Department of Computer Science
Series of Publications A
Report A-2010-5

# Multi-User Resource-Sharing Problem for the Internet

Andrey Lukyanenko

*To be presented, with the permission of the Faculty of Science of the University of Helsinki, for public criticism in auditorium PIII, University of Helsinki, Porthania campus, on November 17th 2010, at 12 o'clock noon.*

University of Helsinki
Finland

## Contact information

Postal address:
Department of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 University of Helsinki
Finland

Email address: postmaster@cs.helsinki.fi (Internet)

URL: http://www.cs.Helsinki.FI/

Telephone: +358 9 1911

Telefax: +358 9 191 51120


ISSN 1238-8645
ISBN 978-952-10-6557-6 (paperback)
ISBN 978-952-10-6558-3 (PDF)
Computing Reviews (1998) Classification: C.2, G.1.0, G.1.6, G.1.8, G.3, I.6
Helsinki 2010
Helsinki University Print

**Multi-User Resource-Sharing Problem for the Internet**

Andrey Lukyanenko

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
andrey.lukyanenko@hiit.fi
http://www.hiit.fi/people/

PhD Thesis, Series of Publications A, Report A-2010-5
Helsinki, October 2010, 81+72 pages
ISSN 1238-8645
ISBN 978-952-10-6557-6 (paperback)
ISBN 978-952-10-6558-3 (PDF)

**Abstract**

In this thesis we study a series of multi-user resource-sharing problems for the Internet, which involve distribution of a common resource among participants of multi-user systems (servers or networks). We study concurrently accessible resources, which for end-users may be exclusively accessible or non-exclusively. For all kinds we suggest a separate algorithm or a modification of common reputation scheme. Every algorithm or method is studied from different perspectives: optimality of protocols, selfishness of end users, fairness of the protocol for end users. On the one hand the multifaceted analysis allows us to select the most suited protocols among a set of various available ones based on trade-offs of optima criteria. On the other hand, the future Internet predictions dictate new rules for the optimality we should take into account and new properties of the networks that cannot be neglected anymore.

In this thesis we have studied new protocols for such resource-sharing problems as the backoff protocol, defense mechanisms against Denial-of-Service, fairness and confidentiality for users in overlay networks. For backoff protocol we present analysis of a general backoff scheme, where an optimization is applied to a general-view backoff function. It leads to an optimality condition for backoff protocols in both slot times and continuous time models. Additionally we present an extension for the backoff scheme in order to achieve fairness for the participants in an unfair environment, such as wireless signal strengths. Finally, for the backoff algorithm we sug-

gest a reputation scheme that deals with misbehaving nodes. For the next problem – denial-of-service attacks, we suggest two schemes that deal with the malicious behavior for two conditions: forged identities and unspoofed identities. For the first one we suggest a novel most-knocked-first-served algorithm, while for the latter we apply a reputation mechanism in order to restrict resource access for misbehaving nodes. Finally, we study the reputation scheme for the overlays and peer-to-peer networks, where resource is not placed on a common station, but spread across the network. The theoretical analysis suggests what behavior will be selected by the end station under such a reputation mechanism.

**Computing Reviews (1998) Categories and Subject Descriptors:**
C.2 Computer-Communication Networks
G.1.0 General
G.1.6 Optimization
G.1.8 Partial Differential Equations
G.3 Probability and Statistics
I.6 Simulation and Modeling

**General Terms:**
Thesis, Game Theory, Backoff Protocol, Denial-of-Service, resource sharing

**Additional Key Words and Phrases:**
Applying Game Theory to the problem of fair resource sharing

*To my lovely wife, Elena, and*

*to my dear parents, Sergey and*

*Natalia Lukyanenko.*

# Preface

This work is dedicated to a wonderful topic of resource sharing. As in the real life resources during networking are very limited. Say we have a pie, and everyone wants a piece from it. Of course, the first one who finds the pie can take the whole thing as a piece (of course if one will not be conscience-stricken). That is what we do not want to allow. All the algorithms which we discuss in this thesis are aiming at the fair division of the pie. In one case we give the same instruction to everyone to follow, i.e. robots slice a piece according to a rule. After that we invite people, who love pies, to choose freely the amount of a piece to slice. How much they will decide to cut from the pie? If after cutting one pie we will get another with same number of people, will they change the desired size of a slice? And further more, imagine that there are a lot of tables with a lot of pies on every. A person after slicing a pie on one table may go to another table. When I see a few known people at my table and a few newcomers, how would I decide on the size of a piece to slice? Tough question, and this thesis exactly about it.

The interesting topic of "pie slicing for robots" was initially suggested to me in University of Kuopio by Prof. Martti Penttonen. It was called Backoff protocol and I was asked to do it in batches for parallel computing. Prof. Evsey Morozov from my former Petrozavodsk State University helped me with theoretical background related to the Backoff protocols - Queueing Theory. With successful study of standard Backoff protocol, I failed to study batches: my short-term grant run out and I was looking for a job. That was the moment when I have found a job at the Helsinki Institute for Information Technology, Networking Research group, leaded by Prof. Andrei Gurtov. Prof. Andrei Gurtov introduced to me a lot of new topics including the Denial-of-Service attacks and Overlays, which in a way still the same "pie slicing" problem as before, but with a different restrictions. The fruitful discussions in the group with other researchers, including Andrey Khurri, Boris Nechaev, Dmitry Kuptsov, Miika Komu, Dr. Dmitry Korzun, Dr. Pekka Nikander, Tatiana Polishchuk, gave me a

lot of new ideas about the topics. Moreover, with collaboration with Russian Academy of Science, Karelian Research Center, Institute of Applied Mathematics and personally with Prof. Vladimir Mazalov, and Dr. Igor Falko, I have found an interesting field - the Game Theory. Now, changing robots to the people who slice pies by own well thought-out decisions with egoistic will to eat as much of the pie as possible we get absolutely new problems, which shows own interesting solutions. Later, with such wide research field, with a lot of problems and criteria for study, I was accepted as a PhD student at University of Helsinki with Prof. Jussi Kangasharju as my scientific advisor. In the University of Helsinki I was able to finally finish the "pie slicing" problem, which I was thinking of during last four years.

