Department of Computer Science and Engineering

# Improving Dependability of Networks with Penalty and Revocation Mechanisms

Dmitriy Kuptsov

Aalto University

DOCTORAL
DISSERTATIONS

Aalto University publication series
**DOCTORAL DISSERTATIONS** 185/2014

# Improving Dependability of Networks with Penalty and Revocation Mechanisms

**Dmitriy Kuptsov**

Doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the lecture hall T2 of the school on December 5th, 2014, at 12 noon.

**Aalto University**
**School of Science**
**Department of Computer Science and Engineering**

**Supervising professor**
Professor Antti Ylä-Jääski

**Thesis advisor**
Professor Andrei Gurtov

**Preliminary examiners**
Professor Randy H. Katz, University of California, Berkeley, USA
Professor Xiaoming Fu, Georg-August University of Goettingen, Germany

**Opponent**
Professor Jussi Kangasharju, University of Helsinki, Finland

Aalto University publication series
**DOCTORAL DISSERTATIONS** 185/2014

ISBN 978-952-60-5961-7 (printed)
ISBN 978-952-60-5962-4 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-5962-4

Unigrafia Oy
Helsinki 2014

Finland

NORDIC ECOLABEL
441 697
Printed matter

A! Aalto University

# Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

**Author**
Dmitriy Kuptsov

**Name of the doctoral dissertation**
Improving Dependability of Networks with Penalty and Revocation Mechanisms

**Publisher** School of Science

**Unit** Department of Computer Science and Engineering

**Series** Aalto University publication series DOCTORAL DISSERTATIONS 185/2014

**Field of research** Telecommunications Software

**Manuscript submitted** 17 December 2013 **Date of the defence** 5 December 2014

**Permission to publish granted (date)** 31 January 2014 **Language** English

□ **Monograph** ☒ **Article dissertation (summary + original articles)**

## Abstract

Both malicious and non-malicious faults can dismantle computer networks. Thus, mitigating faults at various layers is essential in ensuring efficient and fair network resource utilization. In this thesis we take a step in this direction and study several ways to deal with faults by means of penalties and revocation mechanisms in networks that are lacking a centralized coordination point, either because of their scale or design.

Compromised nodes can pose a serious threat to infrastructure, end-hosts and services. Such malicious elements can undermine the availability and fairness of networked systems. To deal with such nodes, we design and analyze protocols enabling their removal from the network in a fast and a secure way. We design these protocols for two different environments. In the former setting, we assume that there are multiple, but independent trusted points in the network which coordinate other nodes in the network. In the latter, we assume that all nodes play equal roles in the network and thus need to cooperate to carry out common functionality. We analyze these solutions and discuss possible deployment scenarios.

Next we turn our attention to wireless edge networks. In this context, some nodes, without being malicious, can still behave in an unfair manner. To deal with the situation, we propose several self-penalty mechanisms. We implement the proposed protocols employing a commodity hardware and conduct experiments in real-world environments. The analysis of data collected in several measurement rounds revealed improvements in terms of higher fairness and throughput. We corroborate the results with simulations and an analytic model. And finally, we discuss how to measure fairness in dynamic settings, where nodes can have heterogeneous resource demands.

**Keywords** Internet architecture, protocol design, network measurements, wireless sensor and edge networks

**ISBN (printed)** 978-952-60-5961-7 **ISBN (pdf)** 978-952-60-5962-4

**ISSN-L** 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

**Location of publisher** Helsinki **Location of printing** Helsinki **Year** 2014

**Pages** 152 **urn** http://urn.fi/URN:ISBN:978-952-60-5962-4

# Preface

In the course of my studies I have met, received support from, and worked together with many brilliant people. All these people contributed to the completion of this dissertation, and I would like to thank them all.

I am grateful to my supervising professor, Antti Ylä-Jääski, and to my advisor, professor Andrei Gurtov. Andrei was the one who initially introduced me to the networking research group at Helsinki Institute for Information Technology (HIIT) and gave me a chance to pursue my goals. I would like to thank Andrei for providing me with the financial support for doing the research, helping me with valuable feedback and motivating me, especially during the last phase of my studies. I would also like to express my deepest gratitude to Antti. Thank you for your great assistance and patience.

I grateful to all my colleagues within HIIT. Part of the research presented in this work is the result of the truly wonderful collaboration with Boris Nechaev and Andrey Lukyanenko. Not only I truly enjoyed collaboration with these brilliant people, but I also learned a lot while working with them! I want to thank separately Dmitry Korzun for his advices and feedback. I also had the pleasure to work and study with Andrey Khurri, Ilya Nikolaevskiy, Tatiana Polishchuk, Sasu Tarkoma, Petri Savolainen, Miika Komu, Alexander Gogolev, Joakim Koskela, Ramya Sri Kalyanaraman, Oleg Ponomarev and Samu Varjonen.

During my study years I also had the opportunity of working in several research groups outside of HIIT. I am grateful to the people from the Distributed Systems Group at RWTH Aachen University. I would especially like to thank Oscar Garcia with whom I co-authored several papers. Thank you for your patience, support and help. It was also the pleasure to work with Tobias Heer, Rene Hummen and Stefan Götz.

I had the opportunity to visit the International Computer Science Institute (ICSI), Berkeley, California. There I had the honor of collaborating with professor Scott Shenker. I will always admire him. I also greatly enjoyed working with Barath Raghavan, Ali Ghodsi and Igor Ganichev. The research visits to ICSI were supported by FICNIA. And I am grateful to Martti Mäntylä, Jussi Kangasharju, Scott Shenker and Kristiina Karvonen, who made these visits possible.

I want to thank my pre-examiners professor Randy Katz and professor Xioaming Fu. I would also like to thank professor Jussi Kangasharju who agreed to serve as my opponent.

I would also like to express my deepest gratitude to my present colleagues for their understanding, patience and support.

Finally, I would like to thank my family. Father, mother thank you for all your guidance, support, love and care. Without you this work would never be possible, and so I dedicate this effort to you. I also want to thank Marina Panova. Your patience helped me to complete this work.

Helsinki, November 18, 2014,

Dmitriy Kuptsov

# Contents

# List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

**I** Dmitriy Kuptsov and Andrei Gurtov. SAVAH: Source Address Validation with Host Identity Protocol. In *Proceedings of the First International ICST Conference on Security and Privacy in Mobile Information and Communication Systems*, pages 190–201, July 2009.

**II** Teemu Koponen, Scott Shenker, Hari Balakrishnan, Nick Feamster, Igor Ganichev, Ali Ghodsi, P. Brighten Godfrey, Nick McKeown, Guru Parulkar, Barath Raghavan, Jennifer Rexford, Somaya Arianfar and Dmitriy Kuptsov. Architecting for Innovation. In *SIGCOMM Computer Communication Review*, vol. 41, pages 24–36, July 2011.

**III** Oscar Garcia, Dmitriy Kuptsov, Andrei Gurtov and Klaus Wehrle. Cooperative Security in Distributed Networks. In *Elsevier Computer Communications (COMCOM)*, vol. 36, pages 1284–1297, August 2013.

**IV** Dmitriy Kuptsov, Boris Nechaev, Andrey Lukyanenko and Andrei Gurtov. How penalty leads to improvement: A measurement study of wireless backoff in IEEE 802.11 networks. In *Elsevier Computer Networks (COMNET), DOI: 10.1016/j.comnet.2014.09.008*, pages 1–21, September 2014.

**V** Dmitriy Kuptsov, Boris Nechaev, Andrey Lukyanenko and Andrei Gurtov. A Novel Demand-Aware Fairness Metric for IEEE 802.11 Wireless

Networks. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC'13)*, pages 603–610, March 2013.

# Author's Contribution

### Publication I: "SAVAH: Source Address Validation with Host Identity Protocol"

Although the idea was not initially proposed by the author, he implemented and analyzed the architecture described in the publication.

### Publication II: "Architecting for Innovation"

The author was primarily involved in the design and analysis of the accountability protocol presented in this paper. In addition, he conducted a literature survey on topics related to accountability and DoS prevention mechanisms.

### Publication III: "Cooperative Security in Distributed Networks"

The contribution of the first two authors is considered comparable. The author of the present thesis proposed the design of the keying material data structure. He was also actively engaged in most aspects related to the analysis of the protocol described in the publication.

### Publication IV: "How penalty leads to improvement: A measurement study of wireless backoff in IEEE 802.11 networks"

The author proposed and implemented the protocols described in the paper. He was the primary person responsible for designing the test-bed and data collection apparatus, designing and conducting the experiments. He was engaged in most aspects of data analysis, study and validation of analytic models and simulation experiments. In addition the author contributed by suggesting and implementing the backoff adaptation protocols described in the paper.

### Publication V: "A Novel Demand-Aware Fairness Metric for IEEE 802.11 Wireless Networks"

The author initially proposed the idea. He actively participated in devising the fairness metric proposed in the paper. In addition, the author designed experiments, prepared and analyzed the data sets.

# 1. Introduction

Over the past few decades the Internet has unleashed an unprecedented wave of transformation. In a relatively short period of time, the Internet grew from a small experimental type of network interconnecting major universities in the United States to a sophisticated web comprising billions of devices ranging from tiny sensors to smart mobile devices and personal computers to huge data centers. Today the Internet is an integral part of our society, delivering vital services to people. Since the early 90's the Internet begun to evolve dynamically, with now virtually any household in developed countries having a connection to the Internet. The ubiquitous deployment of wireless networks, reduction in data rate costs and the explosive growth of smart phones and tablet computers brought the number of the Internet connected devices to $12.5$ billion in 2010 according to a Cisco study [42]. It is unlikely that this rapid growth will stop in the near future: the breakthrough in microelectronics allows companies to manufacture powerful, but small in size microcomputers capable of connecting to the Internet over wireless links. These developments eliminated previously existing boundaries, allowing for novel network applications to emerge. The most promising scenario is the ubiquitous deployment of wireless sensor networks ranging in sizes from small to large. It is roughly projected that the estimated number of connected devices will grow tremendously: the Cisco report [42] predicts that there will up to $50$ billion of such devices by 2020.

Such a continuing spread of the Internet's popularity and its penetration into our daily lives also places high demands on its dependability: Even short outages in network connectivity can lead to serious economic losses. For example, failures occurring due to bugs in software or defects in hardware are commonplace, but they can also appear because of misconfiguration of applications or even more fundamental flaws in system design. Nonetheless, all of this might have unpleasant, if not tragic, consequences. For example, the lack of adequate resource allocation mechanisms may affect the execution of basic networking protocols, such as packet forwarding, in a predictable manner because the principle of fairness in such situations can be undermined: In this setting some users, without malicious intent, can gain more network resources than some other users, making utilization of the network unfair and inefficient. Bugs in software and hardware problems can also lead to improper execution of network protocols since packets can be dropped, altered or delayed in an arbitrary way. Finally, misconfiguration can cause serious outages in network connectivity. One example being misconfiguration in border gateway protocol (BGP), which occurred in the past and lead to network blackouts.

Attacks, on the other hand, are more systematic and deliberate actions. Malicious activity can appear, for instance, after nodes become compromised by an intruder. And among many harmful effects, compromised nodes can exhaust network resources, intentionally making network services unavailable for some users. For example, one particularly noxious problem of today's Internet involves intentional attacks on the network infrastructure. These attacks can range from rather primitive [26] to highly sophisticated ones, such as successful attacks on root DNS servers [148] or attacks utilizing sophisticated network of reflectors [123]. Protecting the network from malicious nodes in a timely and efficient manner plays an immense role in ensuring stable end-to-end connectivity and the proper functioning of various network protocols.

Similar problems may arise in future autonomous networks, such as wireless sensor networks. In these networks each node has to behave in a fair way, *i.e.*, as expected, to ensure the correct operation of the distributed system since misbehaving nodes can disrupt basic functionality, such as routing, time synchronization protocols, or even cause inconsistencies in the collected data. There is no guarantee, however, that we can prevent nodes from misbehaving due to either the nodes being under the control of an attacker or simply because of buggy software or hardware faults. To ensure predictable performance of such systems it is, therefore, can be desirable to evict non-cooperative nodes from the network in a fast and reliable way.

At the other extreme, the quality of end-to-end communication also depends much on last mile connectivity. Today, as the Internet edge becomes increasingly wireless, a lot of users rely on the availability of these networks. In fact according to the Wi-Fi Alliance [142], already today about $200$ million households use Wi-Fi networks, and another $750,000$ Wi-Fi hot-spots installed in public places. As these networks become more crowded, the shared resource – wireless channel time – becomes scarce and if not distributed in a fair way, the performance can be degraded significantly for some of the users. Clearly, ensuring fairness and availability of these networks is as important, if not more so, as ensuring fairness in other parts of the network.

Thus, in this dissertation we investigate several mechanisms for dealing with malicious and non-malicious (but equally harmful) faults. For example, to cope with malicious nodes we study mechanisms which enable the network to isolate the faults by accounting the nodes and evicting those that do not behave according to desired rules. In the latter part of the thesis, the focus is shifted to problems in the wireless edge networks where some users can disrupt proper network functioning by being unfair with respect to other participants. Here, we study node penalty mechanisms that enable the network to limit access to the shared resource for unduly successful users, ensuring that all participants receive *a pro rata* share of the resources and will not suffer from resource starvation. At the same time, we are interested in reduction of the number of packet collisions in the air in order to improve the overall efficiency of the network.

## 1.1 Research Questions and Scope

In present thesis, the main research question is related to the study of several fault mitigation techniques in networks. We conducted the study for the settings in which faults can be malicious and non-malicious. Thus, to deal with malicious faults, we investigate accountability and node revocation frameworks. In contrast, to mask failures in wireless networks, which are not due to malicious behavior, we investigate several penalty-based mechanisms, which allow the system to regulate the access to shared resources fairly for all network participants. All of the above, in one way or the other, is related to *dependability of networked computer systems* - a field of study, which, among many other things, deals with the ways of improving availability and reliability of computer systems by isolating and removing faults [8].

Due to the big scope of the above research question it is impossible to devise a single solution that would be suitable for all scenarios where such problems arise. On one hand, we limit the scope of our study to three different scenarios: resource exhaustion attacks in the Internet, node misbehavior in unattended autonomous networks, and resource sharing problems in wireless edge networks. The solutions we consider are different in designs, but the overall purpose is similar - improve reliability and availability of the target systems. On the other hand, some of the ideas we develop can be potentially applied to other settings. For example, the node revocation protocol that we investigate in the context of the wireless sensor networks can be potentially applied to other types of networks [84] which share common operational principles.

The *first research question* we study in this thesis is: **How to account nodes on the internet-wide scale and what are the requirements for such architecture?** Here we consider a network in which some nodes are assumed to never become compromised and, therefore, always operate as designed. These network elements, spread around the network, take the role of trusted entities and belong to different administrative domains although always cooperating. We investigate what is the required set of changes to the infrastructure is needed for the solution to be efficient and deployable. We study this in the context of the Internet in which compromised nodes can send undesirable traffic, endangering the availability of network resources.

The *second research question* we investigate is: **What are the building blocks of node revocation protocols for the networks which lack a centralized trusted entity?** Here, the type of the network we consider is comprised of nodes that are not operated by humans, and thus it represents an autonomous class of network. Designing protocols for the networks which lack a centralized entity coordinating their functions is a challenging task: In such networks all nodes need to take equal roles and to cooperate in order to carry out the functionality of trusted entities. Here we perform the study in the context of wireless sensor networks. In these networks relatively low-power devices, being unattended and possibly deployed in hostile environments, can become faulty unnoticed or even compromised by an attacker. Such nodes, if not isolated from the network in a timely manner, can inflict tangible damage on the whole network.

Our *final research question* relates to fairness problems in wireless edge networks. The question we investigate here is: **Can short-term penalties improve the fairness and availability of wireless networks and how to fine-tune such mechanism in dynamic environments?**

Here we are concerned with the situations when non-malicious users can cause damage unintentionally by congesting network and thus using resources in an unfair manner. As a solution to this problem we propose the node penalty mechanisms. We implemented these solutions in real hardware and tested our hypothesis in various real-world settings. To confirm our observations we further performed several rounds of simulations and devised an analytic model. In addition, while analyzing the data sets, we noticed that it can be non-trivial to represent fairness quantitatively for the settings where nodes have different resource demands. To untangle this ambiguity we proposed and evaluated a methodological tool.

## 1.2 Methodology

In this thesis we chose measurements as one of our main methodological approach to validate our designs. For example, in our study of IEEE 802.11 wireless networks we have mostly preferred real experiments over simulations because this approach has enabled us to observe the system's behavior in environments similar to those in which such networks typically operate, *e.g.*, office buildings and residential areas. As part of the measurement study, we also use basic principles of exploratory data analysis – a methodological tool which allows us to reveal trends in the data using simple statistics and plotting. Of course, conducting controlled experiments in real-life settings can be challenging or sometimes even impossible. In these cases, approaches based on simulations can become preferential. Simulations can also be used to verify the correctness of the results obtained in real-world experiments. In our work, we use simulations mostly for the latter purpose. In addition, we apply more formal methods in our study. For example, we use such an approach to study security protocols. Here, the goal is to reason about all possible flaws of the design (under given assumptions and constraints), trying to ensure that an attacker cannot exploit these flaws. Finally, we apply some techniques of mathematical modeling to validate our hypotheses. For example, we use elements of mathematical analysis in attempt to derive optimized parameters for some of our designs.

