.e. routers, on the backbone of the provider. Each intermediate hop rewards the previous hop in the forwarding chain unless it is the originator or a gateway. We explain each message in detail below.

In case one of the communication participants is not located in a CASHnet-enabled multi-hop cellular network, but for example in the Internet, the gateway of the partner in the multi-hop cellular network has to act as proxy, i.e. it takes the role of the recipient for the authentication mechanisms and redirects the normal traffic. Thus, the provider has to know about CASHnet-enabled networks. This can be achieved for example with the help of selected address ranges or additional attributes in location lookup services, which are required in any case to find a node from outside the multi-hop cellular network. Figure 4.6 shows the message sequence chart when the recipient is located in the Internet. Instead of the recipient, the gateway sends a certificate reply to the originator. The originator accepts this reply from the proxy gateway and starts the normal communication. The gateway removes the information added by the CASHnet scheme from the payload, i.e. it sends the inner payload from the originator to the recipient.

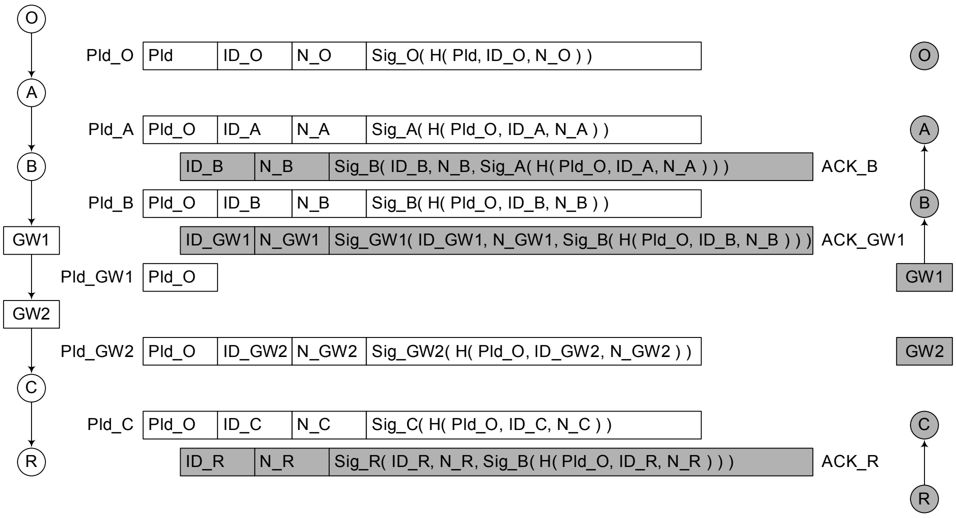

O
Pld_O
Pld
ID_O
N_O
Sig_O(H(Pld, ID_O, N_O))
A
Pld_A
Pld_O
ID_A
N_A
Sig_A(H(Pld_O, ID_A, N_A))
B
ID_B
N_B
Sig_B(ID_B, N_B, Sig_A(H(Pld_O, ID_A, N_A)))
ACK_B
Pld_B
Pld_O
ID_B
N_B
Sig_B(H(Pld_O, ID_B, N_B))
GW1
ID_GW1
N_GW1
Sig_GW1(ID_GW1, N_GW1, Sig_B(H(Pld_O, ID_B, N_B)))
ACK_GW1
Pld_GW1
Pld_O
GW2
Pld_GW2
Pld_O
ID_GW2
N_GW2
Sig_GW2(H(Pld_O, ID_GW2, N_GW2))
C
Pld_C
Pld_O
ID_C
N_C
Sig_C(H(Pld_O, ID_C, N_C))
R
ID_R
N_R
Sig_R(ID_R, N_R, Sig_B(H(Pld_O, ID_R, N_R)))
ACK_R

Figure 4.5: CASHnet operation message payload

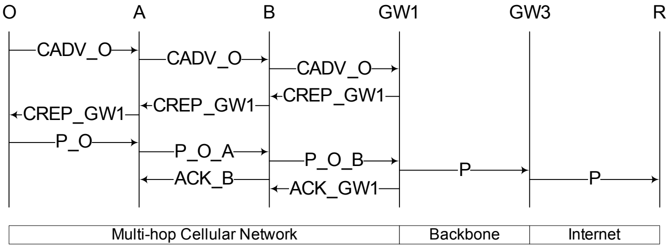


Figure 4.6: CASHnet operation with recipient outside of a Multi-hop Cellular Network

Figure 4.7 illustrates the message sequence chart if the originator is located in the Internet. The gateway sends a certificate advertisement in the name of the originator to the recipient. The recipient answers with a certificate reply. The payload from the originator is then signed by the recipient's gateway and sent to the recipient.

According to Figure 4.8, we distinguish between twelve operation phases in the CASHnet framework and we categorize them into preparation, authentication, charging, rewarding and maintenance. The preparation and the authentication initialize the nodes for the participation in CASHnet. The charging and rewarding handle the accounting for the traffic in the network and the maintenance covers the interaction with the service station, i.e. certificate update and refill of traffic credits. We describe each operation phase in detail as algorithm (see Algorithms 1-12) using the notation listed in Table 4.1.

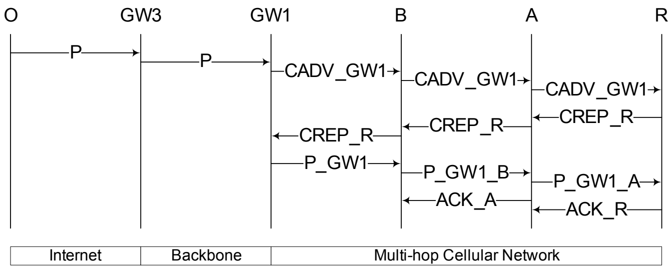


Figure 4.7: CASHnet operation with originator outside of a Multi-hop Cellular Network

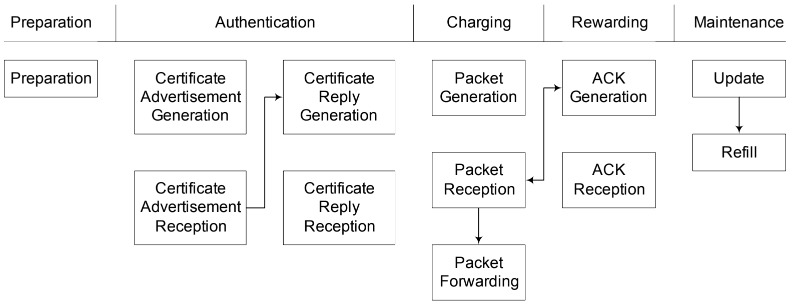


Figure 4.8: CASHnet operation phases

O	Originator	R	Recipient
N	Current node	$N - / + 1$	Previous/next node
$O \rightarrow R$	O sends message to R	$O \leftarrow R$	O receives message from R
Pld_O	Payload generated by O		
ID_O	Unique identifier of O		
K_O/KP_O	Public/private key pair of O		
$Cert_X(O)$	Certificate issued by provider X binding O's identity and public key		
N_O	Nonce chosen by O		
$H(M)$	One-way hash function applied to message M		
$Sig_O(M)$	Digital signature of message M by O		
$\Rightarrow$	leave to another phase and end current phase		
$\rightleftharpoons$	execute other phase and return to current phase		
$\square$	end current phase		

Table 4.1: Notation for the CASHnet operation phases, Algorithms 1-12

4.5.2 Preparation

First, the customer obtains a smart card for her computer from the provider and loads the traffic credits account on her card (see preparation phase, Algorithm 1). The smart cards holds the critical data of the customer, i.e. the unique identifier, the public/private key pair, the certificate from the provider and the provider's public key. The certificate securely binds the identifier and the public key. The certificate

Algorithm 1 Preparation phase

```
1 obtain personal smart card from provider $X$ with an unique identifier $ID_N$, a public/private key pair $K_N/KP_N$, a certificate $Cert_X(O)$ issued by provider $X$ for $N$ and the provider's public key $K_X$
2 load traffic credits account at provider's Service Station by paying with real money
```

can be verified with the provider's public key, i.e. by performing a digital signature verification. Thus, the provider's public key must be inherently trusted. All this critical data can only be modified by the provider at the service station.

In addition, the customer loads the traffic credits account on the smart card using a service station. At first, she can only buy traffic credits for real money. Later, when she has forwarded other node's packets she will be able to trade her helper credits against traffic credits. Both accounts are also stored on the smart card and the provider keeps a copy of the current state for reference when she comes back.

4.5.3 Authentication

In order to perform the charging and rewarding securely, the involved nodes need to be authenticated. The originator of a communication needs to be authenticated to all nodes on the path towards the recipient. In addition, the previous and the next hop of the path need to be authenticated to each intermediate node.

To authenticate itself, a node creates a certificate advertisement, which contains its certificate, a nonce and a digital signature over the hash value of the certificate and the nonce. The nonce protects against replay attacks. The digital signature allows to verify the origin and integrity of the certificate advertisement and is computed by encrypting the hash with the node's private key. The certificate allows to verify the binding between a node's identity and its public key.

We decided, that only the owner of the certificate is allowed to transmit its certificate in order to ensure the freshness of the certificate and in order to avoid the additional complexity resulting from the management of non-authoritative certificate advertisements. In addition, the digital signature allows the tracing of the propagation source of false certificates.

Certificate advertisements are generated upon request, i.e. previous to the start of a communication the originator sends the advertisement to the recipient. In addition, certificate advertisements are sent periodically to one-hop neighbors in order to reduce the number of authentication messages. To complete the mutual authentication, the node waits for the certificate reply from the recipient, which acknowledges the advertisement and indicates an existing route to the recipient (see certificate advertisement generation phase, Algorithm 2).

When a node receives a certificate advertisement, it first verifies the certificate. A certificate typically includes the node identity and its public key as well as an

Algorithm 2 Certificate advertisement generation phase

```
create a payload consisting of a nonce and the certificate of the originator
calculate digital signature over one-way hash from certificate and nonce
append digital signature to payload
transmit certificate advertisement to next hop towards recipient OR broadcast
  it to all one-hop neighbors
repeat
  wait
until certificate reply from recipient $CREP_R$ arrives OR timeout
if $CREP_R$ arrives then
  go to certificate reply reception phase (Algorithm 5) $\Rightarrow$
end if
$O \rightarrow R : CADV_O = Cert_X(O), N_O, Sig_O(H(Cert_X(O), N_O))$
```

Algorithm 3 Certificate advertisement reception phase

```
if certificate for node from $CADV$ valid then
  if signature of node from $CADV$ valid then
    save tuple of identity and public key in authenticated nodes list
    if current node $==$ certificate advertisement recipient then
      go to certificate reply generation phase (Algorithm 4) $\Rightarrow$
    else
      look up next hop towards recipient
      forward certificate advertisement to next hop $\Box$
    end if
  else
    drop $CADV$ $\Box$
  end if
else
  drop $CADV$ $\Box$
end if
$N \leftarrow N-1 : CADV_O = Cert_X(O), N_O, Sig_O(H(Cert_X(O), N_O))$
$N : Tuple_O =< ID_O, K_O >$
$N \rightarrow N+1 : CADV_O$
```

expiration date and information about the issuer of the certificate. It also includes a digital signature over the computed hash value from this information. In CASHnet, we assume the provider to issue certificates. In order to validate a certificate, i.e. verify its digital signature, the provider's public key is required, which we assume to be on the smart card. If the validation fails the certificate advertisement is dropped.

If the validation is successful, the node continues with the verification of the digital signature by decrypting it with the public key of the originator node to retrieve the one-way hash of the originator certificate and the nonce. The node

Algorithm 4 Certificate reply generation phase

```
look up next hop towards originator $O$ which send the certificate reply
create a payload containing a nonce and the certificate of the recipient $R$
calculate digital signature over one-way hash from certificate and nonce
append digital signature to payload
transmit certificate reply to next hop □
$R \rightarrow O : CREP_R = Cert_X(R), N_R, Sig_R(H(Cert_X(R), N_R))$
```

Algorithm 5 Certificate reply reception phase

```
if certificate for node from $CREP$ valid then
  if signature of node from $CREP$ valid then
    save tuple of node identity and public key in authenticated nodes list
    if current node $==$ certificate reply recipient then
      end processing □
    else
      look up next hop towards recipient
      forward certificate reply to next hop □
    end if
  else
    drop $CREP$ □
  end if
else
  drop $CREP$ □
end if
$N \leftarrow N - 1 : CREP_R = Cert_X(R), N_R, Sig_R(H(Cert_X(R), N_R))$
$N : Tuple_R =< ID_R, K_R >$
$N \rightarrow N + 1 : CREP_R$
```

computes the hash over the certificate and the nonce to compare it to the retrieved hash value. If the decryption and comparison is successful, the node knows that the originator has sent the certificate and that it has not been modified during transit. In case the verification is not successful, the certificate advertisement is dropped.

If the verification is successful, the node stores the originator's identity and public key in the authenticated nodes list. If the node is not the destination of the certificate advertisement, it forwards the message to the next hop otherwise it sends a certificate reply to the originator (see certificate advertisement reception phase, Algorithm 3).

In case a node has received a certificate advertisement, it sends back a certificate reply to the originator. The certificate reply contains the recipients certificate and a nonce as well as a digital signature (see certificate reply generation phase, Algorithm 4).

When a node receives a certificate reply, it verifies the certificate and the signature contained in the message. If the verification is unsuccessful the certificate

Algorithm 6 Packet generation phase

```
if packet recipient location == outside of current MCN then
  look up hop count to gateway towards recipient
  calculate transmission fee
  if transmission fee $\leq$ traffic credits account then
    traffic credits account $-$ transmission fee
  else
    drop packet □
  end if
end if
append originator identity and nonce to payload
calculate digital signature over one-way hash from payload, identity and nonce
append digital signature to new payload
transmit packet to next hop towards recipient □
$O \rightarrow R : Pld_O = Pld, ID_O, N_O, Sig_O(H(Pld, ID_O, N_O))$
```

reply is dropped. In case it is successful, the node saves the sender's identity and public key in the authenticated nodes list. If the node is not the destination of the certificate reply, it forwards the message to the next hop otherwise the processing ends (see certificate reply reception phase, Algorithm 5).

4.5.4 Charging

Now that the nodes between originator and recipient are authenticated, the communication between both parties can start. Before transmission, the current transmission fee is calculated. CASHnet supports two type of transceiving costs, dynamic charges related to the hop count from the current node to the gateway or fixed charges equal for each node in the current multi-hop cellular network. The hop count to the gateway can be obtained from any ad hoc routing protocol with gateway discovery support (see Chapter 2, Section 2.5.2 on page 21). If the originator can not afford the transmission, the packet is dropped. Otherwise the transmission fee is debited from the traffic credits account. The originator appends its identity and a nonce to the payload. The nonce helps to detect replay attacks. The originator then applies a one-way hash function to the payload, its identity and the nonce. The resulting hash value is then encrypted (digitally signed) using the originator's private key. Last, the originator adds the digital signature to the extended payload and transmits the packet (see packet generation phase, Algorithm 6). The digital signature allows to validate the origin and the integrity of the payload.

Each intermediate node also appends a proof for its participation (i.e. its identity, nonce and digital signature) to the payload when forwarding a packet to the next node. So, an intermediate node has to remove the identity, nonce and digital signature from the previous node, except if it is the first node towards the destination from the originator's perspective.

Algorithm 7 Packet reception phase

```
if signatures of previous node AND of originator from packet valid then
  if current node $==$ packet recipient AND packet originator location $==$ outside of current MCN then
    look up hop count to gateway towards originator
    calculate reception fee
    if reception fee $\leq$ traffic credits account then
      traffic credits account - reception fee
      deliver packet to recipient's protocol stack
      increase the previous node's packet counter
      if previous node's packet counter $>$ packet counter ACK threshold then
        execute ACK generation phase (Algorithm 9) and end □
      end if
    else
      drop packet
      alert gateway to filter for some time period □
    end if
  end if
  if previous node $\neq$ packet originator AND (packet recipient OR originator location $==$ outside of current MCN) then
    increase the previous node's packet counter
    if previous node's packet counter $>$ packet counter ACK threshold then
      execute ACK generation phase (Algorithm 9) and return $\rightleftharpoons$
    end if
  end if
  go to packet forwarding phase (Algorithm 8) $\Rightarrow$
end if
$N \leftarrow N-1 : Pld_{N-1} = Pld_O, ID_{N-1}, N_{N-1},$
                  $Sig_{N-1}(H(Pld_O, ID_{N-1}, N_{N-1}))$
```

When a node receives a packet, it first verifies the proof of participation from the previous node. It does so by decrypting the digital signature with the public key of the previous node to retrieve the one-way hash of the originator payload, the identity and the nonce. Now the node computes the hash over the originator payload, the identity as well as the nonce and compares it to the retrieved hash value. If the decryption and comparison is successful, the node knows that the originator payload has been forwarded by the previous node and that the payload has not been modified. In case the verification is not successful, the packet is dropped.

Now the node repeats the verification with the originator payload, i.e. it decrypts the digital signature with the public key of the originator, applies the one-way hash function to the inner payload, the originator identity as well as the nonce

Algorithm 8 Packet forwarding phase

```
retrieve encapsulated originator payload $Pld_O$ from current packet
append current node identity and nonce to originator payload
calculate signature over one-way hash from originator payload, identity and nonce
append digital signature to new payload
look up next hop towards recipient
save tuple of digital signature and identity of next hop in expected reward list
transmit packet to next hop □
$N \rightarrow N+1 : Pld_N = Pld_O, ID_N, N_N, Sign_N(H(Pld_O, ID_N, N_N))$ $N : Tuple_N =< Sign_N(H(Pld_O, ID_N, N_N)), ID_{N+1} >$
```

and compares it to the decrypted hash value from the signature. Again, when the decryption and comparison is successful, the node knows that the originator has submitted the inner payload. If the verification fails, the packet is dropped.

If the node is the destination of the packet, it calculates the reception fee for the packet, which can be dynamic based on the hop count to the gateway or a globally fixed cost. In case the node has not enough traffic credits, the packet is dropped and the originator and by that the gateway are notified in order to stop the transmission to the recipient and thereby the needless rewarding of intermediate nodes. The gateway frees a blocked node when packets paid by the node arrive at the gateway or after some time elapses.

When the node can afford the reception, its traffic credits account is debited with the reception fee and the packet is passed to the protocol stack of the node. The node also sends an acknowledgement to the previous node to reward its cooperation. The rewarding can happen immediately per packet (when the packet counter ACK threshold is set to 1) or collectively for several forwarded packets when the packet counter ACK threshold is exceeded (see packet reception phase, Algorithm 7). We describe the rewarding in Section 4.5.5.

In case the node is not the destination, it also sends an acknowledgement to the previous node and prepares to forward the packet. First, the node retrieves the originator payload and adds the proof for its participation (identity, nonce and digital signature). It looks up the next node towards the destination and saves the next node's identity together with the previously computed digital signature into the expected reward list. This list the signature of all forwarded packets and the identity of the corresponding next hop, to which the packet has been forwarded. This information is used to verify the acknowledgements sent by the next hops. Last, the node transmits the packet to the next node (see packet forwarding phase, Algorithm 8).

Algorithm 9 Acknowledgement generation phase

```
create a payload consisting of node identity and nonce
calculate digital signature $Sig_N$ over node identity $ID_N$, nonce $N_N$ and digital signature contained in the payload from the forwarding node $Sig_F$
append digital signature to payload
transmit packet $ACK_N$ to node $F$, which forwarded the packet
$N \rightarrow F : ACK_N = ID_N, N_N, Sig_N(ID_N, N_N,$
                        $Sig_F(H(Pld_O, ID_F, N_F)))$
```

Algorithm 10 Acknowledgement reception phase

```
if digital signature $Sig_N$ from $ACK$ valid then
  retrieve digital signature from rewarding node $N$ contained in the digital signature
  if digital signature $Sig_F$ and identity $ID_F$ as tuple in expected reward list then
    helper credits account + reward
    remove tuple $< Sig_F, ID_F >$ from expected reward list □
  else
    drop $ACK$ □
  end if
else
  drop $ACK$ □
end if
$F \leftarrow N : ACK_N = ID_N, N_N, Sig_N(ID_N, N_N, Sig_F)$
```

4.5.5 Rewarding

The rewarding of cooperative nodes is done on a per-hop basis. That means, each node rewards a node, from which it received a packet - unless this node is the packet originator. The rewarding is either performed for each packet or for a number of packets. The packet counter ACK threshold specifies the granularity of the rewarding and thereby the value of the rewards, i.e. the helper credits. For example if we reward only every fifth packet, the value of a helper credit is five times higher compared to when we reward every single packet. To reward the forwarding node, a node creates a new acknowledgement message, which contains its identity, a nonce as well as a digital signature over its identity, the nonce and the digital signature contained in the payload from the forwarding node. This acknowledgement message is sent to the forwarding node (see acknowledgement generation phase, Algorithm 9).

When a node receives an acknowledgement, it verifies the digital signature by decrypting it with the public key of the node, which sent the acknowledgement. This node is typically a one-hop neighbor and already authenticated, i.e. contained in the authenticated nodes list. If the verification of the acknowledgement is not

Algorithm 11 Maintenance update phase (at the service station)

```
if certificate lifetime expired OR due soon then
    issue a new certificate
end if
go to maintenance refill phase (Algorithm 12) $\Rightarrow$
$ServiceStation : Cert_X(O)$
```

successful, the node drops it. Now, the node retrieves the contained digital signature and the sending node identity and looks for a matching tuple in the expected reward list. If it does not find a matching entry, it drops the acknowledgement. If the node finds one, it adds the reward to the helper credits account and removes the entry from the expected reward list. Because of the removal of the entry, multiple transmissions of the same acknowledgement or the same digital signature of the forwarded packet do not result in additional helper credits (see acknowledgement reception phase, Algorithm 10).

4.5.6 Maintenance

In order to reduce the attractiveness of misuse, the certificates issued to the nodes have a short lifetime. Thus, the customer requires to refresh her certificate regularly at a service station of the provider. When a customer removes the smart card from her computer and enters it in a smart card reader at the service station to refill her traffic credits account, the state of the smart card is checked first. If the certificate is about to expire, a new certificate is issued by the provider to bind the given public key with the node identity (see maintenance update phase, Algorithm 11).

When a node participates in the network, i.e. transmits self-generated packets and forwards other nodes' packets, its traffic credits account empties and its helper credits account fills up over time. To continue participation, the node needs to refill its traffic credits account. In order to ensure the control over the cash flow, the node can only load its traffic credits account at a service station operated by the provider.

At the service station, the customer inserts her smart card and has two possibilities. First, she can exchange her helper credits against traffic credits at the providers helper credits exchange rate. Second, the customer can also buy traffic credits with real money at a given price. Via the helper credits exchange rate and the traffic credits price in real money the provider can control the revenue. In order to limit the number of transactions, we introduce thresholds for the exchange of helper credits and the minimum number of traffic credits to be refilled in a single transaction. The customer can combine helper credits exchange and real money buy in order to reach the minimum traffic credits refill threshold (see maintenance refill phase, Algorithm 12). We also limit the number of storable traffic and helper credits on a smart card. Thereby we want to ensure, that customer regularly visits the service stations and in turn allow the provider to monitor and detect possible misuse.

Algorithm 12 Maintenance refill phase (at the service station)

```
repeat
  if traffic credits account $<$ traffic credits storage threshold then
    if helper credits account $>$ helper credits exchange threshold then
      calculate resulting amount of traffic credits from the exchange of helper credits at the current exchange rate
      if exchanged traffic credits amount $<$ traffic credits refill threshold then
        calculate needed amount of real money to match the traffic credits refill threshold
      end if
      show the customer the precomputed helper credits exchange and real money buy conditions
      offer the customer to modify the conditions within the given bounds
    else
      calculate needed amount of real money to match the traffic credits refill threshold
      show the customer the precomputed real money buy conditions
      offer the customer to modify the conditions within the given bounds
    end if
  else
    inform customer about traffic credits storage threshold □
  end if
until customer accepts refill conditions OR cancels refill process
if customer accepts refill conditions then
  balance traffic credits, helper credits and real money accounts □
end if
```

4.6 Security Analysis

CASHnet - like every other cooperation encouragement scheme - charges the transmission of self-generated packets and rewards the forwarding of other node's packets. The financial properties make motivation-based scheme worthwhile targets for attacks. Adversaries may try to circumvent the charges and/or obtain rewards by fraud. In the following we analyze possible attacks on the CASHnet scheme and how CASHnet can cope with them. We start with a description of the adversary model, which we use in our analysis and continue with general attacks and the analysis of every operation phase in CASHnet.

4.6.1 Adversary Model

The adversary may either be malicious or fraudulent. A malicious adversary acts regardless of the benefits for himself. His primary goal is to cause harm, e.g. to manipulate information (deletion) or to disrupt the availability of services (denial of service). A fraudulent adversary acts in expectance of benefits for herself, e.g. financial (transmission costs, rewards) or resource (energy, bandwidth). Her primary goal is to obtain an advantage and at the same time accepting the possibility of causing harm to others. This behaviour is also called ”rational” in the literature.

An attack can be executed in two ways: passive or active. In a passive attack the adversary analyzes data without modifying it. Passive attacks are difficult to detect and the properties of wireless networks support passive attacks as the medium is shared in an uncontrollable manner. Secrecy via encryption makes the analysis of information more difficult. We do not consider passive attacks, as these can be addressed via higher layer encryption protocols. In an active attack the adversary analyzes and manipulates the data. Active attacks include dropping of packets, replaying packets, modifying packets, impersonating nodes, tunneling of packets and denial of service. Active attacks can be detected when an appropriate level of data protection is available.

4.6.2 General Attacks

In general we address packet drops via rerouting. This makes fraudulent attacks (e.g. to save energy) not beneficial to adversaries, because they will lose forwarding opportunities. Since we use nonces in every message, we can detect replayed packets, which are dropped. The use of digital signatures allows us to validate the origin and the integrity of each packet and thereby detect any modifications. Modified packets are dropped too. The impersonation of nodes requires the possession of the private key, which in turn requires the possession or access to the smart card of the node. Because CASHnet is highly decentralized, the provider has only limited control inside the network. However, the provider can monitor the traffic at the gateways and the cash flow at service stations.

CASHnet does not provide a strong protection against malicious attacks, such as packet drops without benefit. It relies on the routing protocol to avoid these malicious nodes, e.g. by treating this attack as a link break and finding a new route. In a multi-hop cellular network with a high node density, and few adversaries this should not pose a problem. It is clear, that with an increasing number of malicious adversaries, the network stops to function. However, CASHnet provides the possibility to identify an attacker, because in CASHnet, every node needs to be authenticated with the help of its certificate issued by the provider, before it can receive packets. Nodes, which are not authenticated do not receive any packets. Therefore, a node could monitor and report suspicious next hops via the service station to the provider, which in turn can deny the renewal of the accused node’s certificate. One possible indicator for a malicious adversary could be for example periodic link breaks.

Colluding fraudulent adversaries may send packets as ad hoc only traffic (tunneling) from far locations close to the gateway in order to reduce the cost for transceiving in case of dynamic, hop count related charges. This only makes sense when the colluding nodes are rather stationary, e.g. some close to the gateway and some far away eager to reduce the transceiving costs. So instead of a distant node, a node close to the gateway acts as a proxy and pays the transceiving costs. However, intermediate nodes, which forward ad hoc only traffic do not get rewarded and thus the provider has no loss. Further, the traffic has to pass via the gateway and thus must be digitally signed. The provider can monitor the traffic frequency and identify possible colluders with abnormal traffic patterns. The provider can also use the globally fixed transceiving costs.

Colluding fraudulent adversaries may try to artificially increase the route length to earn additional helper credits. This is influenced by the robustness of the routing protocol. Several attempts exist to provide a secure routing protocol as shown by Buttyán and Hubaux [BH03a]. In order to avoid loss, the provider can regulate the revenue via the exchange rate for the helper credits and the cost in real money of the traffic credits.

In the Internet, denial of service attacks are difficult to prevent, because of the missing authentication of the originator of a communication. CASHnet requires every node to sign its packets, thus making the originator identifiable. However, a malicious adversary might attack the signature verification process of a node by sending false signatures. A countermeasure would be a filter with a possible timeout for each entry.

We assume the secure storage of node identity, public/private keys and credits accounts on the card. However, smart cards have been attacked in different ways as published in 1996 by Anderson and Kuhn in [AK96]. Besides mathematical attacks such as cryptanalysis, where attackers try to exploit statistical properties of the used cryptographic algorithms, attacks on the implementation are also possible. Implementation attacks use the leakage of side-channel information, such as power dissipation, timing information or faulty outputs. Messerges et al. [MDS02] give an extensive overview on possible attacks and study power analysis attacks in particular. They conclude, that power analysis attacks are possible and can be prevented by reducing or rendering unusable the leaked side-channel information.

The manufactures of smart cards are well aware of the results from cryptanalysis and continuously improve their products. It takes a considerable effort to perform these attacks and a broken smart card could only be used until the detection by the provider. As every originator needs to sign its packets, in order for them to leave the multi-hop cellular network, the originator's identity is revealed to the gateway. The provider can keep track of traffic from the originators and verify expenses on their account. Thus, a node which is transmitting over a long time period, but never refills its traffic credits account is suspicious.

4.6.3 Attacks on CASHnet Operation Phases

In this section we describe why attacks on the CASHnet operation phases are not beneficial for fraudulent adversaries. As CASHnet interprets missing messages as link break and initiates a reroute, a small number of malicious adversaries can also be circumvented.

Preparation & Maintenance

The preparation phase involves direct interaction with the provider. An adversary might try to obtain a smart card through theft or social engineering from another customer in order to avoid registering his personal information with the provider. This would be discovered by the true owner of the card and the card would be blocked, i.e. it would be impossible for the adversary to use the smart card anymore. The adversary may also try to load the credits accounts by himself. Here the adversary would have to imitate the operation of a service station. This would require trial and error processing and the smart card can react accordingly, e.g. lock or even destroy itself [NPSQ03]. The maintenance phase is similar to the preparation phase and so are the possible attacks.

Authentication

In the authentication phase, a malicious adversary might drop certificate advertisements and/or replies during their transit to the recipient and back to the originator. In both cases, the originator of the certificate advertisement will never receive the certificate reply from the recipient. The originator will try to use another route by triggering a search via the routing protocol.

A fraudulent adversary has no advantage from dropping authentication messages, as she increases the signaling traffic with new certificate advertisements as well as route discoveries, which may even lead to her exclusion from the route. It is therefore not beneficial for a fraudulent adversary to attack the authentication phase.

Charging

In the packet forwarding phase a malicious adversary might drop data packets it has received. The loss of data packets can be interpreted as a link break and can initiate the search for a new route. It is also not beneficial for fraudulent adversary to drop data packets, as she can only obtain helper credits if the next hop receives the packet and thus transmits an acknowledgement.

Rewarding

A malicious adversary might not send the rewards. The lack of reward messages can be interpreted as a link break by the node and initiate appropriate actions, e.g. trigger the routing protocol to search for a new route. A fraudulent adversary may block the rewards in order to save battery power. However, in case the previous hop looks for a new route, she may become excluded from the route and thereby earn less helper credits. This renders dropping the reward messages not worthwhile for a fraudulent adversary.

To disclose the adversary, the node can hand the expected reward list to the provider during the maintenance phase. The provider could react appropriately, e.g. reduce the value of the adversary's helper credits.

4.6.4 Summary

Attacks on CASHnet are not beneficial for fraudulent adversaries and can be circumvented via rerouting in case of a small number of malicious adversaries. It is clear, that once a certain number of malicious adversaries is reached, every multihop wireless network stops functioning.

Although CASHnet is highly decentralized and does not even require source routing, it gives the provider several possibilities of control. The obligatory digital signing of every message allows to identify its originator and ensure the integrity of the message. Neither unsigned nor invalid packets can pass the gateway. Thus, the provider can observe the traffic pattern of every originator at the gateway and compare it with the cash flow at the service stations. As in real-life, being identifiable is a psychological barrier in committing any kind of attack.

The provider can also balance the revenue using the helper credits and real money exchange rates accordingly in order to distribute the loss. By setting globally fixed charges, some fraudulent attacks become unnecessary. The reporting of the expected reward list helps to identify packet dropping attacks. In CASHnet, the smart card contains valuable information. The short certificate life-time and the limitations of the maximum number of traffic and helper credits enforce a regular visit at the service station, where the integrity of the smart card can be verified.

4.7 Resale

In CASHnet the charging and rewarding is decentralized, however the exchange of helper credits or real money into traffic credits is only possible at a service station. In order to allow greater flexibility and independence from the service stations, we introduce the possibility of resale of traffic credits against helper credits. A node can act as a reseller by offering its traffic credits for helper credits to a buyer node. A buyer node in lack of traffic credits can ask its one-hop neighbors for the resale conditions.