I would like to thank all the people without whom this work was hardly possible. First of all, I would like to thank Prof. Andrei Gurtov, who gave me the topics to study and supported the study during the working time. I would like to thank my scientific advisor Prof. Jussi Kangasharju who helped me to prepare the thesis. I would like to thank Prof. Evsey Morozov and Prof. Vladimir Mazalov who helped me with the theory behind the studying topics a lot. I would like to thank Prof. Martti Penttonen for taking me to the University of Kuopio, and guiding me during early time. Additionally, I would like to thank all the researchers whom I was collaborating with: Andrey Khurri, Boris Nechaev, Dmitry Kuptsov, Miika Komu, Dr. Dmitry Korzun, Dr. Pekka Nikander, Tatiana Polishchuk, Dr. Igor Falko. Undoubtedly, I would like to thank reviewers of my thesis for their reviews and comments Prof. Sabine Wittevrongel and Dr. Konstantin Avrachenkov and I would like to thank my future opponent Prof. Leon Petrosjan for the will to be my opponent. Of course, I would like to thank my good friends who were supporting me all the time first of all my Dell desktop PC and my Lenovo notebook, the latter is especially was helpful in the business trips.

Finally, I would like to thank people who were supporting me from behind. I would like to thank my wife Elena Lukyanenko for all the support, my father Sergei Lukyanenko, my mother Natalia Lukyanenko and brother Artem Lukyanenko. All of them helped me a lot during my studies and life.

# Contents

# List of Figures

# List of Tables

# Abbreviation list

| Abbrev. | Meaning |
|---|---|
| ACK | Acknowledgement |
| AIMD | Additive Increase Multiplicative Decrease |
| AP | Access Point |
| BEB | Binary Exponential Backoff |
| BEX | Base Exchange |
| bps | bits per second |
| BRITE | Boston university Representative Internet Topology gEnerator |
| CAN | Content Addressable Network |
| CIDR | Classless Inter-Domain Routing |
| CRISP | Cooperation via Randomized Inclination to Selfish/Greedy Play |
| CSMA/CD | Carrier Sense Multiple Access with Collision Detection |
| CSMA/CA | Carrier Sense Multiple Access with Collision Avoidance |
| CTS | Clear-to-Send |
| CW | Contention Window |
| CWmin | minimal Contention Window |
| CWmax | maximal Contention Window |
| DCF | Distributed Coordination Function |
| DDoS | Distributed Denial-of-Service |
| DNS | Domain Name System |
| DoS | Denial-of-Service |
| DHT | Distributed Hash Table |
| DIFS | DCF Interframe Space |
| DSSS | Direct-Sequence Spread Spectrum |
| EB | Exponential Backoff |
| ESP | Encapsulating Security Payload |
| FHSS | Frequency-Hopping Spread Spectrum |
| FIFO | First In First Out |
| FPE | Fixed-Point Equation |
| HCF | Hybrid Coordination Function |
| HIP | Host Identity Protocol |

| | |
|---|---|
| HIPL | Host Identity Protocol for Linux |
| HIT | Host Identity Tags |
| HTTP | Hypertext Transfer Protocol |
| i3 | Internet Indirection Infrastructure |
| ID | Identity |
| IEEE | Institute of Electrical and Electronics Engineers |
| IETF | Internet Engineering Task Force |
| IP | Internet Protocol |
| IPsec | Internet Protocol Security |
| IR | Infrared |
| Kbps | Kilobits per second |
| MAB | Matrix Adaptive Backoff |
| MAC | Media Access Control |
| MKFS | Most Knocked First Served |
| NS | Network Simulator |
| OSI model | Open System Interconnection Reference Model |
| P2P | Peer-to-Peer |
| PCF | Point Coordination Function |
| RMAB | Reverse Matrix Adaptive Backoff |
| RTS | Ready-to-Send |
| SIFS | Short Interframe Space |
| tBEB | truncated Binary Exponential Backoff |
| TCP | Transmission Control Protocol |

# Publication list

I A. Lukyanenko, A. Gurtov, *Performance analysis of general backoff protocols.* Journal of Communications Software and Systems, 4(1), March 2008, p. 13–21. ISSN: 1845-6421.

II A. Lukyanenko, I. Falko and A. Gurtov, *N-Player Game in a Multiple Access Channel is Selfish.* In Proceedings of Workshop on Networking Games and Management, June 2009, pp. 51–56.

III A. Lukyanenko, E. Morozov, A. Gurtov, *An adaptive backoff protocol with Markovian contention window control.* To appear in Journal of Statistical Planning and Inference, 2010.

IV A. Lukyanenko, A. Gurtov, *Towards behavioral control in multi-player network games.* In Proceedings of the First ICST international Conference on Game theory For Networks (Istanbul, Turkey, May 13 - 15, 2009). IEEE Press, Piscataway, NJ, pp. 683–690, May 2009, ISBN 978-1-4244-4176-1.

V A. Lukyanenko, V. Mazalov, A. Gurtov, I. Falko, *Playing Defense by Offense: Equilibrium in the DoS-attack problem.* 2010 IEEE Symposium on Computers and Communications (ISCC), pp.433–436, 22-25 June 2010. ISBN: 978-1-4244-7754-8.

VI A. Lukyanenko, A. Gurtov, V. Mazalov, *Applying a reputation metric in a two-player resource sharing game.* In Proceedings of The Third International Conference on Game Theory and Management (GTM'09), June 2009. ISBN: 978-5-9924-0052-6.

VII A. Gurtov, D. Korzun, A. Lukyanenko, P. Nikander. *Hi3: An efficient and secure networking architecture for mobile hosts.* Computer Communications, 31, 10 (Jun. 2008), pp. 2457–2467. ISSN: 0140-3664.

# Chapter 1

# Introduction

In this chapter we outline what problems this thesis is dedicated to and what area of research it covers. We also describe how we complete the research and what tools/equipment we are using for that purpose. After that we give a summary of the main contributions of our work and conclude it with the structure of the thesis.

## 1.1 Research area/problem

In this work we are studying the major problems and tasks of today's Internet architecture under the assumptions of tomorrow's evolution. A lot of similar work in the field is dedicated to future Internet architecture, and is based on the performance and processes that are taking place in it. These works try to predict future problems and suggests methods to evade the most significant threats in the future Internet. Some protocols and algorithms that were designed for the Internet decades ago are not as applicable for the new technology trends, as they are supposed to be. The main idea of this work is to add the existence of new problems to these protocols in order to evolve them to a new level.