## 1.3 Contributions

This thesis is a summary of five publications. Here we briefly outline the contributions of each publication. We provide more elaborate summaries in Chapter 3.

Publication I describes our initial view on the architecture enabling node accountability and source address spoofing prevention in the Internet. Publication II provides a more detailed view on the future Internet architecture. It describes a framework which can transform the Internet architecture into a flexible ecosystem allowing fostering of innovation. Among many other aspects, the paper discusses a protocol enabling Internet-wide node accountability and revocation while preserving privacy of the end-users.

Publication III presents our study of the node revocation protocol for the networks lacking centralized, trusted third party. This work describes the analysis of cooperative security protocol.

Publication IV presents an experimental effort with modified backoff mechanisms applied to IEEE 802.11 wireless edge networks. The work in Publication V evolved from the observations made in Publication IV. Here we discuss a possible way of measuring fairness in the settings when nodes have heterogeneous demands for network resources.

## 1.4 Thesis structure

The thesis is structured as follows. In Chapter 2 we present the background relevant to our work and give an overview of the related work. In Chapter 3 we summarize the results obtained in our publications. Finally, Chapter 4 concludes the thesis.

# 2. Background and Related Work

We start with a short overview of the background material. Thus, in Section 2.1 we give a short overview of cryptography and cryptographic protocols. This material is important for understanding concepts presented in Section 2.2 and Section 2.3, where we delve into a review of the literature related to accountability and node revocation protocols.

In the second part (Section 2.4), we first briefly discuss the principles of operation of 802.11 wireless networks which are relevant to our own research. Then we provide the review of the literature, covering a wide spectrum of research related to these networks. In particular, we discuss fairness and performance issues specific to these networks and the ways these problems are tackled.

## 2.1 Cryptography and security protocols

Cryptography forms the basis for securing many computer systems. In essence, cryptography is a practice of techniques for secure communication over insecure channels. In the following paragraphs, we review the basic principles of cryptography, cryptographic algorithms and protocols.

### 2.1.1 Symmetric key cryptography

Parties that are involved in using *symmetric key cryptography* in order to communicate need to share the same key to effectively perform encryption and decryption operations on messages. There are two types of symmetric cryptography algorithms: *stream* and *block* ciphers [141]. Stream ciphers, as the name implies, operate on a stream of bits and perform transformations for each bit individually, whereas block ciphers perform transformations on larger blocks of bits at a time.

There exists an extensive number of block ciphers. However, nowadays only few provide an acceptable level of security. Among these Triple Data Encryption Standard (3DES) [113], Advance Encryption Standard (AES) [115], and Twofish [135] are the most widely used algorithms.

Symmetric key algorithms have their advantages and disadvantages. The main advantage of these ciphers is their computational efficiency. This is mainly because symmetric cryptosystems do not involve complex operations, *i.e.*, big number exponentiation, division and multiplication. Due to these reasons, the majority of end-to-end security protocols, such as IPsec [74], Transport Layer Security (TLS) [36], and Secure Shell (SSH) [156] use symmetric cryptography for securing data plane traffic. Unfortunately, the application area of these cryptosystems is limited by the complexity of a key management process, that among many other operations involves the distribution and revocation of secret keys in a secure way.

### 2.1.2 Public key cryptography

In contrast to symmetric key cryptography, in asymmetric or *public key* cryptography encryption and decryption keys are different. The keys typically exist in pairs [141, 130]: with one part – the *public key* – being open to anyone and used to encrypt the messages; the second part – the *private key* – is always kept secret and is used to decrypt the messages. The fundamental property of any well-established public-key cryptosystem is that the private key cannot be easily obtained from the public key. These properties significantly simplify key management processes making this class of cryptosystems attractive in many application areas. There are many public key cryptography algorithms exist today, however, RSA [130] is the oldest and most widely used algorithm. Elliptic curve cyptosystems (ECC) [105, 78], on the other hand, are more recent and efficient [55].

### 2.1.3 Cryptographic hash functions

*Cryptographic hash functions* are another important building block in modern cryptographic protocols. On the high level, as described in [141], cryptographic hash functions produce a fingerprint – a string of a fixed length (also called a *image*, or *hash value*), from an arbitrary long string, also called a *pre-image*. Any secure hash function must contemplate the following three fundamental properties. *Pre-image resistance*: for any secure hash function it should be computationally hard to find a pre-image

**Table 2.1.** Life cycles of popular cryptographic hashes[2]

| | 1992- | 1994- | 1996- | 1998- | 2000- | 2002- | 2004- | 2008- | 2012- |
|---|---|---|---|---|---|---|---|---|---|
| MD5 | 🟦 | ▨ | ▨ | ▨ | ▨ | ▨ | 🟥 | 🟥 | 🟥 |
| MD2 | 🟦 | 🟦▨ | ▨ | ▨ | ▨ | ▨ | 🟥 | 🟥 | 🟥 |
| SHA-0 | 🟦 | 🟦 | 🟦 | ▨ | ▨ | ▨ | 🟥 | 🟥 | 🟥 |
| SHA-1 | | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | ▨ | ▨ | ▨ |
| RIPEMD-160 | | | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 |
| SHA-2 | | | | | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 |
| SHA-3 | | | | | | | | | 🟦 |
| | 🟦 Unbroken | | ▨ Weakened | | 🟥 Broken | | | | |

that will produce a hash value identical to a given one. *Second pre-image* resistance: given a pre-image it should be computationally hard to find another pre-image such that when both are passed as inputs to the same secure hash function, this function will produce identical hash values. *Collision resistance*: for any secure hash function it should be computationally hard to find two distinguishable pre-image values such that both will map to an identical hash value.

The number of cryptographic hash functions is rife. Nevertheless, only few provide desirable level of security and performance. For instance, widely used in the past, MD5 [129] algorithm is now known to be insecure [153]. More secure versions are therefore suggested for use, such as SHA-2 [114], the newly developed but not standardized SHA-3 [117] or the even less popular RIPEMD-160[1] algorithm for which no successful attacks are known. In Table 2.1, we list several well known hash functions and their corresponding security statuses.

Many keyed versions of different flavors also exist. Examples are HMAC [82], CMAC [116] and PMAC [19]. Keyed versions of hash algorithms can be used to produce message authentication codes (MAC), which resemble a form of digital signatures of messages.

### 2.1.4 Digital signatures and key exchange protocols

According to [141] *digital signature* is a fingerprint that allows an interested party to uniquely identify and distinguish the signer of a message. Thus, once a message is signed, the signer cannot deny its involvement in originating the message [130]. Both public-key and symmetric-key cryptography can be used to produce digital signatures.

The *Merkle signature scheme* is an example of a signature algorithm

[1] http://homes.esat.kuleuven.be/~bosselae/ripemd160.html

[2] Adapted from: http://valerieaurora.org/monkey.html

that relies on symmetric cryptography. This algorithm is based on one-time signatures (such as the one due to the Lamport [89]) and Merkle hash tree [104]. The Merkle hash tree itself is an interesting concept. On a high-level, it is a binary hash tree in which each of $n$ leaf, $L_i$, is calculated as the hash of some value $a_i$, and each internal node $m_{ij}$ is calculated as the hash of the concatenation of its two sibling nodes. Thus, given a root of such tree along with the $log(n)$ elements (on the path from a specific leaf up to the root), it is easy to verify whether a message $a_i$ is authentic or not.

Nevertheless, conventional public-key cryptography allows one to construct more flexible digital signature schemes. Among the many algorithms available, RSA [130], Digital Signature Algorithm (DSA) [119] and its improved elliptic curve cryptography-based variant, ECDSA [119], are commonly used in modern security systems. Many threshold-based variants of these algorithms are also available (for sampling see [138]). These algorithms find their roots in applications where it is important to ensure that the signature was constructed not by a sole holder of a private key, but instead collectively by a group, in which each individual holds just a part of a private key.

*Key agreement* protocols are another integral part of security protocols. These algorithms are indispensable tools as they allow parties to exchange a common (symmetric) secret without requiring a secure side-channel. The examples of such protocols are Diffie-Hellman (DH) [37] and the more efficient Elliptic Curve Diffie-Hellman algorithms [27].

## 2.2 Fault isolation in the Internet

We now move on to the discussion of security threats on the Internet. In this context, we devote much of the attention to the problem of denial of service (DoS) attacks. We then present state-of-the-art solutions that allow network to account and isolate nodes on an Internet-wide scale. Here, we discuss the advantages and limitations of different approaches.

Today there are indications that the Internet in the face of its ever increasing popularity was not sufficiently prepared to repulse certain security threats: The original design of the Internet concealed colossal potential weaknesses that malicious parties are able to exploit nowadays, and so undermine some of the fundamental principles of the Internet. Once the existence of these threats was understood, protecting the Internet became a difficult task because the network already had a complex structure with many nodes attached to it.

Many solutions for securing end-to-end communication such as HIP [108, 109, 57, 80, 111] on the Internet layer, SSL [46] and SSH [156] on the application layer, have been proposed. Undoubtedly, without these initiatives the modern developments of Internet services would not be possible. Despite all the advantages, however, these solutions alone are hardly capable of defending the end-hosts against attacks that were perhaps unimaginable back in early 60's and became commonplace nowadays – DoS and Distributed DoS (DDoS) attacks – attacks aiming to make target systems unavailable for a prolonged period of time.

Overall, it does not require deep technical knowledge to launch admittedly rather primitive DoS attack on the Internet. For example, by analogy to a Smurf attack [26, 143] an intruder can undertake an attack by sending broadcast packets, whereby source IP address is forged and belongs to a victim. In this way, if the packet is reflected by large enough number of hosts [123], the victim would become unavailable on the network as it will be unduly overwhelmed with packet processing routines.

In theory, as pointed out in [81], solving source address spoofing problem, and hence preventing or otherwise limiting the impact of some class of DoS attacks, does not require complex mechanisms either. For example, network operators can verify that the source addresses in packets are valid and reachable from ports at which they were received [45, 11]. This protocol is mainly developed by Cisco, and is known as Unicast Reverse Path Forwarding (uRPF) [44]. The challenge, however, here is in constructing and keeping the filters up to date: note, that maintaining these filters manually is not practical or even feasible in large scale deployments. And although there are works, such as [90], describing protocols for automating the filter construction, as indicated in [81], these proposals still require expensive modifications to the functionality of network elements on the path. Lack of deployment incentives, however, may not come from technical challenges alone, as we shall see next because although source address validation can limit the number of attacks on the Internet, the approach falls short in preventing more sophisticated DDoS attacks.

Probably started as fun projects, it was soon realized by the rogue community that DDoS attacks can generate revenue. Since then, more sophisticated tools appeared. Indeed, the appearance of *botnets* shattered the stability of the Internet significantly. In essence, botnets represent networks of hundreds of thousands of computers that came under control of an attacker. These compromised computers, however, would typically belong to benign users who may not even have suspected that their machines were ruled by the attacker. Thus, most of the time these computers would generate legitimate traffic and only upon command from the attacker would flood a victim with requests degenerating its ability to deliver (perhaps vital) services. Targets of these attacks are not only subjects to the exhaustion of bandwidth on access links. The attacks can also target other network bottlenecks as well as exhaustion of computing and memory resources on both servers and clients or, even on middle boxes.

And although end-host security solutions such as anti-virus applications are widely available, in many cases these solutions can provide only *post facto* cures to the problem, meaning that some exploits are patched only after they were initially discovered and exploited by the attackers. And the time taken from detection of these security breaches until they are finally patched can be sufficient for the attackers to launch several successful attacks. Yet there are millions of networked devices that do not have such a basic security solution installed and thus can be easily compromised by attackers.

It is reasonable to assume that the fundamental shortcoming of the Internet is in the lack of mechanisms which would allow end-points to effectively defend themselves from the receiving of unwanted packets: Today, once under attack, victims have very few tools at their disposal to shut down, or even trace back the origin of the attack. The solution, therefore, might be more controllable and accountable network elements, which in a case of misbehavior can be identified and eventually shut down. Solving this problem in practice, however, is hard. Not only is there a multitude of technical challenges, but the entire ecosystem needs to be more flexible to make it possible for the solution to find their path to large scale deployments.

### 2.2.1 Capability mechanisms

One of the possible ways which can allow users to be in control of the incoming traffic is to employ the so called capability-based mechanisms. As suggested in [3], in these approaches senders obtain relatively short-lived authorization tokens from the receivers. Senders then use these tokens to stamp the packets, whereas the routers discard those packets without valid tokens, and destinations do not renew the tokens if they suspect the sender.

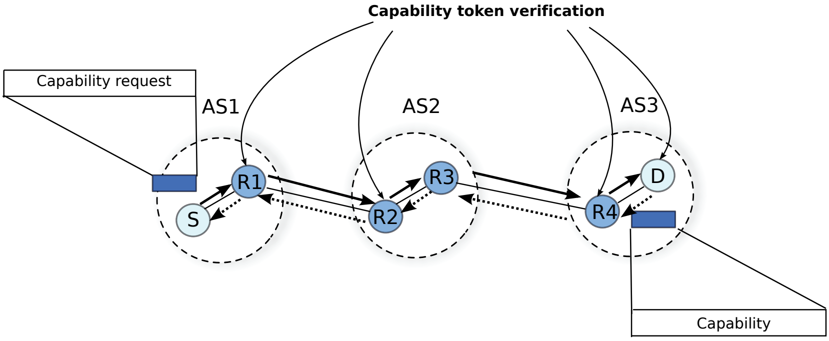

**Figure 2.1.** A capability-based DDoS limiting architecture. Adapted from [155]

The idea of capability-based approaches emerged first from overlay filtering architectures such as SOS [75] and Mayday [3]. In these systems, border gateways authenticate outgoing traffic and assign to it some secret. Verified traffic is then passed to a protected destination through an overlay. These ideas formed the basis for many capability-based Internet architectures. One example is the work in [5], which was further revisited by Yaar *et al.* in [154]. A more complete work describing the capability architecture by Yang *et al.* emerged in [155]. In Figure 2.1 we give a generalized view of such architectures.

There are also other studies that exhibit commonalities with capability-based architectures. For instance, Host Identity Indirection Infrastructure (*hi3*) [112, 59] shares common design principles and can be easily turned into capability-based infrastructure with DDoS-limiting features. A very similar work can be found described by Liu *et al.* in [95]. Other works such as in [24] devise a capability-based solution for flow-level granularity. The authors proposed tunneling the traffic between servers and clients through special *cookie-boxes* which can drop flows without the required capabilities. A similar design is demonstrated in [51]. The authors suggest that flows between clients and servers be moved inside protected tunnels. This ensures that the control over these flows can be acquired at any point, for example, during attacks the sending rate can be decreased artificially.

Certainly, capability-based solutions provide a fine grained and secure way for access to the infrastructure: These approaches can potentially allow ISPs to reduce the scale of resource exhaustion attacks in the Internet but at the cost of verifying cryptographic tokens along the path.

### 2.2.2 Filtering mechanisms

In contrast to capability architectures, in filtering-based mechanisms victims directly request installing filters for suspicious senders. These solutions can be *pro-active* in which the users install filters well before the attacks take place (this is similar to how we punch the holes in our home firewalls) and *reactive* in which users request installing filters once attacks are detected.

One example of reactive approaches is the design in [94]. In their work, the authors suggest *StopIt* architecture – a closed-control, open-service traffic filtering architecture (Figure 2.2). In this architecture, any receiver can use StopIt servers to filter undesired traffic from botnets comprising million of hosts. StopIt is built to protect the network from two main DDoS attacks: destination and link flooding attacks. The system uses a cascade of servers from destination to source. In case of attack, a victim sends filter requests to its nearest StopIt server. It is then the responsibility of these servers to propagate the request as close to the attackers as possible. The accountability mechanism described in Publication II shares some similarities with this approach. Thus, for example, the way we handle shut-up messages through trusted points in the Internet is similar in spirit to the filter requests through cascade of StopIt servers.