Resale among nodes implies the exchange of virtual currency over the wireless network. To secure the exchange, special requirements must be fulfilled. The resale process has to be conducted between trusted entities in a secure environment (e.g. programs also called agents on a smart card). Therefore, such an agent on the smart card requires a dedicated public-/private-key pair together with a verifiable certificate, which binds the agent's identity with its public key and proves the reseller status. The entity's private key must be inaccessible to the user of the node to protect against impersonation attacks on the resale agent. The complete communication must be encrypted in order to ensure the secrecy about the resale.

The protected environment for the resale operation can be provided by Java Cards [BBE+99]. A Java Card is a smart card which runs a virtual machine capable of executing applets programmed in Java Card [Sun05], a small subset of the Java programming language.

In order to avoid the need for constant interaction between the customer and its agent on the smart card, the customer should indicate her preferences for resale, e.g. the minimum acceptable exchange rate for her helper credits and her expected network usage profile. This would help the agent to determine at which account balance the agent should start to buy traffic credits and also the optimal amount.

Figure 4.9 shows the resale operation phase of CASHnet, in which a node trades its helper credits for traffic credits with the help of a reseller. We use the notation from Figure 4.2 on page 56. The reseller offers traffic credits at a specific exchange rate.

The typical course of action for the agents of node and a reseller, which want to perform a resale consists of three steps. Figure 4.10 illustrates all steps in a message sequence chart. The numbered gray markers refer to example positions of the actions from the following list.

1. Advertisement: To advertize its existence, a reseller R periodically broadcasts a reseller advertisement $RADV_R$ to all its one-hop neighbors if enough traffic credits are available. The $RADV_R$ authenticates the reseller to its recipient and the buyer agent B can add the reseller to its known reseller list. Before a resale can take place, the buyer agent needs to authenticate towards the reseller. To do so, it sends a buyer advertisement $BADV_B$ to a known reseller from the list and which is a one-hop neighbor.

2. Offer: When a node has a low traffic credits account, its buyer agent B sends an offer request $OREQ_B$ to the reseller to which it has previously advertized its certificate. The $OREQ_B$ indicates the amount of helper credits the buyer is willing to offer. Upon reception of an $OREQ_R$ the reseller sends back an offer reply $OREP_R$ to the buyer. The $OREP_R$ indicates the amount of traffic credits the reseller is willing to trade for the offered amount of helper credits.

3. Resale: If the buyer agrees with the conditions of the reseller, it sends a resale request $RREQ_B$ to the reseller. Upon reception of a $RREQ_B$ the reseller sends back a resale reply $RREP_R$. When the buyer receives the $RREP_R$ it reduces the helper credits and increases the traffic credits accounts according to the resale conditions. The buyer also acknowledges the reception by sending a resale acknowledgement $RACK_B$ to the reseller. Upon reception of the $RACK_B$, the reseller balances the accounts accordingly.

We distinguish between eight operation phases and divide them into two categories. Figure 4.11 illustrates the relation between the different phases. In the authentication phase, a reseller agent advertizes its ability to trade traffic credits

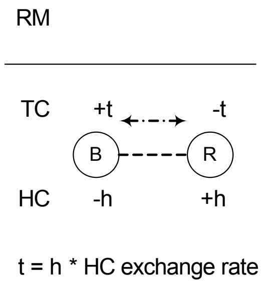


Figure 4.9: CASHnet resale operation

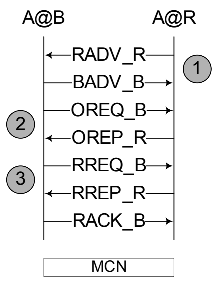


Figure 4.10: CASHnet resale operation

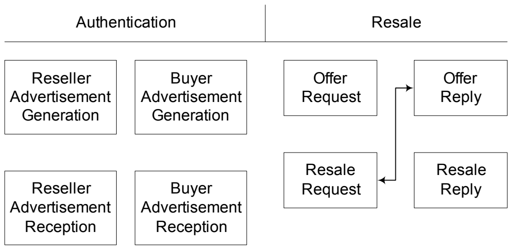

Authentication
Resale
Reseller Advertisement Generation
Buyer Advertisement Generation
Offer Request
Offer Reply
Reseller Advertisement Reception
Buyer Advertisement Reception
Resale Request
Resale Reply

Figure 4.11: CASHnet resale phases

B	Buyer agent	R	Reseller agent
HCO	Helper Credits offer	HCO	Traffic Credits offer
N_B	Nonce chosen by B		
$E_R(M)$	Public key encryption of message M with R's public key		
$Sig_B(M)$	Digital signature of message M by B		

Table 4.2: Notation for Resale Algorithms 13-20

for helper credits. In the resale phase, the exchange between a node's agent and a reseller's agent is performed. We describe each operation phase in detail as algorithm (see Algorithms 13-20) using the notation listed in Table 4.2.

4.7.1 Authentication

In order to announce the presence of reseller agents to buyer agents and to securely exchange credits, the agents on the smart card must authenticate each other. This is done by exchanging certificates between the agents on each smart card. To authenticate itself, an agent creates a reseller advertisement, which contains its certificate a nonce, and a digital signature over the hash value of its certificate and the nonce. The digital signature allows to verify the origin and integrity of the certificate advertisement. The certificate indicates the reseller status of an agent and allows to verify the binding between an agent's identity as well as its public key. Reseller advertisements are broadcasted periodically to all one-hop neighbors (see reseller advertisement generation phase, Algorithm 13).

Algorithm 13 Reseller advertisement generation phase

```
loop
  if reseller advertisement interval elapsed then
    create a payload consisting of reseller agent certificate and nonce
    calculate digital signature over reseller agent certificate and nonce
    append digital signature to payload
    transmit packet $RADV_R$ to all one-hop neighbors
  end if
end loop
$R \rightarrow B : RADV_R = Cert_X(R), N_R, Sig_R(H(Cert_X(R), N_R))$
```

Algorithm 14 Reseller advertisement reception phase

```
if certificate for reseller agent from reseller advertisement $RADV$ valid then
  if signature of reseller agent from $RADV$ valid then
    save tuple of reseller identity and public key in known reseller list
  else
    drop $RADV$
  end if
else
  drop $RADV$
end if
$B \leftarrow R : RADV_R = Cert_X(R), N_R, Sig_R(H(Cert_X(R), N_R))$
$B : Tuple_R =< ID_R, K_R >$
```

When a buyer agent receives a reseller advertisement, it first verifies the certificate and than the digital signature. If the verification of either is not successful, the reseller advertisement is dropped. Otherwise, the agent adds a tuple of the reseller's identity and public key, which are both contained in the certificate, to the known reseller list (see reseller advertisement reception phase, Algorithm 14).

To complete the authentication, the buyer agent needs to advertize its certificate in return to the reseller. This is done before the actual resale starts. The buyer node creates a buyer advertisement, which contains its certificate, a nonce and a digital signature over the hash value from the certificate and the nonce. The buyer transmits the message to an available reseller in its one-hop neighborhood, which it selects from its known reseller list (see buyer advertisement generation phase, Algorithm 15).

When a reseller agent receives a buyer advertisement it verifies the signature and the certificate and - if valid - stores the agent identity and its public key, which are both contained in the certificate, as tuple in its known buyer list. In case, the verification fails, the buyer advertisement is dropped (see buyer advertisement reception phase, Algorithm 16).

Algorithm 15 Buyer advertisement generation phase

```
look up all one-hop neighbors
if one-hop neighbor $\in$ known reseller list then
    create a payload consisting of buyer agent certificate and nonce
    calculate digital signature over reseller agent certificate and nonce
    append digital signature to payload
    transmit packet $BADV_B$ to the selected reseller
end if
$B \rightarrow R : BADV_B = Cert_X(B), N_B, Sig_B(H(Cert_X(B), N_B))$
```

Algorithm 16 Buyer advertisement reception phase

```
if certificate for buyer agent from buyer advertisement $BADV$ valid then
    if signature of buyer agent from $BADV$ valid then
        save tuple of buyer identity and public key in known buyer list
    else
        drop $BADV$
    end if
else
    drop $BADV$
end if
$R \leftarrow B : BADV_B = Cert_X(B), N_B, Sig_B(H(Cert_X(B), N_B))$
$R : Tuple_B =< ID_B, K_B >$
```

4.7.2 Resale

When the traffic credits account of a node falls below a specified threshold and it has enough helper credits, its buyer agent sends an offer request to a known reseller agent within its one-hop neighborhood. To ensure secure bi-directional communication between the buyer and reseller agent, the buyer agent sends a buyer advertisement before the offer request.

The offer request contains the amount of helper credits the buyer is offering for exchange, its identity, a nonce as well as a digital signature over these three fields. This payload is encrypted with the reseller's public key, so that only the reseller can read it. By encrypting after signing we ensure that the signer has knowledge about the encrypted data.

If the buyer does not receive an offer reply within a specified time frame, it tries to contact another available reseller. If the buyer receives an offer reply, it continues with the resale request phase (see offer request phase, Algorithm 17).

When a reseller receives an offer request, it decrypts the message with its private key. It continues to verify the signature of the decrypted offer request, by decrypting it with the public key of the buyer and comparing the result with the remaining message payload. If the comparison fails, the message is dropped.

If it is successful, the reseller calculates the resulting amount of traffic credits

Algorithm 17 Offer request phase

```
loop
  if traffic credits account $<$ traffic credits buy threshold AND helper credits account $>$ helper credits threshold then
    execute buyer advertisement generation phase (Algorithm 15) and return $\rightleftharpoons$
    if one-hop neighbor $\in$ known reseller list then
      create a payload consisting of helper credits offer, node identity and nonce
      calculate digital signature $Sig_B$ over helper credits offer $HCO$, node identity $ID_B$, nonce $N_B$
      append digital signature to payload
      transmit packet $OREQ_B$ to the selected reseller
      repeat
        wait
      until offer reply $OREP$ arrives OR timeout
      if $OREP$ arrives then
        go to resale request phase (Algorithm 19) $\Rightarrow$
      end if
    else if timeout then
      select another reseller
    end if
  end if
end loop
$B \rightarrow R : OREQ_B = E_R(HCO, ID_B, N_B, Sig_B(HCO, ID_B, N_B))$
```

according to its exchange rate and the amount of helper credits offered by the buyer. Then, the reseller agent creates an offer reply, which contains the resulting amount of traffic credits, the offered amount of helper credits, its identity, a nonce and a digital signature of the four fields. The offer reply is encrypted with the buyers public key to ensure confidentiality of the resellers offer. The reseller waits for a resale advertisement from the buyer to arrive and if it does proceeds with the resale reply phase (see offer reply phase, Algorithm 18).

In case the buyer agent receives an offer reply, it decrypts the message using its private key and verifies the signature with the reseller's public key. In case the verification fails, the message is dropped. If it is successful, the buyer agent retrieves the reseller's offer and checks whether it is acceptable. An acceptable minimum threshold is specified by the node owner in advance.

If the buyer agent can accept the reseller's offer, it creates a resale request, which contains the offered amount of traffic credits offered by the reseller, the buyer's identity, a nonce and a digital signature over the three fields. The resulting payload is encrypted with the reseller's public key and transmitted to it. The buyer waits for the reseller to confirm the resale with a resale reply.

Algorithm 18 Offer reply phase

```
decrypt offer request $OREQ$
if signature from $OREQ$ valid then
  if traffic credits account $>$ traffic credits resale threshold then
    retrieve amount of helper credits offered by the buyer
    calculate resulting amount of traffic credits
    send a offer reply $OREP$ to the buyer
    repeat
      wait
    until resale request $RREQ$ arrives OR timeout
    if $RREQ$ arrives then
      go to resale reply phase (Algorithm 20) $\Rightarrow$
    end if
  end if
else
  drop $OREQ$ $\Box$
end if
```

$$R \rightarrow B : OREP = E_B(TCO, HCO, ID_R, N_R,$$
$$Sig_R(TCO, HCO, ID_R, N_R))$$

Upon reception of a resale reply, the buyer decrypts the message and verifies the signature. If an error occurs, the resale reply is dropped. Otherwise, the buyer agent balances the accounts by charging helper credits and debiting traffic credits according to the resale conditions. Finally, the buyer advertizes the finalization of the resale with an acknowledgement (see resale request phase, Algorithm 19).

When the reseller agent receives a resale requests, it decrypts the message using its private key and verifies the digital signature using the buyer agent's public key. The reseller compares the included traffic credits offer with its own and if they are identical, creates a resale reply. The resale reply consists of the reseller identity, the nonce from the reseller and the nonce of the buyer from the resale request as well as a digital signature over the three fields. The resale reply is then encrypted with the buyer's public key to ensure secrecy about the exchange and transmitted to the buyer. The reseller waits for the buyer to announce the resale completion with a resale acknowledgement.

Upon reception of a resale acknowledgement, the reseller agent debits the helper credits and charges the traffic credits accounts according to the condition of the resale (see resale reply phase, Algorithm 20).

4.8 Network Management

The optimal management of multi-hop cellular networks is a major concern of the provider. It includes network planning, i.e. analyzing the current state and identi-

Algorithm 19 Resale request phase

```
decrypt offer reply $OREP$
if signature from $OREP$ valid then
   retrieve resale conditions of the reseller
   if resale conditions within specified acceptance range then
      send a resale request $RREQ$ to the reseller
      repeat
         wait
      until resale reply $RREP$ arrives OR timeout
      if $RREP$ arrives then
         decrypt $RREP$
         if signature from $RREP$ valid then
            helper credits account - helper credits offer
            traffic credits account + traffic credits offer
            send a resale acknowledgement $RACK$ to the reseller □
         else
            drop $RREP$ □
         end if
      end if
   end if
else
   drop $OREP$ □
end if
$B \to R : RREQ = E_R(TCO, ID_B, N_B, Sig_B(TCO, ID_B, N_B))$
$B \to R : RACK = E_R(TCO, ID_B, N_B, Sig_B(TCO, ID_B, N_B))$
```

fying possibilities to improve the service and the revenue. Due to the dynamics of multi-hop cellular networks, the network management requires the frequent verification of the current network state, the reliable identification of changes as well as the fast adoption to the new conditions. In the following sections, we describe the management instruments in a multi-hop cellular network, network state indicators and a possible monitoring architecture. We also perform an exemplary case study.

4.8.1 Network Management Instruments

In CASHnet, the main management instruments for the provider are the number and location of deployed gateways, service stations and resellers. Further, the helper credits/real money exchange rates for traffic credits can be used to influence the behaviour of customers and resellers.

- The *gateways* provide the interconnection of the mobile ad hoc network and the multi-hop cellular network. If correctly placed, multiple gateways can considerably reduce the average route length from a node inside the net-

Algorithm 20 Resale reply phase

```
decrypt resale request $RREQ$
if signature from $RREQ$ valid then
  if traffic credits offer from $RREQ =$ own traffic credits offer then
    send a resale reply $RREP$ to the node
    repeat
      wait
    until resale ACK $RACK$ arrives OR timeout
    if $RACK$ arrives then
      decrypt $RACK$
      if signature from $RACK$ valid then
        helper credits account + helper credits offer
        traffic credits account - traffic credits offer □
      end if
    else if timeout AND retry threshold not exceeded then
      resend the $RREP$ to the buyer
    end if
  end if
else
  drop $RREQ$ □
end if
$R \rightarrow B : RREP = E_B(ID_R, N_R, N_B, Sig_R(ID_R, N_R, N_B))$
```

work to the gateway and thereby increase the per-node throughput. However, gateways are a big expense factor in the multi-hop cellular network of the provider. Thus, an optimal balance between amount and coverage is desirable.

- The *service stations* allow the customers to refill their traffic credits account and update a certificate. At the same time, they enable the provider to monitor and analyze the cash flow as well as reports generated by the node. The service stations only require a low bandwidth connection to the provider and are comparable with a terminal for loading prepaid cards. Also for the service station the correct placement is important in order to ensure short distances and easy accessibility for the customer.

- The *resellers* compensate for the immobility of the service stations. They are able to exchange the helper credits of normal customers into traffic credits. Thus, the provider can use the resellers to virtually extend the coverage of the service stations. We note, that the customers still have to regularly visit the provider to renew their certificates.

- The *exchange rates for traffic credits* directly influence the revenue of the provider. The customer has the possibility to trade traffic credits for helper credits at the service stations and with the reseller. In addition, she can buy traffic credits for real money at the service station. Therefore, the provider can dynamically adapt the exchange rates according to the current network conditions.

4.8.2 Network State Indicators

We identify three indicators in CASHnet for an optimal application of the previously defined instruments: the route length to the gateway, the cash flow and the starvation history of each node. These indicators rely on collected data related to the customer. Due to privacy issues the collected information will most probably be stored anonymously.

- The averaged *hop count* to the gateway per node allows the provider to draw conclusions about the general state of the network. By analyzing the peaks in the route length the phenomena can be isolated. Long routes to the gateway typically indicate that the coverage of the multi-hop cellular network increases, i.e. a change in the distribution of customers. This can be caused by new customers joining the network or customers moving to a new area or a mixture of both. In either case, every additional hop decreases the throughput drastically.

- The analysis of the *cash flow* and the maintenance of the account state history allows the provider to identify changes in the forwarding behaviour of a customer. For example, a reduction in the amount of helper credits traded may indicate that the overall network traffic decreased, and thus a change in the node environment has occurred (e.g. fewer neighbors). Together with the hop count information, the provider can identify possible thin out zones and take appropriate actions.

- Closely related to the cash flow is the *starvation* information. It describes the beginning and ending of a period, where the node was unable to transmit self-generated packets or receive packets destined to it due the lack of traffic credits. High starvation periods on a node may indicate that the node was unable to find a service station or a reseller in time and the provider should try to increase the number of resellers in the network.

The accuracy of these indicators greatly depends on the possibility of mapping them to geographical information of the multi-hop cellular network, which in turn depends on where we collect this information. From the perspective of the gateway, the hop count describes a circle at best. Thus, the direction in which the network is expanding or shrinking is not deducible from the hop count alone. The starvation alone does not indicate where in the network the node starved. However, the distance to the gateway, the distance to the next service station and the number and movement path of the resellers represent important information, which are required to actually derive effective strategies to reduce starvation in the network.

Indicator	Properties	Decentral	Central
Hop count	Location of acquisition Requirements Interface to provider	Node Position (GPS) Service station	Gateway Gateway environment -
Cash flow	Location of acquisition Requirements Interface to provider	Node Position (GPS) Service station	Service station Account state history -
Starvation	Location of acquisition Requirements Interface to provider	Node Position (GPS) Service station	- - -

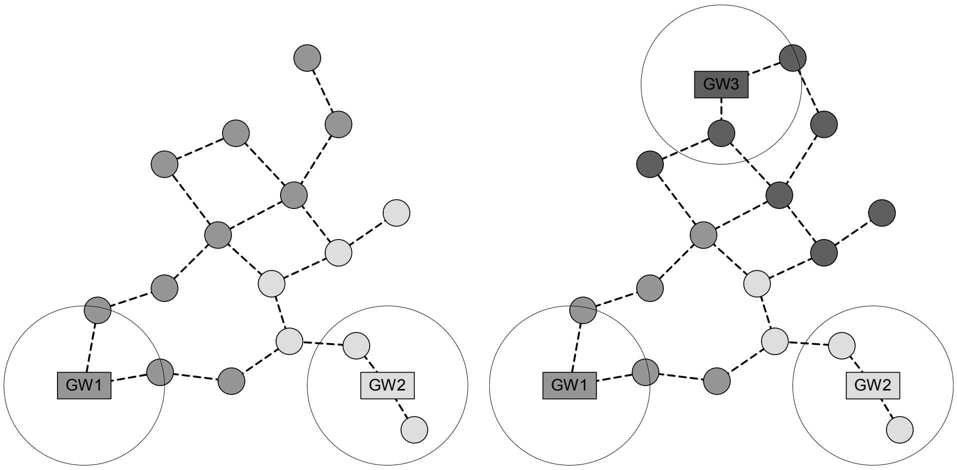
Table 4.3: Network management indicators in CASHnet

4.8.3 Monitoring Architecture

The decentralized characteristics of multi-hop cellular networks make it more difficult to gather information from within the network. On the hand, the provider can use the existing infrastructure, that is the gateways and the service stations, to monitor the average hop count by analyzing the passing traffic as well as the cash flow via the service station. On the other hand, each node can keep track of the same information, which could be stored on the smart card and transferred to the provider via the service station. Also, some information is only available on the node, such as possible starvation occurrences and resales. In practice, a combination of both, the centralized and the decentralized monitoring is most feasible, i.e. to elect some independent customers (e.g. the resellers) as probing nodes and compare their observations with the centrally gathered data.

In both approaches the collected network state information has to be mapped to the geographical location of the network infrastructure (e.g. node position) to be of any use. Without the position information, the reliable evaluation of the indicators as well as the appropriate reaction is not possible. For example, from the starvation indicator alone, the provider does not know where in the network the node starved and thus can not effectively direct the countermeasures (service station or reseller). Table 4.3 gives an overview of the network state indicators and their properties in case of a decentral and a central approach. The centralized approach requires information about the surroundings of the gateways and the estimated coverage area of the network (e.g. park, street, plaza) in order to locate the area of the potential new hot spots. The decentralized approach requires the node position for each collected data set. This can be obtained by an appropriate service (e.g. GPS).

While the centralized approach does not require any additional information from the node, it does not allow precise localization of the monitored phenomena. The decentralized approach is very precise thanks to the position information of the nodes, but may not be feasible in practice due to the unavailability of a positioning service at the node (missing GPS device).



(a) Initial network topology (b) Adapted network topology

Figure 4.12: Network management in CASHnet

4.8.4 Exemplary Case Study

In Figure 4.12a, we show a simplified multi-hop cellular network with two gateways (GW1 and GW2) and several nodes. The nodes are colored according to their respective default gateway, which they use to communicate with nodes located outside the multi-hop cellular network. We can see that the majority of the nodes is connected to GW1. The provider uses centralized monitoring as described above. From the analysis of the traffic passing the gateway and the routing tables he discovers, that the average route lengths at these two gateways is very high. In addition, the provider analyzes the account state history of the nodes in this multi-hop cellular network, which show a high number of exchanged helper credits. Both observations are strong indicators for the deployment of a new gateway.

In order to avoid long routes, which decrease the per-node throughput and high expenses caused by helper credits, the provider decides to deploy an additional gateway. With the help of the environment information of the current gateways, the provider identifies a potential new hot spot and deploys a new gateway (GW3). If the estimations were correct, the average route length at the two older gateways should decrease again, since the routing protocol on the nodes close to the new gateway updates the default route as shown in Figure 4.12b. Because the route length at best describes a circle and no direction, there could be more than one potential hot spot, which the provider would have to test by temporarily deploying more gateways and removing the less used again later on.

In case the provider uses a decentralized monitoring architecture, the interpretation of the observed phenomena becomes much more accurate and so does the localization of new hot spots.

4.9 Conclusion

We presented CASHnet, our cooperation and accounting strategy in hybrid networks. With CASHnet we aim to make the deployment of multi-hop cellular networks feasible and profitable for both providers and customers. We give customers incentives to cooperate and the providers means to control and secure their network service.

With the highly decentralized CASHnet architecture, we retain the flexibility of mobile ad hoc networks by performing the charging on the node of the customer and the rewarding among forwarding neighbors. At the same time we keep the provider in control of the cash flow with the help of service stations, which are similar to immobile, low-bandwidth terminals for prepaid cards and connected to the accounting center of the provider. At the service station, the customer is able to refill its traffic credits account by exchanging helper credits or real money. In addition, we introduce reseller nodes, which are operated by customers with special exchange conditions at the service station. The resellers can exchange traffic credits for helper credits from normal nodes. Still, the provider remains in control of the cash flow as the resellers have to go via the service station to transform the traded helper credits into traffic credits. Thus, the resellers are a beneficial enhancement to the immobile service stations. Further, we describe how the CASHnet framework supports the provider in the network planning process.

Our security analysis shows that fraudulent attacks on CASHnet are not beneficial and malicious attacks can be detected to a certain extent with the help of the digital signatures on all packets and monitoring at the gateways and service stations.

In order to validate our design, we implemented CASHnet in the network simulator ns-2 as well as tested it in a real-life prototype under Linux. We describe the implementation process and the obtained results in the next two chapters.

Chapter 5

Evaluation of CASHnet

5.1 Introduction

CASHnet, our cooperation and accounting strategy for hybrid networks provides a framework to encourage cooperation among nodes in multi-hop cellular networks. It introduces a considerable number of changes to a normal multi-hop cellular network, such as charges to the transmission process of self-generated packets and to the reception process of packets destined to the current node as well as rewards to the forwarding processes of other node's packets. CASHnet also introduces new equipment, the service stations to control the cash flow and at the same time bring reliability in the provision of traffic credits for the network participants. In order to estimate the impact, which CASHnet has on a multi-hop cellular network, we evaluate our cooperation scheme using the network simulator ns-2.

In the remainder of this chapter, we start with the motivation for our approach. We continue with a description of the network simulator ns-2 and our changes to it. Then, we explain our simulation setup, scenario and parameters as well as our evaluation criteria in detail. Also, we identify and analyze the effect of the key parameters of CASHnet. Next, we analyze our simulation results from CASHnet an compare them to Nuglet. Finally we conclude with a summary.

5.2 Motivation

To validate CASHnet we have to measure the performance and impact of our cooperation and accounting framework on a multi-hop cellular network. We started with the evaluation via simulations, because they enable us to find the upper and lower boundaries as well as improve and optimize our scheme by variation of the available parameters. Furthermore, simulations allow us to place our work in relation to other cooperation schemes. All this is difficult to achieve with real-life implementations, as many side-effects and limitations influence the measurements results. Also, due to the application scenario a large and expensive testbed would be required. Nevertheless, we implemented a prototype of the CASHnet framework, which we present in Chapter 6.

At the time of writing the evaluation of cooperation schemes in the literature is mostly restricted to theoretical security and performance analysis leaving out current available technologies and protocols. To best of our knowledge, we are first to show the impact of a cooperation scheme on a multi-hop cellular network build of current technologies and protocols.

5.3 The Network Simulator ns-2

When we searched for a suitable simulator, we had several requirements in mind. The simulator should be established in the research community for comparable results and provide implementations of recent protocols and technologies. It should also support node mobility and be easily extendable. We chose the network simulator ns-2 [BEF$^+$00], because it provides a large library of protocols and is widely used in the research community. It implements a full protocol stack for mobile ad hoc networks and can be extended to also support multi-hop cellular networks.

The network simulator ns-2 is a discrete event simulator targeted at network research. It has been developed by UC Berkeley and UCS/ISI as part of the VINT project. Its source code and documentation are available online [NS204, FV03] and various contributions from other projects have been added over time. Ns-2 is available at no charges.

We use ns-2 allinone version 2.27 with the wireless and mobility extensions from the CMU Monarch project [Ric99] and an extended version of the AODV protocol called AODV+ from Hamidian [Ham03b], which adds Internet gateway discovery support, i.e. support for multi-cellular network. We describe AODV and AODV+ in Section 2.5.1 and 2.5.2 on page 18 and 21 respectively.

Ns-2 maps real world network objects like nodes or links to C++ objects, which can be parameterized to match reality as close as possible. Physical activities like the transmission of a packet are stored as events in a queue. During the simulation run the events are processed from the queue according to the scheduled execution time. The execution of an event in the simulation takes an arbitrary amount of real time on the computer, where the simulator runs. Ns-2 also allows to keep track of all actions and results in so called trace files.

Ns-2 provides two different ways to use it. For the evaluation of already implemented protocols, the scripting language Object Tcl, OTcl allows to specify simulation scenarios as well as parameters at a very detailed level. For the development and evaluation of new protocols, the programming language C++ in which ns-2 is written, can be used. The support of two languages requires the mapping between OTcl and C++ objects.

We illustrate a typical simulation run in Figure 5.1 with markers to indicate the major steps. First, a user writes a script in OTcl, which specifies the simulation scenario including the node properties as well as the movements and passes it to

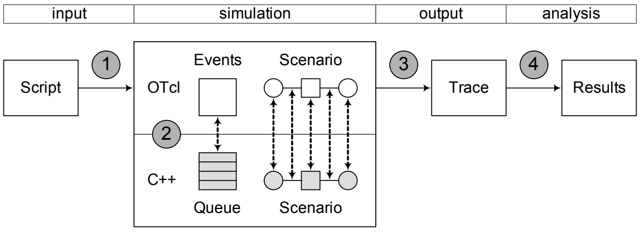


Figure 5.1: Ns-2 exemplary simulation run

the network simulator. Second, ns-2 translates the script using its OTcl interpreter to initialize the corresponding scenario in the C++ space, where the simulation is actually performed. During processing, the simulator writes a trace file describing the events and their results. Last, the user analyzes the trace file with the help of some utilities and according to the evaluation criteria.

5.4 Implementation of CASHnet in ns-2

CASHnet is a framework which relies on cross-layer communication, because the packet flow on the network layer is controlled by an upper layer accounting application. Thus, information needs to be exchanged among different layers and cross-layer control mechanisms need to be established. For the implementation in ns-2 we restrict ourselves to the charging, rewarding, refill and resale functionality of our CASHnet scheme. Thus, we leave out the security mechanisms, i.e. authentication of nodes, signing and verification of packets as well as certificate updates.

We think that these functionalities mainly stress each node locally and their impact on the network performance can be approximated accordingly. Thus, we leave the analysis of the computational and communication overhead introduced by the CASHnet security mechanisms to our real-world implementation described in Chapter 6.

Figure 5.2 shows a subset of the ns-2 class hierarchy. The gray-colored objects indicate the additions and modifications we performed in order to implement CASHnet. Since we require the node to posses CASHnet abilities (i.e. being charged and rewarded), we created a new class *CASHnetNode* which inherits from *CooperationNode* which in turn inherits from *MobileNode*. The rewarding process is implemented as an agent called *CASHnetACK*. The refill process requires a service station *CASHnetServiceStation* and a trigger, i.e. the *CASHnetTradeTimer* to periodically look for trade opportunities.

We replaced the included AODV routing protocol with AODV+ [Ham03a], which supports Internet gateways and thereby allows to transform a mobile ad hoc network into a multi-hop cellular network. We also modified the *AODV+* routing agent and the *Agent* class itself in order to support cooperation functionality. We

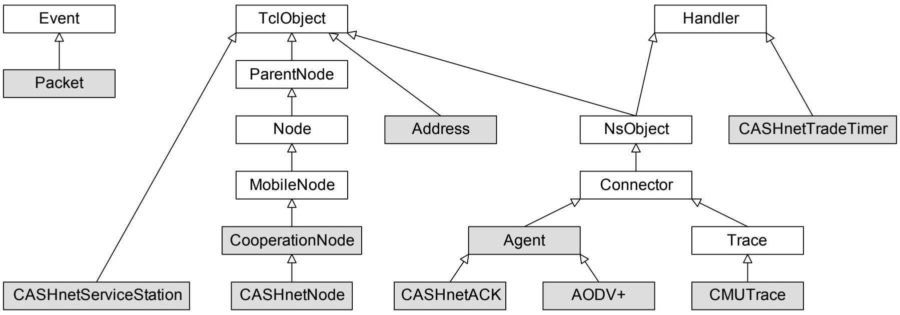


Figure 5.2: Partial ns-2 class hierarchy with CASHnet extensions/modifications

added a new packet type for the transmission of rewards to the *Packet* class, tracing support for CASHnet in *CMUTrace* as well as localization of nodes according to their hierarchical address level in *Address*. In addition, we also extended the scripting interface to easily configure the new objects and their parameters introduced with CASHnet.

Figure 5.3 shows the internal structure of a CASHnet node in ns-2 using hierarchical addressing. The gray-colored objects and the dashed lines illustrate our additions/changes to a normal mobile node in ns-2. The numbered markers indicate the packet flow for an intermediate forwarding node. First, the intermediate node receives the packet from the link layer, where it has arrived from the wireless channel. Second, the packet passes the hierarchical address classifiers and is handed to the routing agent. Third, the routing agent triggers the rewarding of the previous hop via the CASHnet ACK agent and finally passes the packet to the wireless channel.

For further details on the CASHnet implementation process in ns-2, we refer to [Sta04].

5.5 Simulation Setup

We evaluated CASHnet through extensive simulation runs on an AMD Athlon MP 2000+ 1.67 GHz dual-processor machine. In order to make use of the second processor, we ”parallelized” ns-2 by starting two runs at the same time. A complete run consists of three phases: the simulation, the analysis of the results and the archiving of the huge trace file. In order to synchronize the parallelized runs, we created two child processes per phase, one for each run and waited for their termination before entering the next phase.