The new threats that have arisen so far and that we are concerned about include congestion control for public resources, denial-of-service attacks and security. Everything contains questions on fairness among participants and defense against misbehavior of malicious parties. Technologies that in some sense are related to these problems are overlay algorithms (structured Distributed Hash Tables (DHTs) and unstructured), Transmission Control Protocol (TCP) congestion algorithms, medium access methods (including backoff algorithm), publish/subscribe techniques. All those are united by the terms of fair *multi-user resource-sharing problems*. Multi-user resource-

sharing problems are the threats and problems where a common resource is exploited and, thus, shared among many users in a concurrent way. In this thesis multi-user resource-sharing problems include:

i. Backoff protocol — a series of stations share a medium on the physical layer. Messages are primarily sent simply by broadcast of signals. Concurrent broadcasting corrupts the signals on the physical layer.

ii. Denial-of-service attack or congestion problem is a threat when one point of the Internet may refuse to process messages or processes them with low probability because of overload (exhaustion of resources).

iii. Overlays problems — problems, where resources are not placed into one point, instead they are spread among the network nodes, which are consumers and contributors at the same time.

Summarizing, we are studying a series of resource-sharing problems among many users in the scope of existing network technologies under assumptions of future networking trends.

## 1.2 Own approach

To study these different problems, we use well-known techniques of mathematical analysis: queueing theory and game theory. Those techniques evolved during different periods of time, and in a sense the game theory is a more novel tool for networking than the queueing theory. Thus, for defined problems we introduce two (in some cases non-intersecting) ways of thinking. For the first one, we assume that everyone in the network honestly wants to make the system work in the most optimal way (the system itself may define the optimality and the participants simply doing the instructions of the system — protocols). For the second one, we give the participants free will to decide (bounded only by our definition of the possible strategies) what strategy to follow. Based on that for the mentioned problems we give the two measures: optimality, in the sense of the whole system, and equilibrium, i.e., how a participant positions itself in the system. Positions we measure as negative, neutral and positive: the first hampers the system, the second does not affect it, and the last one helps it.

Based on the mathematical theory we suggest a series of extensions for existing protocols where they lead to some achievements, and introduce new protocols where it is needed. These protocols let us deal with problems

of shared resources among many users and the theory itself gives us the metrics to measure how these algorithms behave under the problems.

To support our theoretical analysis and suggested algorithms, we employ simulations, which show the behavior of the different protocols for the problems defined and studied. For this purpose, we use a set of simulation tools that are widely used in the field for model evaluations and study, including NS–3 for wireless models, OMNeT++ for Denial-of-service attacks and BRITE as a realistic topology generator tool. We, of course, are not able to fully support the theory with simulations in the sense of selection of all possible strategies for parties (mostly those are infinite), however, we do analyze the radical and commonly used ones. Additionally, we produce some implementations in real protocols.

## 1.3 Contributions

As was said, in this work we introduce a methodology and suggest a set of algorithms, which are based on criteria of optimality. We address a series of problems on different layers with these algorithms and methods and present required evaluation in order to support our results and conclusions.

For a backoff protocol, we produce the analysis in terms of queueing theory. We present a general form of backoff protocol, which is unifying many forms of backoff schemes to one generalization. For that generalization we solve the optimality problem in slot number and, later, introduce a model for optimization in continuous time space. As an extension to that model we introduce a class of Matrix Adaptive Backoff protocols (MAB) and reversed MAB (RMAB). As the new models differ from our generalized backoff scheme only in distribution over states, we present a theoretical analysis for searching such states. We assume that the new MAB and RMAB models will increase fairness among participants of a shared medium network, in the sense of amount of sent data, among the parties. The simulation results support our assumption. Finally, we study the backoff scheme in terms of selfish behavior of end stations. It is shown that the backoff problem has undesirable performance with selfish end stations involved, however, we show that there exist advanced schemes to neglect the selfishness, i.e. schemes to make end stations work optimally for the system.

For Denial-of-Service attacks we suggest two schemes to deal with misbehaving stations. The first scheme deals with spoofing identity attacks, i.e. an attack where attacking nodes send messages with forged IP addresses, thus the victim-server is not able to use the history of interactions

with this identity in order to punish or encourage it. For such situations we present a novel queueing policy — Most Knocked First Served (MKFS), instead of the classical First-In-First-Out (FIFO) policy. The new algorithm allows good nodes to fight for the right to be served on the server. This scheme is supported by numerical analysis and evaluation on NS-3 simulator, which shows excellent performance of the MKFS algorithm. For the second scheme, as the first algorithm deals with spoofing, we present an algorithm which deals with an attack, when unforged addresses are involved. The addresses are real and, hence, the server may start to deal with clients based on a history of interactions. For such an algorithm, we introduce a reputation scheme and reputation metric that allows to make end stations behave well in the scheme. Misbehaving nodes get server resources in a very restricted amount. For such a scheme we present theoretical analysis based on the principle maximum of Pontryagin and evaluate in an OMNeT++ simulator in order to conform the analysis. Both algorithms support extensions in the form of distributed algorithms. In continuation, we present the overlay use for a more secured way of the public address presentation (hi3 scheme). With that technique the server may separate parties whom it gives the information on its location, and, thus, make the denial-of-service attack harder to implement.

For overlays in general, we introduce a reputation scheme that allows stations to record the history of their interactions in one-dimensional variable — reputation (or probability). The scheme was evaluated in order to check that it makes the participants to behave when they want to gain the most. The end station behavior was studied as a control theory problem first, where only one player maximizes its profit (this analysis is valid under the assumption of many players' interaction) and secondly it was studied as a two-player game theory problem (this analysis is valid under the assumption of few player interactions). For both analyses an optimal trajectory for the players was constructed. However, the first one is given in a more explicit form compared to the second one. Additionally, we show how and under what conditions this scheme deals with free-riding or whitewashing problems of overlay algorithms.

As our contribution, we suggest these new algorithms and techniques under an assumption of future Internet use. These schemes imply problems of many user interactions, fairness among interacting users and additional end-user behavior control schemes for the multi-user resource-shared problem. We believe that these schemes will be asked for in next-generation networks, the ones that the Internet will evolve to.

The current thesis is based on a series of reviewed publications, where

the author was taking part as the first author in all of them, except one. The author contributed the most in these publications. An exception is the hi3 publication [42], where the author made a study of the i3 protocol and implemented the hi3 scheme for a HIPL realization of HIP.

## 1.4 Structure of the Thesis

This thesis overview is organized as follows. In Chapter 2, we will give the necessary background for the thesis with more detailed discussion on the technologies and techniques we are using. Chapter 3 contains the summary of our published papers with a list of the main results and discussions on the results achieved and the significance of them. Lastly, in Chapter 4 we discuss future work and conclude the overview of the thesis. The overview is followed by the original publications.