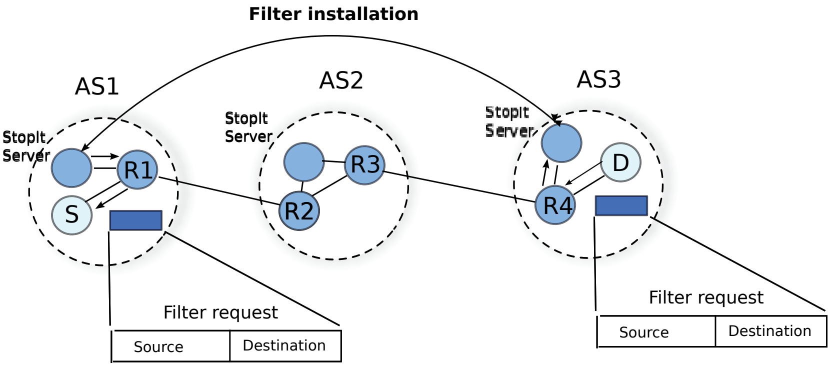

**Figure 2.2.** A StopIt traffic filtering architecture. Adapted from [94]

Though, such filtering-based solutions have several limitations. For instance, AITF [6], being the most complete work on filters, verifies the legitimacy of a filter using a three way handshake: if the link is flooded during attack, the filter setup procedure can fail.

Some other frameworks, such as Pushback [102], instead of filtering specific sources, mitigate attacks by limiting the rate of traffic from aggregated prefixes. While such an approach can be effective, it can also introduce performance impairments for other, legitimate users.

Several pro-active filtering approaches also exist in the literature. For example, in [13] the authors suggest that the hosts explicitly signal routing infrastructure with information about what traffic they are willing to receive and from what hosts, similarly as one would do using its home router, but the filtering is enforced closer to the source. This fact also makes such approaches similar to some capability-based mechanisms.

### 2.2.3 Accountability architectures

Capability and filtering-based approaches can be indispensable against DDoS attacks. However, as mentioned in [4], today many security issues in the Internet are *due to lack of accountability*. Thus, deploying capability and filtering solutions which we discussed previously, can become less useful if there will be no reliable way to identify and locate the attackers. Of course there are solutions which provide for strong accountability of hosts in networks, such as port-based access control [65]. But these mechanisms are designed for rather small scale networks, and they are not fundamental part of the core Internet. Thus, their Internet-wide adoption would require additional engineering effort.

One readily available way of implementing accountability is to rely on IP addresses. For example, an end-host can authenticate itself to the edge router by means that are acceptable within a given domain (for example, using MAC address-based identification). Here the edge router, if it vouches for the end-host, needs to ensure that the address of the end-host is valid within its sub-network. In a similar manner inter-domain accountability is achieved based on validity of observed IP addresses or domain specific identifiers. Source address validation architecture in [152] standardizes some of these ideas (the work in Publication I can be used as part of this framework, for example, to perform intra-domain accountability functions). However, for the approach to work well its ubiquitous deployment is desirable.

Other approaches found in the literature (such as [139]) discuss the possibility of storing packet fingerprints in the routers to allow hosts to trace-back the origin of the attacks, and thus make attackers accountable for their actions. Although, such architectures can be effective in detecting the source of the attacks, they also impose significant burden at the routers.

Most of the cryptographic approaches, however, rely on the non-repudia-

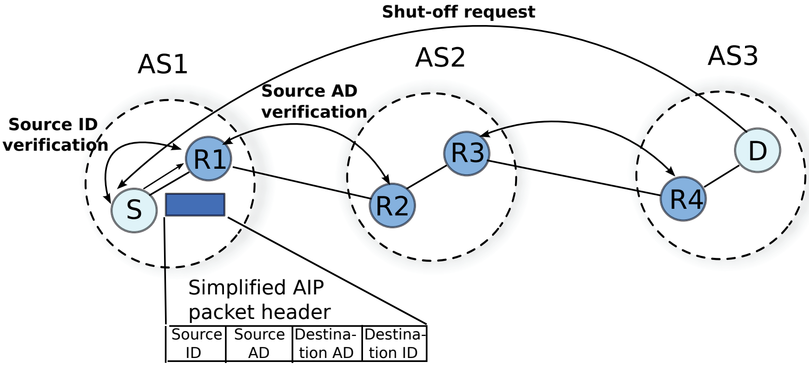

**Figure 2.3.** Interactions in AIP protocol. Adapted from [4]

tion property of signatures attached to packets. Here, the ultimate goal is to allow destinations to verify the signatures, hence the identity of a sender. In comparison to non-cryptographic solutions, on one hand, cryptography-based approaches offer finer-grained accountability. On the other hand, these solutions are also more heavyweight since nodes must be able to generate and verify some sort of cryptographic signatures at line speeds for every packet sent or received correspondingly. Other challenges which are common to all cryptography-based solutions include issues related to privacy and large-scale key management.

Accountable Internet Protocol (AIP) [4] is one example of such architecture. Here, the authors' emphasis is on a fully distributed solution which does not depend on any globally trusted authority. The idea of AIP revolves around *self-certifying* identifiers, which are essentially self-generated public keys used both for routing and accounting purposes. Here, if a router receives a packet from unknown sender it drops the packet and performs address verification procedure by asking the sender to prove that it is a genuine holder of the address. A simplified view of this architecture is shown in Figure 2.3. It is worth noting that the bootstrapping phase in Publication I and Publication II are close in essence to the address verification in AIP.

Similarly to AIP, in packet passports [93], transit and destination domains can securely verify the origin of the packets (this architecture is schematically shown in Figure 2.4). To achieve this, the packet passports architecture uses efficient, symmetric-key cryptography to place tokens on the packets which can be verified by autonomous systems (AS) along the path. Unlike AIP, here border routers first learn the public keys of ASes in the network, which are distributed along with border gateway

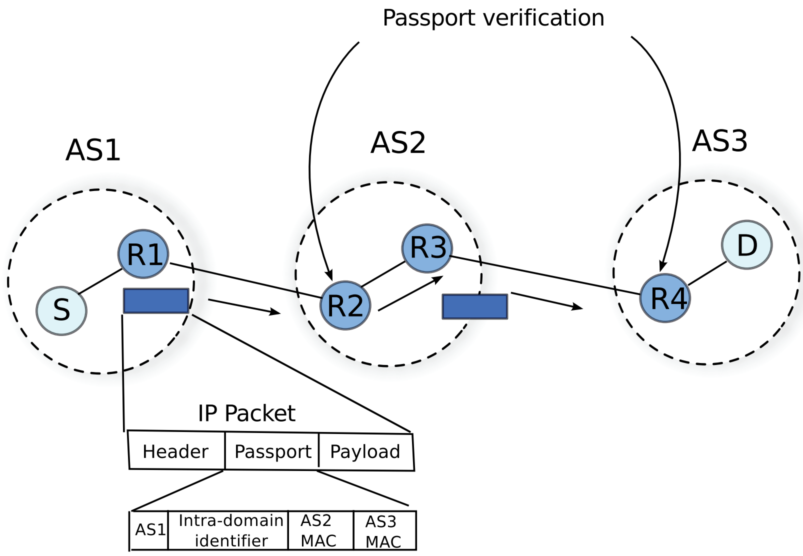

**Figure 2.4.** Packet passports architecture. Adapted from [93]

protocol (BGP) route advertisements. Upon receiving an outbound packet, the border router, if vouching for the sender, stamps the packet with the secure tokens, one for each AS along the path. The tokens here are MACs which can be verified only by the corresponding AS. On one hand, the downside of this approach is that the border router needs to know the entire path the packet will take from source to destination. This might degrade end-to-end performance if there is a discrepancy between the paths assumed by the border router and the actual path used to forward the packet. On the other hand, this allows victim to grasp by looking at any packet exactly from where the attack has originated, and which transit AS forwarded the packets. A very similar solution was proposed in [20].

Unlike the packet passports approach, in PLA or Packet Level Authentication [87] architecture the cryptographic tokens attached to the packets are generated by the sender. The sender uses asymmetric signature algorithms to construct such tokens, while the intermediaries and the destination can then use these tokens to verify the authenticity of the packets. To make this verification process feasible, a sender also sends its certificate along with the packet. PLA also can deal with node revocation. For example, to filter packets from undesirable sources, PLA architecture employs a mechanism that is very similar to the shut-off messages in AIP. In addition, in PLA, routers can blacklist certificates belonging to undesirable sources. These interactions are schematically demonstrated in Figure 2.5.

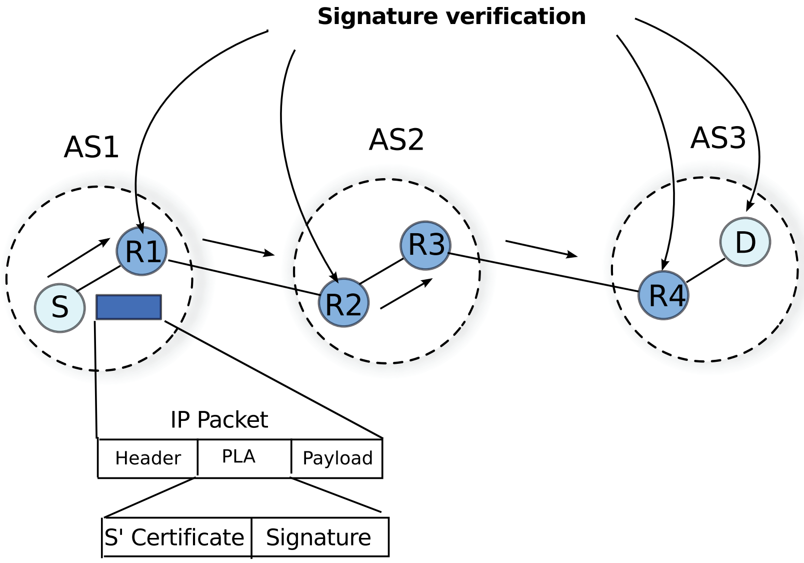

**Figure 2.5.** PLA architecture. Adapted from [87]

### 2.2.4 Other mechanisms

Filtering and accountability are indispensable tools in dealing with DDoS attacks. But there are also other solutions that can complement these approaches. For example, one way to deal with DDoS attacks is to reward good users and to penalize attackers. Thus, in [150] the authors focus on the ways to defend systems against application-level DDoS attacks by incentivising users to increase their resource utilization. The key idea of this approach is not to limit the clients as in capability proposals, but instead to encourage hosts to speak up and consume more resources. Assuming that attackers use their entire available bandwidth, they will not be able to benefit from such encouragement. Benign clients, on the other hand, typically using only a small fraction of the resources to send the requests, will react to encouragement and increase the volume of traffic sent or received. In this setting, good users can naturally penalize the attackers by capturing a much larger portion of resources.

As discussed in [150], there are also proposals in which users are charged in a currency to prevent massive DoS attacks on servers. For instance, designs in which users pay to access the resources all fall into this category. Here, the payments can be based on the computing resources of the clients. Such approaches are typically based on computational puzzles [2, 7, 10, 40, 69]. However, the methods in this category can also use real money in order to restrict the access the infrastructure. Example is the work found in [103]. Other proposals suggest to place clients in a queue and advanced them in this queue based on their contributions, *e.g.*, amount of spent computational resources. For example, in [99] the authors propose *most-knocked first-served (MKFS)* queuing mechanism which preferentially admits users who pay enough with their CPU cycles.

Another way to deal with DDoS attacks is to over-provision the systems. For example, service and network providers can employ additional capacities in order to defend against DDoS attacks [126]. Finally, there are also a multitude of studies dealing with detection of DDoS attacks. For example, some DDoS attacks can be detected by profiling user demands [33] and avoided by blocking the outliers – users with abnormal resource demands. Authors in [150] mention that approaches that preferentially admit only humans are also widely spread. For example, certain DDoS attacks can be detected and prevented if the machines were requested to provide information that can be replayed only by humans. In this setting, bots that are typically programs running in stealth mode on compromised machines would fail to access a resources that required such interaction. A widely known examples of such defense solutions are CAPTCHAs [49, 70, 107]. In fact, CAPTCHAs are probably the simplest of all to deploy in real-life. However, we should note that these approaches do not fully off load the burden away from critical infrastructure. Finally, approaches that filter out packets that contain invalid [68] or suspicious [123] fields can be also used to mitigate attacks or minimize their negative effects on the infrastructure.

## 2.3 Fault removal in wireless sensor networks

There are other distributed systems which can be exposed to similar threats, which we discussed in previous section. Examples are distributed smart environments comprising objects communicating over wireless links. In such networks, each node has to behave in a fair way, *i.e.*, as expected, to ensure the correct operation of various protocols. Here faulty nodes can disrupt basic functionality, such as routing, time synchronization protocols, or even cause inconsistencies in collected data.

### 2.3.1 Cooperative security approaches

One way to deal with the problem is to allow nodes to cooperate and revoke faulty nodes. Thus, cooperative security can be understood as a mechanism in which honest nodes in the network cooperate, ensuring that all nodes behave in a fair way. For example, if the honest nodes detect some unacceptable actions by some other node, they can react and rapidly isolate such a node from the entire network. This concept emerged first in the area of node revocation in wireless sensor networks [29, 28, 47].

One of the early works on distributed node revocation in sensor networks was proposed in [28]. The basic idea suggests that every node in the network be configured with some revocation information against the rest of the devices in the network before deployment. After the deployment, this information is used to revoke misbehaving nodes. Preloading the revocation information during deployment inevitably leads to a need for the rekeying of all the nodes in the network whenever a new node is added. In other words, the scheme is more suitable for static networks.

Over time, more advanced versions of the protocol detailed in [28] appeared. Thus, the limitations of [28] were first addressed in [47]. Specifically, in this work, the authors introduced the *Cooperative Security Protocol (CSP)* concept which uses two voting procedures – one for admission and one for revocation. On one hand, it was this design choice that made it possible to mitigate the problem of high memory requirements. On the other hand, the protocol remains suboptimal in terms of the number of colluding attackers the system can sustain due to the type of keying material data structure used in the protocol.

Several centralized approaches also exist in the literature [38, 92, 41]. In these solutions, a centralized node monitors all nodes in the network either directly or through reports relayed by other nodes in the network. These approaches can overwhelm the network, undermining the overall performance of the system. Moreover, such setups may not be even possible if the network is deployed in a random fashion and lacks a centralized entity responsible for coordination.

### 2.3.2 Miscellaneous

There are also similar studies in the area of node revocation protocols for mobile ad-hoc networks (MANET). We outline several works we found in the literature. The first work we mention [34, 106] advocates the *suicide* node revocation scheme. The idea of the protocol is simple. Whenever a node finds some node to become faulty, it issues a revocation message for both faulty node and itself. Such signed revocation message is then broadcast network wide for the revocation to take effect. The shortcoming of the scheme is the false revocation decisions, which can lead to a fast network depletion.

Another relevant study considers a threshold based public-key cryptography (PKC) [100] for realizing a node revocation protocol. In the protocol, any node in order to join the network should request a set of its neighbors to cooperatively construct a certificate. If the certificate is granted, the node becomes a fully functional part of the network. It is the public-key cryptography that makes the protocol scalable. The protocol exhibits some limitations though. For example, the protocol can sustain a relatively small fraction of faulty nodes due to limited number of nodes that can generate the certificates.

There are also applications of mechanisms similar in spirit to the *cooperative security approach* to secure the Internet routing infrastructure. In this setting, the goals are different from those in node revocation algorithms, but these mechanisms still share similar design principles. For example, in [157] the authors suggest using verifiable voting among neighboring Internet domains to ensure the consistency of the disclosed routing information. There are also proposals that suggest using a variant of cooperative security protocol to secure peer-to-peer networks [84].

Another relevant area is that of studies concerning *group membership protocols*. These protocols belong to a family of distributed protocols in which the processes can in the presence of faults agree on which processes should remain in the group [128, 132]. Unlike cooperative security protocols, these solutions lack the notion of the formation of small groups in the network that perform node admission, monitoring and revocation tasks.

There are also other solutions which are similar in spirit to the protocols we have discussed so far. Thus, *state-machine replication* is an approach used to implement fault-tolerant systems by replicating resources and coordinating requests in a distributed way. Cooperative security is close to the notion of Byzantine state-machine replication in which a set of processors acts in unison masking Byzantine faults. For instance, this is similar to the behavior of nodes in cooperative security in a situation where monitoring nodes ask each other whether a node joining the network has distributed enough revocation information. These ideas appear in the literature starting with Lamport's paper [88] and followed by the contribution of Scheider on fail-stop processors [133]. A comprehensive overview of these concepts is provided in [134]. But again, these concepts lack the formation of processor groups that can monitor and revoke some other processor whenever latter starts to misbehave.

Furthermore, cooperative security is related to the work on *fault-containment* in the context of self-stabilizing algorithms. Here, a group of processors attempts to contain the effects of faults by handling these effects locally so that other processors outside of the group are not affected. One of the first work which investigates these concepts was presented in [48], and a more general work can be found in [63].

The cooperative security protocols also show some links to *failure detectors*. A failure detector aims at isolating the failed processes prior to agreement, instead of directly dealing with them within the agreement algorithm [30, 60]. In principle, intruder detection schemes which are part of many node revocation schemes can use these approaches to detect those nodes that become non-functional.

Finally, the lower-bounds of agreement protocols which are building blocks of many distributed systems in which nodes need to agree on a common action are related to the results on the *Byzantine agreement* and its crypto-variants. Lamport, Shostak, and Pease deserve the credit for their term *Byzantine faults* [124] and their $3t + 1$ lower bound proof. There is a large body of work that suggests several variants to the original algorithm, one example being work by Cachin *et al.* in [22].