As mentioned before, ns-2 supports a variety of protocols and configuration options for mobile nodes. Table 5.1 lists the parameters we used for the mobile nodes in our simulation runs. We specify hierarchical addressing, as it allows us to use subnetworks for different multi-hop cellular networks and thereby identify the location of a node. We use AODV+ (see Section 2.5.2 on page 21) as ad hoc rout-

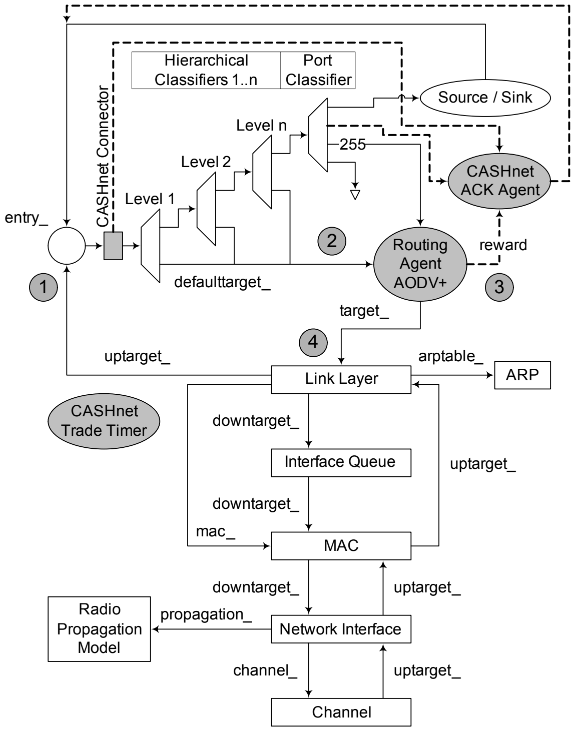


Figure 5.3: Internal structure of a CASHnet node in ns-2

ing protocol with hybrid gateway discovery and a gateway advertisement radius of 4 hops. We select the default link layer type from ns-2 and IEEE 802.11 DCF (see Section 2.4.1 on page 12) as the medium access control scheme. The radio propagation model uses the Friis free space attenuation ($1/r^2$) at near distances and an approximation to two-ray ground ($1/r^4$) at far distances. The approximation assumes specular reflection off a flat ground plane. The interface queue gives priority to routing protocol packets by inserting them at the head of the queue. We keep its default length of 50 packets. The network interface simulates the wireless medium by considering collisions and the radio propagation model when receiving packets. The antenna type is omnidirectional with unity gain. The transmission range for a mobile node is 250 meters.

5.5.1 Node Movements

To simulate node movements we use the random waypoint mobility model. Although it does not represent well the movement of a node (and the person carrying the node) compared to real life, the random waypoint mobility model is widely used in the wireless network research community and agreed upon as a standard reference. In this model the movement pattern of a node is as follows. The node starts in an initial location waiting for a specified pause time. The coordinates of a

Parameter	Value
Addressing type	Hierarchical
Ad hoc routing protocol	AODV(+)
Link layer type	LL
MAC type	Mac/802_11
Radio propagation model	Propagation/TwoRayGround
Interface queue type	Queue/DropTail/PriQueue
Interface queue length	50
Network interface type	Phy/WirelessPhy
Antenna type	Antenna/OmniAntenna
Channel type	Channel/WirelessChannel

Table 5.1: Parameters of a mobile node in our evaluation

location are selected independently and uniformly on the given region. The pause time is selected independently from speed and location. After the pause time has elapsed a new location and a speed are selected. The speed is chosen uniformly from an interval $< v_{min}, v_{max} >$ and independently from current location and destination. After the node arrives at the destination, it either waits for a new pause time or moves to a new location.

As stated before, the random waypoint mobility model is widely used and has been studied extensively. Yoon et al. [YLN03] showed that if the speed is chosen from an interval $< 0, v_{max} >$ the mean speed approaches zero over time. They also find that the probability distribution varies and converges over time to a steady state also called stationary distribution. Thus, they recommend to start the actual simulation after the nodes have moved for a while. Bettstetter et al. [BHPC04] analyze the stochastic properties of the random waypoint mobility model and find the spatial distribution of this model, which is more concentrated in the center of the simulation area. Navidi and Camp [NC04] derived the stationary distributions for location, speed and pause time for the random waypoint mobility model.

Figure 5.4 shows the movement paths of 10 out of 40 nodes, which we obtained from *setdest* and visualized with the help of gnuplot [WK04]. The arrow on each path indicates the starting position of the node and its movement direction. We describe the parameters of our node movements together with the simulation scenario in which we use them below.

5.5.2 CASHnet Parameters

In CASHnet, mobile nodes obtain several new characteristics (e.g. charge, reward, refill and resale), which are reflected in the additional parameters a CASHnet node has. We distinguish between fixed and variable parameters. The first remain unchanged throughout all simulations, the latter have the strongest influence on our scheme and vary between the different simulation runs. In the following list, we

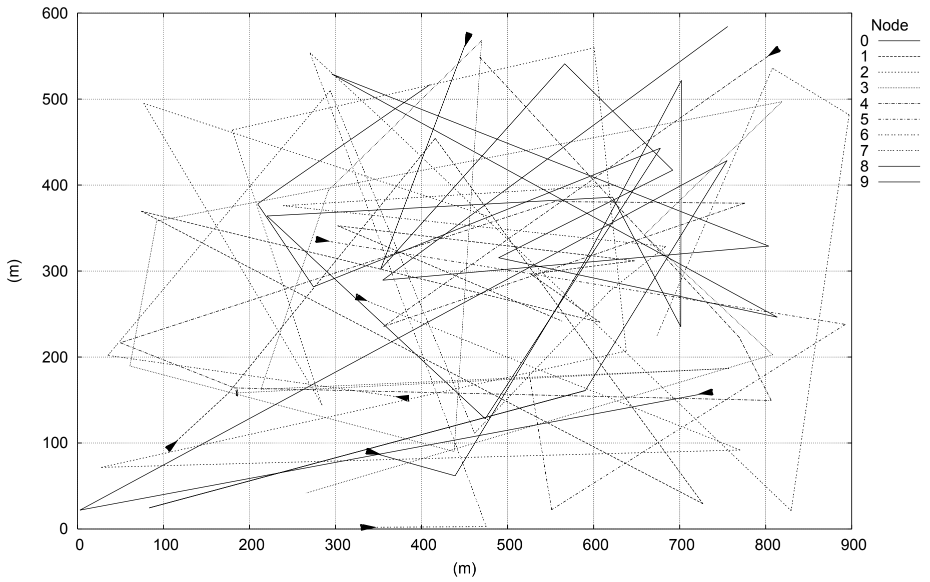


Figure 5.4: Movement paths of 10 nodes in the random waypoint mobility model

describe each parameter in detail and emphasize the variable parameters with a bold font.

- The **packet generation interval** specifies how many seconds elapse between the generation of two packets. A smaller value increases the network load.
- The **service stations** value indicates how many service stations are deployed in the simulation scenario. A high number of service stations increases the probability of a node to be able to refill its traffic credits account and thereby transmit more packets.
- The amount of nodes, which are allowed to resell their traffic credits for helper credits to another node, is set with the **resellers** value. A high number of reseller nodes increases the probability for nodes to exchange their helper credits against traffic credits and thus transmit more packets. All nodes are allowed to engage in a resale with a reseller.
- The **transceiving cost** specifies the amount of traffic credits an originator and a recipient of a packet have to pay. The transceiving cost can be dynamic for each node by setting it equal to the hop count to the gateway or it can be fixed to a global value every node has to pay.
- The **packet counter ACK threshold** defines how many forwarding packets a node has to receive from a single forwarder before it rewards this forwarding node, i.e. sends an acknowledgement message. A high number reduces the load on the network and increases the probability of loss. The packet counter

ACK threshold value is directly connected to the value of the rewards, i.e. the amount of helper credits debited per reward is proportional to the threshold.

- The *initial traffic credits/real money account state* sets the amount of (virtual) currency a node has when the simulation starts. A high amount of traffic credits allows a node to transmit packets for a long time period without the need for a service station in order to refill its account. Real money can only be traded at service stations, helper credits at service stations and with resellers.
- The *trade threshold at the service station* indicates the minimum amount of helper credits a node needs to possess in order to be able to trade with a service station.
- The *trade amount* describes the maximum number of traffic credits a node tries to trade at once.
- The *helper/traffic credits rate* specifies how many traffic credits a node can obtain for its helper credits at the service station.
- The *real money/traffic credits rate* defines how many traffic credits a node can obtain for its real money at the service station.
- The *distance threshold to the service station* sets the radius of the service station inside which a node is able to trade.
- The *reseller amount* specifies the maximum number of traffic credits sold in a single interaction with a reseller.
- When the traffic credits account exceeds the *reseller traffic credits threshold*, the reseller node announces itself to its one-hop neighborhood.
- The *reseller traffic/helper credits rate* indicates how many helper credits a node can receive for its traffic credits from a reseller.
- If the traffic credits account falls below the *buyer traffic credits threshold* and the helper credits account exceeds the helper credits threshold, a node contacts a known and reachable reseller.
- The *exchange interval* specifies how often each node checks whether it is in range of a service station and/or a reseller node.

5.6 Evaluation Criteria

We identified three major evaluation criteria for our CASHnet implementation in the network simulator: the starvation, the packet flow and the cash flow, which we investigate in our simulation runs. In the following we describe the three criteria in detail.

- With **starvation** we describe the state, where a node is unable to transmit a self-generated packet or receive a packet addressed to it due to lack of money. This situation can occur, if a node is unable to use its earned helper credits as traffic credits. The node owner is required to regularly refill its traffic credits account at a service station or with the help of a reseller. A reseller in turn is required to visit the service station to exchange the received helper credits when he has not enough traffic credits for resale. The goal is to keep starvation as low as possible in a multi-hop cellular network.

- The **packet flow** describes the network performance of a simulation run. We analyze the number of transmitted (Sent) and received packets (Received) as well as the amount of rewards sent (CASHnet ACKs sent) and dropped (CASHnet ACKs dropped). We also investigate the different drop reasons for the data packets given by ns-2. A high goodput and a low overhead are the goals regarding the packet flow in a multi-hop cellular network. The drop reasons, which occurred in our simulation runs were lack of money (No Cash), no available route (No Route), routing loop (Loop), link break detection at the routing layer (Callback), a full buffer in the address resolution protocol and a full buffer in the interface queue. When a route is not available in AODV, a route request is sent and the packet will be retained until the route requested succeeds or times out, which leads to the No Route error. The Callback error indicates that the link detection from AODV has noticed a link break. When the address resolution protocol resolve an address for a packet by sending a request, it buffers the packet until the reply arrives. If in the meantime, to many new requests arrive, the buffer is full and the unanswered packets are dropped.

- The **cash flow** summarizes the distribution of (real or virtual) money within the network. We investigate the final state of the traffic credits (TCA), helper credits (HCA) and real money (RMA) accounts. In addition, we keep track of the amount of traffic credits spent (TCS), traded (TCT) and resold (TCR) as well as the amount of real money traded (RMT). A stable distribution of virtual credits (traffic and helper credits) as well as real money spent are the goals for the cash flow.

5.7 Evaluation in a Static Scenario

In order to analyze and present the influence of the key simulation parameters mentioned before we conduct simulation runs using a static chain topology. We use two multi-hop cellular networks with 4 nodes and 1 gateway each as well as an interconnecting router. We place the nodes at 200 meters distance from each other. We install a constant bit rate, CBR traffic source on each node, where each node in the multi-hop cellular network I (1, 2, 3 and 4) transmits packets of 512 bytes

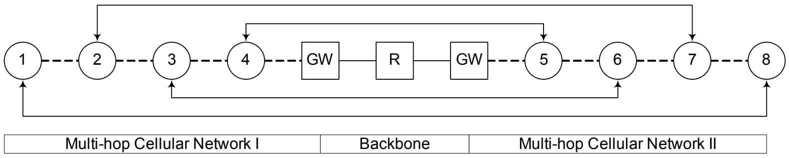


Figure 5.5: Static simulation scenario

	Scenario					
	1	2	3	4	5	6
Parameter	Initial	PGI	SS	R	TC	PCAT
Packet generation interval, PGI [s]	1	2	1	1	1	1
Service Station(s), SS	0	0	8	0	0	0
Reseller(s), R	0	0	0	8	0	0
Transceiving Cost, TC [TC]	hop count	hc	hc	hc	2	hc
Packet counter ACK threshold, PCAT	1	1	1	1	1	4

Table 5.2: Key simulation parameter settings for the 6 static scenarios

length to its opposite node in multi-hop cellular network II (8, 7, 6, and 5) and vice versa as shown in Figure 5.5. We choose a total simulation time of 300 seconds.

We created an initial scenario and five additional scenarios. In each we change a single parameter compared to the initial scenario respectively. Table 5.2 lists the parameter settings for all six scenarios. In the initial scenario, the 8 traffic sources transmit packets every 1 second and neither service stations and nor resellers are available. The transceiving cost for each node are equal to the hop count to the gateway from the respective node. Also, every forwarded packet is rewarded immediately. In the second scenario, we increase the packet generation interval to 2 seconds. Next, we deploy 8 service stations, one next to each node. In the fourth scenario, we allow each node to resell its traffic credits for helper credits. Then, we use an equal transceiving cost of 2 traffic credits for every node. Last, we increase the packet counter ACK threshold to 4, so that only every fourth packet is rewarded.

Table 5.3 lists the parameter values we used for each CASHnet node in the initial scenario. CASHnet is a micro-payment scheme, where each transmitted packet is charged at the originator and recipient as well as rewarded by intermediate nodes. Thus, the traffic credits account is quickly depleted. For the static scenarios we choose a relatively low initial amount of traffic credits and a 1:1 exchange rate between real money and traffic credits in order to provoke packet loss and thus starvation to better demonstrate the influence of the key simulation parameters.

The goal of these simulation runs in a static chain topology, is to show the influence of each CASHnet key parameter on the network. In our analysis, we use the

	Parameter	Value
	Initial traffic credits account state [TC]	500
	Initial real money account state [RM]	500
Trade	Trade threshold at service station [HC]	10
	Trade traffic credits amount [TC]	20
	Helper/traffic credits rate	1:1
	Real money/traffic credits rate	1:1
	Distance threshold to service stations [m]	50
Resale	Reseller traffic credits threshold [TC]	100
	Reseller traffic credits amount [TC]	20
	Reseller traffic/helper credits rate	1:1
	Buyer traffic credits threshold [TC]	50
	Buyer helper credits threshold [HC]	10
	Exchange interval [s]	1

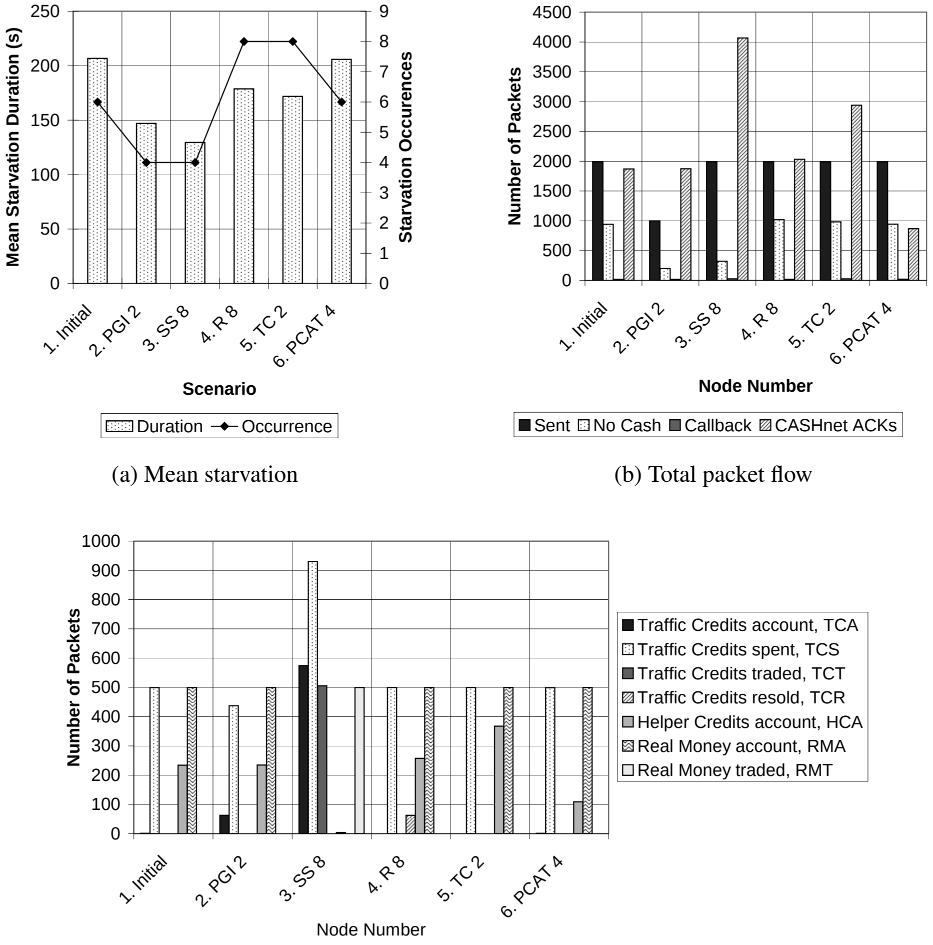
Table 5.3: CASHnet node parameter settings for the initial static scenario

previously defined evaluation criteria. We also visualize our results in a variety of graphs to justify our reasoning. The graphs are divided into two categories: In the first we show the overall network performance for each scenario using mean and total values for the evaluation criteria (see Figure 5.6). Figure 5.6a shows the mean starvation duration and the number of starvation occurrences, Figure 5.6b breaks down the total packet flow, and Figure 5.6c visualizes the mean cash flow. In the second category, we illustrate the individual node performance for each scenario (see Figure 5.7, 5.8 and 5.9 on page 99, 100 and 101 respectively). In the following paragraphs, we describe and analyze the results for each of the six scenarios in detail.

5.7.1 Initial Scenario

In the initial scenario the nodes starve on average 207 seconds, over 2/3 of the simulation time and 6 out of 8 nodes are actually starving as can be seen in Figure 5.6a. From Figure 5.6b we find, that half of the packets sent are dropped due to the lack of traffic credits (No Cash) and the amount of reward messages (CASHnet ACKs) is almost equal to the amount of transmitted packets. The mean cash flow in Figure 5.6c indicates that all traffic credits (TCA, TCS) have been spent, that neither trade (TCT) nor resale (TCR) occurred. On average 230 helper credits (HCA) have been earned.

The number of dropped packets because of traffic credits shortage is very high and thus is the starvation duration (see Figure 5.8a and 5.7a). The closest nodes to the gateway (4 and 5) never starve, because their transceiving cost is equal to 1 traffic credit. With an initial traffic credits account of 500, the cost for the transmission and reception of all packets (2 x 249) is just covered. Therefore, the accounts of



(a) Mean starvation

(b) Total packet flow

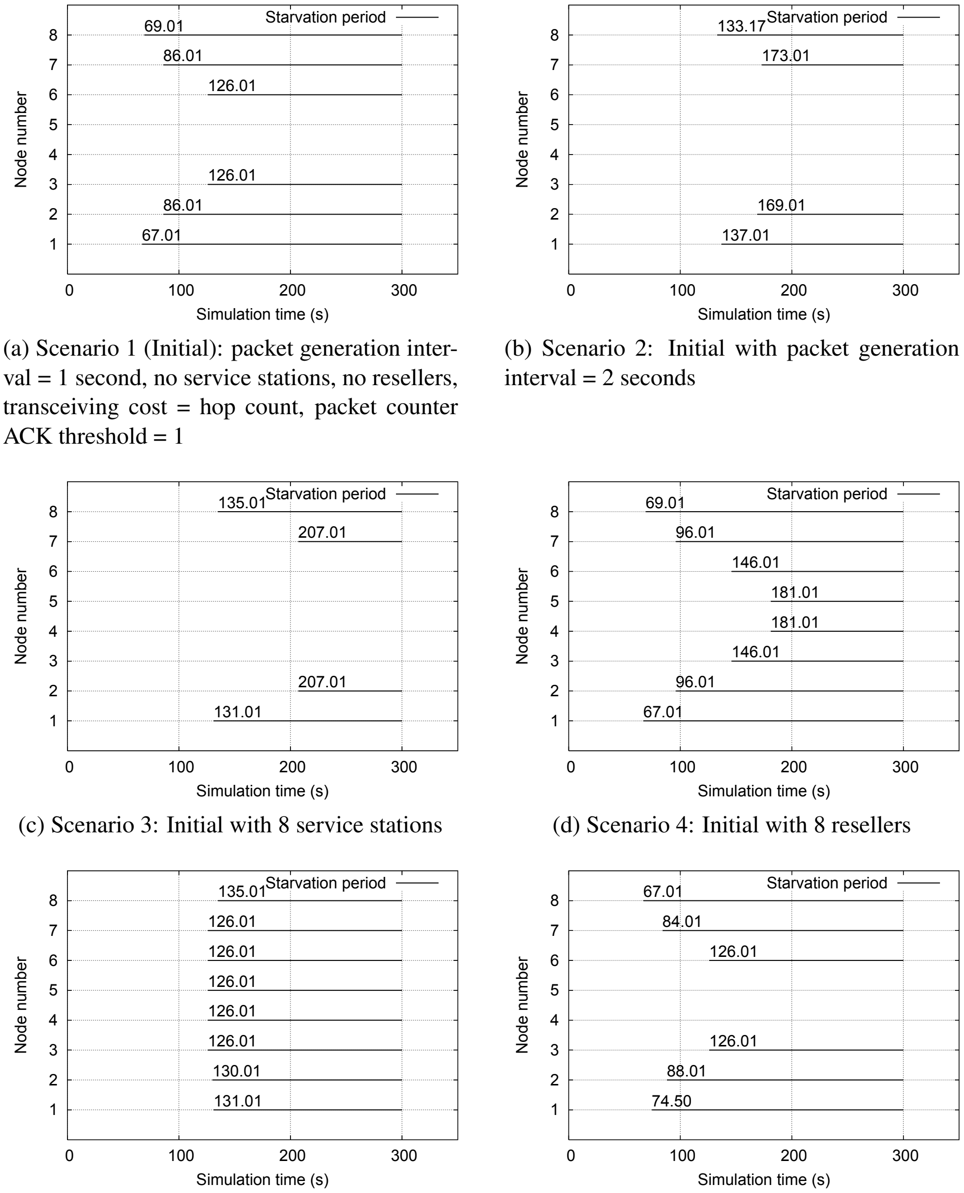
(c) Mean cash flow

Figure 5.6: Overall performance under different parameters in the static scenarios

nodes 4 and 5 are depleted but the nodes do not starve as illustrated in Figure 5.9a.

5.7.2 Packet Generation Interval Scenario

In the second scenario, we increase the packet generation interval (PGI) to 2 seconds, which reduces the total number of transmitted packets by 50%. As a result, the average starvation duration falls to 147 seconds, half of the simulation time and leads to half of the nodes starving. Now, 25% of the packets sent are dropped due to missing traffic credits. The number of reward messages has the double amount of packets sent. The mean cash flow shows, that on average 60 traffic credits per node remain at the end of the simulation. The remaining values are close to the initial scenario.



(a) Scenario 1 (Initial): packet generation interval = 1 second, no service stations, no resellers, transceiving cost = hop count, packet counter ACK threshold = 1

(b) Scenario 2: Initial with packet generation interval = 2 seconds

(c) Scenario 3: Initial with 8 service stations

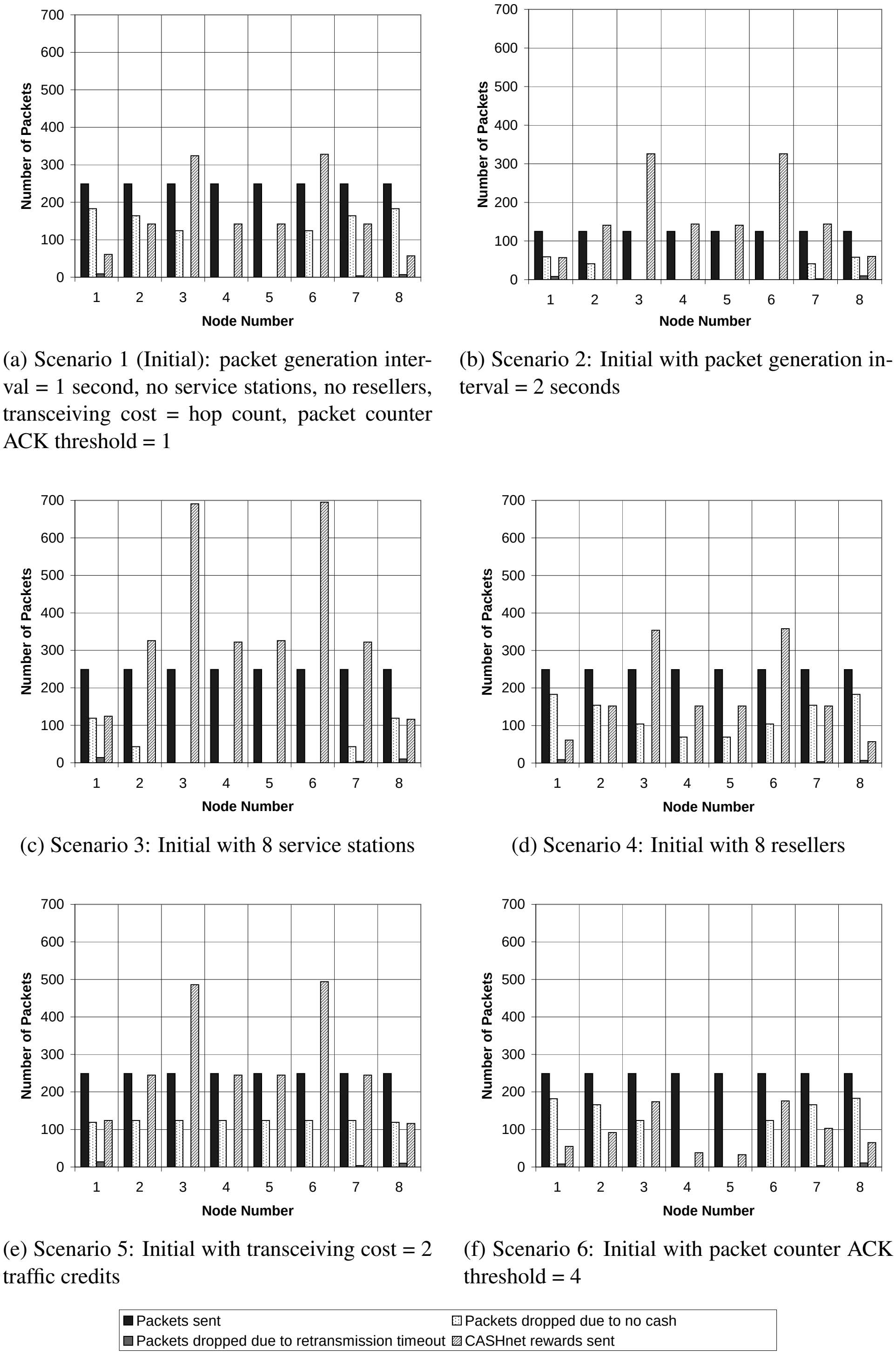
(d) Scenario 4: Initial with 8 resellers

(e) Scenario 5: Initial with transceiving cost = 2 traffic credits

(f) Scenario 6: Initial with packet counter ACK threshold = 4

Figure 5.7: Starvation per node under different parameters in the static scenarios

In this scenario, we can see that not all nodes are able to spend their initial amount of 500 traffic credits before the simulation ends. Because of the increased packet generation interval, which reduces the network load, each node sends and receives 125 packets respectively. Thus, nodes 4 and 5 have 250 traffic credits remaining and nodes 3 and 6 with a transceiving cost of 2 traffic credits just managed



(a) Scenario 1 (Initial): packet generation interval = 1 second, no service stations, no resellers, transceiving cost = hop count, packet counter ACK threshold = 1

(b) Scenario 2: Initial with packet generation interval = 2 seconds

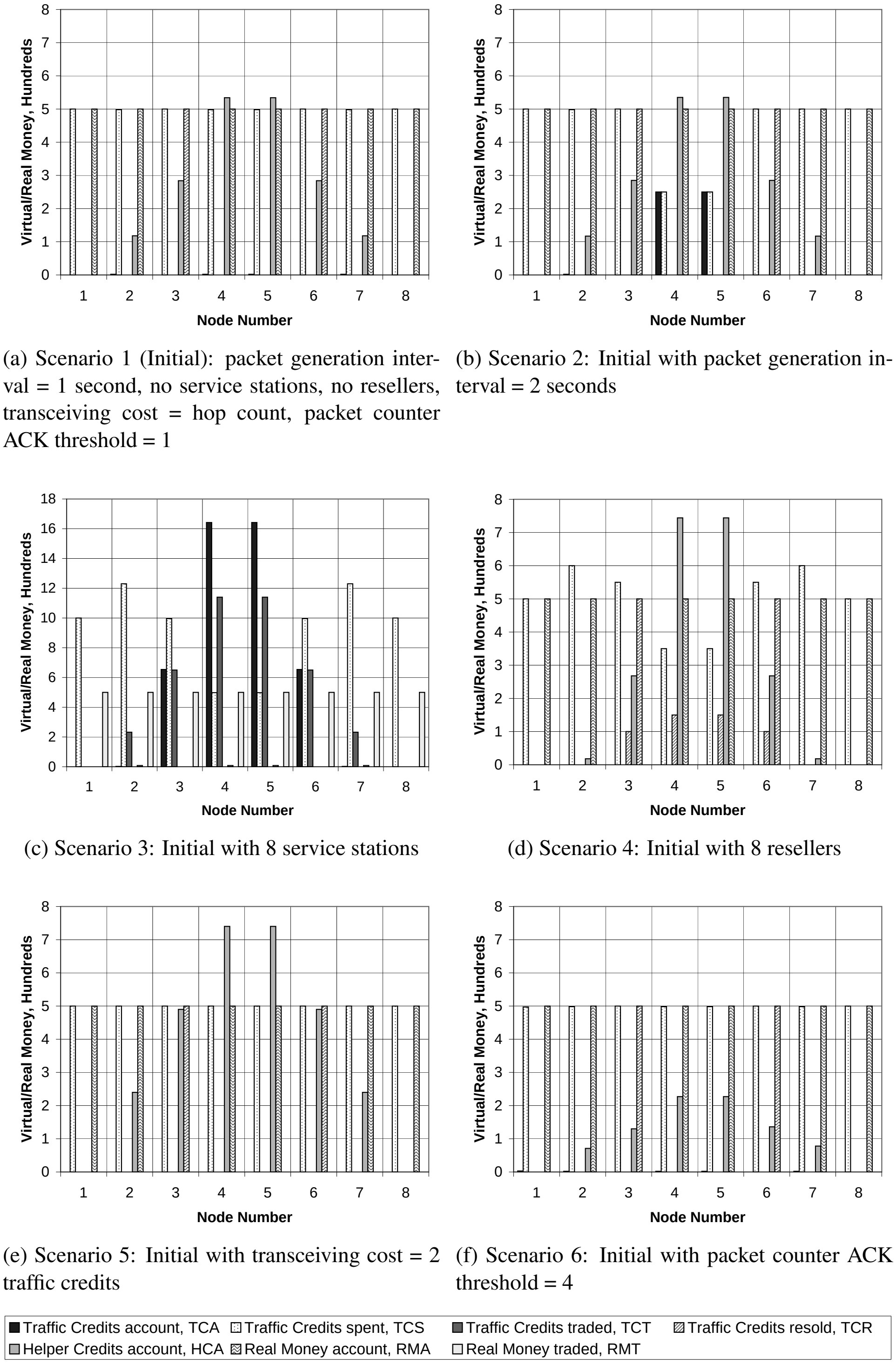
(c) Scenario 3: Initial with 8 service stations

(d) Scenario 4: Initial with 8 resellers

(e) Scenario 5: Initial with transceiving cost = 2 traffic credits

(f) Scenario 6: Initial with packet counter ACK threshold = 4

Figure 5.8: Packet flow per node under different parameters in the static scenarios



(a) Scenario 1 (Initial): packet generation interval = 1 second, no service stations, no resellers, transceiving cost = hop count, packet counter ACK threshold = 1

(b) Scenario 2: Initial with packet generation interval = 2 seconds

(c) Scenario 3: Initial with 8 service stations

(d) Scenario 4: Initial with 8 resellers

(e) Scenario 5: Initial with transceiving cost = 2 traffic credits

(f) Scenario 6: Initial with packet counter ACK threshold = 4

Figure 5.9: Cash flow per node under different parameters in the static scenarios

to pay the last packet as presented in Figure 5.9b. All 4 nodes do not starve as it can be seen in Figure 5.7b. As expected, the packet generation interval controls the network load. We see that the initial 500 traffic credits are still not enough to avoid starvation for all nodes as CASHnet charges per packet.

5.7.3 Service Stations Scenario

In the third scenario, we place a service station (SS) next to each node (8 in total), enabling the nodes to trade traffic credits against helper credits and real money. This reduces the average starvation duration to 130 seconds, 43% of the simulation time. 18% of the transmitted packets are dropped due to shortage of traffic credits. The number of reward messages is again double as high as the number of transmitted packets. In the mean cash flow, we see that on average 570 traffic credits remain at the end of the simulation and 930 traffic credits have been spent per node. Also, 500 traffic credits have been traded for helper credits on average per node. In addition, the complete amount of real money has been used to trade traffic credits at the service stations.