# Chapter 2

# Literature review and background

In this chapter we will give the necessary background for the field of study and overview of the main literature related to that background. First of all, we are going to formulate the main threats and problems of the current Internet and networking communications, which we are going to deal with. After that, we will mention various aspects and theories, based on which we are looking on the problems, these include Markov Chain Theory and Game Theory. We introduce methodologies based on these aspects and used to solve formulated network-related problems. Then, we take a deeper look at the backoff protocol that is used in medium-access control schemes, with an overview of the history of research and survey on gained results so far. After that we list related technologies that are parts of future Internet topics, and mention how they relate to our research. Lastly, we list network simulation and evaluation tools which are widely used for networking analysis.

## 2.1 Internet and network communication problems

During recent decades Internet grew from a small educational university network to a global communication network that connects all parts of the world into one informational domain [64]. It became a part and parcel of day-to-day life, and so did the accompanying problems. At the beginning, during the development stage some of the threats were not thought of at all and some of them were not considered to be of great importance to the technology. Now, as time has shown most of those problems need solutions in today's Internet and are reconsidered as of great importance to the Internet.

As we were able to observe, during previous years the growth of the

technology in the number of users and amount of resources leads to public appearance of predicted, but hidden problems or even leads to new, unforeseen problems popping up. As an example one can see (i) transition IPv4 from class to classless CIDR [34], when the number of networks were not enough to satisfy all demand, (ii) TCP congestion collapse [49] occurred in 1986, when throughput between two nodes dropped from 32 Kbps down to 40 bps, (iii) Denial-of-Service attack [81], when a malicious user artificially creates congestion at some end-point and, hence, produces denial-of-service to process requests from benign users at the node, and so on. The history of the Internet has a full set of examples that shows how the growth of the technology produces new threats and shows new sides of the solved ones.

For all mentioned problems, and for problems that have not been mentioned, but are found in today's Internet, a set of all kinds of solutions were introduced. Some of them fully solve the problems (as far as context will not change again, as we saw in history), some of them partly solve the problems, meaning that the solutions are hardly applicable or are not easy to deploy. The examples give us to understand that any solution of the problems is fully dependent on the context (i.e., number of users, resources and technology). Thus, we study the problems from various angles to make the study of them multifaceted and, hence, with more predictable properties for the future.

Here, we are going to focus on a subset of today's Internet and communication networking problems, namely *multi-user resource sharing problems*, which we are addressing in the following chapters.

### 2.1.1 Congestion control and multiple-access channels

The term congestion control is mainly applicable for two sets of schemes: one is TCP congestion control and the other is a mechanism to deal with collisions on a shared medium — the Medium Access Control (MAC) sublayer of the OSI model. Historically, those were developed in parallel; the first one came with the development of TCP, when the threat of congestion collapse was considered serious [30, 49, 82]. The second one came from development of the ALOHA protocol [12, 14], and then the Ethernet protocol [79].

In spite of the fact that these schemes are dealing with a common in spirit congestion problem, the actual technology has important distinctions and dictates different solutions. In TCP, packets from one source are sent in bunches and congestion there means that some of the bunch will be lost, while congestion in medium means that every station sends only one frame at a time and all (or most) sent frames in the network collide (a frame

is a message of the MAC layer). Collision in medium is on the physical layer meaning physical damage of part of the signal. On the other hand, TCP collisions are mostly happening when some intermediate node has an overflown buffer, then the node deliberately discards the packets. One can find more details on comparison of TCP, for example, in [30].

Our main focus during the work will be on MAC schemes to deal with collisions implemented in the following two protocols IEEE 802.11 or Wi-Fi [47] and IEEE 802.3 or Ethernet [46]. More precisely we are going to study the core of these standards — *the backoff protocol*, that gives the core principle on how to deal with and avoid collisions.

### 2.1.2 Selfishness

In the previous section we talked about congestion control in terms of protocol work. But what will happen if the users start to select or adjust the protocols based on their own preferences? If one is free to select which congestion control to adopt then in some cases the aggressive congestion control (or neglect of any congestion control, just selfish bulks of data) will be selected.

Selfishness becomes a more and more significant aspect to take into account, when a new protocol is designed. It was shown by recent development of the network communication field that selfishness of nodes in some cases cannot be ignored. The adaptation of Game Theory for this problem was widely introduced recently. The selfishness of nodes itself can be formulated using some notions of the theory. The formulation and the application of the game theory for resource-sharing problems will be made in the following sections.

In general, without any reference to the theory, an optimal protocol is designed based on the assumption that all users of the system work accordingly. However, in practice it is hard to restrict users to choose only a necessary strategy over others, and not to deviate from it. Mainly these assumptions are produced by the hardness of the end user to change the actual protocol, although some adjustments are in any case available for the end user. The adjustments that the user may produce, can affect the protocol greatly or can be inconspicuous for the system. This fully depends on the protocol, and a designer of the protocol needs to take into account the trade-offs of system optimality and individual selfishness.

### 2.1.3 Denial-of-service attack

A Denial-of-Service attack (DoS) or Distributed Denial-of-Service attack (DDoS) [81] becomes a more dangerous threat with development of the Internet as a political and economical tool. Internet websites (or servers in general) are used as fundamental business platforms, for example as Internet shops [18] or auctions [28]; and it is also used for delivery of political news or mass media information. All these are potential targets for malicious deeds both as for mercenary ends as well as for simple fun. In general, DoS is an attack on an Internet node, the main purpose of which is to make the resource unaccessible for benign users making it perform as if it is heavily congested or out-of-service. In this sense the DoS attack resembles the congestion of a network: if the node has the same amount of benign users it will be congested in the same way. However, the crowd is created on the server artificially and, hence, it does not bring profit for the end system, it only increases maintenance costs and affects the satisfaction of service for the clients.

The DoS attack becomes feasible mostly because of the original concept of the Internet, main principles of which are (i) delivery of packets is based on the best effort basis and (ii) there is no global control on the operation level [64]. Without check for the original source address any node can insert any packet in the network and it will be delivered to the destination node without any global validation for the source address and for legitimacy of the traffic (in the following sections we will discuss publishing/subscription techniques). Based on it, the IP address spoofing becomes one of the serious tools for performing DoS attack on a server. Recent trends show new techniques to perform defense mechanisms. Mainly the defense is based on

(i) source address validation schemes, e.g. [66, 105],

(ii) classification of the traffic, e.g. [45],

(iii) pushback mechanisms, when a server informs the closest routers about packets which should be dropped, after that the routers inform second-level routers, and so on, e.g. [48],

(iv) traceback-marking mechanisms, where routers put some mark into upstream packets and, thus, the server after some period of time receiving these packets may construct an attacking graph, e.g. [40,62,83,99] for some discussion on ineffectiveness see [102],

(v) filtering of packets based on hop-counting [50] or by the routers' defense line [106, 107], sometimes with the help of benign clients [103].