## 2.4 Resource sharing problems in wireless edge networks

We now turn our attention to the problems in 802.11 wireless networks. A thorough description of the operational principles of these networks can be found in [120]. And in what follows we describe *distributed coordination function (DCF)* since its principles have utmost importance to our own research.

The IEEE 802.11 DCF protocol was designed to reduce contention in the wireless networks. For this purpose, in the standard implementation a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) algorithm is used to schedule the access to the shared resource. In this protocol, a node, before transmitting a frame, first checks if the channel is idle or busy. If the channel is not idle, the station chooses a uniformly random backoff interval from the currently used contention window and waits for the selected time before attempting to access the channel again. If, however, the channel is idle, the station attempts to transmit the frame and waits for the acknowledgment packet to arrive. A missing acknowledgment packet is an indication of a failed delivery. At this point the station attempts to recover from the failure by retransmitting the packet again. The retransmission continues at most six times after which the packet is discarded.

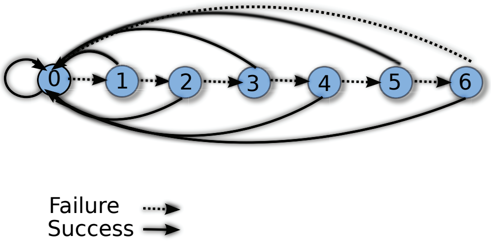

**Figure 2.6.** IEEE 802.11 backoff protocol

After every failed transmission, a stations exponentially increases its contention window to increase its odds at the next retransmission. Once the packet is successfully delivered or discarded the sender resets its retransmission counter and starts with the smallest contention window for transmission of a next packet. The Figure 2.6 shows this process for the IEEE 802.11 wireless networks. And although the protocol copes fairly well with its functions when the number of users is small (typically no more than two, and not considering a setting involving hidden stations), its dynamics can potentially lead to unfair resource usage for a larger number of users.

Overall, there are several well-known problems with communication in wireless LANs (WLANs) that affect the stability of these networks in terms of throughput, delay and fairness. In particular, since nodes in such networks use a shared medium in an unlicensed radio spectrum to transmit the frames, collisions are possible. A measurement study of large scale enterprise WLANs [31] showed that in their test-bed nearly $15\%$ of sender-receiver pairs experienced significant loss due to collisions. Whereas the measurement study in [131] indicated that in their network the retransmissions can account for as much as $28\%$ of all data transmissions and $46\%$ of data transmission time. Another problem is fairness. The study in [39] showed that 802.11 networks have good short-term fairness when the number of contending stations is small, *e.g.* just two, and becomes worse for an increasing number of stations. Furthermore, the study in [79] shows that collisions, and hence frequent backoff phases, can negatively impact the performance of TCP flows: The authors observed that in WLANs, the TCP can receive bursts of acknowledgment (ACK) packets and in response send bursts of data packets. Such an anomaly, as mentioned in [79], impacts the performance of the network in several ways, including an increased number of packet losses and significant network under-utilization. These results support the observations made in [56, 58], that TCP in general performs poorly over wireless links experiencing packet losses.

### 2.4.1 Performance modeling

The studies on modeling the performance of the standard IEEE 802.11 protocol find their roots in Ethernet technology as both share common design principles. For example, both technologies use exponential backoff to avoid collisions on the medium. There is an extensive body of work in which the performance of the protocol is modeled analytically. For example the studies in [16, 83, 32, 97, 43, 85] can be a good starting point. In these papers the authors proposed several assumptions that can be made about wireless networks and derive analytic frameworks for modeling such important characteristics of IEEE 802.11 networks as throughput and delay. Overall, most of the studies agree that the performance of this protocol, although not optimal, can be improved by tuning certain parameters. For example, one way to improve the throughput would be to properly choose initial backoff windows depending on the number of contending stations [16]; another way would be to vary the backoff factors accordingly [83]. In [39] the authors take a step forward and adduce several key factors that impact the stability and the performance of wireless networks in one way or the other. Thus, using their model, the authors prognosticate that backoff protocol alone reduces the performance of these wireless networks by as much as $15\%$. The authors further mention that at high transmission rates, packet losses increase significantly. Finally, the authors also indicate that the short-term fairness in wireless networks becomes worse for an increasing number of stations due to exponential backoff.

Overall, the distinctive lineaments of the majority of theoretical studies are the underlying set of assumptions and the tools used to validate the theoretical models. For example, many works rely on controlled experiments in simulation frameworks rather than gathering empirical evidence in real-world deployments. Typically such an approach is preferred over real-world experiments to exclude the different artifacts that can appear during the course of experiments. But, on the other hand, although modern simulation frameworks are powerful tools on their own, they still cannot represent real-world environments in their full depth. Thus, real-world measurement studies can give a possibility to look at the performance of these networks from different angle.

### 2.4.2 Protocol optimization

There exists a considerable number of works that attempt to improve stability and fairness in wireless networks by using non-standard backoff schemes. Thus, the work described in [140, 64] in one way or the other suggests using non-standard contention windows. The key idea is to either remove exponential backoff completely and use fixed contention windows or non-standard backoff factors to reduce packet collisions. More radical approaches exist such as in [25, 98] where the authors proposed using non-standard state transitions. These works are similar in spirit to our own solutions presented in this thesis.

Other studies proposed using frequency domain backoff [145] to improve the performance of the wireless networks. The basic idea is to control the maximum number of sub-channels that one node can access based on the observed collision level. Thus if collisions are too frequent, a station will back off (using either binary exponential backoff (BEB) or additive increase/multiplicative decrease (AIMD) strategy) and reduce the number of used sub-channels.

Certainly, collisions are not the only source of packet losses in wireless networks. Indeed, packet transmissions can fail either because of time-varying wireless channels (such as frequent changes of signal quality) or contention (a race for channel access by two or more hosts, which in practice can lead to simultaneous packet transmissions). In these settings, backoff protocols help to avoid simultaneous transmissions, but they are not designed to combat packet losses due to degraded channel conditions. Instead, to aid the receiver to successfully decode corrupted packets, some redundant information is typically transmitted along with the original data. However, the amount of transmitted redundant information depends on environmental conditions and is usually controlled by physical layer solutions. To choose the right bit rate, the nodes can estimate channel quality with probing [18, 101] or using feedback from the receivers [149]. Although these approaches can be accurate they can, as indicated in [53], also incur significant overhead and can, therefore, negatively affect system performance. To reduce this overhead, in [53] the authors advocate sending the packets without any rate as a stream of symbols representing a linear combination. Once the needed amount of symbols is received, the packet can be decoded and its reception is acknowledged to the sender.

The idea of using various network coding techniques recently received a lot of attention from the wireless network community. The work by Katti *et al.* [73] was one of the earliest to explore this direction in an innovative way. In their work, proposed a way to reduce the number of transmitted packets (hence contention in the medium) by using a simple xor operation on packets destined for different hosts. Later, the principles of network coding became the basis for several other innovative designs. For example, the studies in [50, 72, 71] all use smart network coding techniques to improve packet delivery in wireless networks.

Another research direction that received much attention in recent years is related to the possibility of utilizing the radio spectrum more efficiently. Examples are designs that use multiple input-output antennas [91, 127] to improve robustness and capacity of the wireless networks. These mechanisms, although orthogonal to the backoff protocol proposals, illustrate one way of coping with simultaneous transmissions in the wireless networks.

Dynamically tuning the parameters of wireless networks is a separate concern. The ability to accurately estimate these parameters based on current load and the number of attached users can be a cornerstone in determining the performance of such networks. One important aspect here is how to make the adaptations, based on the number of active stations. For example, Bianchi *et al.* [17] in their seminal work suggested using busy slots to estimate the number of active stations. Cali *et al.* [23] investigate this direction further, and derive a metric which estimates the number of active stations based on the observed number of idle slots. An empirical evaluation of the ideas similar in spirit to those in [23] was presented in [52]. And although the approach has its merits, the accuracy of such estimation in the presence of hidden terminals remains questionable.

Other relevant studies consider using a centralized controller [151] in enterprise wireless networks. The controller periodically collects the information about used channel time, available bandwidth, *etc.*. Once this information is processed at some central server, it can be used to fine-tune wireless access points. Such approaches are suitable for closed-controlled deployments since it is feasible to gather a global knowledge of the entire network state and make accurate adaptations accordingly.

### 2.4.3 Measurement studies

There is a large body of work which employs a measurement approach to illuminate the performance of large scale wireless networks. For example, the studies in [12, 67] are an attempt to characterize wireless users in a single but relatively large scale network (with an average of 12 active stations attached to an access point) in a conference setting. In [146] the authors attempt to illuminate such characteristics as the types of devices used and type of traffic being transferred. The authors make several interesting observations. For example, they discovered that the amount of traffic in their settings in a download direction was prevailing over the traffic in an upload direction, but the opposite tended to be true during peak throughput periods.

Other wireless measurement papers focus on even more diverse scenarios. For example, Rodrig *et al.* [131] measure overhead, retransmissions and the dynamics of bit rate adaptation algorithms in wireless hotspot networks. In [31] the authors conduct the research of IEEE 802.11 wireless networks using data collected from 150 radio monitors. This work is interesting in the context of this thesis since it provides insights on a technique for merging logs collected from different nodes. Henderson *et al.* [62] investigate an even larger network comprising over 550 access points and 7000 users involved. And perhaps the work by LaCurts *et al.* [86] constitutes the largest study of real-world 802.11 networks. Thus, their data set contained information from over 1400 access points from all over the world. In the study, the authors take a step forward and try to observe the existence of invariant properties in wireless networks, *i.e.*, properties that do not change from network to network.

Although the aforementioned studies make important contributions toward understanding and improving the behavior of wireless networks, a limited number of papers discuss the empirical investigation of modified backoff protocols in real-world deployments using cheap commodity hardware. One such research is discussed in [52]. Here the authors report some practical implementation and evaluation of the modified backoff protocol using proprietary firmware and a small number of wireless stations. Another piece of research [147] considers the implementation of MAC protocols in general on commodity hardware. Thus, a better understanding of the real-life performance of modified MAC protocols on commodity hardware still needs experimental evidence. In this thesis we attempt to make a step in this direction.

### 2.4.4 Fairness and metrics

Fairness is an important performance characteristic of computer networks: Several studies indicate that fairness has a direct impact on the stability of wireless networks. And in the next few paragraphs we discuss some of these works. For example, the study in [14] indicates that fairness is extremely important in wireless networks for attaining low latency and high availability objectives. Unfairness can also provoke an avalanche of impairments at the upper layers. For example, the performance of TCP connections might be severely degraded because of delayed data and ACK segments. In [39] the authors mention that short term unfairness almost certainly always leads to a longer term performance pathology in wireless networks, impacting attainability of wireless hosts. In [96] the authors investigate mixed upload download scenario in 802.11 wireless networks. They observed significant unfairness: the stations performing upload obtained considerably higher throughput than stations downloading.

In general, fairness deals with the distribution of network resources among participants in a fair way, whereas max-min fairness [15, 35] is a typical approach (in a single resource setting) to ensure such allocation. To measure quantitatively the effectiveness of resource allocation, however, several useful tools exist. The study of short term fairness in IEEE 802.11 networks dates back to the early paper by Koksal *et al.* [79]. In their work, the authors proposed using Jain's index [66] with a sliding window to characterize short term fairness in wireless networks. Such a method was widely used by the community to study fairness in wireless networks [39, 52, 64, 9]. To measure fairness quantitatively in [14] the authors derive a novel metric based on the number of packet intertransmissions. In their work, the proposed metric is compared with the sliding window used with Jain's index under homogeneous conditions: no host is disadvantaged by its signal quality, traffic pattern, or spatial position.

On the other hand, some studies that focus on investigating the fairness of transport layer protocols, such as TCP, over IEEE 802.11 networks consider longer term fairness issues. Here a meaningful single value statistic is used to describe fairness. For example, in [125, 110] the authors measure the fairness of TCP over wireless networks using average throughput. And in [144] the fairness in wireless networks is assessed using both average throughput and channel occupancy time. In [110] the authors indicated that measuring fairness in real environments can be challenged by, for example, hosts being sending packets at different rates. Thus, prior to conducting controlled measurements a separate calibration step should be performed.

## 2.5 Summary

In this chapter we introduced the essential background of this thesis. We began with the introduction to *cryptography and cryptographic protocols*. We then discussed security threats in today's Internet. As the next step, we reviewed the approaches designed to mitigate DoS and DDoS attacks on an Internet-wide scale. Here we introduced ways of dealing with the problem by employing *filtering*, *capabilities* and *node accountability* mechanisms, and whenever applicable we compared these solutions to our own approach.

Next we moved on to the second area of interest in this thesis – *node revocation in wireless sensor networks*. Here we covered approaches which allow nodes to admit and revoke nodes in the network in a secure way. In addition we compared these solutions to other approaches from related areas. Thus, we showed their relationship to node revocation in mobile adhoc networks, group membership protocols, fault-containment solutions in distributed systems and several others.

We then moved to our third area of interest – *resource sharing problems* in IEEE 802.11 wireless networks. Here, we first introduced the general principles of the a backoff mechanism designed to manage and reduce contention in these networks. Then we introduced the studies on the performance modeling of these networks and covered literature related to the various optimizations of these networks. We concluded the chapter with the review of the measurements studies and a discussion of fairness issues in IEEE 802.11 networks.

# 3. Summary of Results

This chapter discusses the published results of this thesis. First, we answer the questions related to node accountability and revocation in the Internet and describe our architectural effort in this area. Second we present our design and analysis effort related to the node revocation protocol for autonomous networks. Finally, we present the design of our penalty mechanism for wireless edge networks and discuss our experimental experience. We conclude the chapter with open questions and suggestions for future work.

## 3.1 Accountability, fault isolation and revocation

In these sections we describe our work related to the revocation of malicious nodes from the network. We study this problem for two different settings. In the first case, we consider that the underlying network is of the Internet scale in which some nodes are assumed to be always trusted. In the latter case, the assumed network comprises wireless sensor nodes, none of which are assumed to be always trusted.

### 3.1.1 Accountability and revocation at the Internet scale

In publication Publication I and Publication II, the goal was to design a node accountability mechanism for a network comprising multiple politically and financially independent domains with thousands of end-hosts attached to each such domain. The design of such systems is challenging for the following reasons.

*Scalability*. This requirement is stipulated by the nature of the Internet, which dictates that there cannot be a single authority solely coordinating its functions. Therefore, the designs must ensure the needed level of scalability by requiring that there need not be any single trusted third party

(TTP) – an entity which carries out such functions as identity and key management – but all TTPs in the Internet must be globally reachable and identifiable in a secure way. Each host in the network should be provided with a default TTP by its domain although the hosts may choose to use another TTP, which must be approved by its current domain.

*Computational efficiency*. The majority of accountability designs need to deal with the verification of some sort of information generated using cryptogrphic algorithms. To this end, verification of such accountability information must be a lightweight operation to ensure efficient packet processing. Imposing a significant burden on every forwarding element on the data path would increase the cost of the architecture, and can complicate its adoption. It is desirable that the complexity of packet signing are imposed on senders and to some extent on the border routers behind which the senders are located. However, the accountability information verification routines are not to be done on the data path and instead are offloaded to some external entities such as, for example, TTPs. This will reduce the complexity of the forwarding infrastructure.

*Uniformity across domains*. If a domain implements the accountability interface, then every packet crossing its border should contain a sufficient state, which will allow destinations and intermediary forwarders to reliably identify who to contact to report an attack incident; as for the TTP, this state should contain enough information about the identity of an ultimate source to allow for the shut-up messages to reach the source of the packets. Moreover, the shut-up messages must be a fundamental part of the system rather than a domain specific security mechanism: although, domains can adopt their own intra-domain DoS defense mechanisms, there needs to exist a unifying mechanism which is accepted globally because DoS attacks typically cross domain boundaries.

*Support for privacy*. It is also important to preserve the privacy of the senders. Identity of a source should not be easily distinguishable from packet headers by all but only by responsible TTPs. Revealing only the TTP identifier and not the individual source in the packet makes it hard for the third parties to track the sources. Of course, TTPs should know the identities of their users, but there should be a guarantee that that this information will be kept private. At the same time, TTPs should be able forward shut-up message between each other. This is to ensure that if a victim sends a shut-up request to its TTP, it will eventually reach the TTP of the ultimate packet sender.

*Flexibility.* Perhaps one of the biggest problem with the current Internet architecture is its inability to accommodate radical changes ( there are significant costs associated with even small scale changes to the infrastructure). The feasibility of the discussed frameworks depends much on how easily these designs can be integrated into infrastructure. Thus, the Internet ecosystem must be open enough to accommodate these changes. Moreover, the proposed designs must be themselves flexible. For example, it should be feasible for the TTPs, end-hosts and domains to upgrade in order to support newer versions of accountability protocol. These changes should not require global agreement, thus enabling coexistence of different versions and making gradual transitions possible.