The service stations greatly improve the overall network performance as presented in Figure 5.8c. All nodes are able to exchange their helper credits and real money for traffic credits. However, as can be seen in Figure 5.9c the two nodes at the border (1 and 8) never obtain any helper credits since they have no packets to forward. Combined with the high transceiving cost of 4 traffic credits per packet, these nodes start to starve soon and are followed by their neighbor nodes (2 and 7) as shown in Figure 5.7c. Service stations have a huge positive impact on the overall network performance, as they are the source for new traffic credits for nodes. Of course their placement needs to be well planned so that the nodes can reach them.

5.7.4 Resellers Scenario

In the fourth scenario, we allow each node to resell (R) its traffic credits for helper credits (8 in total). The average duration of starvation is now 180 seconds, and all 8 nodes are starving. The packet flow performance is similar to the initial scenario, with a little more drops caused by traffic credits shortage and more reward messages. The cash flow is also similar to the initial scenario, with the exception of 60 traffic credits resold on average per node.

We see that the reseller functionality distributes the traffic credits in the network. Nodes with a small hop count to the gateway (3, 4, 5 and 6) fall at a later time below the reseller traffic credits threshold and can thus resell their traffic credits to their one-hop neighbors for a longer time period. The final helper credits account state of all nodes in Figure 5.9d illustrates this situation very well. This leads to all nodes starving, while the starvation duration for the intermediate nodes (2, 3, 6 and 7) is reduced compared to the initial scenario as show in Figure 5.7d. Because resellers can not introduce new traffic credits in the network, the overall performance is very similar to the initial scenario. But we find that resellers allow us to balance the distribution of traffic credits.

5.7.5 Transceiving Cost Scenario

In the fifth scenario, we use equal transceiving costs (TC) of 2 traffic credits for all nodes instead of the individual hop count to the gateway. Although the packet flow is similar to the initial scenario, the average starvation duration is decreased to 170 seconds, but all 8 nodes starve. The average cash flow is also similar to the initial scenario, with the exception of an average final account state of 370 helper credits.

An equal transceiving cost leads to a balanced consumption of traffic credits in the network. Therefore, all nodes approximately run out of traffic credits at the same time resulting in almost equal starvation durations for each node as depicted in Figure 5.7e. For the nodes far away from the gateway, the equalization of transceiving cost to 2 traffic credits is an advantage as they are now able to transmit more packets. This gives the intermediate nodes the possibility to earn more helper credits by forwarding packets for the outbound nodes as shown in Figure 5.9e and Figure 5.8e. However, the nodes can not take advantage of the helper credits due to the lack of service stations and so the overall performance stays close to the initial scenario. We find, that the equal transceiving cost removes a node's dependency on its location, as the costs are not anymore bound to the hop count towards the gateway.

5.7.6 Packet Counter ACK Threshold Scenario

In the last scenario, we increase the packet counter ACK threshold (PCAT), so that only every 4th forwarded packet from the same forwarder is rewarded. The results for the average starvation duration, the total packet flow and the average cash flow are almost equal to the initial results, with the exception in the number of reward messages, which is now less than half of the transmitted packets.

The increased packet counter ACK threshold greatly reduces the number of transmitted rewards and thus the network load as can be seen in Figure 5.8f. However, as the main drop reason in the initial scenario is the unavailability of traffic credits, rewarding only every 4th packet has almost no effect on the starvation duration as illustrated by Figure 5.7f. Figure 5.9f shows that the amount of helper credits earned is much less compared to the initial scenario. However as explained before the helper credits value at the service station is bound to the packet counter threshold. Under low network load, the reduction of reward messages has no considerable impact.

5.7.7 Summary

The evaluation of the key simulation parameters demonstrates the impact of each parameter. The results from the increase of the packet generation interval emphasizes the micro-payment properties of CASHnet. A node requires a large amount of traffic credits to be able to continuously transmit under high network load until it reaches a service station to refill its account. The service stations play a major role in CASHnet, as they allow the node to reload its traffic credits account by exchanging helper credits or real money. The optimal deployment of the service stations is important, especially under node mobility. The usage of resellers balances the distribution of traffic credits in the network. They can be used in addition to service station to help a node overcome a period of traffic credits shortage.

When we set a fixed and equal transceiving cost, the charges become independent of a node's location in the network. It is important for the provider to choose a global transceiving cost which covers the average expenses. For example, the average hop count can be used as an indicator for the expenses and the revenue can be regulated via the value of the trade rates for the traffic credits. In addition, the (accurate) hop count information from the routing protocol is not required anymore. The packet counter threshold aims to reduce the load put on the network. Because it reduces the amount of reward messages sent, it increases their value at the same time. Thus, the loss of such a message is more costly. The best solution would be an adaptive threshold, however this requires information about the current and future network condition which is beyond the scope of our CASHnet scheme.

5.8 Evaluation in a Mobile Scenario

CASHnet provides a cooperation and accounting framework for hybrid networks. We want to analyze the impact of CASHnet on a multi-hop cellular network and evaluate its performance as well as the distribution of the virtual currencies. To do so, we analyze the starvation, the packet flow and the cash flow for all nodes as well as for each individual node. The results from the evaluation of the key simulation parameters in the previous section give us several indicators of the parameters' impact on the overall network performance. Using these hints as a base, we perform extensive evaluations of CASHnet under a variety of scenarios.

In order to create a more realistic simulation environment than the previously used chain topology, we require node mobility over a larger area. We generated the node movements based on the random waypoint mobility model as described in Section 5.5.1 with the help of the program called *setdest* included in ns-2. We used the movement parameters listed in Table 5.4 to obtain the movement files. In total, we generate 20 movement files, to average our simulation results. We distribute 40 nodes over an area of 900x600 meters. The nodes move with an average speed of approximately 5 meters per second and take an average pause of 10 seconds for a total simulation time of 900 seconds. This results in a rather dense node distribution, which is required to achieve a minimum connectivity among the nodes and thus create a multi-hop cellular network. In the literature, the minimum number of neighbors, a node needs to be connected to in order for the total network to be connected, is assumed to be between 6 and 8. Takagi and Kleinrock [TK84]

Parameter	Value
Number of nodes	40
Speed type	uniformly distributed within $< v_{min}, v_{max} >$
Minimum speed v_{min}	1 m/s
Maximum speed v_{max}	10 m/s
Simulation time	900 s
Pause type	uniformly distributed within $< 0, 2 * p >$
Pause time (median) p	10 s
X dimension of space	900 m
Y dimension of space	600 m

Table 5.4: Parameters of the node movement generation

Parameter	Value
Packet generation interval [s]	0.5, 1, 2
Service station(s)	1, 2, 5, 9
Reseller(s)	0, 4
Transceiving cost [TC]	hop count, 3
Packet counter ACK threshold	1, 5, 10, 15, 20
Movement scenarios	1..20

Table 5.5: Key simulation parameters for the mobile scenarios

derived these numbers for randomly distributed packet radio terminals.

Like in the static simulation scenarios, we vary the five key parameters, which we introduced before. We use packet generation intervals of 0.5, 1, and 2, which correspond to packet generation rates of 2, 1 and 0.5 packets per second per node. We also change the number and location of deployed service stations in a movement scenario between 1 (A), 2 (B), 5 (C) and 9 (D). Figure 5.10 illustrates the four variations of service stations and their respective scenario letter. In each scenario a gateway with both a wireless and a wired network interface is placed at the right border. The gateway provides the interconnection between the multi-hop cellular network and the backbone of the provider.

Further, we allow 0 or 4 nodes to act as resellers. Also, we test the effect of dynamic and fixed transceiving cost. For the dynamic transceiving cost, we use the node's current hop count to the gateway, for the fixed transceiving cost we use 3 traffic credits, a value which we derived from the analysis of the route lengths in our plain simulation scenario (see Section 5.8.1). In addition, we investigate the effect of different packet counter ACK thresholds, i.e. the number of forwarded packets a node receives, before it sends a reward. A node rewards a forwarding node after receiving 1, 5, 10, 15 or 20 packets. Table 5.5 illustrates the various key simulation parameter settings.

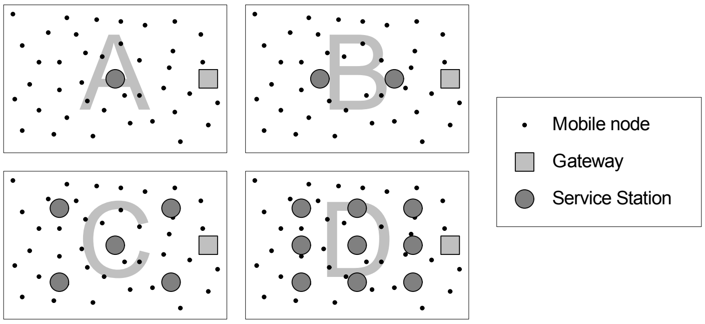


Figure 5.10: Service station variations and resulting scenarios A, B, C and D

Table 5.6 lists the parameter values of each CASHnet node in our mobile simulation scenarios. We give each node an initial value of 1000 traffic credits. For example, with an average hop count of 4 to the gateway, a node can send 250 packets before its account is depleted. Also, each node has 200 real money, which is worth 4000 traffic credits at a service station, since we set the respective exchange rate to 20:1. We specify a maximum trade amount of 500 traffic credits for a single transaction in order to allow nodes to sustain periods where they don't encounter any service station. Traffic credits and helper credits are exchanged equally. We consider the customer of a node able to engage in a trade, if she is within 50 meters distance to a service station.

In real life reseller nodes will have special discounts with the provider, because their trade volume is much higher compared to a normal node. To match the properties of resellers, we give them a higher initial traffic credits and real money account. We also set the exchange rates for helper credits and real money higher than for normal nodes. Since a reseller generates packets like a normal node, the agent allows the resale only if more than 500 traffic credits are available. A reseller exchanges a maximum of 50 traffic credits at once, giving more nodes the opportunity to engage in a resale. We set the same exchange rate for traffic/helper credits exchange rate as the service station, because the reseller makes profit from the advantageous trade conditions with the provider at the service station.

Finally, we specified the exchange interval so that every second a node checks whether a service station or a reseller is within its vicinity, i.e. within the node's 50 meter radius or in its one-hop neighborhood respectively.

5.8.1 Plain Multi-hop Cellular Network Performance

As a basis for comparison we evaluate a multi-hop cellular network without any CASHnet functionality, i.e. we assume the nodes cooperate (forward packets) without any cooperation scheme. Only the mobile node settings in Table 5.1, the different packet generation intervals and the 20 movement scenarios are used. This allows us to compare a multi-hop cellular network without a cooperation framework to one with CASHnet. Figure 5.11 shows our plain mobile simulation sce-

	Parameter	Value
		Standard (Reseller)
	Initial traffic credits account state [TC]	1000 (10000)
	Initial real money account state [RM]	200 (1000)
Trade	Trade threshold at service station [HC]	10
	Trade traffic credits amount [TC]	500
	Helper/traffic credits rate	1:1 (1:2)
	Real money/traffic credits rate	1:20 (1:30)
	Distance threshold to service stations [m]	50
Resale	Reseller traffic credits threshold [TC]	500
	Reseller traffic credits amount [TC]	50
	Reseller traffic/helper credits rate	1:1
	Buyer traffic credits threshold [TC]	100
	Buyer helper credits threshold [HC]	10
	Exchange interval [s]	1

Table 5.6: CASHnet node parameter settings for the mobile scenario

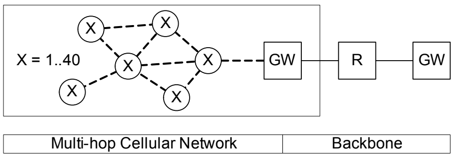


Figure 5.11: Plain mobile simulation scenario

nario. 40 nodes roam around in an area of 900x600 meters and transmit packets at a constant bit rate and with 512 bytes length towards the gateway at the specified packet generation rate. We analyze the packet flow and the route lengths in this plain mobile simulation scenario.

Packet Flow

The packet flow is an indicator for the network performance. We measure it by analyzing the trace files from ns-2 and consider the total packet flow of the whole network. Figure 5.12 illustrates the mean goodput for all nodes and for each packet generation interval averaged over the 20 movement scenarios. For a packet generation interval of 0.5 seconds, we measure a mean goodput of 63% with a standard deviation of 3%. For both packet generation intervals of 1 and 2 seconds, we measure a goodput of 81% and a standard deviation of 2% respectively. The drop reasons give us some hints as to where and why the losses occur.

Figure 5.13 breaks down the average number of dropped packets into their drop reasons for the 20 movement scenarios. We include the following packet drop

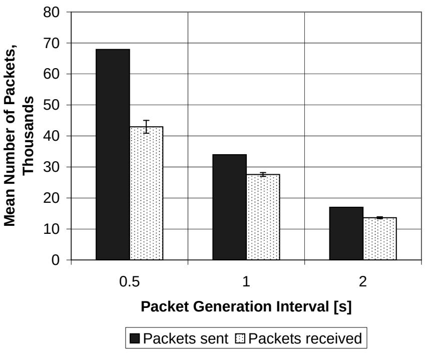


Figure 5.12: Goodput in the plain mobile scenario

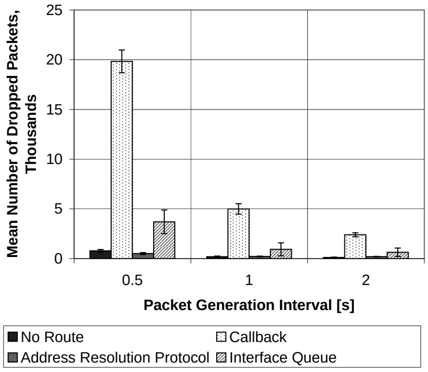


Figure 5.13: Drop reasons in the plain mobile scenario

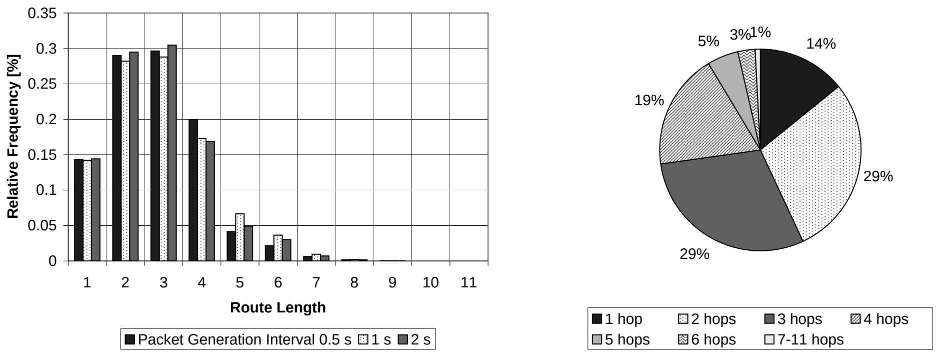
reasons: No Route, Callback, Address Resolution Protocol and Interface Queue, which we explain in Section 5.6 on page 94. We exclude the packets dropped due to routing loops (Loop) as their number is very small (< 27, < 7 and < 2 for the packet generation interval 0.5, 1 and 2 respectively).

The main drop reason is Callback, an event which occurs then AODV receives a link break notification from the link layer in ns-2. Link breaks are typically caused by mobility. However, when we compare the values for the different packet generation intervals, we see that doubling the packet generation rate leads to an exponential increase in packet drops due to Callback. Thus, we conclude that also congestion leads to a link break notification. The second highest number of drops occur in the interface queue of each node. This is mainly caused by the length of the queue, which is set to the default value of 50 packets. However, simply increasing the queue length leads to an increase of outdated packets, because the network topology changes quickly. This leads to an increase in Callback drops, as the neighbors have moved away and a link break notification is given.

We attribute the No Route drops to the node mobility and the hybrid gateway discovery. We specified a gateway advertisement zone with a radius of 4 hops. Therefore, nodes located at 5 or more hops from the gateway, will not receive the gateway advertisements and have to reactively acquire a route to the gateway, which causes a delay. The address resolution protocol (ARP) drops indicate a delay in the reply of the address resolution protocol. While waiting for the ARP reply for a specific address, ns-2 drops any further packet for the same destination.

Route Length

The route length is a good indicator for the network size and the geographical distribution of transmitting nodes. We measure it by registering the number of hops to the gateway on each node for each transmitted packet. Figure 5.14 presents the relative frequency of route lengths, which we obtained by summing up the hop



(a) Route length distribution for packet generation interval 0.5, 1 and 2 seconds

(b) Combined route length distribution

Figure 5.14: Relative frequency of route lengths in the plain mobile scenario

count occurrences on each node in all 240 permutations resulting from 3 packet generation intervals, 20 movement patterns and 40 nodes.

Figure 5.14a shows the route length distribution for each packet generation interval. A smaller packet generation interval results in a higher packet generation rate and thus in a higher hop count occurrence. Also, the geographical transmission location of a node changes among the different packet generation intervals. However, we see that the route length distribution is very similar among the different intervals.

Figure 5.14b shows the combined route length distribution. We can see that 58% of the occurrences are for route lengths of 2 and 3 hops. That means in our simulation area, almost 2/3 of all transmissions occur at a distance of 2 or 3 hops from the gateway. If we look at route lengths between 1 and 4 hops, we comprise 91% of all occurrences. The next 8% cover route lengths of 5 and 6 hops. The high percentage of short routes can be explained with the size of the simulation area of 900x600 meters and the node density of 40 nodes. We obtain an average route length of 2.83 hops, which we use as an indicator for the fixed transceiving cost in CASHnet. Short average route lengths improve the quality of the connection, because every additional hop decreases the throughput. At the same time, the high node density indicates a crowded wireless medium, which decreases the quality of the connection.

Summary

We find, that in a plain multi-hop cellular network with AODV+ in hybrid gateway discovery mode, the network performance is not optimal, especially under high network load, mobility and congestion decrease the throughput to 63% while with a packet generation interval of 1 and 2 seconds the goodput is 81%. Although obtained in a simulation, these results show, that the technology and protocols for

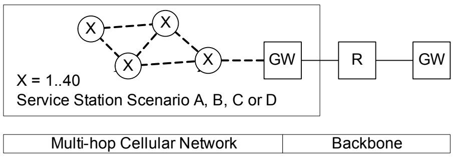

X
X
X
X
GW
R
GW
X = 1..40
Service Station Scenario A, B, C or D
Multi-hop Cellular Network
Backbone

Figure 5.15: CASHnet mobile simulation scenario

mobile ad hoc and multi-hop cellular networks still leave room for improvements.

As we want to investigate the impact of CASHnet on a multi-hop cellular network with current technologies, we take these results as a base for our comparison. Any improvements in the base technologies and protocols of multi-hop cellular networks will of course have a positive effect on the results of CASHnet.

5.8.2 CASHnet Performance

After analyzing the performance of a plain multi-hop cellular network without any cooperation scheme, we deploy CASHnet in the network. We create our simulation scenarios by varying the key parameters listed in Table 5.5 on page 105, which are the packet generation interval (PGI), the number of service stations (SS) and reseller (R) nodes, the transceiving cost (TC), the packet counter ACK threshold (PCAT) and the movement scenarios (MS). This results in a large number of simulation scenarios, i.e. 3 PGI x 4 SS x 2 R x 2 TC x 5 PCAT x 20 MS = 4800 simulation runs. We use the CASHnet parameters shown in Table 5.6 on page 107.

Figure 5.15 shows the schematic for our CASHnet mobile simulation scenario, which is identical to the one of the evaluation of the plain scenario except that we deploy service stations as shown in Figure 5.10. 40 nodes roam around in an area of 900x600 meters and transmit packets at a constant bit rate and with 512 bytes length towards the gateway at the specified packet generation rate. We analyze the starvation, the packet flow and the cash flow in these CASHnet mobile simulation scenarios.

In the following figures, we show mean values, which we obtained by averaging the results from the 20 simulation scenarios and - in case of the starvation and the cash flow - the 40 contained nodes. The packet flow results already represent the overall network performance. We also show the corresponding standard deviation. Thus, we obtain averaged results for 240 scenarios (PGIxSSxRxTCxPCAT). To present the results of one scenario we require one figure for the starvation, three for the packet flow and one for the cash flow. Each figure includes all values from the combinations of the packet generation interval and the number service stations (3x4) indicated in the second and first line in the x-axis title respectively. As we can not show every possible combination of the other key parameters, we restrict ourself to the most significant cases.

In Table 5.7 we list the parameter settings of the selected combinations. In the first scenario we use no reseller, transceiving costs related to the hop count

	Scenario				
Parameter	1	2	3	4	5
Packet generation interval, PGI [s]	0.5, 1, 2				
Service Station(s), SS	1, 2, 5, 9				
Reseller(s), R	0	0	4	4	0
Transceiving Cost, TC [TC]	hop count				3
Packet counter ACK threshold, PCAT	1	10	1	10	1

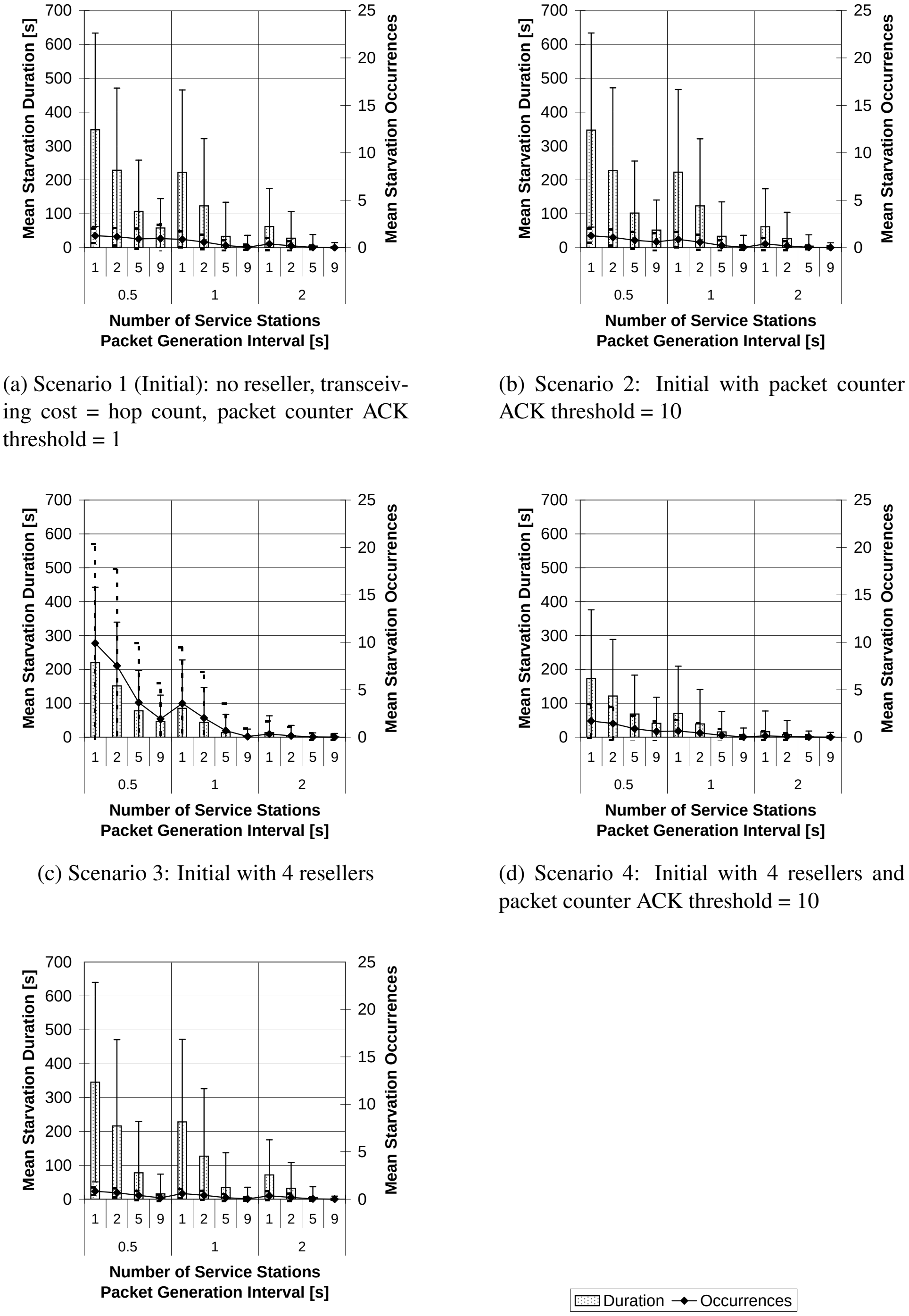
Table 5.7: Parameter settings for the CASHnet mobile scenarios

and acknowledge every packet. In the second, we change the packet counter ACK threshold, so that every tenth packet is acknowledged. Note, that this implicates an increase of the value of a single acknowledgement by 10. From our results, we found the performance to be inferior for lower (1, 5) and higher (15, 20) packet counter ACK thresholds compared to 10. In case of the lower values, we attribute this to the still high overhead, in case of the higher values, to the increased value of lost acknowledgements. Next, we give 4 nodes (out of 40) resale abilities. The resellers have special CASHnet parameter settings, which we presented in Table 5.6 on page 107. In the fourth scenario, we combine resellers and the packet counter ACK threshold. Finally in the last scenario, we show the impact of a globally fixed transceiving cost set to 3 traffic credits. We derive this value from the average hop count, which we determined in the previous evaluation of the plain simulation scenario.

Starvation

The starvation indicates how long and how often a node is neither allowed to transmit self-generated packets nor receive packets destined to the node over the simulation time of 900 seconds. Starvation can be affected by different influences. The direct cause is the lack of traffic credits at the moment of transmission or reception. This in turn, can be caused by few traffic credits refill opportunities (e.g. service stations) on the movement path of each node. Another reason can be a high average hop count towards the gateway, which leads to a fast decrease in traffic credits. And last, the loss of acknowledgement leads to fewer helper credits on the node and thus less traffic credits when exchanging the helper credits at the service station. Figure 5.16 shows the mean starvation duration and occurrences for the 5 scenarios, which we previously introduced. The figures on starvation show the mean duration as bars and the occurrences as points with standard deviation in bold and dashed lines respectively.

From the results of first scenario in Figure 5.16a, we see that for the highest traffic load (packet generation interval 0.5 seconds) and 1 service station, the mean starvation duration per node is around 2/3 of the simulation time with an average of 1.2 starvation occurrences (periods). Increasing the number of service stations



(a) Scenario 1 (Initial): no reseller, transceiving cost = hop count, packet counter ACK threshold = 1

(b) Scenario 2: Initial with packet counter ACK threshold = 10

(c) Scenario 3: Initial with 4 resellers

(d) Scenario 4: Initial with 4 resellers and packet counter ACK threshold = 10

(e) Scenario 5: Initial with transceiving cost = 3 traffic credits

Figure 5.16: Starvation in CASHnet for different parameters

to 9, reduces the average starvation duration to 58 seconds. As expected, reducing the network load also reduces the average starvation duration significantly. In the scenario with a packet generation interval of 2 seconds, almost no starvation occurs with 5 and 9 service stations. The standard deviation for the duration is very high. The average number of starvation occurrences per node is around 1. The high standard deviation indicates that some nodes almost never starve, while others starve for almost the double of the average duration. This is due to the random movement of the nodes, which decides about their probability of meeting a service station. The low number of average occurrences indicates that the few and long starvation periods occur.

In the second scenario, we increase the packet counter ACK threshold to reward only every 10th packet. Figure 5.16b shows the results. Compared to the first scenario, we notice a slight reduction in the starvation duration, e.g. for the scenario with the highest traffic load and 9 service station, the average starvation duration is now 52 seconds. The standard deviation is very similar to the first scenario.

Figure 5.16c shows the results of the third scenario, where we allow 4 nodes out of 40 to act as resellers. We notice the highest impact for scenarios with only 1 or 2 service stations, where the starvation duration is reduced by 40 to 50% compared to the first scenario. At the same time, the number of starvation occurrences increase, which indicates that on average the nodes starve more often, but for shorter time periods. The standard deviation for the duration is high as in the initial scenario, for the occurrences it is significantly increased. This reflects that some nodes are able to meet a reseller, where they can exchange their helper credits for traffic credits and thereby interrupting their starvation period more often.

The results in Figure 5.16d show the impact from the combination of resellers and packet counter ACK threshold of the fifth scenario. We find a considerable decrease of the starvation duration compared to the first scenario in all combinations of traffic load and number of service stations. We note especially the positive influence compared to the previous fourth scenario for the cases with only 1 and 2 service station. Also, the standard deviation for the occurrences is considerably reduced. The reduction of reward messages and the implied increase in their value to 10 helper credits supports best the scenarios with high traffic load.

In the last scenario, we set the transceiving cost to a fixed value of 3 traffic credits. Every node, wherever it is located in the network, has to pay this price whenever it wants to transmit a self-generated packet or receive a packet destined to it. Figure 5.16e presents the results. Compared to the first scenario, the results are slightly inferior. We attribute this to the fact, that some nodes have an average route length shorter than 3 hops towards the gateway, so that in this scenario, they actually have to spend more traffic credits and thus starve faster. This is underlined by the observation, that 43% of all routes to the gateway in the movement scenario are of length 1 or 2 as discovered in the analysis of the route length distribution in Figure 5.14 on page 109.

Packet Flow

The packet flow allows us to analyze the overall network performance of CASHnet and the specific reasons. The packet flow can be affected by many effects. The most direct is the number of simultaneously transmitted packets, either data or signaling (CASHnet acknowledgment). Other influences come from the size of the interface queues and the effects resulting from the interaction of the different protocols.

We analyze the goodput, which we define as the number of received packets, the different drop reasons for data packets and the overhead in terms of acknowledgements sent. Figure 5.17, 5.19 and 5.18 on page 115, 117 and 116 respectively illustrate our results. In the following, we combine the discussion of goodput, drop reasons and overhead for every of the five scenarios. We'd like to recall the results from the plain mobile scenario, where we measured a goodput of 63% for the packet generation interval of 0.5 second and 81% for 1 and 2 seconds respectively.

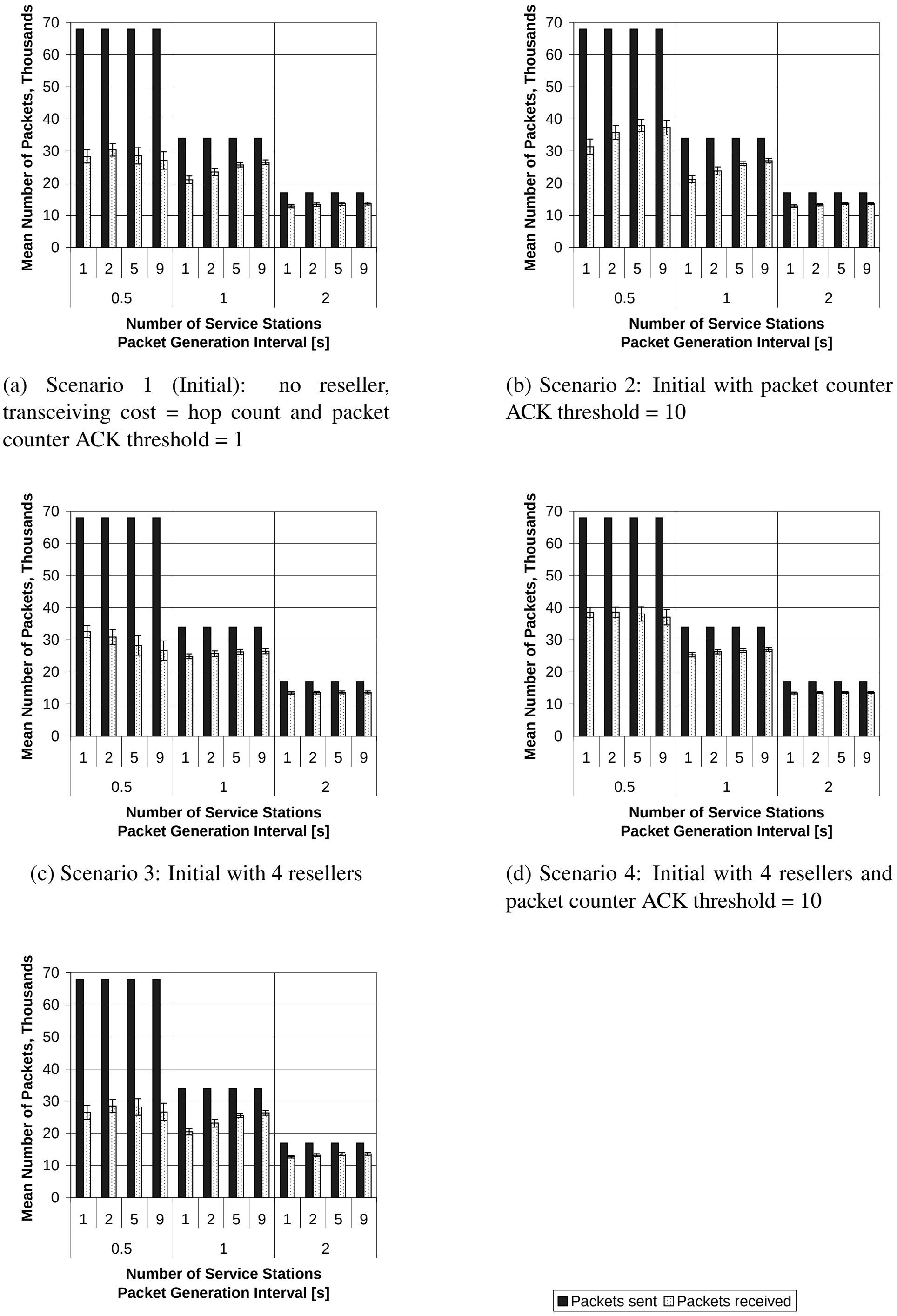
In the first scenario, we see in Figure 5.17a that the goodput is around 42% for the scenarios with high traffic load (packet generation interval 0.5 s). Even an increase in the number of service stations does not ameliorate the situation. To the contrary, the throughput decreases slightly. When we look at the drop reasons in Figure 5.19a, we find that while the number of packets dropped due to lack of money (No Cash) decreases, the packets dropped in the interface queue and the initiated drops by the MAC layer (Callback) increases. We found, that the link layer trigger in ns-2 interprets congestion as link break and commands AODV to search for a new route, which worsens the situation. A higher number of service stations leads to a higher number of acknowledgements, which we can see in Figure 5.18a.