However, those techniques are still in the deployment or pending stage and they do not seem to be deployed in the near future. Additionally to spoofing IP, botnets produce a real threat; the sizes of some found botnets are huge, more than 10 million poisoned computers, [51] and even without spoofing they may produce heavy attacks on the end-nodes. All these and some more make the DDoS attacks a threat worth attention and study.

### 2.1.4 Confidentiality

The last problem we are going to address is confidentiality. Confidentiality is an ambiguous term itself. We define it as a restriction on the network resource availability and access only for authorized users, whom we checked and gave the right to access the resource. In other words for every piece of data we divide all users into two groups: ones who have access to the data and ones who do not. Even the request to access the resource can be a subject for confidentiality.

As we have seen in the previous section, the DDoS attack is in some sense achievable because everyone can freely see what address the destination host has and send a packet directly to it. Hence, there is no confidentiality of the resource's location, nor the confidentiality of packets' delivery. Another example is SPAM [27] when anyone can freely send an advertisement by e-mail or instant messenger, for example. In that case there is no confidentiality of the users' IDs, when there is no method to separate ID access for authorized and unauthorized users.

In the following sections we are going to talk about Host Identity Protocol (HIP), overlays and publish/subscribe techniques that give a cause to some rethinking of the design of the Internet in terms of confidentiality and provides the address/location splitting.

## 2.2 Markov chains and Queueing theory

In this section we are going to give the basic definitions and models starting from "what is the probability space" and ending with the "ergodicity for Markov chains". The same definitions may be found in any book on the theory of probability. The main theorems on potential functions and ergodicity are most important for the work and those we repeat with our own interpretation from Meyn's and Tweedie's excellent book [80]. For the Queueing Theory we will also outline only brief principles and definitions because it is a vast topic, for details on different models and analysis principles see the book of Asmussen [19].

### 2.2.1 General formulations

Let us have some *probability space* $\langle \Omega, \mathfrak{F}, \mathrm{P} \rangle$, where sample space $\Omega$ is a non-empty set of elementary events, $\mathfrak{F}$ — is a $\sigma$-algebra on $\Omega$, i.e., is a collection of all combinations of events from $\Omega$, and, finally, P — is a probabilistic measure on $\sigma$-algebra $\mathfrak{F}$, which measures all possible events on $[0, 1]$ space. The sample space $\Omega$ is fully based on an experiment, and outcomes of the experiment can be expressed using one element of $\Omega$. $\mathfrak{F}$ is $\sigma$-algebra on $\Omega$ if

(a) $\mathfrak{F}$ contains the whole sample space: $\Omega \in \mathfrak{F}$.

(b) $\mathfrak{F}$ is closed under complements: If $\mathrm{A} \in \mathfrak{F}$, then $\overline{\mathrm{A}} \in \mathfrak{F}$, where $\overline{\mathrm{A}} = \Omega \setminus \mathrm{A}$.

(c) $\mathfrak{F}$ is closed under countable unions: If $\mathrm{A}_n \in \mathfrak{F}$ for all $n$, then $\cup_{n=1}^{\infty} \mathrm{A}_n \in \mathfrak{F}$ and $\cap_{n=1}^{\infty} \mathrm{A}_n \in \mathfrak{F}$.

Finally, mapping $\mathrm{P} : \mathfrak{F} \to [0, 1]$ is a probability measure on $\mathfrak{F}$ if

(a) $\mathrm{P}(\mathrm{A}) \geq 0$ for any $\mathrm{A} \in \mathfrak{F}$.

(b) $\mathrm{P}(\Omega) = 1$.

(c) If a set of events $\mathrm{A}_n \in \mathfrak{F}$ is pairwise disjoint, i.e., $\mathrm{A}_i \bigcap \mathrm{A}_j = \varnothing$ for any $i \neq j$, then

$$\mathrm{P}\left(\bigcup_{n=1}^{\infty} \mathrm{A}_n\right) = \sum_{n=1}^{\infty} \mathrm{P}(\mathrm{A}_n).$$

$\xi$ is called *a random variable* if it is a measurable function that maps sample space on real number axis $\xi : \Omega \to \mathbb{R}$, i.e., if preimage $\xi^{-1}(\mathrm{B}) = \{\omega : \xi(\omega) \in \mathrm{B}\}$ for any Borel set $\mathrm{B} \in \mathfrak{B}$ is an element of $\sigma$-algebra $\mathfrak{F}$. We say that $\xi$ is doing measurable mapping of $\langle \Omega, \mathfrak{F} \rangle$ onto $\langle \mathrm{B}, \mathfrak{B} \rangle$. For random variable $\xi$ *probability measure* is defined as $\mathrm{P}_\xi(\mathrm{B}) = \mathrm{P}(\xi \in \mathrm{B})$ and *distribution function* is defined as $F_\xi(x) = \mathrm{P}(\xi < x)$, i.e., when $\mathrm{B} = (-\infty, x)$.

For random variable $\xi$ we define *expectation* and *dispersion (or variance)* as

$$\mathrm{E}\xi = \int_\Omega \xi(\omega)\mathrm{P}(\mathrm{d}\omega) = \int_\mathrm{B} x\mathrm{P}(\mathrm{d}x) = \int_\mathrm{B} x\mathrm{dF}(x),$$

and

$$\mathrm{D}\xi = \mathrm{E}(\xi - \mathrm{E}\xi)^2,$$

where the integration operation is Lebesgue integral.