Our initial design of accountability architecture partially fulfilling the requirements discussed in the previous section was presented in Publication I. In this work, we did not use a public key cryptography to construct accountability fields for each packet. Instead, we make use of the public key cryptography only during a bootstrap process – the phase when end-hosts attach to the network and register with the edge or border router. In latter phases, end-hosts use learn symmetric keys to create an accountability fields for the packets. The edge or border routers verify this field during packet forwarding. This solution enables edge or border routers to keep the binding between the cryptographic identities of an end-host and its more ephemeral identity revealed to the Internet, which is useful for privacy.

It is, however, desirable to preserve cryptographically generated state in the packets even after they cross the domain boundaries. Preserving such state all the way to the destination is useful in several ways. For example, it makes possible to account for end-hosts across different communication sessions or when they roam from domain to domain. Thus, in Publication II, the challenge was in designing an accountability field that would provide the TTPs with information sufficient to shut-up hosts during attacks, while not revealing this information to other parties in order to preserve the privacy of an end-user.

Accordingly, when a node joins the network it first registers with a border router and its TTP. At this point, via a key exchange procedure source, border router, and TTP establish shared symmetric keys. All three parties also need to learn and verify the longer term identities of each other to prevent various attacks: Such an approach prevents the source from constructing bogus accountability fields containing false information about the sender. On the other hand, in this way the source can also ensure that it reveals its information to the correct party. At this point, the TTP also learns the address of the source and border router such that the location of both can be looked up in future.

After completing the bootstrap phase, the source can start its normal communication. Here, when source sends a packet via its border router, it attaches a valid cryptographic tag to the packet parameterized with the secret key it shared with TTP and border router. To this end, the grand purpose of such a tag is to prove that the source indeed vouches for its packet. When border router forwards the packet, its task is to verify the tag and generate an encrypted source address using the key it shares with the TTP. Such an encrypted source address ensures that border router verified the identity of the end-host, on the other hand it also hides the sender's identity from all but the responsible TTP. Together tag, encrypted source address, TTP's identifier, and some other random information (known to both, the border router and TTP), when attached by border router to the packet will allow any intermediary and destination to request a shut-up from the TTP.

In the context of the architecture presented in Publication II, the network deals with unwanted traffic by allowing the victims to tell an attacking machine to stop sending packets to it via a shut-up message (SUM). The concept of SUM was introduced in several studies [4, 94, 54, 137], whereas in Publication II it is augmented with the support for privacy. To enforce the SUM messages, however, a secure control-point somewhere in the network close to the source can be used. Here, middle boxes installed at the ISP's premises or hardware NIC installed on the end-hosts can be used to prevent the sources from spoofing, as well as from sending unwanted packets after receiving a valid SUM messages.

The prototypes of both architectures were built to demonstrate their overall feasibility. Thus, in Publication I we experiment with an end-user connected to an edge router which assumed to be controlled by an ISP. The edge router functionality was implemented in a low power router running the Linux distribution OpenWrt [122]. We choose such a setting to favor deployments in which accountability functionality is enforced as close to the sender as possible (in this manner our approach is similar to the architecture in [61], which uses wireless edge routers to authenticate packets of attached end-users). We use implementation of Host Identity Protocol for Linux (HIPL) [1] to negotiate the pair of keys between the end-host and edge router, suitable for signing and verifying the data plane traffic using the HMAC algorithm. A proof of a concept implementation was also presented in Publication II. The goal of the implementation was not to demonstrate the performance of separate components, but rather to show the overall feasibility of the approach.

### 3.1.2 Cooperative node revocation

We now move to the next area in which faulty nodes can undermine fairness and availability, and thus their timely isolation can in this way play an immense role. In Publication III, we present a protocol designed to deal with this issue in wireless sensor networks (although the applicability of this protocol can be broader). On a high level, the protocol allows nodes to cooperate and revoke faulty nodes in the network (Figure 3.1). There are several key difference from the settings we discussed so far: First, in the previous section we assumed that all TTPs are actively engaged in the protocol execution, whereas in this case we assume that the presence of the TTP is not guaranteed. Second, we assume that the nodes comprising this type of network have limited computational, communication and energy capabilities.

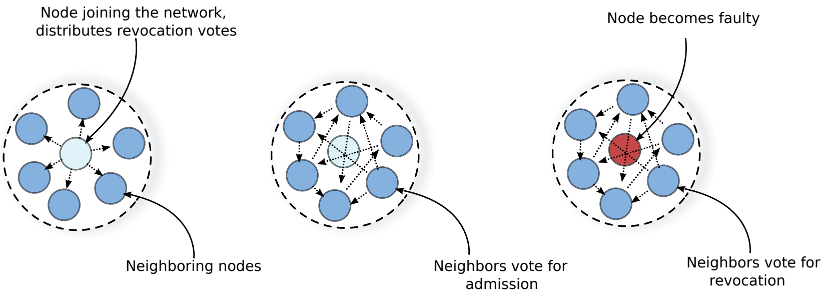
Node joining the network, distributes revocation votes
Node becomes faulty
Neighboring nodes
Neighbors vote for admission
Neighbors vote for revocation

**Figure 3.1.** Cooperative node revocation architecture. Adapted from [47]

The protocol consist of three phases: admission, normal operation and revocation. The operation of the protocol starts with the *admission* phase. At this point, a node can start to communicate with other nodes in the network if a set of its neighbors agrees on its admission. After successful admission, the node starts its normal operation, *i.e.*, executes the functions for which it was designed. We denote such a phase as the *communication session* of the node. Later, if the node is found to be faulty by its neighbors (with the help of an intruder detection system (IDS)) an *isolation* phase is initiated. To this end, if a positive agreement is reached, the node is revoked network-wide. Otherwise the node drops its current communication session and reattempts to join the network. According to the protocol, all nodes play a dual role – they act as nodes joining the network as well as participate in admission, monitoring and revocation of other nodes.

A mandatory condition for a node to join the network is the distribution of partial revocation votes (PRVs) – cryptographically verifiable secret tokens – to its neighbors. Thus, prior to starting a new communication session, a node distributes fresh PRVs to its neighbors via unicast messages over secured channels (for example, depending on capabilities of sensor nodes, one can employ a suitable variant of protocol described in [77] to establish pair-wise keys). After this step, the neighbors need to decide whether a sufficient number of such PRVs was disclosed. If these nodes can find a positive agreement, they will admit the node into the network and form its *Dynamic Trusted Security Domain (DTSD)*. Each neighbor, if it receives a PRV and participates in admission voting, also agrees to participate in the monitoring of the node and, if needed, to carry out the revocation procedure in the future. Finally, during the revocation voting the nodes in the DTSD exchange the PRVs to reconstruct the revocation vote (RV).

One of the fundamental building blocks of the protocol is the underlying keying material data structure. We consider its design as one of the core contributions in Publication III since the properties of the protocol depend much on the choice of its structure. Thus, the emphasis was on the following aspects.

*Reduced number of nodes engaged in the protocol*: In our work, the PRVs and RV represent the points on the polynomial of the degree $t$. The PRVs are the values computed, using this polynomial at points other than zero. The RV is a special value and is computed using the same polynomial at point zero. This design choice allows nodes to reconstruct the RV from $t+1$ PRVs using the approach described in [136]. Here $t$ is a configuration parameter, and its choice depends on the size of the network and desired level of resiliency – $t$ is also the upper bound for the possible number of colluding attackers in the DTSD. For comparison, in [47] the ratio of minimum DTSD size and number of faulty nodes is significantly higher.

The possibility of reusing the keying material in the admission and revocation phases was another crucial goal. This reduces the amount of needed keying material as well as the communication and computation complexities of the protocol. To achieve this we allow the nodes to use the double hash values of the PRVs as votes during the admission voting, whereas only the hash of PRVs and the actual PRVs are used during the revocation voting. In this way, the votes disclosed in different phases can be easily linked together. For example, a PRV or its hash value can be compared to the double hash value of the PRV disclosed during the admission phase using a single hash function evaluation.

*Space efficiency*: We were also interested in a data structure that is succinct enough and can be distributed among the nodes in the network in a such way that each node holds only its small portion without compromising other properties. Thus, the space requirements per node for the proposed data stricture are logarithmic with respect to the number of nodes in the network. For comparison, the data structure in [28] has storage requirements which are linear with respect to the number of nodes in the network.

*Computational efficiency vs. scalability*: The design of keying material in Publication III relies on symmetric cryptography. This type of cryptography is very suitable for the scenarios involving nodes with limited computational capabilities [76]. Thus, in Publication III, the PRVs and RVs for different nodes and their different communication sessions are authenticated using Merkle trees. Nevertheless, public key cryptography can be used in the protocol to favor the settings in which better scalability is desired. For instance, instead of having fixed sized Merkle trees, the PRVs and RVs can be secured with asymmetric signature algorithms.

Another important building block of the protocol discussed in Publication III are the voting algorithms. We considered two different voting strategies which rely on the keying material presented in previous paragraphs. The first algorithm we considered is an agreement based on a reliable broadcast of the double hash values of PRVs. The second algorithm is based on a simple disclosure of the hash value of PRVs or plain PRVs.

To reach a consensus, nodes can employ an agreement scheme using the double hash of PRVs for admission, whereas the mechanism based on the disclosure of the hash of PRVs or plain PRVs can be only used during revocation. Here, the choice of a voting algorithm during revocation depends much on the underlying IDS. If it is *biased* (can produce erroneous decisions) two rounds of revocation voting are needed. During the first round the nodes seek an consensus by exchanging the hash values of the PRVs. The second round starts if a positive decision on the revocation is found (*i.e.*, a sufficient number of such PRVs hashes are exchanged).

At this point, the nodes can safely disclose the actual values of the PRVs and reconstruct the final RV value. Here the two rounds are necessary to prevent false node revocations from the network. If, however, the IDS is *perfect*, nodes can omit the first round, and directly disclose the PRVs.

We have analyzed the proposed protocol for two different settings. In the first case, we assumed that the underlying IDS is perfect. In the second case that the IDS was biased. We first showed that the protocol is correct, *i.e.* fulfills the properties of the cooperative security protocol under the presence of $c$ colluding attackers. Here we also devised the bounds for the minimum DTSD size, the total number of needed PRVs and the maximum number of colluding attackers. Thus, the system operates correctly when the number of colluding attackers does not exceed $t$ and the number of PRVs is at least $3t + 1$ and the minimum size of the DTSD is $2t + 1$ nodes. This ensures that during the revocation, even if $t$ nodes are faulty, the RV can still be reconstructed by disclosing $t + 1$ PRVs. On the other hand, when the IDS is biased, these parameters depend on the probability of false positive revocation decision (made by node's IDS) and can be selected accordingly. The next bit of analysis was related to the ability of nodes to propagate the revocation messages through the DTSD when there are $t$ colluding attackers present it. Thus, we analyzed this property for a randomly formed network and for the network in which each node has a direct communication channel with any other node in the network. Next, we analyzed the protocol execution time. We demonstrated this for the setting where all operations, such as message delivery, IDS fault detection, were bounded. The last bit of analysis that we performed was related to the comparison of voting algorithms. We compared the (message and communication) complexity of voting algorithms which rely on the proposed keying material with an approach that does not use this keying material, neither during admission nor during revocation. We concluded that the usage of the proposed keying material can potentially simplify the studied revocation protocol.

## 3.2 Mitigating faults in wireless edge networks with penalties

In Publication IV we dealt with different type of faults from those discussed so far. Here we consider a wireless edge network in which nodes use a shared medium in an unlicensed radio spectrum for communication. We assume that nodes in this setting are non-malicious, but nonetheless can use resources unfairly. Here we investigate the possibility of enforcing fair resource sharing by applying penalties to nodes.

### 3.2.1 Penalty backoff protocols

Unlike our previous approaches, here we are not dealing with deliberate misbehavior but rather with failures that are more transient in nature and caused by faults during a design phase of the protocol. Thus, we do not suggest revoking unfair nodes from a network for a long period of time. Instead, to avoid failures we propose to give nodes short-term penalties. In this spirit, we proposed to incorporate a self-penalty mechanism in backoff function of the IEEE 802.11 networks. The underlying principle of the proposed algorithms is simple: to penalize overly successful nodes by attempting to increase their silent periods and accordingly reward unsuccessful nodes with smaller waiting times. We hypothesized, using also previous knowledge found in [98] as the bases, that such an approach could allow stations to utilize network resources more efficiently and fairly. To test this conjecture, we implemented the proposed protocols in real hardware, conducted multiple rounds of real-life experiments and analyzed the collected data. In attempt to ensure the correctness of the obtained results we also repeated the experiments in the simulation framework.

Thus, in the context of this work, we experimented with two novel backoff protocols and compared them with existing solutions. We briefly describe each protocol in the paragraphs that followed. The first protocol which we studied was the standard IEEE 802.11 backoff protocol. This protocol is used in almost all 802.11 wireless network deployments. We used this protocol as a benchmarking baseline and compared it with other algorithms. To meet our needs, though, we introduced one modification to the protocol: in addition to experiments with a standard backoff factor of $2.0$, we also conducted the experiments with a broader range of values ($[1.2, 2.6]$) for this parameter.

The first non-standard algorithm, which we investigated in the present thesis was *penalty backoff*. According to this algorithm, after a successful transmission that does not require retransmissions, a station chooses the largest available contention window for the consecutive transmission. This is the self-penalty phase. In contrast, if the station fails to transmit a packet without retransmission, its behavior is similar to that in the standard backoff: the station starts to exponentially increase its contention window and reattempt the transmission; and for the transmission of a consecutive packet the station starts again with the smallest contention window. In this way, we attempt to increase the odds of unsuccessful stations to transmit packets fast enough.

*Rollback backoff* is another modified version of the protocol that we investigated. In contrast to the penalty backoff algorithm, stations here always start with a state that corresponds to the largest contention window (but optimized with respect to the current number of active stations). If the station fails to transmit a packet, it exponentially decreases its contention window and attempts to retransmit it. In this way unsuccessful stations are rewarded. In essence this protocol can be viewed as the reversed version of the standard backoff protocol whose principles were covered in Section 2.4.

Finally, we also implemented and experimented with a *backoff protocol with fixed contention windows* [64]. This protocol is different from all the protocols described above in that the contention window changes only with the number of stations in the network. In other words, for all transmissions (including retransmissions), stations use the same contention window size as long as the number of wireless stations does not change. The contention window must be selected properly with respect to the current number of active stations in the network.

We implemented the aforementioned protocols in open source firmware [121] for Broadcom B43 wireless cards. This firmware features the implementation of standard 802.11g Medium Access Control (MAC) mechanisms for Broadcom/Airforce chipsets. For our experimental test-bed, we used 12 wireless cards, four commodity computers, an Ethernet switch, and a wireless access point running a Linux distribution OpenWRT. We dedicated a single computer to play the role of a *master node*. This machine was responsible for sending commands to slave nodes to trigger experiments and also participated in receiving and sending test traffic from and to the slave nodes. This machine was also responsible for the synchronization of the log collection process.

The other three machines were used as *slave nodes*. These nodes were provisioned with a single wired connection and multiple (up to 5) wireless cards. We also configured these machines with policy based routing to send all control traffic such as commands and calibration packets through wired interfaces. Experimental traffic was, however, carried over wireless interfaces. Such a setup allowed us to separate control and experimental

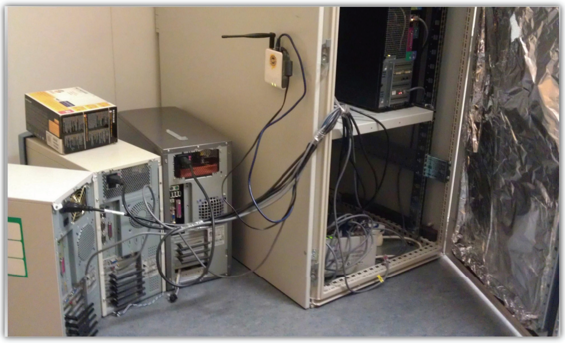

**Figure 3.2.** An example of test-bed setup. Adapted from Publication IV

traffic. In Figure 3.2 we show one of the deployments of the test-bed.

To collect the data, we instructed the kernels on the slave machines to log on a per packet basis the information about the number of retries, acknowledgment flags, packet sizes, used contention windows and backoff intervals. In doing so we encountered a problem with the Linux kernel, which did not allow us to log this information too frequently. To overcome the issue, we recompiled the kernel with an increased ring buffer size for debug messages and also increased the kernel *printk* rate limit. Upon receiving the packet transmission status notification from the firmware, the kernel registers the event and logged it into ring buffer. The ring buffer was periodically (every 0.1 seconds) read and dumped into a file. Each event was also flagged with the wireless interface ID and a timestamp.