We also observe, that the number of dropped acknowledgements increases with the number of service stations under high network load even though the number of transmitted acknowledgements does not increase. We attribute this phenomena to the congestion. With more service stations, the nodes are able to transmit more packets. As the network is already congested, these packets place an additional burden on the interface queues and thus block the transmission of acknowledgements.

In the second scenario, we witness the positive effect of the packet counter ACK threshold under high network load. Figure 5.17b shows an increase in goodput for these scenarios. When we compare the drop reasons depicted in Figure 5.19b with the one from first scenario, we see that the number of packets dropped by the interface queue is significantly decreased. From Figure 5.18b we find a reduction of about 30-40% in transmitted acknowledgements.

The third scenario investigates the influence of resellers. Figure 5.17c presents the results. We note a general increase in goodput compared to the first scenario, especially in the case with packet generation interval of 0.5 and 1 second. When we compare the drop reasons in Figure 5.19c to the first scenario, we see that the number of packets dropped due to lack of traffic credits (No Cash) is considerably decreased.

With the combination of 4 resellers and the increased packet counter ACK



(a) Scenario 1 (Initial): no reseller, transceiving cost = hop count and packet counter ACK threshold = 1

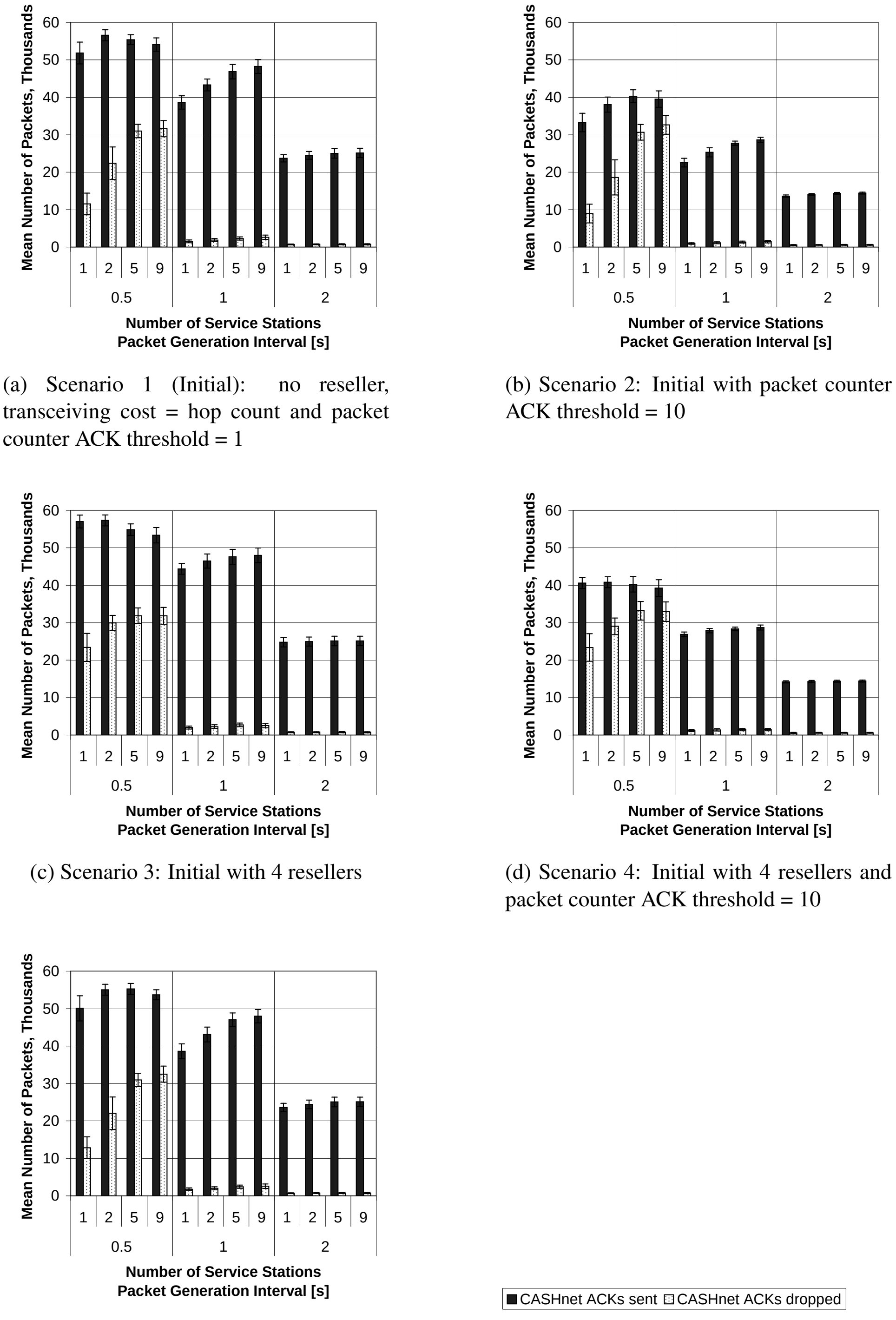
(b) Scenario 2: Initial with packet counter ACK threshold = 10

(c) Scenario 3: Initial with 4 resellers

(d) Scenario 4: Initial with 4 resellers and packet counter ACK threshold = 10

(e) Scenario 5: Initial with transceiving cost = 3 traffic credits

Figure 5.17: Goodput in CASHnet for different parameters



(a) Scenario 1 (Initial): no reseller, transceiving cost = hop count and packet counter ACK threshold = 1

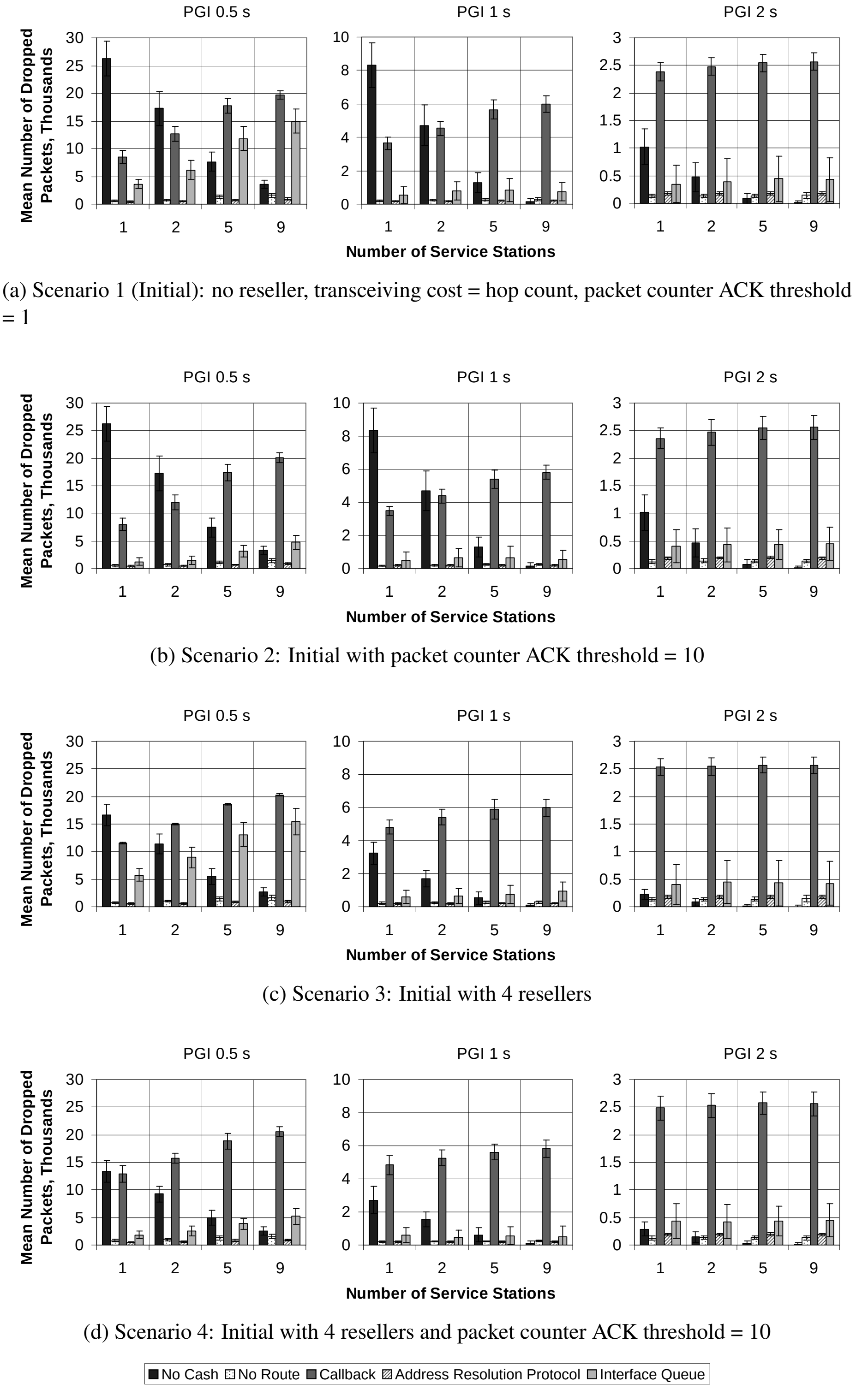
(b) Scenario 2: Initial with packet counter ACK threshold = 10

(c) Scenario 3: Initial with 4 resellers

(d) Scenario 4: Initial with 4 resellers and packet counter ACK threshold = 10

(e) Scenario 5: Initial with transceiving cost = 3 traffic credits

Figure 5.18: Overhead in CASHnet for different parameters



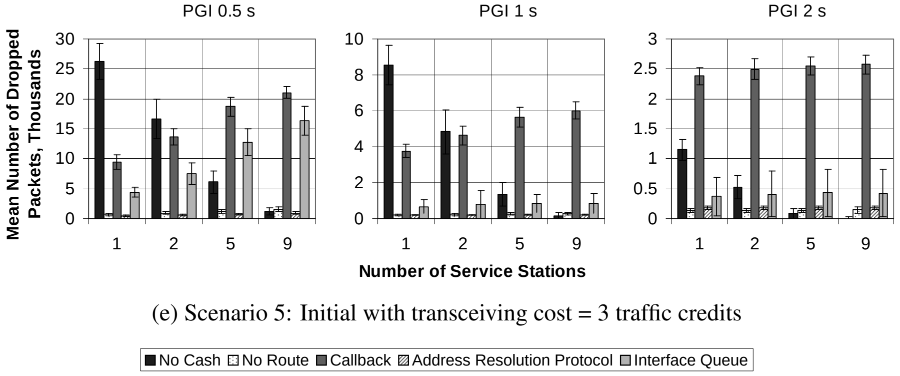
(a) Scenario 1 (Initial): no reseller, transceiving cost = hop count, packet counter ACK threshold = 1

(b) Scenario 2: Initial with packet counter ACK threshold = 10

(c) Scenario 3: Initial with 4 resellers

(d) Scenario 4: Initial with 4 resellers and packet counter ACK threshold = 10

Figure 5.19: Drop reasons in CASHnet for different parameters 1/2



(e) Scenario 5: Initial with transceiving cost = 3 traffic credits

Figure 5.19: Drop reasons in CASHnet for different parameters 2/2

threshold, we achieve the best results in this fourth scenario as shown in Figure 5.17d. The goodput is now at around 57% for the packet generation interval of 0.5 seconds and around 80% for 1 and 2 seconds respectively. These results are very close to the plain mobile scenario without any cooperation framework. In Figure 5.19d we see the reduction in dropped packets due to lack of money (No Cash) as well as due to a full interface queue compared to the first scenario. The overhead shown in Figure 5.18d is similar to the previous third scenario.

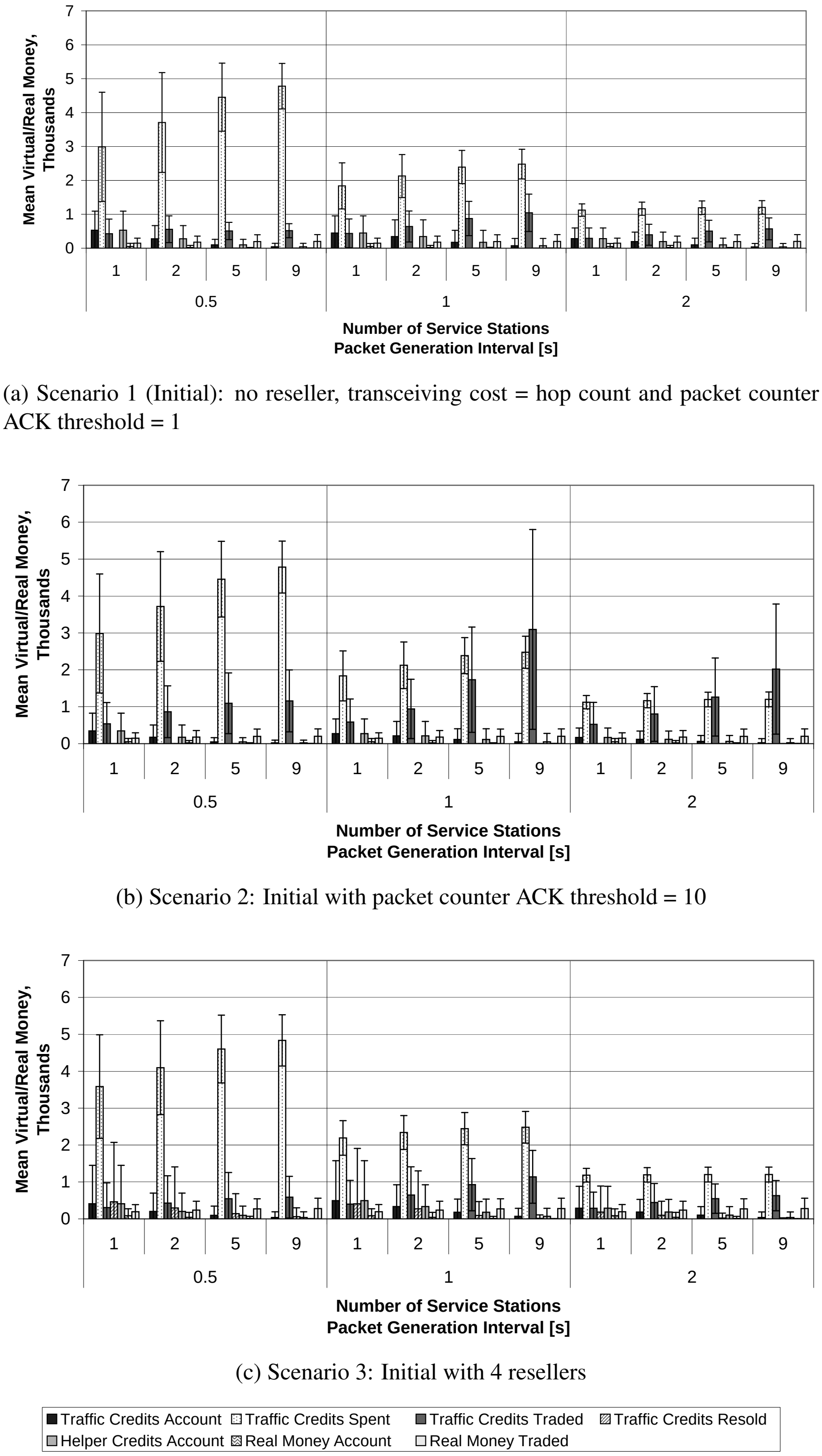
The last scenario which uses the globally fixed transceiving cost of 3 traffic credits gives very similar results to the first scenario. The values for the goodput in Figure 5.19e, the drop reasons in Figure 5.18e and the overhead in Figure 5.17e are slightly lower compared to the first scenario. Again, we attribute this to nodes with an average route length below 3. The results show us however, that it is feasible to use a globally fixed transceiving cost and obtain very similar results compared to dynamic hop count related charges.

Cash Flow

The cash flow provides important information about the distribution of virtual and real money in the network. This allows us to optimize the different CASHnet parameters in order to optimize the operation of our incentive based cooperation framework. The cash flow can be affected by different influences. The number of service stations as well as the number of resellers has a direct impact. Also, the amount of network traffic has a direct impact on the consumption of traffic credits and the distribution of helper credits.

Figure 5.20 on page 119 displays the cash flow in the five scenarios. We analyze the average final account states of the traffic/helper credits and the real money per node. In addition, we show the average amount of traffic credits spent, traded at the service station, resold by a reseller and the amount of real money traded.

The first scenario demonstrate the effect of CASHnet in Figure 5.20a. We observe that the average final account state for traffic credits is well above zero, how-

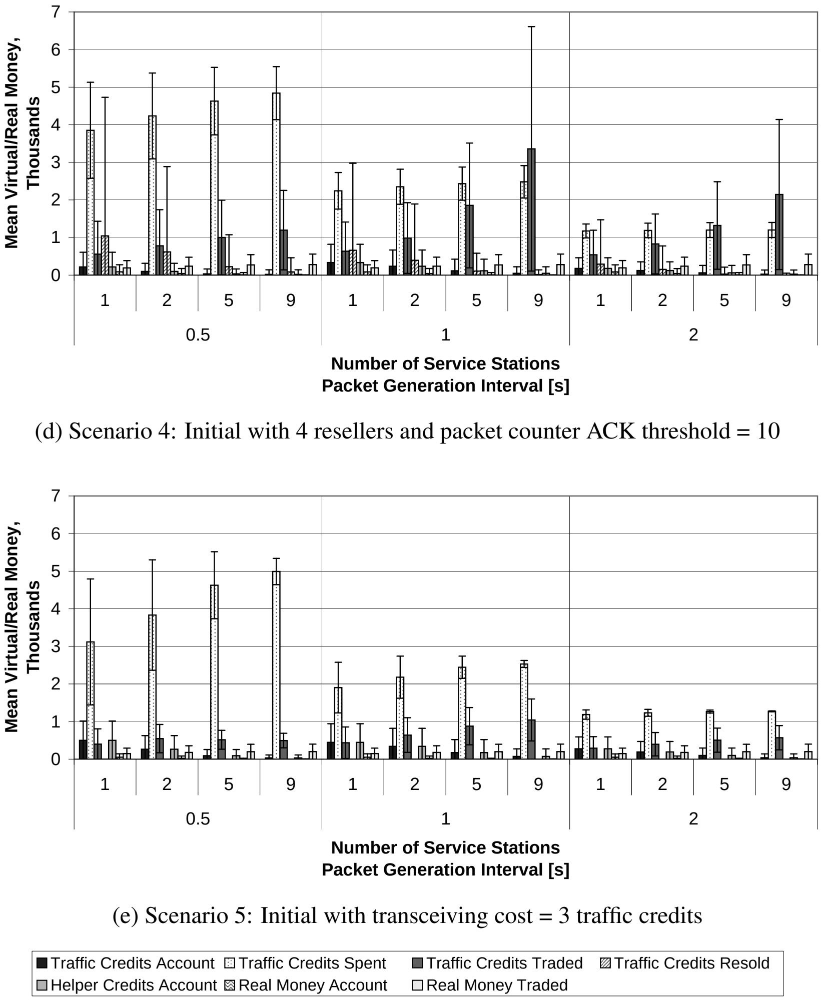


(a) Scenario 1 (Initial): no reseller, transceiving cost = hop count and packet counter ACK threshold = 1

(b) Scenario 2: Initial with packet counter ACK threshold = 10

(c) Scenario 3: Initial with 4 resellers

Figure 5.20: Cash flow in CASHnet for different parameters 1/2



(d) Scenario 4: Initial with 4 resellers and packet counter ACK threshold = 10

(e) Scenario 5: Initial with transceiving cost = 3 traffic credits

Figure 5.20: Cash flow in CASHnet for different parameters 2/2

ever the standard deviation is very high indicating that some nodes have a depleted traffic credits account. As expected, the amount of traffic credits spent increases with the number of service stations, because the nodes have more opportunities to refill their account, which is also indicated by the reduced final state account of helper credits. However, as we previously saw the increased number of traffic credits spent does not automatically result in an improved goodput under high network load. Also, the number of traffic credits traded is very similar under high network load, which is caused by the high amount of acknowledgements lost due to congestion.

Figure 5.20b shows the results of the second scenario with the packet counter ACK threshold of 10. The values are similar to the first scenario with a significant difference in the amount of traffic credits traded. The reduced number of transmitted acknowledgements implies their increase in value to 10 helper credits and at the same time reduce the traffic load in the network. Again, the high standard deviation results from the random node movement path.

As the resellers have a positive effect on the throughput compared to the first scenario, the amount of traffic credits spent increases as depicted in Figure 5.20c. Since the resellers are deployed in addition to the service station, their effect is reduced in all cases with a high number of service stations (5 or 9).

In the fourth scenario, the combination of reseller and packet counter ACK threshold gives the results shown in Figure 5.20d. Both factors lead to an overall increase in traffic credits spent, traffic credits traded and traffic credits resold compared to the first scenario. We also note that the reduced number of acknowledgements leads to a better use of resellers in scenarios with a low number of service stations.

Figure 5.20e presents the results of the fifth scenario. Compared to the first scenario, the globally fixed transceiving cost of 3 traffic credits leads to a higher average amount of traffic credits spent in every single case. So, while the average packet flow performance is slightly inferior, the average revenue is higher compared to a dynamic transceiving cost equal to hop count. Otherwise, the cash flow is similar to the first scenario.

Summary

The extensive analysis of the CASHnet mobile simulation scenarios in terms of starvation, packet flow and cash flow gives us a deep insight in the performance of our cooperation scheme. The obtained results allow us to draw several conclusions regarding the limitations of CASHnet and give us indicators for the minimal requirements of CASHnet to perform almost equal to a plain multi-hop cellular network.

As expected, the traffic load and the number of service stations directly affect CASHnet. However, an increase in the number of service stations does not result in a linear performance increase. This is caused by the immobility of the service station as well as the random node movements. In real life however, we can imagine nodes would start to move directly to a service station before or the moment they run out of traffic credits.

Because CASHnet introduces additional overhead with the acknowledgements, the reduction of the frequency of transmitted acknowledgements using the packet counter ACK threshold is very helpful. However, as mentioned before, increasing the packet counter ACK threshold automatically increases the value of each acknowledgement and thereby gives a possible loss of an acknowledgement a higher impact. Acknowledging every 10th packet provided the best results in our simulation scenarios.

The reseller nodes compensate for the immobile service stations. They have the highest impact in scenarios with few (1 or 2) deployed service stations. However, as we chose the resellers from the available nodes, they also move according to the random waypoint mobility model. Because of that, we suspect their impact to be higher in real life. The combination of resellers and the packet counter ACK threshold set to 10 delivered the best results for most cases of packet generation interval and number of service stations.

The effect of the globally fixed transceiving cost is remarkable in a sense that compared to dynamic hop count related charges, it increases the number of traffic credits spent, when we set the transceiving cost to the average hop count from a node to the gateway in the multi-hop cellular network. The increase is most visible in scenarios with a high network load. Here, the frequent transmission of packets at locations distant from the gateway and thus high transceiving cost quickly depletes the traffic credits account of the node. We find, that with fixed transceiving cost, the nodes are able to transmit more packets. In real life, this would translate into an increased sales volume for the provider and should also positively affect the revenue. However, due to the congestion interpreted as link break in the network simulator, it does not automatically translate into a higher goodput.

5.8.3 Comparison with Nuglet

In order to compare CASHnet with another cooperation scheme, we also implemented Nuglet in the network simulator ns-2. We described Nuglet extensively in Chapter 3, Section 3.5.1 on page 38. Nuglet is an incentive-based cooperation framework, with only one virtual currency called Nuglets and a single source of income for Nuglets, that is the forwarding of other node's packets. Thus, Nuglet relies on a self-perpetuating cycle of spending and earning virtual money. As mentioned before, this limitation occurred to us as a major drawback and motivated our work on the resulting CASHnet scheme.

In Nuglet, only the transmission of a self-generated packet is charged. The cost is equal to the number of intermediate nodes. When a node forwards a packet it receives one nuglet. In Nuglet each node keeps a pending nuglet counter for every neighbor and increases it in case the node receives a forwarded packet from a neighbor. The counters on all nodes are periodically synchronized by sending the pending nuglets to the respective node. We implemented the nuglet synchronization with the help of a timer which is triggered according to the nuglet synchronization interval parameter.

Figure 5.21 shows the schematic of our Nuglet mobile simulation scenario. It is identical to the plain mobile scenario presented in Section 5.8.1 on page 106 except that the nodes have Nuglet functionality. We use the same movement files as in the plain and CASHnet mobile scenarios, where 40 nodes roam around in an area of 900x600 meters and transmit packets at a constant bit rate and with 512 bytes length towards the gateway at the specified packet generation rate.

Table 5.8 lists the Nuglet node parameters. We set the amount of initial virtual money to 1000 nuglets. We note that real money does not exist in the Nuglet scheme and is not equal to virtual money, as it must be exchanged first. Therefore, we believe the comparison to be fair in a sense that both schemes have the same

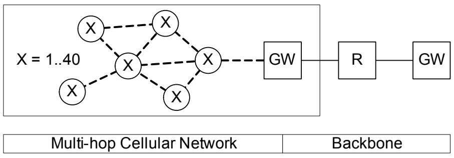


Figure 5.21: Nuglet mobile simulation scenario

Parameter	Value
Initial nuglets account state [N]	1000
Nuglet synchronization interval [s]	5

Table 5.8: Nuglet node parameter settings for the Nuglet mobile scenario

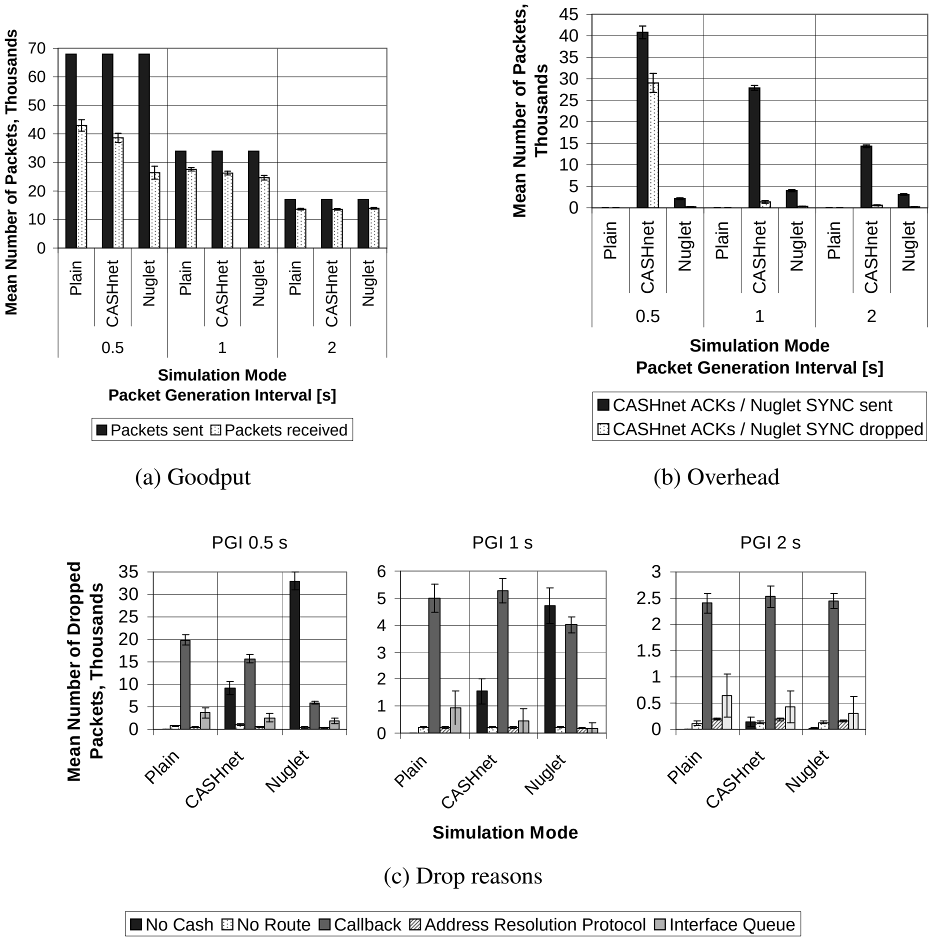
initial situation according to their abilities. In the Nuglet scheme, a node needs to find other nodes and forward their packets to earn nuglets. In the CASHnet scheme, a node needs to find a service station to exchange the helper credits and the real money against traffic credits. The weakness in Nuglet lies not in the initial amount of virtual money, but in the way of earning additional nuglets later on during operation. The nuglet synchronization process is triggered every 5 seconds.

We analyze the starvation, packet flow and cash flow of Nuglet and compare the results to the values obtained in the CASHnet mobile scenario and - where applicable - to the results of the plain mobile scenario, in which no cooperation scheme is active and all nodes are cooperative. As before, we present the averaged results over the 20 movement scenarios with their standard deviation. For comparison, we choose a CASHnet scenario with the following key simulation parameters: 2 service stations are deployed, 4 nodes our of 40 act as resellers, the transceiving cost is equal to the hop count to the gateway and for every tenth forwarded packet an acknowledgement is sent (SS=2, R=4, TC=hop count, PCAT=10).

Packet Flow

Figure 5.22 illustrates the packet flow consisting of goodput, drop reasons and overhead. From Figure 5.22a, we see that the goodput for the plain mobile scenario is the highest with 63%, 81% and 80% for the packet generation interval of 0.5, 1 and 2 seconds. Then follows CASHnet with 57%, 79% and 80% respectively. For Nuglet we obtain the according values of 39%, 73% and 82%. We find that under high traffic load, CASHnet is able to come closest to the plain mobile scenario performance, while Nuglet stays far behind. For lower traffic loads, the results of Nuglet and CASHnet come very close to the plain scenario. We also note the curious result, that for the packet generation interval of 2 seconds, CASHnet and Nuglet have slight increase in goodput, which we explain below.

When we look at the drop reasons in Figure 5.22c, we observe that under high network load, the main drop reason in Nuglet is the lack of virtual money (No



(a) Goodput

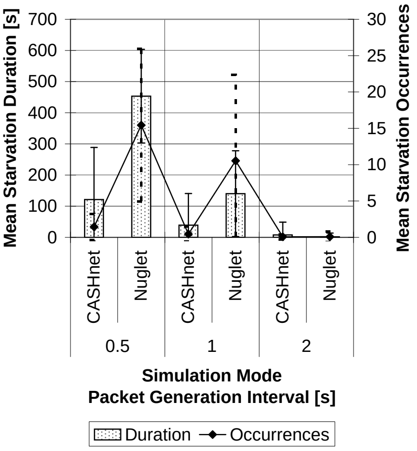
(b) Overhead

(c) Drop reasons

Figure 5.22: Mean packet flow in the Plain, CASHnet and Nuglet scenario

Cash) followed by drops triggered from the link layer (Callback). For CASHnet, we find the inverse situation. For the packet generation interval of 2 seconds, we observe an increased amount of data packet drops in the interface queue for the plain mobile scenario. This leads to a slightly inferior goodput compared to CASHnet and Nuglet. We explain this as follows. In the plain mobile scenario, no signaling overhead is present and thus, more data packets can be transmitted at the same time and therefore also lost. Whereas in both cooperation schemes, the data packets compete with the signaling traffic, i.e. the acknowledgements CASHnet ACK and synchronization messages Nuglet SYNC, which is not accounted for in the drop reason figures.