Events $A_1, A_2, \ldots A_n \in \mathfrak{F}$ for any $n \geq 2$ are *independent* if and only if $\mathrm{P}(A_1 \cap A_2 \cap \ldots A_n) = \mathrm{P}(A_1)\mathrm{P}(A_2)\ldots\mathrm{P}(A_n)$. In the same way random variables $X_0, X_1, \ldots, X_n$ are called *independent random variables* if and

only if events $\{\xi_1 \in \mathrm{B}_1\}$, $\{\xi_2 \in \mathrm{B}_2\}, \ldots, \{\xi_n \in \mathrm{B}_n\}$ are independent events for any Borel sets $\mathrm{B}_1, \mathrm{B}_2, \ldots, \mathrm{B}_n$, i.e.

$$\mathrm{P}(\xi_1 \in \mathrm{B}_1, \xi_2 \in \mathrm{B}_2, \ldots, \xi_n \in \mathrm{B}_n) = \mathrm{P}(\xi_1 \in \mathrm{B}_1)\mathrm{P}(\xi_2 \in \mathrm{B}_2) \ldots \mathrm{P}(\xi_n \in \mathrm{B}_n).$$

We define *conditional probability for events* $A, B \in \mathfrak{F}$ as $\mathrm{P}(A|B) = \frac{\mathrm{P}(A \cap B)}{\mathrm{P}(B)}$. Analogously, we define *conditional probability for random variables* $\xi, \eta$ as $\mathrm{P}(\xi|\eta) = \frac{\mathrm{P}(\xi,\eta)}{\mathrm{P}(\xi)}$.

A random variable $\xi$ is called *discrete* if for a finite or countable set of points $\{x_n\}_{n\geq 0}$ the following holds

$$\sum_{n\geq 0} \mathrm{P}_\xi(x_n) = 1,$$

if it is not possible then a random variable is called continuous or mixed. Any point from the set $\{x_n\}_{n\geq 0}$ above is called *a state*, while the whole set of points under the equation is called *the state space*. We will refer to it as X.

### 2.2.2 Markov chains and ergodicity

The following construction will help us to study trends of random variables with discrete time. Let us have a sequence of discrete random variables $\{X_t\}_{t\geq 0}^{\infty}$, the sequence forms a Markov chain if

$$\mathrm{P}(X_t = j|X_{t-1} = i, X_{t-2} = k_{t-2}, \ldots, X_1 = k_1, X_0 = k_0) = \mathrm{P}(X_t = j|X_{t-1} = i) \equiv p_{ij}(t), \quad (2.1)$$

for $t \geq 1$, sometimes it is interpreted as time or step. Equation (2.1) is called *a Markov property*, it indicates that the probability to be in state $j$ at the moment $t$ depends on the previous state (at the moment $t-1$), and independent from states before previous $(t-2, \ldots, 0)$ with given previous state (at the moment $t-1$).

Values of $i, j$ are called states and all possible states form the state space (as we defined above), however, for convenience, we will enumerate all states starting from 1 (i.e., $\mathrm{X} = \{1, 2, \ldots, n\}$), as the random variables are discrete, the number of states is countable or finite and such enumeration is allowed. For simplicity we define conditional probabilities 2.1 at time $t$ as $p_{ij}(t)$. The notation $p_{ij}(t)$ is also called a transition probability from state $i$ to state $j$ for $t$ timesteps. As one can see it is possible to represent them as a graph with vertices as states $i, j$ and weighted edges as $p_{ij}(t)$.

For any state we define the *initial distribution* to be in that state $\pi_i^0 = \mathrm{P}(X_0 = i)$, where $\sum_i \pi_i = 1$, or in vector form $\pi^0 = \mathrm{P}(X_0)$. The transition probabilities can be viewed as a matrix (countable or finite) $P(t)$, where an element in row $i$ column $j$ of matrix $P(t)$ is equal to $p_{ij}(t)$. If we have initial probability $\pi^0$ and the transition probabilities $P(t)$ then the probability of a random variable at any moment of time to be in some state can be found using a matrix form

$$\mathrm{P}(X_t) = \pi^0 \prod_{i=1}^{t} P(i). \tag{2.2}$$

If transition probabilities do not depend on time $t$ then the Markov chain is called homogeneous Markov chain, $P = P(i)$, for any $i$. In that case equation (2.2) may be written as

$$\mathrm{P}(X_t) = \pi^0 P^t(i). \tag{2.3}$$

We follow the definitions from Meyn's and Tweedie's book [80] for irreducibility, aperiodicity and ergodicity of Markov chains.

**Definition** (Irreducibility)**.** *Markov chain $\{X_t\}_{t\geq 0}^{\infty}$ is called irreducible Markov chain if for any $i, j \in \mathrm{X}$ exists $t$ such that $p_{ij}(t) > 0$.*

**Definition** (Aperiodicity)**.** *An irreducible Markov chain $\{X_t\}_{t\geq 0}^{\infty}$ on a countable space $\mathrm{X}$ is called aperiodic, if $d(x) = 1$, $i \in \mathrm{X}$, where $d(x) = \gcd\{t \geq 1 : p_{ii}(t) > 0\}$.*

**Definition** (Occupation time)**.** *For any set $\mathrm{A} \in \mathfrak{B}$, the occupation time $\eta_{\mathrm{A}}$ is the number of visits by $\{X_t\}_{t\geq 0}^{\infty}$ to $\mathrm{A}$ after time zero, and is given by*

$$\eta_{\mathrm{A}} \equiv \sum_{t=1}^{\infty} \mathbb{I}\{X_t \in \mathrm{A}\},$$

*where indicator function $\mathbb{I}$ is defined as following*

$$\mathbb{I}(A) = \begin{cases} 1, \ \textit{if } A \\ 0, \textit{otherwise.} \end{cases}$$

**Definition** (First return and hitting time)**.** *For any set $\mathrm{A} \in \mathfrak{B}$, the variables*

$$\tau_{\mathrm{A}} \equiv \min\{t \geq 1 : X_t \in A, \ X_0 \in A\} \tag{2.4}$$

$$\sigma_{\mathrm{A}} \equiv \min\{t \geq 0 : X_t \in A\} \tag{2.5}$$

*are called the first return and hitting times on $\mathrm{A}$, respectively.*

**Definition** (Recurrence). *The state $i$ is called a recurrent state if* $\mathrm{E}[\eta_i] = \infty$*, i.e., the expected number of visits to the state is infinite. If every state is recurrent, the chain is called recurrent.*

**Definition** (Positive recurrence). *The state $i$ is called a positive recurrent state if* $\mathrm{E}[\tau_i] < \infty$*, i.e. the expected return time to the state is finite. If every state is positive recurrent, the chain is called positive recurrent.*

**Definition** (Transience). *The state $i$ is called a transient state if* $\mathrm{E}[\eta_i] < \infty$*, i.e. the expected number of returns to the state is finite. If every state is transient, the chain is called transient.*

**Definition** (Invariance). *For homogeneous Markov chain $\{X_t\}_{t\geq 0}^{\infty}$ a probability distribution $\pi$ is called invariant if $\pi = \pi P$, i.e. one step of transition matrix does not change the distribution.*

**Definition** (Positivity). *Homogeneous Markov chain $\{X_t\}_{t\geq 0}^{\infty}$ is called a positive chain if it is irreducible and admits an invariant probability distribution $\pi$.*

**Definition** (Ergodicity). *Markov chain $\{X_t\}_{t\geq 0}^{\infty}$ is called an ergodic chain if it is positive recurrent and aperiodic (same for a single state).*

**Theorem 1** (Theorem 8.3.4 from [80]). *If $\{X_t\}_{t\geq 0}^{\infty}$ is irreducible, then it is either recurrent or transient.*

Next we will give Theorem 14.0.1 from [80] on potential function.