In total we used 12 wireless cards installed on three slave nodes. For the majority of experiments, we have used 3, 6, 9 and 12 concurrently active clients. We have balanced the usage in such a way that for any number of active clients we have employed all three slave nodes in our test-bed. Our eventual goal was to study the combined performance of all active clients, which required merging the logs recorded on different machines. Since the clocks on the machines were not in sync, we had to find a way to correctly align our logs. The solution was to send calibrating beacons from the master node to all slave machines via wired interfaces. In principle, it would have sufficed to send a single beacon at the beginning of each experiment, which the slave nodes would have recorded as a reference time frame. Then subtracting this value from each packet's timestamp would yield a relative packet's timestamp in the merged log file. Alas, this solution is not perfect since in prolonged experiments the clock drift among different machines would cripple relative packet timings by putting some of them unduly further into the future or the past. To eliminate the effect of clock drift we instructed the master node to send beacons periodically, with an interval of 10 msec. This made it possible to do the re-alignment on short timescales.

Even though the beacons were sent with a strict 10 msec interval, there was no guarantee that they were recorded by slave nodes with exactly the same intervals. In fact, the various network, NIC or OS, effects could also cause perfect inter-arrival time to drift. Incorrect beacon inter-arrival times could then result in imperfect binning and thus undermine any analysis that relies on the assumption of constant bin size. To assess the possible drift in beacon timestamps, we calculated beacon inter-arrival times for all logs. To our relief, inter-arrival times turned out to be sharply clustered around 10 msec although the figure still showed rare outliers. This could lead to a drift in cumulative beacon intervals among several machines. However, we also calculated differences between respective beacon inter-arrival times on different machines. It turned out that the distribution was centered at zero, highly clustered and symmetric proving that bins calculated based on beacons remain equally sized in the long run.

After ensuring that the data was properly collected and calibrated, we turned to an analysis of the data sets. Our research agenda was to observe the behavior of all the protocols in various environments. Thus, we were interested in the results for aggregated throughput, fairness, collision probability, and the delays obtained for different protocols in four various scenarios. In each of the experiments described below we varied such parameters as number of active stations and used a backoff factor. Accordingly, we review the key results of our experiments and discuss them in the next few paragraphs.

*Experiment with close proximity setting*. Our first data set contained data for the experiment in which the nodes were placed close to an access point. Using this setting, we tried to imitate real-life, dense deployments of wireless stations. In summary, our experiments revealed that the penalty backoff and rollback backoff both achieve significant improvements in throughput characteristics in comparison with standard backoff. For example, the average improvement of rollback backoff (configured with the optimized parameter for backoff factor) over the standard backoff (with the backoff factor 2.0) was 77%. Significant improvement was also achieved for backoff with penalty (also when configured with the optimized parameter for the backoff factor). Another observation was related to backoff with fixed contention windows. In this setting the protocol showed results comparable (or even slightly worse) to those of the standard backoff protocol.

The results for packet collision rate resemble similar trends. Thus, the proposed protocols had a collision rate twice as small as the standard protocol. For example, the median packet collision rate for the experiments with the different number of contending stations varied between $0.14$ and $0.2$ for the backoff with penalty and between $0.15$ and $0.21$ for rollback backoff respectively. The results for the standard backoff were considerably higher and were between $0.3$-$0.4$ marks.

Closely resembling trends were also observed for fairness. Our data showed that all protocols but the standard (with a standard value for the backoff factor) had a nearly perfect fairness characteristic. We also observed that the standard backoff protocol showed slightly better fairness when it was configured with smaller values for the backoff factor. We concluded that such a result was expected because when contention windows are small enough, the chances of an arbitrary station capturing the channel for a long period of time were insignificant.

The data set that we used to derive the results discussed in previous paragraphs represented the setting in which wireless stations used the dynamic rate adaptation algorithm [4]. It was our next step to repeat the same experiment involving $12$ stations but now setting the wireless transmission rate to a fixed value. The bottom line here was that all four protocols were achieving comparable aggregated throughput. However, penalty backoff and rollback backoff showed nearly perfect fairness in all experiments. The same results were not achievable for the standard backoff protocol.

*Sparse deployment experiment*. To corroborate our observations in the close proximity setup, we conducted a set of additional experiments where nodes were placed apart from each other by as much as $30$ meters. After analyzing the data sets we found that the trends in these experiments closely resembled those in the close proximity setup. Thus, both the penalty backoff and the rollback backoff showed nearly perfect results for fairness. Similarly to the experiments in close proximity environment, we also calculated the average improvement of the penalty based pro-

[4] In the test-bed, the stations were configured with dynamic rate adaptation algorithm

tocols over the standard backoff protocol. Thus, it turned out that the improvement for the rollback backoff was > $70\%$ in comparison with standard backoff. The result for the backoff with penalty was also significantly better in comparison with standard backoff protocol.

*Experiment with hidden stations.* One particular adverse scenario which can occur in the IEEE 802.11 network is when two (or more) spatially separated nodes cannot reliably receive the signals from each other. This phenomena is often called the *hidden station problem*, entailing in its turn significant fairness and throughput impairments. This particular scenario was immensely interesting for this reason. Thus, we sat down to experiment with the two hidden stations. Regarding the goal of the experiment, here we wanted to observe whether the penalty mechanisms, built-in in our backoff protocols, could solve the problem without requiring any additional mechanisms such as RTS/CTS.

In this setting, the penalty and rollback backoff protocols when configured with the optimize backoff parameters achieved far better throughput than the standard backoff protocol. The fairness characteristic, however, turned out to be well below the limits that were considered satisfactory for all three protocols. However, there was another very significant observation that was made. We observed that nearly perfect fairness was achievable when the values for the backoff factor parameter exceeded the threshold value of $1.6$. Remarkably, even when configured with the non-optimal parameter of $1.7$, the throughput results for the penalty and rollback backoff protocols were comparable to those obtained for the standard backoff protocol. In comparison, the fairness characteristic came to around $0.9$ for the penalty and the rollback backoff protocols versus $0.5 - 0.6$ for the standard backoff protocol (In our work to represent the fairness quantitatively, we used Jain's fairness index [66]. According to this metric the value of $1.0$ corresponds to perfect fairness and $1/N$ (where $N$ is the number of users in the system) is the indication of total unfairness).

*Experiment with download traffic.* Our next experiment involved stations performing bulky downloads. After analyzing the data sets, we observed that both the penalty and rollback backoff marginally outperformed the standard backoff protocol. But again, the fairness characteristic for these two non-standard protocols was far superior than for the standard backoff protocol.

*Experiment with delay sensitive traffic.* Our final experiment was designed to understand the impact of our protocols on the per-packet delays.

This experiment involved a traffic mixture comprising a low but fixed rate UDP flow generated by a single station and several bulky TCP flows generated by multiple stations. All flows were generated in an upstream direction from slave nodes to the master node.

The bottom line in this experiment was that the per-packet delays for UDP flow were comparable for all three protocols. Moreover, for the optimized configuration of the protocols, the delays were sharply clustered around 10ms, which corresponded to the original packet generation rate. On the other hand, we observed that these delays tended to increase with the growth of the number of stations. We concluded that this problem was related to capacity planning rather than being an issue of the proposed designs.

To corroborate the results obtained in our experiments, we also conducted several simulation experiments using the NS-3 [118] framework. We simulated the two main deployment modes described previously: the close proximity setting and the setting with hidden stations. Our first simulation setup included 12 stations attached to an access point, with each station performing a TCP upload to a machine attached to a wired network. Overall, the trends observed in the simulations supported our previous empirical evidence. For example, the trend we saw in the real life experiments in the close proximity settings was comparable to the trends we saw in the simulations: the median aggregated throughput was 6.5 and 9.5 Mb/s for the standard and penalty backoff protocols respectively; Similarly, the collision rate in simulations was around 13% and 4% for the standard and penalty backoff protocols respectively. These numbers are smaller than the collision rates we saw in real experiments, which is certainly to be expected since the simulation provides an idealization of many real mechanisms such as timers, queues, etc. Nevertheless, the overall trend clearly persisted.

We also conducted the simulations for the setting with two hidden stations. The experiment was performed for the penalty backoff with the backoff factor set to 1.2 and 1.7 and standard backoff (with the backoff factor being set to its default value of 2.0). Upon analyzing the data we concluded that real-life experiments and simulations both showed similar trends. For example, while the median throughput was 1.6 and 1.9 Mb/s for standard backoff and penalty backoff configured with $r = 1.7$ respectively. The penalty backoff also achieved almost perfect fairness (0.93) verses 0.69 for the standard backoff protocol.

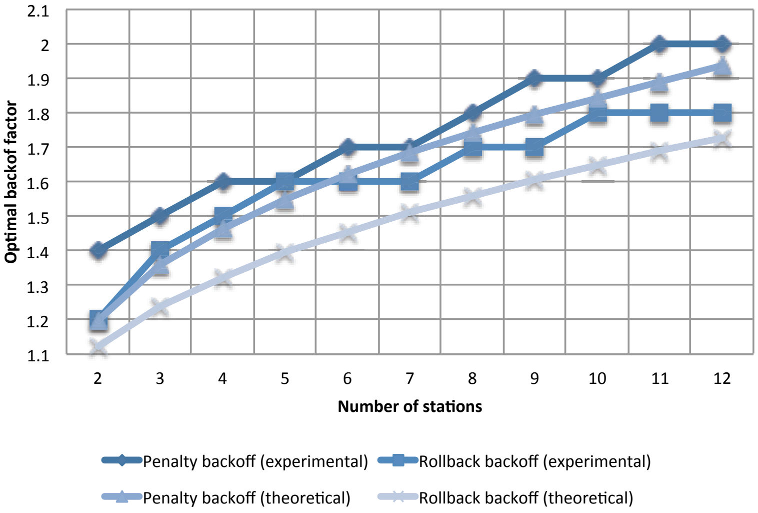

**Figure 3.3.** Comparison of the optimized backoff factor parameters found experimentally and analytically. Adapted from Publication IV

When working with the empirical data, it became clear to us that the proposed protocols achieve better performance when the values for the backoff factor are carefully selected. Furthermore, we noticed that these values depended on the number of active stations in the network. In the experiments described these values were found empirically. To confirm these observations, we devised a mathematical model for each protocol and found the optimized values for these parameters analytically. We compared these results with values obtained empirically for a varying number of wireless stations. For example, in Figure 3.3 we show the comparison of the values obtained experimentally and analytically for the first 12 stations.

In practice, the backoff factor parameters for the suggested protocols should be adapted dynamically, based on the current system load and number of stations communicating. This requires an additional mechanism enabling correct protocol operation in dynamic environments. To fill the gap, we designed and implemented two different algorithms that allow the access point to choose proper parameters and configure the wireless stations accordingly.

We opted out of using the approaches based on counting an observed number of idle slots as other works suggest [23, 52, 64]. We concluded that these algorithms are harder to implement, but they can also be inaccurate if hidden stations are present. Instead, we turned to approaches in which an access point estimates the number of active stations using information about the number of associated stations and amount of traffic each such station generates. Thus, in the first algorithm, the access point simply counts a station as *active* if this station occupies the channel for a duration of time longer than some pre-configured threshold. The choice of the threshold in this algorithm is purely empirical. The second algorithm is more complex, yet it allows the access point to estimate the number of active stations more accurately. According to this algorithm, any station that occupies the channel for a time greater than or equal to a fair share is considered as active. This coarse-grained estimate is then augmented with an estimate of the stations that occupied the channel for a time less than a fair share. Together these two values comprise a more precise estimate for the number of active stations. Mathematically this can be represented as follows:

$$N_{active} = \sum_{\forall i} I(\tau_i \geq x) + \lfloor \frac{\sum_{\forall j} \tau_j I(\tau_j < x)}{x} \rfloor$$

where $I(\cdot)$ is an indicator function, $\tau_j$ is the duration of time (in a given window $T$) a station $j$ occupies the channel and $x$ is the fair share calculated as $x = \frac{\sum_{\forall i} \tau_i}{N_{associated}}$.

We implemented these algorithms in the Linux distribution OpenWrt using *hostap daemon* - a piece of software that realizes wireless access point functionality. We also introduced a new management frame. The access point used this frame to convey the estimates to all stations in its vicinity. Wireless stations, on the other hand, used it to select the correct value of the backoff factor.

The final step was to validate these designs. In the experiment we employed all $12$ stations out of which $6$ of them followed an *on-off* pattern and were sending traffic every other $30$ seconds for a $30$ second period. The other $6$ stations were continuously sending traffic. The experimental data revealed that the performance (time to complete the experiment) for the setup with a simple adaptation algorithm and penalty backoff was around $20\%$ better than the performance for the setup with the standard protocol. The performance gain for the setup with the second algorithm was even more prominent.

### 3.2.2 Fairness and dynamic environments

During the course of our measurement study, we concluded that measuring fairness quantitatively in dynamic IEEE 802.11 wireless networks (by dynamic settings here we understand environments in which nodes are typically generating different amounts and types of traffic), can be a challenging task. This process can be complicated in multiple ways. For example, in real environments it is common that different hosts have different demands for a resource: some users can perform bulky upload, others can be involved in communication sessions requiring sending few packets but at a constant rate. This poses a question: *How can we measure fairness quantitatively in dynamic environments in which hosts might have uneven demands for the network resource?* To overcome this hurdle, in Publication V we attempt to devise a fairness metric that can be applied to the above mentioned scenarios. We showed experimentally how the results obtained with this metric are different from other commonly used approaches.

Furthermore, we discussed a possible way how to use this metric in existing wireless networks to ensure better resource allocation. For example, based on the outcome of this metric, a wireless access point can drop packets or even impose penalties through other mechanisms (such as by marking packets with a special flag) for some users, ensuring overall fair resource usage. In other words we discussed the possibility of decongesting the wireless network according to the demands of the users and the amount of congestion caused by each user.

## 3.3 Open research questions

Having reviewed the main results of our research, we will now try to articulate several possible future directions:

In Publication I and Publication II we presented several possible solutions for node accountability and their possible deployment paths. Furthermore, the work in Publication II covers many other building blocks of future, evolvable Internet architecture. In this context, a further understanding of how these potential building blocks can be implemented and incorporated into existing Internet infrastructure is of the utmost importance. Here, real implementations in *software defined networks* and a larger scale deployments of these protocols is an interesting research direction. Another direction can be investigation of how to combine filtering, capability and accountability approaches in order to build an efficient and salable network auditing framework: Such framework could be used to debug and resolve various network problems as well as isolate potential sources of attacks.

In Publication III we have explored the distributed security protocol. We have limited our efforts to a single type of network in which this protocol can be applicable – wireless sensor networks. There are several other types of networks which exhibit similar properties. Thus, a possible future research can be related to application of the protocol to these networks. For example, real-life implementation of the protocol for a peer-to-peer network could be interesting.

Although our work in Publication IV covers a wide range of experiments in different settings, this work can be still extended in multiple ways. For example, we have demonstrated several mechanisms enabling protocol adaptation in a setup comprising a single operating network. Here, it can be interesting to investigate how to adapt the protocol for multiple networks operating in the shared environment. Understanding how different networks comprising modified and legacy protocols can coexist deserves at least some attention: Performing a wider range of experiments or even modeling such scenarios theoretically can entail ideas for more efficient designs of wireless networks. Applying the discussed protocols to most recently developed variants of WLAN networks could be also interesting. For example, one could try to incorporate the ideas of the penalty backoff into the design presented in [145].

And finally, in Publication V we merely scratched the surface when we discussed a possible mechanism for improving fairness by giving penalties to wireless stations based on their demands and actual usage of the resources. We have attempted some preliminary investigations of these ideas. However, this work remains to be far from complete, and thus one could pave a further way in this direction. One could also investigate how these ideas are related to the approach described in [21].

# 4. Conclusions

In this thesis we addressed the problem of malicious and non-malicious faults that impact the stability and availability of network services and applications. Our main objective was to investigate several penalty and revocation mechanisms designed to mitigate these faults, ensuring an efficient and fair network resource utilization.

Thus, in Publication I and Publication II, we considered that fairness and availability in the Internet can be undermined by nodes which are compromised and so deliberately exhaust resources on servers, clients and other network bottlenecks. To counter these nodes we have designed architectures to account for the actions of the nodes and to shut off malicious nodes during attacks. Here, we investigated what are the needed requirements for such Internet-wide accountability and node revocation frameworks.

Next we moved on to a similar problem in wireless sensor networks in which compromised nodes (or otherwise nodes that are non-malicious but still faulty) can endanger the correct functioning of the network. To overcome this hurdle, in Publication III we designed and analyzed the cooperative node revocation protocol – a security protocol which allows nodes to cooperate, ensuring fair utilization of network resources by preferentially admitting only trusted nodes and revoking those nodes that have forfeited this trust.

In Publication IV, we investigated a different, but related (in terms of availability) problem. Here we considered a wireless edge network in which some nodes can behave in an unfair manner, endangering resource availability for some users. To counter such unfair nodes, we proposed using several penalty mechanisms incorporated into the backoff function of IEEE 802.11 networks. We showed the effectiveness of these mechanisms through real-life experiments and simulations. Furthermore, to facilitate the optimal operation of these protocols in dynamic settings, we devised and evaluated protocol adaptation mechanisms.