Figure 5.22b presents the overhead of all three mobile simulation scenarios. We see, that the synchronization overhead for Nuglet is much lower compared to the acknowledgement overhead in CASHnet. The plain mobile scenario has of



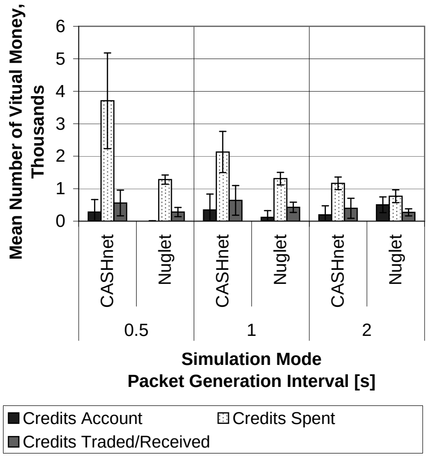


Figure 5.23: Mean starvation duration and occurrences in the CASHnet and Nuglet scenario

Figure 5.24: Mean cash flow in the CASHnet and Nuglet scenario

course no signaling traffic as no cooperation scheme is deployed.

Starvation and Cash Flow

Both CASHnet and Nuglet are incentive-based cooperation schemes and therefore we can compare their starvation and cash flow performance. Figure 5.23 depicts the average starvation duration and occurrences for both cooperation schemes. We find, that in Nuglet the starvation duration is considerably higher and as are the occurrences. This shows, that in Nuglet on average nodes starve more often, which is caused by the periodic synchronization message, which distributes nuglets throughout the network. In CASHnet, the distributed rewards can not be used immediately and must be exchanged at a service station or with a reseller.

Figure 5.24 compares the cash flow for both schemes. We show only the comparable values, such as the final credits account state, the credits spent as well as the credits traded/nuglets received. We note that, for packet generation interval of 0.5 and 1 seconds, CASHnet has more traffic credits left. We attribute this to the fact, that CASHnet has two sources of income, the helper credits and the real money as well as two refill opportunities, the service stations and the resellers. This can be seen in the amount of credits traded/nuglets received which is considerably higher for CASHnet. In addition, the amount of credits spent is also higher in CASHnet compared to Nuglet. Considering the initial amount of 1000 virtual currency units per node in both schemes, we find that in Nuglet, the money in the network diminishes over time. This effect has also been noticed by the authors of Nuglet in [BH03b]. The effect is further increased by the lack of an additional source of income for Nuglets. This makes the long-term operation of Nuglet difficult, as virtual money would have to be distributed regularly among the nodes.

Summary

In the comparison of all three mobile simulation scenarios, we identified the benefits and limitations of each cooperation scheme. We showed, that CASHnet outperforms Nuglet in all analyzed categories and for almost every simulation scenario. Only for low traffic load, Nuglet performs equal to CASHnet. The better performance of CASHnet, is due to the availability of additional sources of income in the network as well as the distribution of refill opportunities in the network. The resellers are moving randomly throughout the network, while the service stations are immobile.

5.9 Conclusion

We implemented CASHnet and Nuglet in the network simulator ns-2 and performed extensive analysis of our cooperation scheme. We explained the different key parameters and motivated our evaluation criteria. In order to find the upper boundaries for the performance characteristics of multi-hop cellular networks, we conduct simulation runs with a plain multi-hop cellular network, where no cooperation scheme is deployed and the node forward packets out of good will. From the various results, we identified optimal CASHnet parameter settings.

In our evaluation we showed, that CASHnet comes close to the performance of a plain multi-hop cellular network, when we use a low or moderate number of service stations and resellers as well as a packet counter ACK threshold of 10. A globally fixed transceiving cost equal to the average route length gives similar results compared to the dynamic cost related to the current hop count and leads to an increase in sales volume, which should translate into a higher revenue for the provider.

From the plain mobile scenario, we learned that the upper boundaries set by the characteristics of the implemented multi-hop cellular network protocols (AODV) and technologies (IEEE 802.11) as well as the movement scenarios, are quite low. We expect, that new wireless technology as presented in Chapter 2, Section 2.4 on page 11 will provide additional increase in performance. Also, the ongoing research in the correct detection of the network link state will considerably increase the performance of the routing protocols.

While it is meaningful to obtain an impression of the CASHnet performance in a simulator, we consider the evaluation of an implementation in a real world environment as much as insightful and even more appropriate to draw more realistic conclusions. In the next chapter, we present the Linux implementation of CASHnet as well as its performance analysis.

Chapter 6

Implementation of CASHnet in a Real Environment

6.1 Introduction

The simulations in the network simulator allowed us to analyze and optimize our CASHnet scheme. While we obtained indicators for the general performance and the influence of different parameters, we have no information about the processing delay caused by the security operations of CASHnet. We left out the security functionality in our network simulator implementation, because we think that the impact of the security functionality is best evaluated in a testbed consisting of real computers. Therefore, we implemented a prototype of the CASHnet framework under Linux and evaluated it using small test scenarios.

In the remainder of this chapter we first motivate our approach. We continue with a description of netfilter/iptables, which provides the basis functionality for our work. Next, we explain the design of our CASHnet implementation. Then, we describe the testbed, the test scenarios and finally our measurement results.

6.2 Motivation

Processing overhead is best analyzed under real conditions with real computers and networks. Therefore, we decided to implement a prototype of our cooperation scheme. CASHnet affects the handling of every packet, be it received, forwarded or generated. Also, CASHnet provides a network layer service, i.e. it encourages the forwarding of packets, by charging traffic generators and rewarding the forwarders. Thus, we require access to each packet after it enters and before it leaves a node. In addition, CASHnet requires cross-layer knowledge (i.e. security and accounting information). We decided to use Linux as development environment, because it provides us with the greatest flexibility in accessing the packets on the network stack.

We had to make several design decisions regarding performance and transparency. We decided to implement CASHnet at the network layer instead of creating an overlay network at the application layer. While the latter would give us more independence from the underlying network and thereby increase the interoperability, the existing mobile ad hoc network technologies and protocols have very limited resources and render the introduction of additional cross-layer communication unfeasible for now.

We decided to implement CASHnet as a user space daemon instead of a kernel module. Thereby, we accept a speed penalty caused by the additional communication between kernel and user space, but we avoid the complexity and the rigidity of the monolithic Linux kernel. As a user space daemon, we have no permission to directly access the packets on the network stack. Therefore, we require the help of a program which establishes a bridge between kernel and user space. We use the netfilter/iptables [RW$^+$05] package to perform this task.

For the test environment we require a mobile ad hoc routing protocol with gateway functionality as described in Chapter 2, Section 2.5.2 on page 21. Several implementations exist, but only two are recommended by the authors of AODV on their website [AOD05]. The first implementation is called Kernel AODV [KB04]. As the name suggests, the complete routing logic is implemented as a kernel module and runs in kernel space. The second called AODV-UU [Nor04] implements the routing logic in a user space daemon and uses netfilter/iptables to access the packets. We chose AODV-UU, because we found more documentation about its architecture and internal operation, such as in [Wib02]. [CBR04] describes the different approaches to implement the AODV routing protocol and presents some implementations.

6.3 Netfilter/iptables

Netfilter and iptables [RW$^+$05] are building blocks of a packet processing framework for the Linux kernel versions 2.4 and 2.6. Netfilter/iptables is widely used as firewall and for network address translation (NAT).

Netfilter provides a set of hooks inside the network stack of the Linux kernel, that allow a kernel module to register callback functions. A registered function is called every time a packet traverses the respective hook in the network stack. Also, each hook allows to alter packets. Iptables provides a generic table structure for the definition of rule sets. A rule within iptables consists of a number of classifiers and an associated action, which is executed when a packet matches the classifiers. Thus, iptables allows to control the packet flow and netfilter provides the required access to the network stack of the Linux kernel.

In Figure 6.1 we show the different netfilter hooks and their location in the Linux kernel network stack as well as three example packet flows. We distinguish between packets destined for the current node (local delivery), packets to be forwarded (forwarding) and packets generated by the current node (local genera-

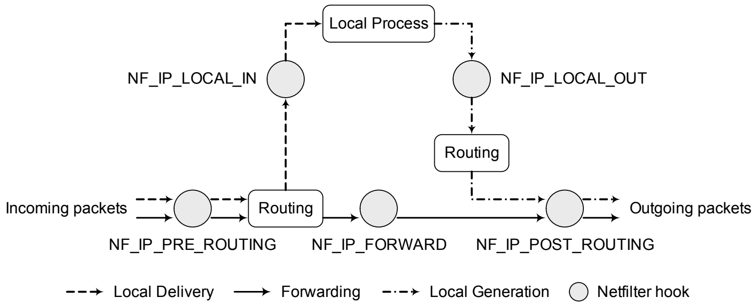


Figure 6.1: Netfilter hooks

Table	Chain	Netfilter Hook
filter	INPUT FORWARD OUTPUT	NF_IP_LOCAL_IN NF_IP_FORWARD NF_IP_LOCAL_OUT
nat	PREROUTING OUTPUT POSTROUTING	NF_IP_PRE_ROUTING NF_IP_LOCAL_OUT NF_IP_POST_ROUTING
mangle	PREROUTING OUTPUT	NF_IP_PRE_ROUTING NF_IP_LOCAL_OUT

Table 6.1: Tables, chains and hooks in iptables/netfilter

tion). An incoming packet first passes the NF_IP_PRE_ROUTING hook. Then, the kernel routing process decides whether this packet is to be forwarded to another node or delivered to a local process. In case the packet is destined for a local process, it traverses the NF_IP_LOCAL_IN hook before it is delivered to the local process. If the packet is to be forwarded, it traverses the NF_IP_FORWARD and the NF_IP_POST_ROUTING hook before it leaves the network stack. A locally generated packet first traverses the NF_IP_LOCAL_IN hook. Then, the kernel routing process decides about how to route the packet (e.g. next hop, network interface). Before the packet leaves the network stack it passes the NF_IP_POST_ROUTING hook.

Iptables consists of tables, chains and rules. A table is a combination of chains for packet processing. A chain is a combination of rules applied to each packet traversing a specific netfilter hook. A rule describes the matching criteria (e.g. packet type) and the resulting action for a matched packet also called target (e.g. accept or drop). Iptables contains three pre-configured tables: filter, nat and mangle. Each of them provides access to packets at specific places in the network stack. Table 6.1 lists the tables, their chains and the hooks respectively. AODV-UU and CASHnet use different tables, which we explain in the next section.

Because packet processing is done in the network stack of the kernel space, a user process can not access the packets directly. However, netfilter/iptables provides a mechanism to pass a packet out of the stack and queue it in user space. There, the user process can modify the packet and define an action before it is inserted back into the kernel. A queue handler is responsible for passing the packets to and from user space. The standard queue handler in Linux is the kernel module ip_queue. Once this module is loaded, a new target named QUEUE becomes available in iptables. The library libipq provides the interface to this queue for user processes.

6.4 Implementation of CASHnet under Linux

The CASHnet framework consists of different components and functionalities. A CASHnet node runs a daemon, which is responsible for charging the generated traffic and rewarding the forwarded traffic. The daemon also exchanges certificates for the authentication and is responsible for the creation and verification of digital signatures. In order to reduce the complexity, our implementation does not include the secure storage of cryptographic keys, i.e. we omitted the interaction with the smart card. Our main focus in this implementation lies on measuring the introduced computational overhead and resulting delay in a network.

We implemented CASHnet in C++ under Linux using the GNU Compiler Collection, GCC as well as the libipq library from netfilter/iptables. For the cryptographic functionality such as the creation and verification of digital signatures, we use the RSA reference implementation called RSAREF [RSA93]. As netfilter operates on the network layer, we deal with IP packets. In particular, we add, remove and verify digital signatures in the IP packet payload. Figure 6.2 shows the interaction of CASHnet and netfilter/iptables in the context of the Linux system. CASHnet runs in user space and has no direct access to the network stack of the Linux kernel. In order to access all locally received, forwarded and generated packets, we use the respective chains of the filter table from iptables (INPUT, FORWARD and OUTPUT), which in turn use the respective netfilter hooks. All packets traversing these hooks are sent to the QUEUE target, a buffer in the user space. The libipq library allows us to initialize this queue, manipulate each packet as well as decide about its verdict, i.e. accepting or dropping. In addition, we use an UDP socket for the generation of the CASHnet control messages, i.e. the certificate advertisement $CADV$ and reply $CREP$ as well as the acknowledgement ACK. We receive these control messages by intercepting them via the INPUT chain, because netfilter had difficulties in delivering them to the UDP socket.

Our CASHnet implementation consists of several classes for the different tasks. Figure 6.3 illustrates the classes of our CASHnet implementation. The main program *acfi* creates the daemon process and initializes the other classes. The *Configuration* class reads in a configuration file, which specifies the CASHnet parameters on the current node. The *Filtering* class handles all packets, which traverse the designated netfilter hooks. Depending on the classification of a packet, a specified

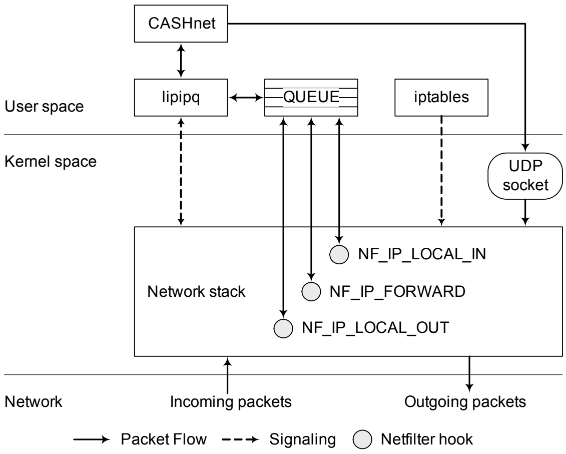


Figure 6.2: CASHnet and netfilter/iptables interaction in the Linux system

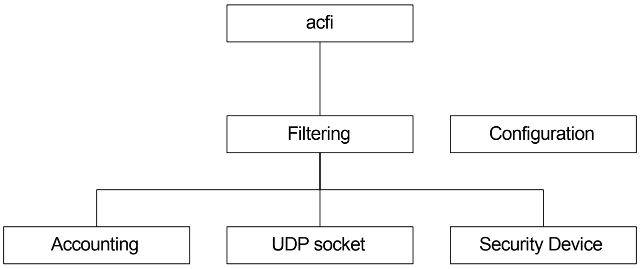


Figure 6.3: CASHnet implementation classes

action is taken. The *Security Device* class is responsible for managing certificates as well as creating and verifying signatures. The *UDP socket* class is used to generate all the CASHnet control messages. The *Accounting* class contains all charging and rewarding functionality and is used to manage the credits accounts.

Figure 6.4 presents the detailed operation of the CASHnet implementation. In the flow chart we distinguish between four end states, which are gray-colored. We can either tell netfilter to accept or to drop a packet. When a signature for the verification of a packet is missing, we queue the packet and decide later about its verdict. In case we receive unknown packets, we enter an error state.

In the following, we describe the main procedures from the flow chart according to the different packet types handled by CASHnet. We distinguish between AODV, CASHnet control ($CADV$, $CREP$ and ACK) as well as data packets. Although, the packets are passed to the same QUEUE, libipq allows us to find out via which iptables chain they entered. AODV packets are immediately accepted and not further processed.

When we receive CASHnet control messages three possibilities exist. For an acknowledgement ACK destined for the current node we reward the node, otherwise the message is forwarded. When a certificate advertisement $CADV$ is intercepted, we add the certificate to the authenticated nodes list. If the current node is the destination, we generate a certificate reply $CREP$. In case a $CREP$ is filtered, we add the certificate to the authenticated nodes list. If the current node is not the destination, we forward the certificate reply.

In case we intercept a data packet, also three possibilities exist. When the current node generated this packet, we charge the node's account as well as sign and transmit the packet. If the current node is the destination, we also charge the node as well as remove all additional data and deliver the payload to the local process. In case neither is true, the current node is forwarding the packet. Thus, we remove the signature from the previous hop and add one from the current node.

For further details on the CASHnet implementation process under Linux, we refer to [Lat05].

6.5 Testbed Setup

In order to evaluate our CASHnet implementation we set up a small testbed shown in Figure 6.5. In this testbed four laptops (A, B, C and GW) are interconnected in a chain topology. One laptop (GW) acts as a gateway between the wireless and the wired network. Each laptop has a recent processor: Node A and C have an Intel Celeron 2.40 GHz with 128 KB cache, node B has an Intel Pentium M 1.86 GHz with 2 MB cache and the gateway has an Intel Pentium M 1.4 GHz with 1 MB cache. The laptops all have an internal wireless network interface card compliant to IEEE 802.11g, which allows a maximum gross data rate of 54 Mbit/s. The gateway has an Ethernet link with 100 Mbit/s (IEEE 802.3u) to the backbone. All laptops run Slackware Linux 10.1 [The05] with the Linux Kernel 2.6.8, netfilter/iptables 1.2.11, NdisWrapper 1.2 [FP05] for the driver of the wireless cards and AODV-UU 0.9 [Nor04]. AODV-UU uses the nat table from iptables and therefore receives packets from the POSTROUTING, OUTPUT and PREROUTING chains. In addition, AODV-UU tunnels all packets leaving the multi-hop cellular network via a gateway. To do so, it intercepts the packets via the POSTROUTING chain and adds the tunnel information, i.e. an own protocol number and recomputes the CRC. However, CASHnet intercepts the packets via the OUTPUT chain to digitally sign them. Because AODV-UU modifies the packet for tunneling after CASHnet digitally signs it, we can not verify the signature correctly at intermediate nodes. In order to avoid the tunneling of AODV-UU, we had to restrict the test range from laptop A to the laptop operating as gateway. Fortunately, this does not affect our evaluations, as the gateway performs the verification and removal of the signature as if a backbone host would be the recipient. The only difference is, that instead of forwarding the packet, it is delivered to the local process.

Both, AODV-UU and our CASHnet implementation require netfilter/iptables to filter packets and queue them to user space for further processing. So we enable the respective options in the Linux kernel (CONFIG_NETFILTER, CONFIG_IP_NF_QUEUE and CONFIG_IP_NF_IPTABLES). To simulate a chain topol-

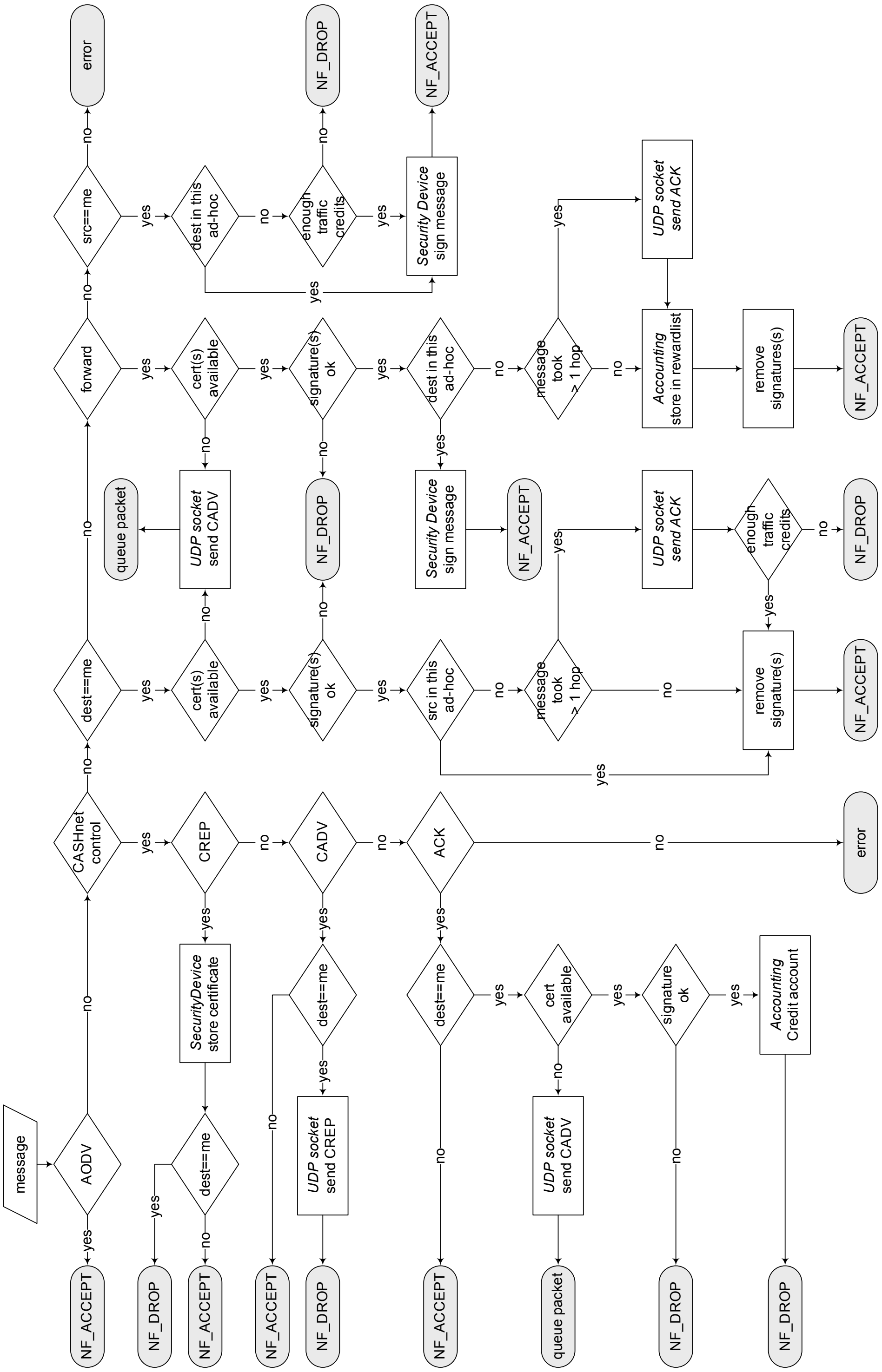


Figure 6.4: CASHnet implementation operation

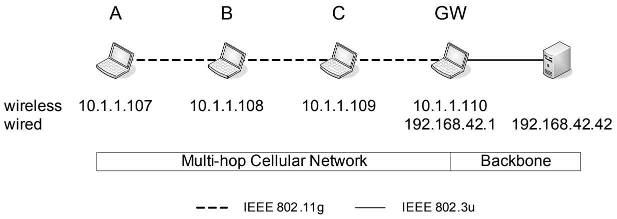


Figure 6.5: Testbed

ogy in our laboratory room, we set up a MAC address filter such that each node only receives packets from its direct one-hop neighbor(s). On the gateway laptop we enable the gateway mode of AODV-UU and specify a locality netmask prefix to let AODV-UU distinguish the mobile ad hoc network from the normal network. For simplicity we use private addresses. On each laptop, we generate a public-/private-key pair as well as the corresponding digital certificate using RSA and MD5 for the hash generation. We use a key length of 1024 bit, which we regard as a reasonable trade-off between security and performance considering the short certificate lifetime on a node, which we set to 5 minutes in the test scenario.

We identified five variable key parameters for the evaluation of our CASHnet Linux implementation. These parameters have a strong influence on the CASHnet implementation and provide us with high flexibility when conducting performance measurements. The following list explains the parameters in detail.

- *Source & destination:* We vary the source as well as the destination of our delay measurements between the four Laptops. The distance between source and destination expressed in the number of intermediate hops strongly influences the end-to-end delay measurements, because each intermediate node processes the packet, i.e. it verifies and creates digital signatures.

- *Data/acknowledgements signed:* As mentioned before, we distinguish between different packet types. In order to measure the impact of the cryptographic functions, we allow to specify whether data packets and/or acknowledgements are digitally signed in the respective test scenario. This directly affects the processing time on the node, and thus the end-to-end delay between source and destination.

- *Packet counter ACK threshold:* The packet counter ACK threshold has the same effect as described in the previous Chapter 5, Section 5.5.2 on page 92. It defines how many forwarding packets a node has to receive from a single forwarder before it rewards this forwarding node, i.e. sends an acknowledgement message.

6.6 Evaluation Criteria

With the Linux implementation of CASHnet, we want to measure the performance of our scheme under real-life conditions. We are interested in the end-to-end delay and jitter of a connection as well as the packet processing time on a node.

- The **delay** specifies the time a packet takes to travel from one point to another point in a network. While the delay does not affect the quality of the received data, it degrades the perceived quality in real-time applications. For example, in IP telephony or video conferencing, delay introduces a disturbing walky talky effect. The ITU Telecommunication Standardization Sector (ITU-T) published the recommendation G.114 [ITU03] for one-way transmission time of voice. It recommends a maximum delay of 150 ms (i.e. a round-trip time of 300 ms) for good quality and if echo cancellers are used to remove the own voice from the returning audio stream. With a delay between 150 and 400 ms acceptable quality can be achieved, if the administrators are aware of the impact on quality. Delays above 400 ms are generally unacceptable for most applications.

  The delay is typically introduced by the packet processing on the nodes in the network (e.g. routers or in our case CASHnet nodes). The **jitter** describes the variance of the delay measured over time from one point to another. This fluctuations in the delay affect streaming applications, which rely on a rather constant arrival time of packets. For example, when streaming audio or video data, a high jitter results in the client application not being able to restore the respective signal in time, which in turn leads to unintelligible audio or video. With buffers in the network equipment and the client application, the effects of jitter can be reduced.

  The delay can be measured in either one-way or round-trip delay. One-way delay measurements are difficult to obtain as they require expensive equipment, round-trip delay measurements are fairly simple to conduct. The round trip delay consists of the time, the packet requires to travel to the destination, the time of generating a reply at the destination, and the time for the reply to travel back to the requestor. A common approach is to divide the measured round-trip time by 2 in order to obtain the one-way delay.

  The Internet Control Message Protocol, ICMP is used for diagnostic information exchange among network layer devices and is an integral part of every Internet Protocol, IP implementation. ICMP also implements the echo request and echo response commands. A network layer node, which receives an echo request packet must answer with an echo response packet. The program *ping* generates echo requests, transmits them to a given destination and waits for the echo replies. It measures the round-trip time between the calling node and the destination.

- The **packet processing time** describes the amount of time, a packet spends inside the CASHnet implementation. Because CASHnet uses public key cryptography the processing can take quite a long time. The packet processing time of each node adds to the end-to-end delay between two communication partners. We do not measure the additional overhead resulting from the communication between user and kernel space.

We measure the processing time for each of the five packet types distinguished by our CASHnet implementation, i.e. AODV, CASHnet control (certificate advertisement $CADV$, certificate reply $CREP$ and acknowledgement ACK) and data packets. This allows us to draw conclusions regarding the impact on processing delay of each packet type and their possible interactions, i.e. data signed by an unknown originator triggering the generation of certificate advertisements. We measure the packet processing time by logging each packet with a timestamp, when it enters and exits the CASHnet implementation.

6.7 Evaluation of the CASHnet Implementation

The security functionality contained in the Linux prototype implementation enables us to test the real-life performance of our cooperation scheme. In the previous Chapter 5 we focussed our simulation analysis on the starvation, packet flow and cash flow in static and mobile scenarios. With the Linux implementation of CASHnet, we are interested in the end-to-end delay and jitter as well as the packet processing time on each node. For simplicity, we deactivate the accounting functionality of CASHnet. In our tests, we use ping to transmit 2000 packets at a rate of 1 packet per second.

Table 6.2 lists the various key parameters for the testbed scenarios. We vary the source and the destination of our measurements between all the four laptops and illustrate the effect of the number of intermediate hops. In order to show the impact of the different security mechanisms in CASHnet, we distinguish between securing the data packets and/or the acknowledgement packets with digital signatures, i.e. data (not) signed and acknowledgement (not) singed. As in the previous chapter, we investigate the effect of different packet counter ACK thresholds, i.e. the number of forwarded packets a node receives, before it sends a reward. A node rewards a forwarding node after receiving 1, 5, 10, 15 or 20 packets.

As in the previous evaluations, we restrict the presentation of results to five main test scenarios, which we list in Table 6.3. In the first scenario (plain) no CASHnet functionality is activated. Here, we obtain an indicator for the optimal performance of our testbed. In the second scenario (pass), we activate CASHnet, but do not use the security functionality. This scenario gives us information about the delay caused by processing the packet in user space. The third scenario (partial) allows us to measure the delay caused by digitally signing data packets. With the fourth scenario (full) we study the delay caused by the complete security operations of CASHnet, i.e. data and acknowledgements are digitally signed. In these four

Parameter	Value
Source	A, B, C, GW
Destination	A, B, C, GW
Data signed, DS	yes, no
ACK signed, AS	yes, no
Packet counter ACK threshold, PCAT	1, 5, 10, 15, 20

Table 6.2: Key parameters for the testbed scenarios

	Scenario				
	1	2	3	4	5
Parameter	Plain	Pass	Partial	Full	PCAT
Source	A, B, C	A, B, C	A, B, C	A, B, C	A
Destination	GW	GW	GW	GW	GW
Data signed, DS	-	no	yes	yes	yes
ACK signed, AS	-	no	no	yes	yes
PCAT	-	1	1	1	1, 5, 10, 15, 20

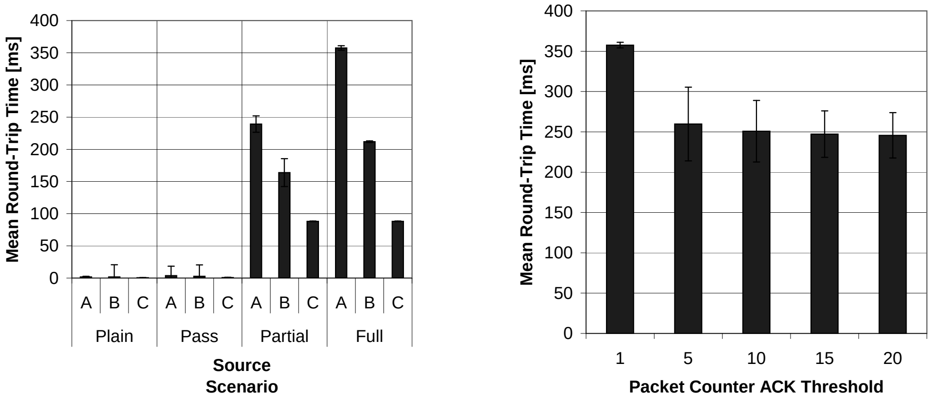
Table 6.3: Parameter settings for the testbed scenarios

scenarios, we let each node ping the gateway consecutively, resulting in three sub scenarios for each main scenario. In the fifth scenario, we analyze the delay caused by different packet counter ACK thresholds.

In the following, we compare different scenarios for the round-trip time measurements and packet processing times. We note that in a real test environment the number of possible influences is huge and that phenomena are difficult to isolate. During our measurements, we found that the simple channel scanning on the wireless medium by other computers greatly affected the performance. Also, we measure sporadic outliers independent of the scenario, which we attribute to retransmission on the medium access control, MAC layer or possible variations in the processing time of intermediate nodes. The first is typically caused by transmission errors on the wireless medium, the latter by the packet processing applications, e.g. CASHnet and AODV-UU. Nevertheless, we think that the results give some indication as to how CASHnet performs in a real environment and show possible improvements.

6.7.1 Round-Trip Time

The round-trip time is an indicator for the end-to-end delay and jitter between two communication partners. We measure it by analyzing the output from the ping program under each testbed scenario. Figure 6.6 presents the mean round-trip time as well as the standard deviation. The expected behavior would be a round-trip time decrease when approaching the gateway in the sub scenarios and an increase



(a) Scenario 1 - 4: Variation of enabled CASHnet functionality between none (plain), pass, partial and full when node A, B and C ping the gateway

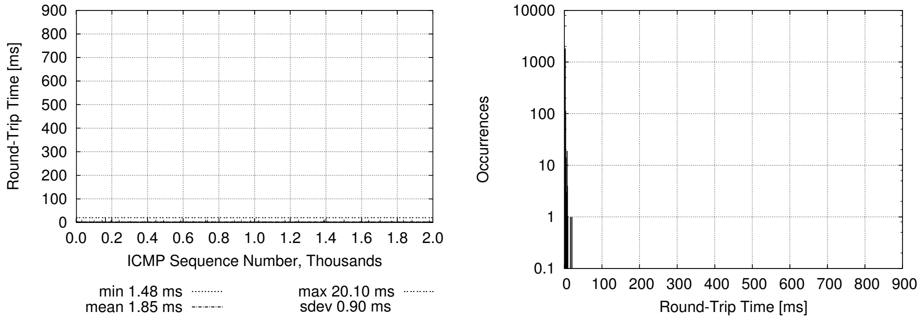
(b) Scenario 5: Variation of different packet counter ACK thresholds when node A pings the gateway

Figure 6.6: Mean round-trip times and standard deviation for all testbed scenarios

between main scenarios, when including more and more security functionality.