**Theorem 2** (Potential function theorem). *Suppose that the chain $\{X_t\}_{t\geq 0}^{\infty}$ is irreducible and aperiodic, and let $f \geq 1$ be a function on* X*. Then the following conditions are equivalent:*

1. *The chain is positive recurrent with invariant probability measure $\pi$ and*
$$\pi(f) \equiv \int \pi(dx) f(x) < \infty.$$
2. *There exists some finite set $C \in \mathfrak{B}$ such that*
$$\sup_{x\in C} E_x \left[ \sum_{n=0}^{\tau_C - 1} f(X_n) \right] < \infty.$$

The last theorem was very useful for analysis of the backoff protocol by Hastad et al. [44]. In this thesis we will often refer to the stability of Markov chains, in many cases it means that the corresponding chain is ergodic.

### 2.2.3 Markov process and embedded Markov chains

Previously we defined discrete time Markov chains (as we used a countable number of random variables). It is also possible to define *continuous-time Markov chain* or *Markov process with discrete state space* equivalently to the previous definition. Let us have a random process $\{X(t) = X(t, \omega)\}_{t \geq 0}$ ($t$ now continuous) for some probability space $\langle \Omega, \mathfrak{F}, \mathrm{P} \rangle$, then $X(t)$ is a Markov process if for any time moments $t_0 < t_2 < \cdots < t_n \leq t$

$$\mathrm{P}(X(t) = x(t) | X(t_n) = x(t_n), X(t_{n-1}) = x(t_{n-1}), \ldots, X(t_0) = x(t_0)) = \mathrm{P}(X(t) = x(t) | X(t_n) = x(t_n)). \tag{2.6}$$

In spite of the fact that continuous-time Markov chains are seen very often (for example, calls on a telephone station are sometimes assumed to be a Poisson process), it is easier to work with discrete-time Markov chains. If we define a new Markov chain based only on the moments when a station changes the states, then we will be able to get a discrete-time Markov chain associated with the Markov process. The new Markov chain is called *an embedded Markov chain*. If the new chain is ergodic then we are able to find its stationary distribution, which in turn says about the probability to be in each state of the original Markov process. Multiplying them by average time we will get the average time to stay in each state. It is used, for example, in queueing theory when the Markov process is studied using Markov chains based on the number of messages in the queue.

### 2.2.4 Queueing theory and average service time

In computer science, queueing theory makes an important mathematical basis for analyzing queueing systems. By queueing systems we mean systems that have hierarchically (and sometimes cyclically) connected processes, the work of one is dependent on a set of others. Any of such processes may need to wait for the previous one to finish its job before receiving data to process or the next one before sending new data to it. The connections between processes often have the forms of queues (first in first out) and a piece of data that is stored in these queues is a message. Any process has a service procedure and the process starts to serve whenever it has at least one message in the input queue, it also may produce an output stream of served messages. Based on that, in queueing theory, an important role is played by the number of messages in the queues and the time that they spend in the queue.

In classical queueing theory (for example, [55]) the input and the service processes are often defined as Poisson processes with parameters $\lambda$ and

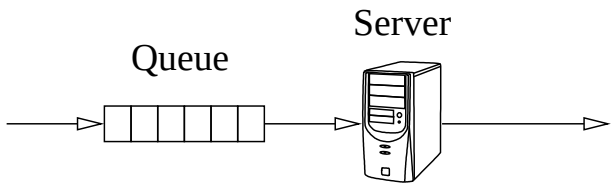



Figure 2.1: Basic queueing system with input process (parameter $\lambda$) and output process (parameter $\mu$). Server policy is to serve only one message at a time.

$\mu$, respectively. Let random variable $\eta_t$ show the number of events that happened up to time $t$, we say that $\eta_t$ is a Poisson process with parameter $\lambda$ if the following conditions hold:

(a) $\mathrm{P}(\eta_{t+h} - \eta_t = 0) = 1 - \lambda h + o(h)$,

(b) $\mathrm{P}(\eta_{t+h} - \eta_t = 1) = \lambda h + o(h)$,

(c) $\mathrm{P}(\eta_{t+h} - \eta_t \geq 2) = o(h)$,

where $o(h)$ - is little-o of $h$. The definition gives us $\mathrm{P}(\eta_t = x) = \frac{e^{-\lambda t}(\lambda t)^x}{x!}$. The Poisson distribution also have a *memoryless property*:

$$\mathrm{P}(\eta_t > x + y | \eta_t > x) = \mathrm{P}(\eta_t > y).$$

Based on the input and service processes an embedded Markov chain can be used - the number of messages in the queue. In the simplest case, as shown in Figure 2.1, $\lambda < \mu$ is a sufficient condition for stability (that the number of messages in the queue is finite) for the Markov chain. For such a process waiting time can also be introduced, which is the time that messages which came at random time wait in the input queue before being served. Waiting time depends on the size of the current queue and sometimes is called *unfinished work*. The process itself can be formulated in terms of renewal and regenerative processes (for details on the processes, see [80]). Average waiting time is fully defined by the input random process and service random process. Stability in that case is achieved if the average waiting time is less than infinity (in the simplest case and some more complicated it is again achieved if $\lambda < \mu$ is fulfilled). This value is of great interest for us, because reduction of the average service time makes the system perform in a more optimal way.

## 2.3 Game Theory for communication problems

In this section we are going to talk about game theory and its application to communication problems. First of all we will give the main definitions of the theory and will give one important example of game theory in matrix form. However, we are not interested in matrix games and in the following section we will give necessary background for differential games. Differential and iterated games are very useful when the profit of a player heavily depends on the sequence of strategies the player chooses in time. Here we are going to formulate briefly differential games and methods to solve them as we are formulating some communication problems in a differential form in the next chapter. More information on these kinds of games can be found in [20,33]. Differential games in some sense are a generalization of control theory and maximum principle of Pontryagin [84]. This famous result from the control theory is utilized a lot in the differential games. Lastly, we will show how game theory is used in network communications and what results it shows in the field.