And finally, in Publication V we took a closer look at the fairness problem in wireless networks. Here, we argued that fairness can be assessed in a better way by taking into consideration the resource demands of the users and the levels of congestion these users cause to the wireless network. We showed how this metric is different from other approaches with experiments and examples, and, finally, we discussed how our approach can be used in existing wireless networks to ensure better fairness by imposing penalties (such as by dropping packets or marking them with a special congestion bit) for unfair users.

# Bibliography

[1] Host Identity Protocol for Linux http://hipl.hiit.fi/. Online, 2013.

[2] ABADI, M., BURROWS, M., MANASSE, M., AND WOBBER, T. Moderately hard, memory-bound functions. *ACM Trans. Internet Technol. 5*, 2 (May 2005), 299–327.

[3] ANDERSEN, D. G. Mayday: distributed filtering for Internet services. In *Proceedings of the 4th conference on USENIX Symposium on Internet Technologies and Systems - Volume 4* (Berkeley, CA, USA, 2003), USITS'03, USENIX Association.

[4] ANDERSEN, D. G., BALAKRISHNAN, H., FEAMSTER, N., KOPONEN, T., MOON, D., AND SHENKER, S. Accountable Internet Protocol (AIP). *ACM SIGCOMM Computer Communication Review 38*, 4 (Aug. 2008), 339–350.

[5] ANDERSON, T., ROSCOE, T., AND WETHERALL, D. Preventing Internet denial-of-service with capabilities. *ACM SIGCOMM Computer Communication Review 34*, 1 (Jan. 2004), 39–44.

[6] ARGYRAKI, K., AND CHERITON, D. R. Active Internet traffic filtering: real-time response to denial-of-service attacks. In *Proceedings of the annual conference on USENIX Annual Technical Conference* (Berkeley, CA, USA, 2005), ATEC '05, USENIX Association, pp. 135–148.

[7] AURA, T., NIKANDER, P., AND LEIWO, J. DOS-Resistant Authentication with Client Puzzles. In *Revised Papers from the 8th International Workshop on Security Protocols* (London, UK, UK, 2001), Springer-Verlag, pp. 170–177.

[8] AVIZIENIS, A., LAPRIE, J.-C., RANDELL, B., AND LANDWEHR, C. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Transactions on Dependable and Secure Computing 1*, 1 (Jan. 2004), 11–33.

[9] BABU, A., AND JACOB, L. Performance analysis of IEEE 802.11 multirate WLANs: time based fairness vs throughput based fairness. In *Proceedings of International Conference on Wireless Networks, Communications and Mobile Computing* (June 2005), vol. 1, pp. 203–208.

[10] BACK, A. Hashcash - A Denial of Service Counter-Measure http://www.cypherspace.org/adam/hashcash/hashcash.pdf. Tech. rep., 2002.

[11] BAKER, F., AND SAVOLA, P. Ingress Filtering for Multihomed Networks. RFC 3704, 2004.

[12] BALACHANDRAN, A., VOELKER, G. M., BAHL, P., AND RANGAN, P. V. Characterizing user behavior and network performance in a public wireless LAN. In *Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems* (New York, NY, USA, 2002), SIGMETRICS '02, ACM, pp. 195–205.

[13] BALLANI, H. AND CHAWATHE, Y. AND RATNASAMY, S. AND ROSCOE, T. AND SHENKER, S. Off by Default! In *Proc. of workshop on Hot Topics in Networks (HotNets-IV)* (November 2005).

[14] BERGER-SABBATEL, G., DUDA, A., GAUDOUIN, O., HEUSSE, M., AND ROUSSEAU, F. Fairness and its impact on delay in 802.11 networks. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'04)* (Dallas, USA, Nov. 2004), vol. 5, pp. 2967–2973.

[15] BERTSEKAS, D., AND GALLAGER, R. *Data Networks*. Prentice-Hall, Englewood Cliffs, New Jersey, 1992.

[16] BIANCHI, G. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications 18*, 3 (Mar 2000), 535–547.

[17] BIANCHI, G., FRATTA, L., AND OLIVERI, M. Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs. In *Proceedings of Seventh IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 1996. PIMRC'96* (oct 1996), vol. 2, pp. 392–396 vol.2.

[18] BICKET, J. C. Bit-rate selection in wireless networks. Tech. rep., Master's thesis, MIT, 2005.

[19] BLACK, J., AND ROGAWAY, P. A block-cipher mode of operation for parallelizable message authentication. In *Proceedings of Advances in Cryptology - EUROCRYPT 2002. Lecture Notes in Computer Science* (2002), Springer-Verlag, pp. 384–397.

[20] BREMLER-BARR, A., AND LEVY, H. Spoofing prevention method. In *Proceedings of IEEE INFOCOM '2005* (2005), pp. 536–547.

[21] BRISCOE, B. Nice traffic management without new protocols, 2012. `http://www.bobbriscoe.net/presents/1210isoc/1210isoc-briscoe.pdf`.

[22] CACHIN, C., KURSAWE, K., AND SHOUP, V. Random oracles in constantinople: Practical asynchronous byzantine agreement using cryptography. In *Proceedings of 19th ACM Symposium on Principles of Distributed Computing (PODC* (2000), pp. 123–132.

[23] CALÌ, F., CONTI, M., AND GREGORI, E. IEEE 802.11 Wireless LAN: Capacity Analysis and Protocol Enhancement. In *Proceedings IEEE Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM'98* (1998).

[24] CASADO, M., CAO, P., AKELLA, A., AND PROVOS, N. Flow-Cookies: Using Bandwidth Amplification to Defend Against DDoS Flooding Attacks. In *Quality of Service - IWQoS 2006: 14th International Workshop, IWQoS 2006, New Haven, CT, USA, 19-21 June 2006, Proceedings* (2006), IEEE, pp. 286–287.

[25] CELIK, G. D., ZUSSMAN, G., KHAN, W. F., AND MODIANO, E. MAC for Networks with Multipacket Reception Capability and Spatially Distributed Nodes. *IEEE Transactions on Mobile Computing 9*, 2 (Feb. 2010), 226–240.

[26] CERT. Smurf IP Denial-of-Service Attacks. Online, 2000. `http://www.cert.org/historical/advisories/ca-1998-01.cfm`.

[27] CERTICOM. Sec 1: Elliptic curve cryptography. Online, 2000. `www.secg.org/collateral/sec1_final.pdf`.

[28] CHAN, H., GLIGOR, V. D., PERRIG, A., AND MURALIDHARAN, G. On the distribution and revocation of cryptographic keys in sensor networks. *IEEE Transactions on Dependable and Secure Computing 2* (2005), 233–247.

[29] CHAN, H., PERRIG, A., AND SONG, D. Random key predistribution schemes for sensor networks. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy* (Washington, DC, USA, 2003), SP '03, IEEE Computer Society, pp. 197–213.

[30] CHANDRA, T. D., AND TOUEG, S. Unreliable failure detectors for reliable distributed systems. *Journal of ACM 43*, 2 (Mar. 1996), 225–267.

[31] CHENG, Y.-C., BELLARDO, J., BENKÖ, P., SNOEREN, A. C., VOELKER, G. M., AND SAVAGE, S. Jigsaw: solving the puzzle of enterprise 802.11 analysis. In *Proceedings of ACM SIGCOMM '06* (2006), pp. 39–50.

[32] CHO, J.-W., AND JIANG, Y. Fundamentals of the Backoff Process in 802.11. *CoRR abs/0904.4155* (2009).

[33] CISCO SYSTEMS. Cisco guard. `http://www.cisco.com/en/US/prod/collateral/vpndevc/ps5879/ps6264/ps5888/prod_white_paper0900aecd8011e927.html`. Online, 2013.

[34] CLULOW, J., AND MOORE, T. Suicide for the common good: a new strategy for credential revocation in self-organizing systems. *SIGOPS Opererating Systems Review 40*, 3 (July 2006), 18–21.

[35] DENDA, R., BANCHS, A., AND EFFELSBERG, W. The fairness challenge in computer networks. In *Proceedings of the First COST 263 International Workshop on Quality of Future Internet Services* (London, UK, UK, 2000), QofIS '00, Springer-Verlag, pp. 208–220.

[36] DIERKS, T., AND ALLEN, C. The TLS Protocol Version 1.0. RFC 2246, January 1999.

[37] DIFFIE, W., AND HELLMAN, M. E. New Directions in Cryptography. *IEEE Transactions on Information Theory IT-22*, 6 (1976), 644–654.

[38] DINI, G., AND SAVINO, I. M. An efficient key revocation protocol for wireless sensor networks. In *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks* (Washington, DC, USA, 2006), WOWMOM '06, IEEE Computer Society, pp. 450–452.

[39] DUDA, A. Understanding the performance of 802.11 networks. In *Proceedings of IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC'08* (sept. 2008), pp. 1 –6.

[40] DWORK, C., GOLDBERG, A., AND NAOR, M. On memory-bound functions for fighting spam. In *Proceedings of Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings* (2003), vol. 2729 of *Lecture Notes in Computer Science*, Springer, pp. 426–444.

[41] ESCHENAUER, L., AND GLIGOR, V. D. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM conference on Computer and communications security* (New York, NY, USA, 2002), CCS '02, ACM, pp. 41–47.

[42] EVANS, D. The Internet of Things How the Next Evolution of the Internet Is Changing Everything. Online, 2011. `http://www.cisco.com/web/about/ac79/iot/index.html`.

[43] FELEMBAN, E., AND EKICI, E. Single Hop IEEE 802.11 DCF Analysis Revisited: Accurate Modeling of Channel Access Delay and Throughput for Saturated and Unsaturated Traffic Cases. *IEEE Transactions on Wireless Communications 10*, 10 (2011), 3256–3266.

[44] FERGUSON, P., AND SENIE, D. Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. IETF RFC 2267, 1998.

[45] FERGUSON, P., AND SENIE, D. Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2827 (Best Current Practice), May 2000. Updated by RFC 3704.

[46] FREIER, A., KARLTON, P., AND KOCHER, P. The Secure Sockets Layer (SSL) Protocol Version 3.0. RFC 6101, 2011.

[47] GARCIA-MORCHON, O., BALDUS, H., HEER, T., AND WEHRLE, K. Cooperative security in distributed sensor networks. In *Proceedings of CollaborateCom* (2007), IEEE, pp. 96–105.

[48] GHOSH, S., HERMAN, T., AND PEMMARAJU, S. V. A fault-containing self-stabilizing algorithm for spanning trees. In *Proceedings of Journal of Computing and Information* (1996), pp. 322–338.

[49] GLIGOR, V. D. Guaranteeing Access in Spite of Distributed Service-Flooding Attacks (Discussion). In *Proceedings of Security Protocols Workshop* (2003), B. Christianson, B. Crispo, J. A. Malcolm, and M. Roe, Eds., vol. 3364 of *Lecture Notes in Computer Science*, Springer, pp. 97–105.

[50] GOLLAKOTA, S., AND KATABI, D. Zigzag decoding: combating hidden terminals in wireless networks. In *Proceedings of ACM SIGCOMM '08* (2008), pp. 159–170.

[51] GREENHALGH, A., HANDLEY, M., AND HUICI, F. Using routing and tunneling to combat DoS attacks. In *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet on Steps to Reducing Unwanted Traffic on the Internet Workshop* (Berkeley, CA, USA, 2005), SRUTI'05, USENIX Association, pp. 1–1.

[52] GRUNENBERGER, Y., HEUSSE, M., ROUSSEAU, F., AND DUDA, A. Experience with an implementation of the idle sense wireless access method. In *Proceeding of ACM CoNEXT '07*, pp. 24:1–24:12.

[53] GUDIPATI, A., AND KATTI, S. Strider: automatic rate adaptation and collision handling. In *Proceedings of the ACM SIGCOMM 2011 conference* (New York, NY, USA, 2011), SIGCOMM '11, ACM, pp. 158–169.

[54] GUHA, S., FRANCIS, P., AND TAFT, N. ShutUp: End-to-end containment of unwanted traffic. Tech. rep., 2008.

[55] GURA, N., PATEL, A., WANDER, A., EBERLE, H., AND SHANTZ, S. C. Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs. In *Proceedings of Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004.* (2004), vol. 3156 of *Lecture Notes in Computer Science*, Springer, pp. 119–132.

[56] GURTOV, A. TCP Performance in the Presence of Congestion and Corruption Losses. Tech. rep., Master's Thesis, University of Helsinki, Department of Computer Science, 2000.

[57] GURTOV, A. *Host Identity Protocol (HIP): Towards the Secure Mobile Internet*. Wiley and Sons, 2008.

[58] GURTOV, A., AND KORHONEN, J. Effect of Vertical Handovers on Performance of TCP-Friendly Rate Control. In *ACM Mobile Computing and Communications Review* (2004), vol. 8, pp. 73–87.

[59] GURTOV, A., KORZUN, D., LUKYANENKO, A., AND NIKANDER, P. Hi3: An efficient and secure networking architecture for mobile hosts. *Computer Communimcations 31*, 10 (June 2008), 2457–2467.

[60] HAYASHIBARA, N., DEFAGO, X., YARED, R., AND KATAYAMA, T. The $\phi$ accrual failure detector. In *Proceedings of the 23rd IEEE International Symposium on Reliable Distributed Systems* (2004), IEEE, pp. 66–78.

[61] HEER, T., LI, S., AND WEHRLE, K. PISA: P2P Wi-Fi Internet Sharing Architecture. *IEEE International Conference on Peer-to-Peer Computing* (2007), 251–252.

[62] HENDERSON, T., KOTZ, D., AND ABYZOV, I. The changing usage of a mature campus-wide wireless network. *Comput. Netw. 52*, 14 (Oct. 2008), 2690–2712.

[63] HERMAN, T., AND PEMMARAJU, S. Error-detecting codes and fault-containing self-stabilization. *Inf. Process. Lett. 73*, 1-2 (Jan. 2000), 41–46.

[64] HEUSSE, M., ROUSSEAU, F., GUILLIER, R., AND DUDA, A. Idle sense: An optimal access method for high throughput and fairness in rate diverse wireless LANs. In *Proceedings of ACM SIGCOMM* (2005), ACM Press, pp. 121–132.

[65] IEEE. Local and Metropolitan Area Networks: Port-Based Network Access Control. IEEE Standard 802.1X, September 2001.

[66] JAIN, R. *The Art of Computer Systems Performance Analysis*. John Wiley and Sons, 1991.

[67] JARDOSH, A. P., RAMACHANDRAN, K. N., ALMEROTH, K. C., AND BELDING-ROYER, E. M. Understanding link-layer behavior in highly congested IEEE 802.11b wireless networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis* (New York, NY, USA, 2005), E-WIND '05, ACM, pp. 11–16.

[68] JIN, C., WANG, H., AND SHIN, K. G. Hop-Count Filtering: an effective defense against spoofed DDoS traffic. In *Proceedings of the 10th ACM conference on Computer and communications security* (New York, NY, USA, 2003), CCS '03, ACM, pp. 30–41.

[69] JUELS, A., AND BRAINARD, J. G. Client Puzzles: A Cryptographic Countermeasure Against Connection Depletion Attacks. In *Proceedings of Network and Distributed System Security Symposium* (1999).

[70] KANDULA, S., KATABI, D., JACOB, M., AND BERGER, A. W. Botz-4-Sale: Surviving Organized DDoS Attacks That Mimic Flash Crowds. In *2nd Symposium on Networked Systems Design and Implementation (NSDI'05)* (Boston, MA, May 2005), pp. 287–300.

[71] KATTI, S., GOLLAKOTA, S., AND KATABI, D. Embracing wireless interference: analog network coding. In *Proceedings of ACM SIGCOMM '07* (2007), pp. 397–408.

[72] KATTI, S., KATABI, D., BALAKRISHNAN, H., AND MEDARD, M. Symbol-level network coding for wireless mesh networks. In *Proceedings of ACM SIGCOMM '08* (2008), pp. 401–412.

[73] KATTI, S., RAHUL, H., HU, W., KATABI, D., MÉDARD, M., AND CROWCROFT, J. Xors in the air: practical wireless network coding. In *Proceedings of ACM SIGCOMM '06* (2006), pp. 243–254.

[74] KENT, S., AND ATKINSON, R. Security Architecture for the Internet Protocol. RFC 2401, November 1998.

[75] KEROMYTIS, A. D., MISRA, V., AND RUBENSTEIN, D. SOS: secure overlay services. *ACM SIGCOMM Computer Communication Review 32*, 4 (Aug. 2002), 61–72.

[76] KHURRI, A. *Evaluating IP Security on Lightweight Hardware*. PhD thesis, Espoo, Finland, 2011.

[77] KHURRI, A., KUPTSOV, D., AND GURTOV, A. On application of Host Identity Protocol in wireless sensor networks. In *IEEE 7th International Conference on Mobile Adhoc and Sensor Systems (MASS)* (2010), IEEE, pp. 358–345.