As expected the first plain scenario shows the best performance with average round-trip times below 2, for both node A and B, and 1 ms for node C pinging the gateway. In the second scenario, all packets from the respective chains are processed by CASHnet in user space, however the security functionality is disabled. Therefore, the average round-trip times slightly increase compared to the first scenario. When we start to apply a part of the security operations in scenario 3, i.e. digitally sign data packets, the round-trip times increase considerably. We obtain mean round-trip times of 239, 164 and 88 ms for node A, B, and C pinging the gateway. In the fourth scenario, we use the normal CASHnet operation mode, where data and acknowledgements are digitally signed. Here, the results are 358, 212 and 88 ms. The results from our fifth scenario, where we test different packet counter ACK thresholds, when node A pings the gateway are as follows: We observe round-trip times of 358, 260, 251, 247 and 246 ms for packet counter ACK thresholds of 1, 5, 10, 15, and 20 respectively.

The delay on node C does not increase between scenario 3 and 4, because in CASHnet the packet originator does not receive a reward. Since there is no intermediate forwarding hop between node C and the gateway, i.e. C is the originator of the echo requests and the gateway is the originator of the echo responses, no packets get acknowledged. When node B pings the gateway in scenario 4, the gateway and node B send an acknowledgement to node C for every echo request and echo response respectively. This increases the delay by 41 ms compared to scenario 3. In case node A pings the gateway in scenario 4, node C and the gateway acknowledge every echo request to node B and C respectively. In addition, node A and B



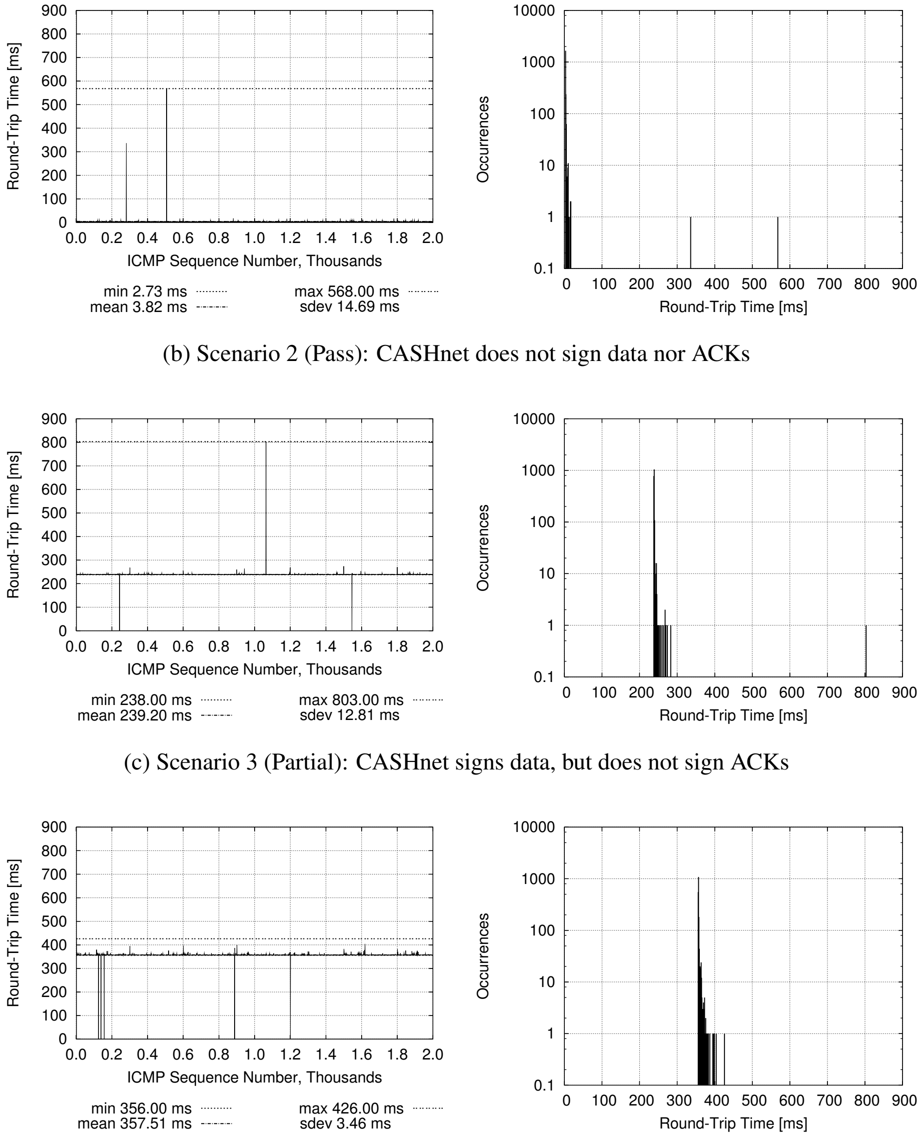
(a) Scenario 1 (Plain): CASHnet is not running

Figure 6.7: Round-trip times for scenario 1 - 4 when node A pings the gateway 1/2

acknowledge every echo response to node B and C respectively. These operations increase the delay by 119 ms compared to scenario 3. The results of scenario 5 show a significant decrease in delay by 98 ms when we acknowledge every fifth instead of every single forwarded packet. The continuous increase of the packet counter ACK threshold decreases the delay in small steps. In fact, the delay approaches the 239 ms, which we measured in scenario 3, where we did not digitally sign any acknowledgement.

In the following figures, we analyze selected round-trip time measurements on different nodes in detail. Figure 6.7 compares the performance of node A in the scenarios 1, 2, 3 and 4. In Figure 6.8 we show the impact of the full CASHnet functionality (scenario 4) in relation to no CASHnet functionality at all (scenario 1) when node B and C ping the gateway. Figure 6.9 shows the results of scenario 5 on node A. For each node we use two figures. The first figure shows the measured round-trip time for each ping identified by its ICMP sequence number. It also presents the mean, maximum and minimum as horizontal lines. Further, the standard deviation is included in the legend. Round-trip time values, which fall below the minimum line represent dropped packets. The second figure shows the frequency of measured round-trip times using a logarithmic scale on the y-axis.

Figure 6.7a shows the round-trip time in the first scenario, when node A pings the gateway without CASHnet running. We notice a very low end-to-end delay as well as a very small jitter, where 1834 measurements have a round-trip time of 2 ms. When we let CASHnet process the packets without performing the security operations, the round-trip time slightly increases to 4 ms as can be seen in Figure 6.7b for scenario 2. The jitter also slightly increases, where 1657 measurements have a delay of 3 ms. We observe two outliers, which we attribute to retransmissions on the MAC layer. Besides the two outliers, no significant differences to the first scenario appear. This changes in scenario 3 presented in Figure 6.7c, where the data is digital signed and verified by the nodes. We notice an increase in the average round-trip time by 234 ms as well as in the jitter. The first can be explained by the digital signature calculated for every outgoing packet. The latter, by the signature



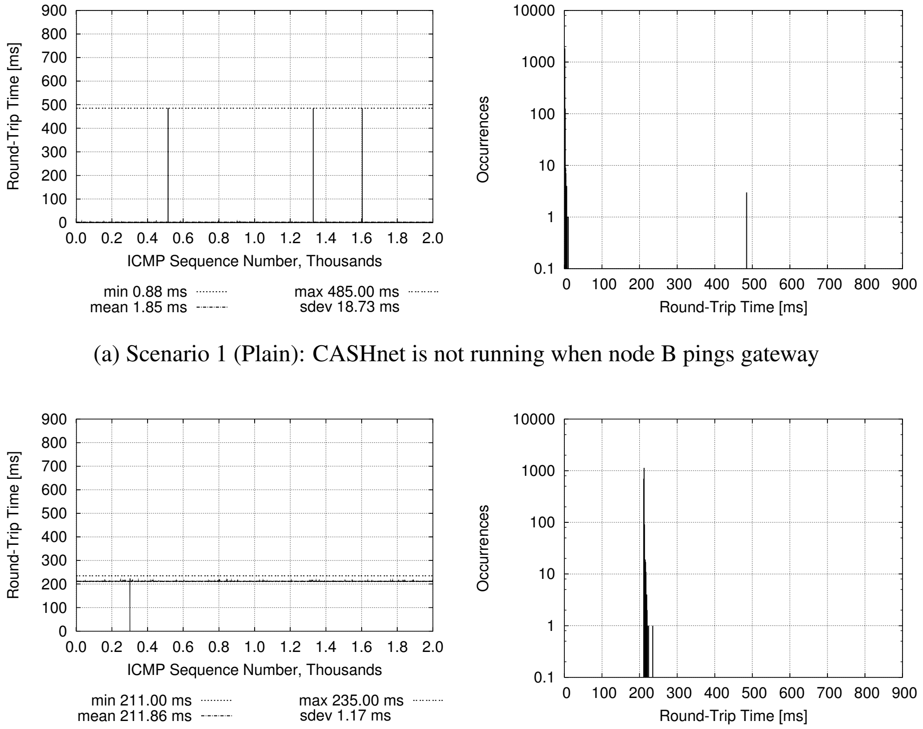
(b) Scenario 2 (Pass): CASHnet does not sign data nor ACKs

(c) Scenario 3 (Partial): CASHnet signs data, but does not sign ACKs

(d) Scenario 4 (Full): CASHnet signs data and ACKs

Figure 6.7: Round-trip times for scenario 1 - 4 when node A pings the gateway 2/2

verification process, where the required certificate might not be available immediately and the packet must be queued until a certificate reply has arrived. Also, 2 packets are lost, which we attribute to the exceeded number of allowed retransmissions on the MAC layer (7 by default in IEEE 802.11). In Figure 6.7d, we see the effect of digitally signing both data and acknowledgement packets. Compared to the previous figure, the average round-trip time increases by 119 to 358 ms due to the additional signature generation and verification for the acknowledgements on



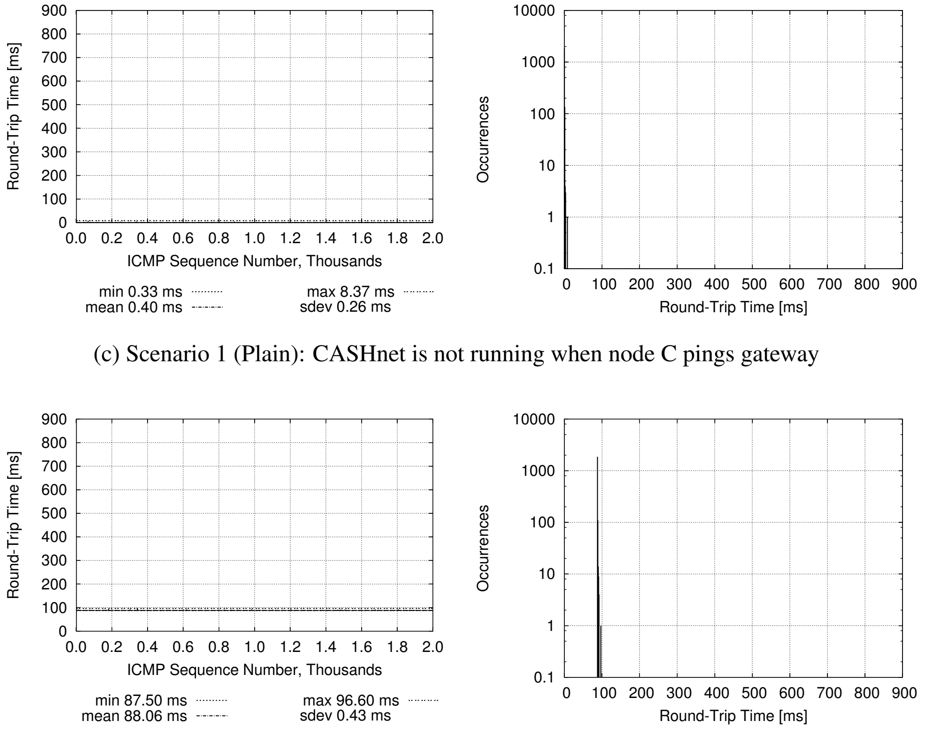
(a) Scenario 1 (Plain): CASHnet is not running when node B pings gateway

(b) Scenario 4 (Full): CASHnet signs data and ACKs when node B pings gateway

Figure 6.8: Round-trip times for scenario 1 and 4 when node B and C ping the gateway 1/2

the respective nodes. Also, the jitter increases, with a peak of 1082 occurrences for 357 ms delay.

Next, we compare the first (plain) and the fourth (full functional CASHnet) scenario on node B and C to see the impact of our cooperation scheme in dependence of the number of intermediate nodes towards the gateway. We like to recall Figure 6.7a and Figure 6.7d from the previous discussion, which illustrate the results when node A pings the gateway. In these figures we observe a considerable increase in the delay to 359 ms and a slight increase in the jitter introduced by CASHnet. Figure 6.8a and Figure 6.8b show the round-trip time, when node B pings the gateway without any and with full CASHnet functionality. As expected, the average round-trip time decreases for both scenario 1 and 4. We observe three equal outliers of 485 ms in the plain scenario, which indicate an equal source for this delay. We suspect an internal packet processing delay in either the Linux network stack or AODV-UU as a disturbance on the wireless channel is not likely to produce three independent outliers with equal delays. We measure 1846 occurrences for a delay of 1 ms in scenario 1 and 1135 occurrences for a delay of 212 ms in scenario 4. This is a slight decrease in the jitter compared to the results of node



(c) Scenario 1 (Plain): CASHnet is not running when node C pings gateway

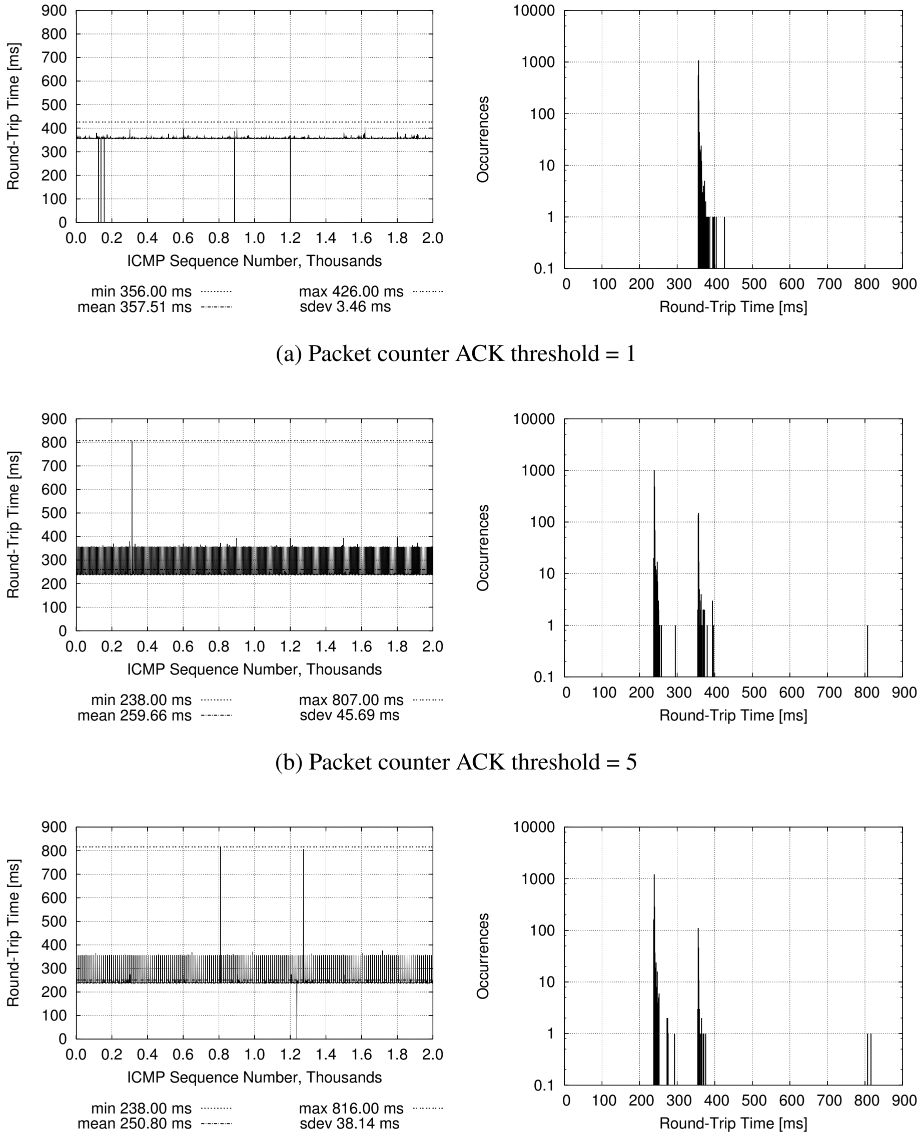
(d) Scenario 4 (Full): CASHnet signs data and ACKs when node C pings gateway

Figure 6.8: Round-trip times for scenario 1 and 4 when node B and C ping the gateway 2/2

A. The results for node C are presented in Figure 6.8c and Figure 6.8b. Here, we observe a very similar jitter for the plain and the full scenario. The delay in the full scenario increases to 88 ms. We find that the increase in delay from 0 to 1 intermediate hop is 124 ms and from 1 to 2 intermediate hops is 147 ms. The difference of 23 ms is caused by the different number of acknowledgements, which have to be created and verified by the corresponding nodes. Thus, when we increase the number of intermediate hops in the full scenario, we observe an increase in the delay and in the jitter compared to the plain scenario.

The total number of generated acknowledgements depends on the value of the packet counter ACK threshold. If we acknowledge every packet, it is equal to the number of transmitted packets times the number of intermediate nodes from the current node to the gateway. For example, in scenario 4 when node A pings the gateway (2 intermediate hops) the total number of transmitted acknowledgements is 8000, because ping consists of two messages (echo request and response).

We continue with the analysis of different packet counter ACK thresholds in scenario 5, when node A pings the gateway and all nodes have full CASHnet functionality. We already described Figure 6.9a, where every single packet is acknowl-



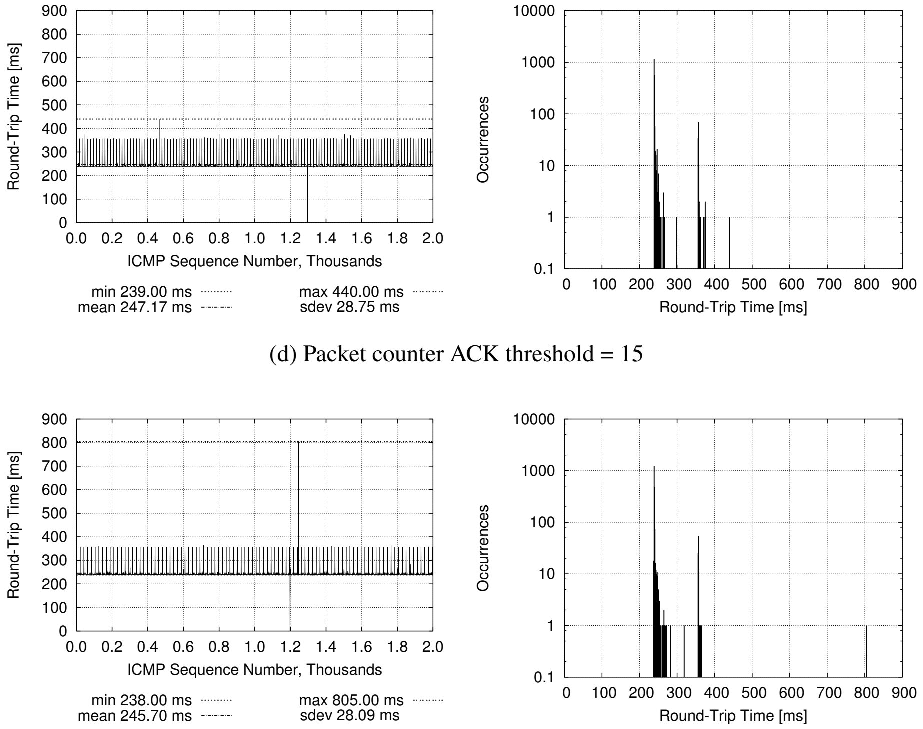
(a) Packet counter ACK threshold = 1

(b) Packet counter ACK threshold = 5

(c) Packet counter ACK threshold = 10

Figure 6.9: Round-trip times for scenario 5 under different packet counter ACK thresholds 1/2

edged. When we acknowledge every fifth packet, we generate a total of 1600 acknowledgements instead of 8000. As shown in the Figure 6.9b, the average round trip time decreases by 98 ms from 358 to 260 ms. We note an oscillating effect on the measured round-trip time and thus a clustering of the jitter with two peaks at approximately 239 ms with 1012 occurrences and at 356 ms with 150 occurrences. The other measured delays lie close to one of the two peaks. The first peak



(d) Packet counter ACK threshold = 15

(e) Packet counter ACK threshold = 20

Figure 6.9: Round-trip times for scenario 5 under different packet counter ACK thresholds 2/2

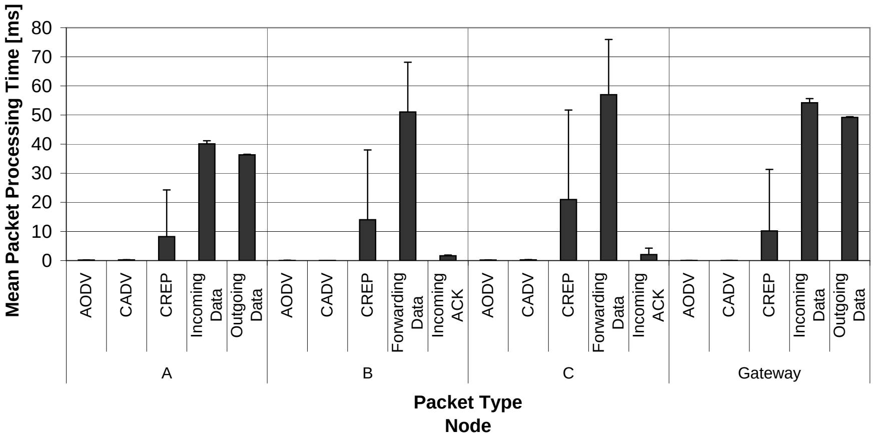
represents the packets, which do not trigger an acknowledgement (1-4) and the second peak for the fifth packet in a row from the same forwarder, which trigger the generation of an acknowledgement. In case we increase the packet counter ACK threshold to 10 (total of 800 acknowledgements) as shown in the Figure 6.9c, we witness another decrease in the mean delay by 9 to 251 ms. Here, the first peak slightly increases to 1219 occurrences and the second peak decreases to 111 occurrences. This is due to the reduction in the frequency of the acknowledgement generation. In Figure 6.9d we acknowledge every 15th packet and thus generate a total of 533 acknowledgements, which results in an average round-trip time of 247 ms. We also notice an increase in the jitter of the first peak and a further reduction of the second peak. In Figure 6.9e we use a packet counter ACK threshold of 20 (total of 400 acknowledgements) and measure a mean delay of 246 ms. The slow down in the reduction of the delay is caused by the linear increase of the packet counter ACK threshold. Our results from scenario 3 (partial, no acknowledgements signed), which we previously described in Figure 6.7c, indicates that the lower boundary for the mean delay on node A is approximately 239 ms. A further increase in the packet counter ACK threshold will approach this delay.

6.7.2 Packet Processing Time

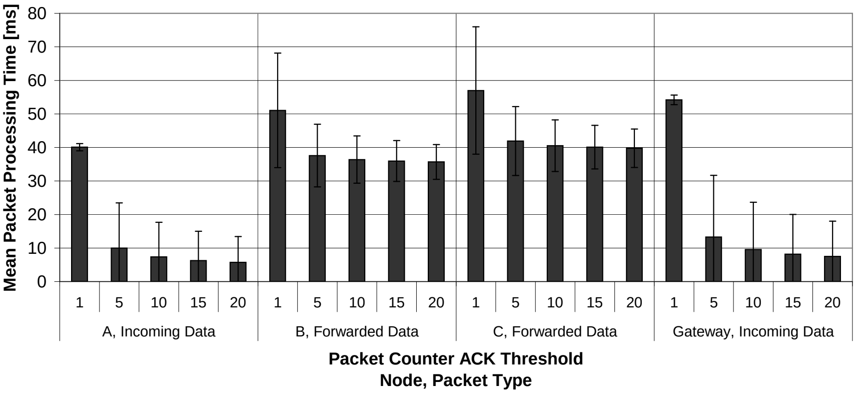
The packet processing time directly affects the delay on the network and gives us information about the impact of specific CASHnet operations in relation to the packet type. We note, that because of the computationally expensive security functions, the processor type in the node affects the measured results. We distinguish the data packets according to the chain via which we intercepted them, i.e. incoming, forwarded or outgoing data. In addition, we gather the processing time for acknowledgements on nodes which receive them. Besides, we monitor how much time the processing of certificate advertisements $CADV$, certificate replies $CREP$ and $AODV$ packets consumes. Figure 6.10 contains the average packet processing time for all nodes in scenario 5, where node A pings the gateway under different packet counter ACK thresholds. In this scenario node A generates the echo requests and the gateway generates the echo responses. Both nodes do not receive any acknowledgements because they are the endpoints of the bidirectional communication. Node B and C are intermediate forwarding nodes and therefore receive acknowledgements. Figure 6.10a shows the processing time for all packets on the respective nodes in the case, where we acknowledge every packet. Figure 6.10b shows the mean processing time on each node for those data packet types, where we measured considerable changes for the different packet counter ACK thresholds. Changes occur in the processing of the incoming and the forwarding data on the respective node, because the processing time for data packets includes the time to generate and transmit an acknowledgement.

From Figure 6.10a we see the time it takes to process the different packet types on each node, when we acknowledge every packet in scenario 5. On all nodes, we measure very low average processing times for AODV and certificate advertisement packets. Both vary between 50 and 200 μs. We explain the variation with the impact of the operating system scheduler. As illustrated in Figure 6.4 on page 133, AODV packets are immediately accepted, i.e. handed back to the network stack, upon interception. Certificate advertisements are also immediately accepted unless the current node is the destination. In this case, the node creates a certificate reply.

In scenario 5, node A is the originator of the echo requests. CASHnet requires on average 36 ms to digitally sign these packets on node A (outgoing data). The processing of this echo request at the gateway requires 54 ms (incoming data), which includes the time to generate an acknowledgement for the last forwarding node C. The gateway sends back an echo response and CASHnet takes an average of 50 ms to digitally sign it (outgoing data). Node A requires 40 ms to process this response (incoming data). Again, this duration includes the generation of an acknowledgement for the last forwarding node B. We note, that the processing of both incoming and outgoing data takes 14 ms longer at the gateway compared to node A. We explain this difference with different processor speeds on node A (2.4 GHz) and the gateway (1.4 GHz). The intermediate nodes B and C only forward data. Thus, they receive acknowledgements from their respective next hops. CASHnet on node B requires 52 ms to verify and digitally sign the data (echo requests and



(a) Node A, B, C and the gateway with packet counter ACK threshold = 1



(b) Data on node A, B, C and the gateway for different packet counter ACK threshold

Figure 6.10: Mean packet processing times and standard deviation of all nodes for scenario 5

responses) as well as acknowledgement the responses forwarded by node C. The same processes takes 57 ms on node C. Again, we attribute the difference to the distinct processors of node B and C. The first is a new Pentium M with 2 MB cache, the latter an older Celeron with 128 KB cache. The total processing delay for data packets in CASHnet on all nodes sums up to 289 ms. From the round-trip time measurements we obtained a delay of 358 ms. We explain the missing 69 ms with influences from outside CASHnet in the operating system or from the possible interaction of AODV and CASHnet while using netfilter. The verification of received acknowledgements (incoming ACK) last 2 ms on both nodes.

In addition, the time to process a certificate reply changes between the different nodes. Every 5 minutes or 300 pings, the certificates must be refreshed and thus the nodes send a certificate advertisement and answer with a certificate reply. The reception of a certificate reply causes the processing (digital signature) of eventually queued packets. Depending on the number of packets in the queue this can cause a significant delay, which is included in the processing time of the certificate reply. From Figure 6.10a we observe that the average processing delay on the sender and receiver nodes is lower than on the intermediate nodes. In contrast to node A and the gateway, the intermediate nodes B and C receive acknowledgements for their forwarding and have to process them. Thus, in case a certificate becomes invalid more packets wait in the queue and must be processed when the certificate reply arrives, which in turn results in higher delays. As for the forwarding data, we attribute the difference between node B and C to the more powerful processor in node C.

Figure 6.10b shows the average data packet processing time on all nodes. Node A and the gateway generate and receive data. However, only incoming data shows a change in the processing time for different packet counter ACK thresholds. We see that, an increase in the packet counter ACK threshold reduces the processing time for the incoming data, e.g. from 40 to 10 ms on node A and from 54 to 13 ms on the gateway, when we acknowledge every fifth instead of every single packet. When CASHnet receives a packet destined to the current node, it rewards the forwarding node before it passes the packet to the local process. Thus, the reduced number of acknowledgements, which need to be transmitted for the same amount of data received, reduce the processing time for the incoming data packets. As in the previous round-trip time measurements we noticed a slow-down in the decrease for higher packet counter ACK thresholds (10, 15, 20), because the number of generated acknowledgements decreases slowly.

Because B and C are intermediate node, they receives data packets to be forwarded. The processing time for forwarded data packets decreases from 51 to 38 ms on node B and from 57 to 42 ms on node C, when we increase the packet counter ACK threshold from 1 to 5. Compared to node A and the gateway the reduction is not so high, because the intermediate nodes still have to verify and digitally sign data packets in both directions. Node A and the gateway have no packets to forward and thus only need to digitally sign outgoing packets. In addition, the intermediate nodes have to process incoming acknowledgements from their one-hop neighbors.

6.7.3 Summary

In the tests with our CASHnet implementation, we showed the impact of the different operations on the packet processing time on each node as well as the round-trip time between communicating nodes. We found the average round-trip time for a communication over three hops to be 358 ms. We could reduce this delay to 246 ms by acknowledging only every 20th packet. With a one-way transmission time of 123 ms, we are below the recommended maximum delay of 150 ms for good voice quality using echo cancelers. We note, that these results are from our first implementation of CASHnet and that we see some room to improve the performance of the code. We found our test environment and in particular the processing power of the laptops as well as the disturbances on the wireless medium to influence the results. The latter are difficult to detect without disturbing the ongoing measurement.

6.8 Conclusion

We implemented CASHnet under Linux using netfilter/iptables and conducted several evaluations in a small testbed using laptops with wireless network interfaces in order to measure the end-to-end delay imposed by CASHnet. To support the identification of the cause for the delay, we analyzed the processing time for each packet type distinguished by the CASHnet implementation. The results of the evaluation indicate high but acceptable end-to-end delays for communications over three hops in our testbed.

The implementation helped us to adapt our CASHnet algorithm to a real environment and showed us possible limitations. The processing power of the nodes is a limiting factor, as the security functionality is computationally expensive. Also, high variations in the processing power of the nodes causes certain nodes to take longer to process packets from queues, which in turn increases the jitter. We see possible improvements in the optimization of the code, such as the queue handling as well as the test environment, e.g. longer interval between periodic certificate one-hop broadcasts and shorter key length with an appropriate certificate lifetime.

Chapter 7

Conclusions and Outlook

7.1 Conclusions

In this thesis, we studied cooperation schemes for multi-hop cellular networks, that stimulate the packet forwarding among nodes with the help of incentives. We proposed and designed a complete cooperation and accounting framework, which gives the provider a secure and profitable way of operating a multi-hop cellular network and the customer the benefits of increased wireless broadband coverage. We implemented our cooperation scheme in the network simulator to analyze and evaluate its performance as well as to identify its optimal operation parameters. In addition, we performed an implementation under Linux and conducted evaluations in a testbed with laptops.

As we motivated in Chapter 3, cooperation among nodes must be ensured - otherwise connections over multiple hops become impossible and the network falls apart. This is true, especially when we consider the commercial application of multi-hop cellular networks, where individual customers have no pre-established social links and thus no reason to forward packets from other nodes. Instead, the customer will act selfish and give priority to her own packets, because she is concerned about the power consumption on her device. Cooperation can be ensured in two ways: detection-based approaches use the fear of punishment in case of selfishness, motivation-based approaches use the hop for rewards in case of cooperation. We believe, that motivation-based approaches are more suitable in civilian application scenarios. Most detection-based approaches have a decentralized design, while the majority of motivation-based approaches rely on a centralized design. However, centralization takes away the flexibility and dynamics of the multi-hop cellular network architecture and thereby limits the possible application scenarios. This situation motivated us to research an architecture, which retains the flexibility of the mobile ad hoc communication pattern and at the same time assures cooperation among the individual customers.