[78] KOBLITZ, N. Elliptic Curve Cryptosystems. *Mathematics of Computation 48*, 177 (1987), 203–209.

[79] Koksal, C. E., Kassab, H., Balakrishnan, H., and Balakrishnan, H. An analysis of short-term fairness in wireless media access protocols. In *Proceedings of ACM Sigmetrics* (2000), pp. 118–119.

[80] Komu, M., Tarkoma, S., Kangasharju, J., and Gurtov, A. Applying a Cryptographic Namespace to Applications. In *Proceedings of the 1st ACM Workshop on Dynamic Interconnection of Networks* (New York, NY, USA, 2005), DIN '05, ACM, pp. 23–27.

[81] Kopponen, T. *A Data-Oriented Network Architecture*. PhD thesis, Espoo, Finland, 2008.

[82] Krawczyk, H., Bellare, M., and Canetti, R. HMAC: Keyed-Hashing for Message Authentication. RFC 2104, February 1997.

[83] Kumar, A., Altman, E., Miorandi, D., and Goyal, M. New insights from a fixed-point analysis of single cell IEEE 802.11 WLANs. *IEEE/ACM Trans. Netw. 15*, 3 (2007), 588–601.

[84] Kuptsov, D., Garcia, O., Wehrle, K., and Gurtov, A. On applications of cooperative security in distributed networks. In *Proceedings of IFIPTM 2010 - 4th International Conference on Trust Management* (June 14-18, 2010, Morioka, Japan, 2010).

[85] Kwak, B.-J., Song, N.-O., and Miller, L. E. Performance analysis of exponential backoff. *IEEE/ACM Trans. Netw. 13*, 2 (2005), 343–355.

[86] LaCurts, K., and Balakrishnan, H. Measurement and analysis of real-world 802.11 mesh networks. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement* (New York, NY, USA, 2010), IMC '10, ACM, pp. 123–136.

[87] Lagutin, D. *Securing the Internet with digital signatures*. PhD thesis, Espoo, Finland, 2010.

[88] Lamport, L. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM 21*, 7 (July 1978), 558–565.

[89] Lamport, L. Constructing digital signatures from a one-way function. Tech. rep., Oct. 1979.

[90] Li, J., Mirkovic, J., Wang, M., Reiher, P., and Zhang, L. SAVE: Source address validity enforcement protocol. In *Proceedings of IEEE INFOCOM '2002* (2002), pp. 1557–1566.

[91] Lin, K. C.-J., Gollakota, S., and Katabi, D. Random access heterogeneous MIMO networks. In *Proceedings of the ACM SIGCOMM 2011* (New York, NY, USA, 2011), SIGCOMM '11, ACM, pp. 146–157.

[92] Liu, D., Ning, P., and Sun, K. Efficient self-healing group key distribution with revocation capability. In *Proceedings of the 10th ACM Conference on Computer and Communications Security* (New York, NY, USA, 2003), CCS '03, ACM, pp. 231–240.

[93] Liu, X., Li, A., Yang, X., and Wetherall, D. Passport: secure and adoptable source authentication. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation* (Berkeley, CA, USA, 2008), NSDI'08, USENIX Association, pp. 365–378.

[94] LIU, X., YANG, X., AND LU, Y. To filter or to authorize: network-layer DoS defense against multimillion-node botnets. In *Proceedings of the ACM SIGCOMM 2008 conference on Data communication* (New York, NY, USA, 2008), SIGCOMM '08, ACM, pp. 195–206.

[95] LIU, X., YANG, X., AND XIA, Y. Netfence: Preventing internet denial of service from inside out. *SIGCOMM Comput. Commun. Rev. 40*, 4 (Aug. 2010), 255–266.

[96] LOPEZ-AGUILERA, E., HEUSSE, M., GRUNENBERGER, Y., ROUSSEAU, F., DUDA, A., AND CASADEMONT, J. An Asymmetric Access Point for Solving the Unfairness Problem in WLANs. *IEEE Transactions on Mobile Computing 7*, 10 (Oct. 2008), 1213–1227.

[97] LUKYANENKO, A., AND GURTOV, A. Performance analysis of general backoff protocols. *Journal of Communications Software and Systems 4*, 1 (2008), 13–21.

[98] LUKYANENKO, A., GURTOV, A., AND MOROZOV, E. An Adaptive Backoff Protocol with Markovian Contention Window Control. *Journal of Communications in Statistics - Simulation and Computation 41*, 7 (2012), 1093–1106.

[99] LUKYANENKO, A., MAZALOV, V., GURTOV, A., AND FALKO, I. Playing Defense by Offense: Equilibrium in the DoS-attack problem. In *Proceedings of the 15th IEEE Symposium on Computers and Communications, ISCC 2010, Riccione, Italy, June 22-25, 2010* (2010), IEEE, pp. 433–436.

[100] LUO, H., KONG, J., ZERFOS, P., LU, S., AND ZHANG, L. URSA: ubiquitous and robust access control for mobile ad-hoc networks. *IEEE/ACM Trans. Netw. 12*, 6 (Dec. 2004), 1049–1063.

[101] MADWIFI. Onoe rate control. `http://madwifi-project.org/browser/madwifi/trunk/ath_rate/onoe/`. Online, 2012.

[102] MAHAJAN, R., BELLOVIN, S. M., FLOYD, S., IOANNIDIS, J., PAXSON, V., AND SHENKER, S. Controlling high bandwidth aggregates in the network. *SIGCOMM Comput. Commun. Rev. 32*, 3 (July 2002), 62–73.

[103] MANKINS, D., KRISHNAN, R., BOYD, C., ZAO, J., AND FRENTZ, M. Mitigating Distributed Denial of Service Attacks with Dynamic Resource Pricing. In *Proceedings of the 17th Annual Computer Security Applications Conference* (Washington, DC, USA, 2001), ACSAC '01, IEEE Computer Society, pp. 411–421.

[104] MERKLE, R. C. *Secrecy, authentication, and public key systems.* PhD thesis, Stanford, CA, USA, 1979. AAI8001972.

[105] MILLER, V. S. Use of elliptic curves in cryptography. In *Proceedings of Advances in Cryptology—CRYPTO 85* (New York, NY, USA, 1986), Springer-Verlag New York, Inc., pp. 417–426.

[106] MOORE, T., CLULOW, J., NAGARAJA, S., AND ANDERSON, R. New strategies for revocation in ad-hoc networks. In *Proceedings of the 4th European conference on Security and privacy in ad-hoc and sensor networks* (Berlin, Heidelberg, 2007), ESAS'07, Springer-Verlag, pp. 232–246.

[107] MOREIN, W. G., STAVROU, A., COOK, D. L., KEROMYTIS, A. D., MISRA, V., AND RUBENSTEIN, D. Using graphic turing tests to counter automated DDoS attacks against web servers. In *Proceedings of the 10th ACM conference on Computer and communications security* (New York, NY, USA, 2003), CCS '03, ACM, pp. 8–19.

[108] MOSKOWITZ, R., AND NIKANDER, P. Host Identity Protocol architecture. IETF RFC 4423, May 2006.

[109] MOSKOWITZ, R., NIKANDER, P., AND JOKELA, P. Host Identity Protocol. RFC 5201, 2008.

[110] NG, A. C. H., MALONE, D., AND LEITH, D. J. Experimental evaluation of TCP performance and fairness in an 802.11e test-bed. In *Proceedings of the 2005 ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis* (New York, NY, USA, 2005), ACM, pp. 17–22.

[111] NIE, P., VÄHÄ-HERTTUA, J., AURA, T., AND GURTOV, A. Performance Analysis of HIP Diet Exchange for WSN Security Establishment. In *Proceedings of the 7th ACM Symposium on QoS and Security for Wireless and Mobile Networks* (New York, NY, USA, 2011), Q2SWinet '11, ACM, pp. 51–56.

[112] NIKANDER, P., ARKKO, J., AND OHLMAN, B. Host identity indirection infrastructure (Hi3). In *Proceedings of Second Swedish National Computer Networking Workshop (SNCNW), Karlstad, Sweden* (2004).

[113] NIST. FIBS PUB 46-3: Data Encryption Standard (DES). Online, 1999. http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf.

[114] NIST. FIBS PUB 180-4: Secure Hash Standard (SHS). Online, 2001. http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf.

[115] NIST. FIBS PUB 197: Advanced Encryption Standard (AES). Online, 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[116] NIST. Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. Online, 2001. http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf.

[117] NIST. FIBS PUB 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions , 2014. http://csrc.nist.gov/publications/drafts/fips-202/fips_202_draft.pdf.

[118] NS-3. http://www.nsnam.org/. Online, 2013.

[119] OF STANDARDS, N. I., AND TECHNOLOGY. FIBS PUB 186-3: Digital Signature Standard (DSS), 2009. http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf.

[120] O'HARA, B., AND PETRICK, A. *IEEE 802.11 Handbook : a designer's companion*. IEEE standards wireless networks series. IEEE, New York, 2005.

[121] OPENFWWF. http://www.ing.unibs.it/~openfwwf/. Online, 2013.

[122] OPENWRT. https://openwrt.org/. Online, 2013.

[123] PAXSON, V. An analysis of using reflectors for distributed denial-of-service attacks. *ACM SIGCOMM Computer Communication Review 31*, 3 (July 2001), 38–47.

[124] PEASE, M., SHOSTAK, R., AND LAMPORT, L. Reaching agreement in the presence of faults. *Journal of the ACM 27*, 2 (Apr. 1980), 228–234.

[125] PILOSOF, S., RAMJEE, R., RAZ, D., RAMJEE, R., SHAVITT, Y., AND SINHA, P. Understanding TCP Fairness over Wireless LAN. In *Proceedings of IEEE INFOCOM* (2003), pp. 863–872.

[126] PROLEXIC. http://www.prolexic.com/. Online, 2013.

[127] RAHUL, H. S., KUMAR, S., AND KATABI, D. JMB: scaling wireless capacity with user demands. In *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication* (New York, NY, USA, 2012), SIGCOMM '12, ACM, pp. 235–246.

[128] REITER, M. K. A secure group membership protocol. *IEEE Trans. Softw. Eng. 22*, 1 (Jan. 1996), 31–42.

[129] RIVEST, R. The MD5 Message-Digest Algorithm. RFC 1321, April 1992.

[130] RIVEST, R. L., SHAMIR, A., AND ADLEMAN, L. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM 26*, 1 (Jan. 1983), 96–99.

[131] RODRIG, M., REIS, C., MAHAJAN, R., WETHERALL, D., AND ZAHORJAN, J. Measurement-based characterization of 802.11 in a hotspot setting. In *Proceedings of the 2005 ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis* (New York, NY, USA, 2005), E-WIND '05, ACM, pp. 5–10.

[132] SAXENA, N., TSUDIK, G., AND YI, J. H. Identity-based access control for ad-hoc groups. In *Proceedings of the 7th international conference on Information Security and Cryptology* (Berlin, Heidelberg, 2005), ICISC'04, Springer-Verlag, pp. 362–379.

[133] SCHLICHTING, R. D., AND SCHNEIDER, F. B. Fail-stop processors: An approach to designing fault-tolerant computing systems. *ACM Transactions on Computer Systems 1* (1983), 222–238.

[134] SCHNEIDER, F. B. Implementing fault-tolerant services using the state machine approach: a tutorial. *ACM Comput. Surv. 22*, 4 (Dec. 1990), 299–319.

[135] SCHNEIER, B., KELSEY, J., WHITING, D., WAGNER, D., HALL, C., AND FERGUSON, N. *The Twofish encryption algorithm: a 128-bit block cipher*. John Wiley & Sons, Inc., New York, NY, USA, 1999.

[136] SHAMIR, A. How to share a secret. *Communications of the ACM 22*, 11 (Nov. 1979), 612–613.

[137] SHAW, M. Leveraging good intentions to reduce unwanted network traffic. In *Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet - Volume 2* (Berkeley, CA, USA, 2006), SRUTI'06, USENIX Association, pp. 9–9.

[138] SHOUP, V. Practical threshold signatures. In *Proceedings of the 19th international conference on Theory and Application of Cryptographic Techniques* (Berlin, Heidelberg, 2000), EUROCRYPT'00, Springer-Verlag, pp. 207–220.

[139] SNOEREN, A. C., PARTRIDGE, C., SANCHEZ, L. A., JONES, C. E., TCHAKOUNTIO, F., KENT, S. T., AND STRAYER, W. T. Hash-based IP traceback. In *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications* (New York, NY, USA, 2001), SIGCOMM '01, ACM, pp. 3–14.

[140] STARZETZ, P., HEUSSE, M., ROUSSEAU, F., AND DUDA, A. Hashing backoff: A collision-free wireless access method. In *Proceedings of IFIP-TC 6 NETWORKING '09*, pp. 429–441.

[141] STINSON, D. R. *Cryptography: Theory and Practice, Second Edition*. Chapman & Hall/CRC, Feb. 2002.

[142] SYSTEMS, C. The Future of Hotspots: Making Wi-Fi as Secure and Easy to Use as Cellular. Online, 2012. `http://www.cisco.com/c/en/us/solutions/collateral/service-provider/service-provider-wi-fi/white_paper_c11-649337.pdf`.

[143] SYSTEMS, C. A Cisco Guide to Defending Against Distributed Denial of Service Attacks. Online, 2014. `http://www.cisco.com/web/about/security/intelligence/guide_ddos_defense.html`.

[144] TAN, G., AND GUTTAG, J. Time-based Fairness Improves Performance in Multi-rate WLANs. In *Proceedings of USENIX'04* (Boston, MA, 2004).

[145] TAN, K., FANG, J., ZHANG, Y., CHEN, S., SHI, L., ZHANG, J., AND ZHANG, Y. Fine-grained channel access in wireless LAN. In *Proceedings of ACM SIGCOMM'10* (2010), pp. 147–158.

[146] TANG, D., AND BAKER, M. Analysis of a local-area wireless network. In *Proceedings of the 6th annual international conference on Mobile computing and networking* (New York, NY, USA, 2000), MobiCom '00, ACM, pp. 1–10.

[147] TINNIRELLO, I., BIANCHI, G., GALLO, P., GARLISI, D., GIULIANO, F., AND GRINGOLI, F. Wireless MAC processors: Programming MAC protocols on commodity hardware. In *Proceeding of INFOCOM'12* (2012), pp. 1269–1277.

[148] VIXIE, P. Events of 21-Oct-2002. Online, 2002. `http://c.root-servers.org/october21.txt`.

[149] VUTUKURU, M., BALAKRISHNAN, H., AND JAMIESON, K. Cross-layer wireless bit rate adaptation. In *Proceedings of the ACM SIGCOMM 2009 conference* (New York, NY, USA, 2009), SIGCOMM '09, ACM, pp. 3–14.

[150] WALFISH, M., VUTUKURU, M., BALAKRISHNAN, H., KARGER, D., AND SHENKER, S. DDoS defense by offense. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications* (New York, NY, USA, 2006), SIGCOMM '06, ACM, pp. 303–314.

[151] WU, G., AND CHIUEH, T. Passive and accurate traffic load estimation for infrastructure-mode wireless LAN. In *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems* (New York, NY, USA, 2007), MSWiM '07, ACM, pp. 109–116.

[152] WU, J., BI, J., REN, G., XU, K., AND WILLIAMS, M. A Source Address Validation Architecture (SAVA) Testbed and Deployment Experience. IETF RFC 5210, 2008.

[153] XIE, T., AND FENG, D. How To Find Weak Input Differences For MD5 Collision Attacks. Cryptology ePrint Archive, Report 2009/223, 2009. http://eprint.iacr.org/.

[154] YAAR, A., PERRIG, A., AND SONG, D. SIFF: A Stateless Internet Flow Filter to Mitigate DDoS Flooding Attacks. In *Proceedings of IEEE Symposium on Security and Privacy* (2004), pp. 130–143.

[155] YANG, X., WETHERALL, D., AND ANDERSON, T. A DoS-limiting network architecture. In *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications* (New York, NY, USA, 2005), SIGCOMM '05, ACM, pp. 241–252.

[156] YLONEN, T., AND LONVICK, C. The Secure Shell (SSH) Protocol Architecture. RFC 4251, January 2006.

[157] ZHAO, M., ZHOU, W., GURNEY, A. J., HAEBERLEN, A., SHERR, M., AND LOO, B. T. Private and verifiable interdomain routing decisions. In *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication* (New York, NY, USA, 2012), SIGCOMM '12, ACM, pp. 383–394.

Aalto-DD 185/2014

ISBN 978-952-60-5961-7 (printed)
ISBN 978-952-60-5962-4 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

**Aalto University**
**School of Science**
**Department of Computer Science and Engineering**
**www.aalto.fi**

BUSINESS +
ECONOMY

ART +
DESIGN +
ARCHITECTURE

SCIENCE +
TECHNOLOGY

CROSSOVER

**DOCTORAL**
**DISSERTATIONS**