Our research led to a cooperation and accounting strategy for hybrid wireless networks called CASHnet, which we proposed in Chapter 4. CASHnet is motivation-based and therefore introduces rewards as well as charges in the form of virtual money. We raise charges for both, the transmission of self-generated packets and the reception of packets destined to the current node. And we issue rewards for forwarded packets. The architecture and mechanism for the charging and rewarding is completely decentralized and operates independently on each node. In order to keep the provider of the multi-hop cellular network in control and reduce the risk of misuse, we designed a centralized refill process. We introduced a new component called service station, where the customer can load up her virtual money account. The service station is an immobile terminal with a low-bandwidth connection to the accounting center of the provider, much like a terminal for prepaid cards. To ensure the regular visit of service stations, we decided that the rewards can not be directly used to cover the charges, but must be exchanged at the service station first. In addition, the customer can buy additional virtual money at the service station using real money. To compensate for the immobile characteristics of service stations, we introduce resellers, which are allowed to trade the rewards from other nodes for their own virtual money.

In order to ensure the correct tracking of the rewards and charges, we must protect CASHnet against misuse. Our security mechanisms rely on public-key cryptography and we use digital signatures to ensure the integrity of messages, as well as the authentication of their origin. The virtual money accounts are stored on a tamper-resistant device along with credentials.

CASHnet has the advantage, that it gives more freedom to the provider and the customer at the same time. First, CASHnet does not rely on source routing, it only requires the hop count to the gateway - or in case of globally fixed charges not even that information is required. So the provider has a free choice in the deployed routing protocol. Second, CASHnet does not require a centralized authentication based on sessions at the gateways, which provide the interconnection to the backbone network. Thus, neither customer nor provider need to maintain a security session. Third, the support for sender and receiver-based charges allows the provider to separate the costs between incoming and outgoing traffic for the respective multi-hop cellular network. This is especially important considering, that a large percentage of traffic is directed downstream, from the gateway to the customer and sessions would need to be maintained in order to correctly attribute the involved costs. Also, all three characteristics increase the level of supported node mobility. Further, CASHnet provides instruments to support network management, in particular network planning.

The probably biggest disadvantage of CASHnet is the use of public-key cryptography, because this results in a high computational overhead. However, in contrast to secret-key cryptography, public-key cryptography truly supports a decentralized and distributed authentication infrastructure with the use of the certificates. Another possible drawback of CASHnet is, that, due to the decentralized accounting, it requires the customer to visit a service station or contact a reseller periodically. However, as the reseller is mobile, he can go to the customers and offer his service on site. Last, CASHnet does not provide strong protection against malicious attacks from adversaries. It does however provide the possibility of identifying an attacker, because every node has to authenticate before it can take part in the network. Further, the service station can be used to report suspicious nodes and the short certificate lifetime ensures that even the attacker has to regularly visit the service station, where the provider can deny the renewal of the certificate.

In order to evaluate CASHnet regarding its impact on the network performance, we implemented it in the network simulator ns-2. We described the implementation process and analyzed the simulation results in Chapter 5. We also implemented Nuglet, an incentive-based cooperation scheme for mobile ad hoc networks which inspired our work and compared it with the CASHnet performance. From our simulation results we find, that CASHnet can come close to the performance of a multi-hop cellular network without any cooperation mechanisms deployed. We also identified the optimal operation parameters for CASHnet in the respective simulation scenario. But we also found, that the decentralization imposes a considerable burden on the network performance. In particular, the normal data traffic has to compete with the reward messages.

While the simulations allowed us to evaluate and optimize CASHnet, we were interested in the real-life performance of our cooperation scheme and therefore conducted a Linux implementation, which we described in Chapter 6. In particular we analyzed the introduced end-to-end delay and jitter as well as the computational overhead imposed on a node. In our first implemented prototype, we found the introduced delay to be tolerable (within ITU recommendations for good voice quality) for a route length of 3 hops from the node to the gateway. Longer routes to the gateway imply more intermediate forwarding nodes, which introduce more delay, because of the computationally expensive security functions. However, we see room for optimizations in the used cryptographic library as well as in the message processing on the node.

We summarize the main conclusions from the work performed as follows. The attractiveness of multi-hop wireless network clearly lies in their low infrastructure requirements, and the thereby gained flexibility which is expressed in the decentralized design pattern of these networks. However, most of the existing incentive-based cooperation schemes for multi-hop wireless networks follow a centralized paradigm. In order to keep track of the node participation (transmission, forwarding or reception), they use traces, which are collected at central locations. This typically leads to restrictions, which reduce the flexibility of multi-hop communications, e.g. by requiring source routing. In the decentralized paradigm, the authentication and accounting is performed among the nodes. Here, each node requires some tamper proof device to securely store the virtual money and credentials. The only decentralized cooperation scheme so far is targeted at mobile ad hoc networks and assumes a self-perpetuating cycle of virtual money in the network. To find the right balance between a centralized and decentralized paradigm for the cooperation schemes is a challenging task.

We proposed CASHnet, a framework to ensure cooperation in multi-hop wireless networks, which follows a decentralized design pattern. We designed CASHnet to address the drawbacks of existing cooperation schemes, that it retains the flexibility of multi-hop networks and ensures an almost continuous availability of virtual money. We achieved the flexibility by using decentralized authentication as well as charging and rewarding mechanisms. For a continuous flow of virtual money we incorporate service stations, where customers can refill their virtual money accounts. Wee see CASHnet as a possibility to make multi-hop cellular networks commercially viable.

7.2 Outlook

While we studied incentive-based cooperation schemes for multi-hop cellular networks, we discovered several issues directly related to cooperation or to multi-hop wireless networks in general, which were however out of scope of this thesis. Cooperation clearly is a cross layer subject, like management and security. In the following, we briefly analyze these open research questions.

Incentive-based cooperation schemes use the hope for rewards of the customer in order to motivate their cooperation. In contrast to detection-based approaches, punishment of nodes based on neighborhood monitoring is typically not intended. However, by introducing incentives in the fundamental network operation, the attractiveness for fraudulent adversaries increases and thus the risk of corresponding attacks. Therefore, the protection of the charging and rewarding mechanisms is of great importance. Unfortunately, the decentralized characteristics of multi-hop cellular networks make this a challenging task. With multi-hop wireless connections, the provider has no direct control to what happens beyond the first hop from his gateways. After the first hop, the provider can only rely on reports conducted by monitoring network participants. To be of any use, the non-repudiation of these reports must be assured, i.e. their originator must be identifiable and their integrity verifiable.

In CASHnet, we digitally sign all transmitted messages and thereby ensure their non-repudiation. The possibility of being identifiable reduce the attractiveness of fraudulent and also malicious attacks. However, we did not further investigate a possible monitoring framework for the provider of a multi-hop wireless network. We believe this to be a challenging and interesting task for future research.

Closely related to the monitoring framework is the management of multi-hop wireless networks. A provider requires means to ensure the correct operation of his network as well as indicators for the future planning. We briefly described possible planning indicators (both general and specific to CASHnet) and how to interpret them. However, this is far from complete and it remains a challenging task to make multi-hop wireless networks manageable and thereby support the commercial success.

With the current version of CASHnet, we target multi-hop cellular networks operated by a provider. An interesting direction would be to extend our framework to also stimulate cooperation in mobile ad hoc networks, which are currently tolerated as coexistent. For security reasons, we envision an architecture with a central instance such as a certificate provider to whom possible misuse could be reported. Again, this requires a monitoring framework with possible reporting functionality (at the service stations) to the provider.

A point, which we encountered during the implementation phase in the network simulator as well as under Linux were the problems caused by the non-existing or weak cross-layer communication. In the simulator, congestion was interpreted as link break and caused the routing protocol to search for new paths, which further increased the load on the network. Also, in real implementations, the link layer triggers are not optimal and current routing information of AODV is difficult to obtain. Cross-layer design is an emerging field, that gained particular attention with the increase of research in the area of multi-hop communication, e.g. in mobile ad hoc and wireless sensor networks. Cross-layer tasks like cooperation, management and security will surely benefit from the eased access to information and triggers in such a design.

In the area of cooperation, we see possibilities for further conceptual research. CASHnet focusses on individual customers in small multi-hop networks with 50 to probably 100 nodes. An interesting task will be the adoption of CASHnet to large scale ad hoc networks. A solution could be the formation of groups to cooperation clusters, for example based on pre-established social links. Here, two types of cooperation exist, intra-cluster and inter-cluster cooperation. In general, the cooperation level inside the cluster will reflect on the cooperation between clusters, because if nodes inside a cluster do not forward packets, no packet can travel over multiple clusters (containing many selfish nodes). Therefore, it suffices for the provider of the incentive-based cooperation framework to handle the clusters and charge for their originating traffic and reward the transit traffic. In this scenario, we see room for a variety of problems to research.

Further, we'd like to comment on the privacy issue, which is very popular, especially in the area of (multi-hop) wireless networks. In these networks, the communication passes via computers operated by individual customers, and not only by some providers. An individual customer must therefore extend her trust to a multitude of unknown people, which can retrieve information about her communication, i.e. the virtual identity of originator and destination as well as the communication content. We think, that communication is by default not anonymous. There is always an originator of and a recipient for the exchanged information. Further, we believe that there is no anonymity, only pseudonymity in communication. The link between the pseudonym and the true identity is only protected via trust to the pseudonym provider. Incentive-based cooperation schemes rely on the collection of charges and the distribution of rewards. To correctly attribute these actions to the concerned people, a unique and verifiable identity is required. Another advantage of the verifiable identification of a network participant is, that it reduces the attractiveness of misuse from his side as the actions can be easier tracked to its source. To find a good balance between security and privacy remains a challenging task.

List of Figures

List of Tables

List of Algorithms

Bibliography

[ACG04] G. Anastasi, M. Conti, and E. Gregori. IEEE 802.11 Ad Hoc Networks: Protocols, Performance and Open Issues. In S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, eds., *Mobile Ad Hoc Networking*, chap. 3, pp. 69–116. Wiley-IEEE Press, 2004.

[AFN04] M. K. A. Aziz, P. N. Fletcher, and A. R. Nix. Performance analysis of IEEE 802.11n solutions combining MIMO architectures with iterative decoding and sub-optimal ML detection via MMSE and Zero forcing GIS solutions. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1451–1456. Atlanta, GA, USA, Mar. 2004.

[AK96] R. Anderson and M. Kuhn. Tamper Resistance - a Cautionary Note. In *Proceedings of 2nd USENIX Workshop on Electronic Commerce*, pp. 1–11. Oakland, CA, USA, Nov. 1996.

[AMM01] Ananthapadmanabha R., B. S. Manoj, and C. S. R. Murthy. Multi-hop Cellular Networks: The Architecture and Routing Protocols. In *Proceedings of 12th IEEE International Symposium on Personal, Indoor, Mobile Radio Communications (PIMRC)*, pp. 78–82. San Diego, CA, USA, Sep.–Oct. 2001.

[AOD05] Ad hoc On Demand Distance Vector. Website, 2005. URL http://moment.cs.ucsb.edu/AODV/. [visited 2 Sep. 2005].

[ASSC02] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A Survey on Sensor Networks. *IEEE Communications Magazine*, vol. 40(8):102–114, Aug. 2002.

[AWW05] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Elsevier Journal of Computer Networks*, vol. 47(4):445–487, Mar. 2005.

[Axa05] Axalto e-gate USB smart card. Website, 2005. URL http://www.axalto.com/infosec/egate.asp. [visited 31 Aug. 2005].

[Axe85] R. Axelrod. *The Evolution of Cooperation*. Basic Books, reprint edn., 1985.

[Bar64] P. Baran. On Distributed Communications Networks. *IEEE Transactions on Communications*, vol. 12(1):1–9, Mar. 1964.

[BBE+99] M. Baentsch, P. Buhler, T. Eirich, F. Höring, and M. Oestreicher. JavaCard - From Hype to Reality. *IEEE Concurrency*, vol. 7(4):36–43, Oct.-Dec. 1999.

[BBHJ03] N. Ben Salem, L. Buttyán, J.-P. Hubaux, and M. Jakobsson. A Charging and Rewarding Scheme for Packet Forwarding in Multi-hop Cellular Networks. In *Proceedings of 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 13–24. Annapolis, MD, USA, Jun. 2003.

[BBHJ05] N. Ben Salem, L. Buttyán, J.-P. Hubaux, and M. Jakobsson. Node Cooperation in Hybrid Ad hoc Networks. *IEEE Transactions on Mobile Computing*, 2005. To appear.

[BCG05] R. Bruno, M. Conti, and E. Gregori. Mesh Networks: Commodity Multihop Ad Hoc Networks. *IEEE Communications Magazine*, vol. 43(3):123–131, Mar. 2005.

[BEF+00] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu. Advances in Network Simulation. *IEEE Computer*, vol. 33(5):59–67, 2000.

[BH00] L. Buttyán and J.-P. Hubaux. Enforcing Service Availability in Mobile Ad-Hoc WANs. In *Proceedings of 1st ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 87–96. Boston, MA, USA, Aug. 2000.

[BH03a] L. Buttyán and J.-P. Hubaux. Report on a Working Session on Security in Wireless Ad Hoc Networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7(1):74–94, Jan. 2003.

[BH03b] L. Buttyán and J.-P. Hubaux. Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks. *ACM Mobile Networks & Applications*, vol. 8(5):579–592, Oct. 2003.

[BHPC04] C. Bettstetter, H. Hartenstein, and X. Prez-Costa. Stochastic Properties of the Random Waypoint Mobility Model. *ACM/Kluwer Wireless Networks*, vol. 10(5):555–567, Sep. 2004.

[Big05] BigChampagne Online Media Measurement. Website, 2005. URL http://www.bigchampagne.com/. [visited 22 Jul. 2005].

[BL02] S. Buchegger and J.-Y. Le Boudec. Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes - Fairness In Dynamic Ad-hoc NeTworks). In *Proceedings of 3rd ACM International*

Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), pp. 226–236. Lausanne, Switzerland, Jun. 2002.

[BL03] S. Buchegger and J.-Y. Le Boudec. The Effect of Rumor Spreading in Reputation Systems for Mobile Ad-Hoc Networks. In *Proceedings of 1st Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*. Sophia-Antipolis, France, Mar. 2003.

[BL05] S. Buchegger and J.-Y. Le Boudec. Self-Policing Mobile Ad Hoc Networks by Reputation Systems. *IEEE Communications Magazine*, vol. 43(7):101–107, Jul. 2005.

[BR04] E. Belding-Royer. Routing Approaches in Mobile Ad Hoc Networks. In S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, eds., *Mobile Ad Hoc Networking*, chap. 10, pp. 275–300. Wiley-IEEE Press, 2004.

[BRSP01] E. M. Belding-Royer, Y. Sun, and C. E. Perkins. Global Connectivity for IPv4 Mobile Ad hoc Networks. Internet-Draft, Nov. 2001. URL http://www.cs.ucsb.edu/~ebelding/txt/globalv4.txt. Expired. [visited 8 Aug. 2005].

[CBR04] I. D. Chakeres and E. M. Belding-Royer. AODV Routing Protocol Implementation Design. In *Proceedings of 24th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pp. 698–703. Hachioji, Tokyo, Japan, Mar. 2004.

[CCL03] I. Chlamtac, M. Conti, and J. J.-N. Liu. Mobile Ad Hoc Networking: Imperatives and Challenges. *Elsevier Journal of Ad Hoc Networks*, vol. 1(1):13–64, July 2003.

[CJL$^+$01] T. Clausen, P. Jacquet, A. Laouiti, P. Mühlethaler, A. Qayyum, and L. Viennot. Optimized Link State Routing Protocol. In *Proceedings of 5th IEEE International Multitopic Conference (INMIC)*, pp. 62–68. Lahore, Pakistan, Dec. 2001.

[CMTG04] M. Conti, G. Maselli, G. Turi, and S. Giordano. Cross-Layering in Mobile Ad Hoc Network Design. *IEEE Computer*, vol. 37(2):48–51, Feb. 2004.

[DBSW67] D. W. Davies, K. A. Bartlett, R. A. Scantlebury, and P. T. Wilkinson. A Digital Communications Network For Computers. In *Proceedings of 1st ACM Symposium on Operating Systems Principles (SOSP)*, pp. 2.1–2.17. Gatlinburg, TN, USA, Oct. 1967.

[DTH02] O. Dousse, P. Thiran, and M. Hasler. Connectivity in ad-hoc and hybrid networks. In *Proceedings of 21st Annual Joint Conference* *of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 2, pp. 1079 – 1088. New York, NY, USA, Jun. 2002.

[FHB05] M. Félegyházi, J.-P. Hubaux, and L. Buttyán. Nash Equilibria of Packet Forwarding Strategies in Wireless Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 2005. To appear.

[FL01] J. A. Freebersyser and B. Leiner. A DoD perspective on mobile ad hoc networks. In C. E. Perkins, ed., *Ad Hoc Networking*, pp. 29–51. Addison-Wesley, 2001.

[FMP04] M. Frank, P. Martini, and M. Plaggemeier. CineMA: Cooperation Enhancement in Manets. In *Proceedings of 29th IEEE Conference on Local Computer Networks (LCN)*, pp. 86–93. Tampa, FL, USA, Nov. 2004.

[FP05] P. Fuchs and G. Pemmasani. NdisWrapper, 2005. URL http://ndiswrapper.sourceforge.net/. [visited 2 Sep. 2005].

[FV03] K. Fall and K. Varadhan, eds. *The ns Manual (formerly ns Notes and Documentation)*. The VINT Project, Dec. 2003. URL http://www.isi.edu/nsnam/ns/doc/. [visited 20 Jul. 2005].

[GK00] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, vol. 46(2):388–404, Mar. 2000.

[GWAC05] A. Ghosh, D. R. Wolter, J. G. Andrews, and R. Chen. Broadband Wireless Access with WiMax/8O2.16: Current Performance Benchmarks and Future Potential. *IEEE Communications Magazine*, vol. 43(2):129–136, Feb. 2005.

[Ham03a] A. Hamidian. AODV+, 2003. URL http://www.telecom.lth.se/Personal/alexh/. [visited 11 Aug. 2005].

[Ham03b] A. Hamidian. *A Study of Internet Connectivity for Mobile Ad Hoc Networks in NS 2*. Master's thesis, Lund University, Sweden, Jan. 2003.

[HCW04] E. Huang, J. Crowcroft, and I. Wassell. Rethinking Incentives for Mobile Ad Hoc Networks. In *Proceedings of ACM SIGCOMM Workshop on Practice and Theory of Incentives and Game Theory in Networked Systems (PINS)*, pp. 191–196. Portland, OR, USA, Aug.–Sep. 2004.

[HL00] Y.-C. Hsu and Y.-D. Lin. Multihop Cellular: A New Architecture for Wireless Communications. In *Proceedings of 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pp. 1273–1282. Tel Aviv, Israel, Mar. 2000.

[HP01] Z. J. Haas and M. R. Pearlman. ZRP: a hybrid framework for routing in Ad Hoc networks. In C. E. Perkins, ed., *Ad Hoc Networking*, pp. 221–253. Addison-Wesley, 2001.

[HS02] H.-Y. Hsieh and R. Sivakumar. On Using the Ad-hoc Network Model in Cellular Packet Data Networks. In *Proceedings of 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 36–47. Lausanne, Switzerland, Jun. 2002.

[I1105] IEEE 802.11 Working Group on Wireless Local Area Network Standards. Website, 2005. URL http://www.ieee802.org/11/. [visited 20 May 2005].

[I1505] IEEE 802.15 Working Group on Wireless Personal Area Network Standards. Website, 2005. URL http://www.ieee802.org/15/. [visited 20 May 2005].

[I1605] IEEE 802.16 Working Group on Broadband Wireless Access Standards. Website, 2005. URL http://www.ieee802.org/16/. [visited 20 May 2005].

[I2005] IEEE 802.20 Working Group on Mobile Broadband Wireless Access Standards. Website, 2005. URL http://www.ieee802.org/20/. [visited 20 May 2005].

[IEE99] IEEE Standards for Information Technology - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. ANSI/IEEE Std. 802.11, 1999 Edition (R2003), Sep. 1999.

[IEE02] IEEE Standard for Information technology - Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs). IEEE Std 802.15.1-2002, Jun. 2002.

[IEE03a] IEEE Standard for Information technology - Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPAN). IEEE Std 802.15.3-2003, Sep. 2003.

[IEE03b] IEEE Standard for Information technology - Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs). IEEE Std 802.15.4-2003, Oct. 2003.

[IEE04] IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems. IEEE Std 802.16-2004, Oct. 2004.

[IEE05] Draft Amendment to IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems- Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands. IEEE Std 802.16e Draft Version 8, 2005.

[ISO94] ISO/IEC. Open Systems Interconnection - Basic Reference Model. ISO/IEC 7498-1:1994, Nov. 1994.

[ITU03] ITU-T Recommendation G.114 - One-way Transmission Time. ITU-T G.114 (05/03), May 2003. URL http://www.itu.int/. [visited 9 Sep. 2005].

[JHB03] M. Jakobsson, J.-P. Hubaux, and L. Buttyán. A Micro-Payment Scheme Encouraging Collaboration in Multi-Hop Cellular Networks. In *Proceedings of 7th International Financial Cryptography Conference*, pp. 15–33. Gosier, Guadeloupe, Jan. 2003.

[JMB01] D. B. Johnson, D. A. Maltz, and J. Broch. DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. In C. E. Perkins, ed., *Ad Hoc Networking*, chap. 5, pp. 139–172. Addison-Wesley, 2001.

[JMH04] D. B. Johnson, D. A. Maltz, and Y.-C. Hu. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR). Internet-Draft, Jul. 2004. URL http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt. [visited 9 Sep. 2005].

[JSAC01] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen. A Survey of Energy Efficient Network Protocols for Wireless Networks. *ACM/Kluwer Wireless Networks*, vol. 7(4):343–358, Aug. 2001.

[KB04] L. Klein-Berndt. Kernel AODV, May 2004. URL http://www.antd.nist.gov/wctg/aodv_kernel/. [visited 22 Apr. 2005].

[KT03] U. C. Kozat and L. Tassiulas. Throughput Capacity of Random Ad Hoc Networks with Infrastructure Support. In *Proceedings of 9th ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 55–65. San Diego, CA, USA, Sep. 2003.

[KV00] Y.-B. Ko and N. H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. *ACM/Kluwer Wireless Networks*, vol. 6(4):307–321, Jul. 2000.

[Lat05] C. Latze. *Implementation and Evaluation of CASHnet in a Real-World Scenario*. Master's thesis, Univeristy of Bern, 2005. To appear.

[LBB04] S. Lee, S. Banerjee, and B. Bhattacharjee. The Case for a Multi-hop Wireless Local Area Network. In *Proceedings of 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pp. 894–905. Hong Kong, China, Mar. 2004.

[LHO$^+$00] Y.-D. Lin, Y.-C. Hsu, K.-W. Oyang, T.-C. Tsai, and D.-S. Yang. Multihop Wireless IEEE 802.11 LANs: A Prototype Implementation. *Journal of Communications and Networks*, vol. 2(4):372–378, Dec. 2000.

[LLT03] B. Liu, Z. Liu, and D. Towsley. On the Capacity of Hybrid Wireless Networks. In *Proceedings of 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pp. 1543–1552. San Francisco, CA, USA, Mar.–Apr. 2003.

[LPW03] B. Lamparter, K. Paul, and D. Westhoff. Charging support for ad hoc stub networks. *Elsevier Journal of Computer Communications*, vol. 26(13):1504–1514, Aug. 2003.

[LPW05] B. Lamparter, M. Plaggemeier, and D. Westhoff. Estimating the value of co-operation approaches for multi-hop ad hoc networks. *Elsevier Journal of Ad Hoc Networks*, vol. 3(1):17–26, Jan. 2005.

[LRS$^+$03] H. Luo, R. Ramjeey, P. Sinhaz, L. E. Liy, and S. Lu. UCAN: A Unified Cellular and AdHoc Network Architecture. In *Proceedings of 9th ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 353–367. San Diego, CA, USA, Sep. 2003.

[MDS02] T. S. Messerges, E. A. Dabbish, and R. H. Sloan. Examining Smart-Card Security under the Threat of Power Analysis Attacks. *IEEE Transactions on Communications*, vol. 51(5):541–552, May 2002.

[MGLB00] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In *Proceedings of 6th ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 255–265. Boston, MA, USA, Aug. 2000.

[MM02] P. Michiardi and R. Molva. CORE: A Collaborative Reputation Mechanism to enforce node cooperation in Mobile Ad hoc Networks. In *Proceedings of 6th IFIP Conference on Communications and Multimedia Security (CMS)*, pp. 107–121. Portoroz, Slovenia, Sep. 2002.

[Mot05] Motorola Mobile MeshNetworks. Website, 2005. URL http://www.meshnetworks.com. [visited 23 Aug. 2005].

[MR02] S. Micali and R. L. Rivest. Micropayments Revisited. In *Proceedings of The Cryptographer's Track at RSA Conference on Topics in Cryptology (CT-RSA)*, pp. 149–163. San Jose, CA, USA, Apr. 2002.

[MWH01] M. Mauve, J. Widmer, and H. Hartenstein. A Survey on Position-Based Routing in Mobile Ad Hoc Networks. *IEEE Network*, vol. 15(6):30–39, Nov.–Dec. 2001.

[Nas50] J. F. Nash. Equilibrium points in N-Person Games. *Proceedings of the National Academy of Sciences*, vol. 36(1):48–49, Jan. 1950.

[NC04] W. Navidi and T. Camp. Stationary Distributions for the Random Waypoint Mobility Model. *IEEE Transactions on Mobile Computing*, vol. 3(1):99–108, Jan.–Mar. 2004.

[Nor04] E. Nordström. AODV-UU, Dec. 2004. URL http://core.it.uu.se/AdHoc/AodvUUImpl/. [visited 10 May 2005].

[NPSQ03] M. Neve, E. Peeters, D. Samyde, and J.-J. Quisquater. Memories: a Survey of their Secure Uses in Smart Cards. In *IEEE International Security in Storage Workshop (SISW)*, pp. 62–72. Washington, DC, USA, Oct. 2003.

[NS204] The Network Simulator ns-2, Jan. 2004. URL http://www.isi.edu/nsnam/ns/. [visited 20 Jul. 2005].

[PB94] C. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *Proceedings of ACM Annual Conference of the Special Interest Group on Data Communication (SIGCOMM)*, pp. 234–244. London, UK, Sep. 1994.

[PBRD03] C. E. Perkins, E. M. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, Jul. 2003. URL http://www.ietf.org/rfc/rfc3561.txt. [visited 15 Jul. 2005].

[PR99] C. E. Perkins and E. M. Royer. Ad-hoc On-Demand Distance Vector Routing. In *IEEE Workshop on Mobile Computing Systems and Applications, (WMCSA)*, pp. 90–100. New Orleans, LA, USA, Feb. 1999.

[PRP02] B. Patil, P. Roberts, and C. E. Perkins. IP Mobility Support for IPv4. RFC 3344, Aug. 2002. URL http://www.ietf.org/rfc/rfc3344.txt. [visited 15 Jul. 2005].

[PW02] K. Paul and D. Westhoff. Context Aware Detection of Selfish Nodes in DSR based Ad-hoc Networks. In *Proceedings of IEEE GLOBECOM*, pp. 178–182. Taipei, Taiwan, Nov. 2002.

[PWS$^+$04] R. Pabst, B. H. Walke, D. C. Schultz, P. Herhold, H. Yanikomeroglu, S. Mukherjee, H. Viswanathan, M. Lott, W. Zirwas, M. Dohler, H. Aghvami, D. D. Falconer, and G. P. Fettweis. Relay-Based Deployment Concepts for Wireless and Mobile Broadband Radio. *IEEE Communications Magazine*, vol. 42(9):80–89, Sep. 2004.

[Ric99] Rice Monarch Project. Wireless and Mobility Extensions to ns-2, 1999. URL http://www.monarch.cs.cmu.edu/cmu-ns.html. [visited 8 Aug. 2005].

[RSA93] RSA Data Security. RSAREF, 1993.

[RW$^+$05] P. Russell, H. Welte, et al. netfilter/iptables, 2005. URL http://www.netfilter.org/. [visited 1 Aug. 2005].

[SLD$^+$05] K. Sanzgiri, D. LaFlamme, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer. Authenticated Routing for Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, vol. 23(3):598–610, Mar. 2005.

[SNCR03] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao. Cooperation in Wireless Ad Hoc Networks. In *Proceedings of 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pp. 808–817. San Francisco, CA, USA, Mar.–Apr. 2003.

[SPA05] SPANworks MultiPeer. Website, 2005. URL http://www.spanworks.com/. [visited 23 Aug. 2005].

[Sta04] T. Staub. *Implementating a Cooperation and Accounting Strategy for Multi-hop Cellular Networks*. Master's thesis, University of Bern, Nov. 2004.

[Sun05] Sun Microsystems Java Card Technology. Website, 2005. URL http://java.sun.com/products/javacard/. [visited 22 Jun. 2005].

[The05] The Slackware Linux Project. Slackware Linux, 2005. URL http://www.slackware.org/. [visited 2 Sep. 2005].

[TK84] H. Takagi and L. Kleinrock. Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals. *IEEE Transactions on Communications*, vol. 32(3):246–257, Mar. 1984.

[UBG03] A. Urpi, M. Bonuccelli, and S. Giordano. Modelling cooperation in mobile ad hoc networks: a formal description of selfishness. In *Proceedings of 1st Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*. Sophia-Antipolis, France, Mar. 2003.

[WB04a] A. Weyland and T. Braun. CASHnet - Cooperation and Accounting Strategy for Hybrid Networks. In *Proceedings of 2nd Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, pp. 423–424. Cambridge University Press, Cambridge, UK, Mar. 2004.

[WB04b] A. Weyland and T. Braun. Cooperation and Accounting Strategy for Multi-hop Cellular Networks. In *Proceedings of 13th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN)*, pp. 193–198. Mill Valley, CA, USA, April 2004.

[Wib02] B. Wiberg. *Porting AODV-UU Implementation to ns-2 and Enabling Trace-based Simulation*. Master's thesis, Uppsala University, Dec. 2002.

[WK04] T. Williams and C. Kelley. Gnuplot, Apr. 2004. URL http://www.gnuplot.info/. [visited 5 Aug. 2005].

[WM04] K. Wrona and P. Mähönen. Analytical Model of Cooperation in Ad Hoc Networks. *Telecommunication Systems*, vol. 27(2-4):347–369, Oct. 2004.

[WMP$^+$05] R. Wakikawa, J. T. Malinen, C. E. Perkins, A. Nilsson, and A. J. Tuominen. Global connectivity for IPv6 Mobile Ad Hoc Networks. Internet-Draft, Jul. 2005. URL http://www.ietf.org/internet-drafts/draft-wakikawa-manet-globalv6-04.txt. [visited 8 Aug. 2005].

[WQDT01] H. Wu, C. Qiao, S. De, and O. Tonguz. Integrated Cellular and Ad Hoc Relaying Systems: iCAR. *IEEE Journal on Selected Areas in Communications*, vol. 19(10):2105–2115, Oct. 2001.

[WSB04] A. Weyland, T. Staub, and T. Braun. Liveliness Evaluation of a Cooperation and Accounting Strategy in Hybrid Networks. In *Proceedings of 4th Workshop on Applications and Services in Wireless Networks (ASWN)*. Boston, MA, USA, Aug. 2004.

[WSB05] A. Weyland, T. Staub, and T. Braun. Comparison of Incentive-based Cooperation Strategies for Hybrid Networks. In *Proceedings of 3rd International Conference on Wired/Wireless Internet Communications (WWIC)*, pp. 169–180. Xanthi, Greece, May 2005.

[YLN03] J. Yoon, M. Liu, and B. Noble. Random Waypoint Considered Harmful. In *Proceedings of 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*. San Francisco, CA, USA, Apr. 2003.

[YML02] H. Yang, X. Meng, and S. Lu. Self-Organized Network-Layer Security in Mobile Ad Hoc Networks. In *ACM Workshop on Wireless Security (WiSe)*, pp. 11–20. Atlanta, GA, USA, Dec. 2002.

[ZCY03] S. Zhong, J. Chen, and Y. R. Yang. Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks. In *Proceedings of 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 3, pp. 1987–1997. San Francisco, CA, USA, Mar.–Apr. 2003.

[ZdV05] A. Zemlianov and G. de Veciana. Capacity of Ad Hoc Wireless Networks With Infrastructure Support. *IEEE Journal on Selected Areas in Communications*, vol. 23(3):657–667, Mar. 2005.

Curriculum Vitae

1976	Born on October 15, in Friedrichroda, Germany
1983 - 1987	Primary school in Eisenach, Germany
1987 - 1991	Polytechnic secondary school in Eisenach
1991 - 1995	Ernst-Abbe Gymnasium in Eisenach
1995 - 1998	Major in Computer Science and Minors in Mathematics and Electrical Engineering & Automation at Technical University Ilmenau, Germany
1998 - 1999	Practical training at Telscom AG in Bern, Switzerland
1999 - 2002	Continuation of studies at University of Bern
2002	M.Sc. in Computer Science, University of Bern
2002 - 2005	Research Assistant and Ph.D. Student at the Institute for Computer Science and Applied Mathematics, University of Bern