### 2.3.1 General formulation

We define a game as a combination of three sets (see [33]):

(i) Set of players $i \in \mathfrak{P}$, which for simplicity are labeled by a set $\{1, 2, \ldots, I\}$.

(ii) Pure-strategy space $S_i$ for each player $i$, where $s_i \in S_i$ is the strategies of the player $i$.

(iii) Payoff functions $u_i$ which gives a player the income based on the selected strategies ($u_i(s)$, $s = (s_1, s_2, \ldots, s_I)$).

I.e., a game in a normal form is a combination of these $\Gamma = \langle \mathfrak{P}, \mathrm{S}, \mathfrak{I} \rangle$, where $\mathfrak{P} = \{1, 2, \ldots, m\}$, $\mathrm{S} = \{S_1, S_2, \ldots, S_m\}$ and $\mathfrak{I} = \{u_1, u_2, \ldots, u_m\}$. Every player receives income based on what strategy that player chooses and what pure strategies are chosen by others. The chosen element of $S_i$ for player $i$ ($s_i \in S_i$) is called *a pure strategy*. We also define *mixed strategies* $\sigma_i$ as the probability distribution over pure strategies (it may be discrete or a continuous depending on the space), i.e., it is some probability measure on $S_i$ for player $i$. By $\sigma_{-i}$ we define strategies of all players, except player $i$, i.e., $\sigma_{-i} = \{\sigma_1, \sigma_2, \ldots, \sigma_{i-1}, \sigma_{i+1}, \ldots \sigma_I\}$.

**Definition** (Nash equilibrium in non-cooperative game)**.** *Let us have a game* $\Gamma = \langle \mathfrak{P}, \mathrm{S}, \mathfrak{I} \rangle$ *then a mixed-strategy profile* $\sigma_i^*$ *is a Nash equilibrium if,*

| | C | D |
|---|---|---|
| C | 1/1 | 10/0 |
| D | 0/10 | 6/6 |

Table 2.1: Prisoner's Dilemma outcome table. The strategy space is to cooperate (C) or defect (D). Rows show strategies of the first player, columns of the second.

*for all players i,*

$$u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(s_i, \sigma_{-i}^*) \text{ for all } s_i \in S_i.$$

One of the important examples of games is called *Prisoner's Dilemma.* Two players ("prisoners") have two strategies: cooperate (C) with another player, or defect (D), testify against another. They cannot communicate with each other. The matrix of the games is shown in Table 2.1. If both cooperate with each other they will receive minimal punishment, if one defects, another remain with cooperating strategy, the one who defects will be set free, while the other will receive severe penalty. If both players defect, then both of them will get "middle" punishment. For both players to defect is the Nash equilibrium in form of pure strategies: (D,D), but it is clearly not optimal in sum $u_1(D, D) + u_2(D, D) < u_1(C, C) + u_2(C, C)$. However, the game changes when it has iterated form. Based on the behavior of another, a prisoner may punish the opponent for defecting, while cooperating when another player cooperates. The formulated game in iterated form has a lot of applications in network communications.

### 2.3.2 Differential games

In order to define a game in differential form, as previously, let us have $I$ players and $I$ pure strategies – "actions" or "controls" $a(t) = \{a_1(t), \ldots, a_I(t)\}$, where $a_i(t) \in S_i$ for every $t$. Additionally for every player we define "coordinate" or "trajectory" – $x(t) = \{x_1(t), \ldots, x_n(t)\}$. These controls and trajectories are connected into a dynamic system $\dot{x}_i(t) = f_i(t, x(t), a(t))$ with initial condition $x(0) = x_0$. A player may variate only controls $a(t)$ (possible variations are strategy space), while variables $x(t)$ are fully defined by the system above, thus controls are selected from the set $S_i \in \mathbb{R}^{m_i}$. The trajectories $x_i(t)$ take all values of the dynamic system without any restrictions (in case such restrictions are needed a more complicated form of the formulation and solution available in the literature, for control theory see [84]). Based on the current coordinate and action a player receives

gain $u_i = \int_0^T g_i(t, x(t), a(t))dt + q(T, x(T))$ $i \in \{1, \ldots, I\}$, which sums up intermediate profit $g_i$ depending on the trajectories and actions, and final profit $q$ that depends on the final position. We also predefined the time required for game $T$.

For such systems Hamiltonians are defined using costate variables $\lambda_i = \{\lambda_{i1}, \ldots, \lambda_{in}\}$ for all $i \in \{1, \ldots, I\}$:

$$H_i(t, \lambda, x, u) \equiv g_i(t, x(t), a(t)) + \sum_j \lambda_{ij}(t) f_j(t, x(t), a(t)).$$

Based on which co-state variables and optimal controls (i.e., Nash equilibrium) are defined by the following system:

$$\left.\begin{array}{rcl} \dot{\lambda_{ij}}(t) & = & -\frac{\partial H_i(t,\lambda,x^*,a^*)}{\partial x_j}, \\ \lambda_{ij}(T) & = & \frac{\partial q_i(T,x^*)}{\partial x_j}, \\ a_i^*(t) & = & \arg\max_{a_i} H_i(t, \lambda, x^*, a). \end{array}\right\}$$

The previous results follow from the maximum principle of Pontryagin (in some literature *the minimum principle*) and in the case when $I = 1$ it is a problem of control theory for which the maximum principle was developed. The above is applied under a set of restrictions on functions: the functions $f$,$g$ and $q$ are continuously differentiable in $x$ and continuous in $t$ and $a$.

### 2.3.3 Application to the networking

Recently application of the Game Theory to the networking analysis erupted a number of results on the known protocols. For example, analysis of TCP games, where the players are allowed to control parameters $\alpha, \beta$ – the additive increase value and multiplicative decrease of AIMD scheme [15] shows that corresponding Nash equilibrium in many cases is undesirable. Another work [35] shows that the current resource-sharing mechanisms in Internet either encourage selfish behavior, or are oblivious to it. Analysis in [108] supports previous results. Everything is based on the assumption of cooperative behavior of the network nodes. However, selfish routing under some models and conditions [85, 86, 93] still may have close-to-optimal behavior. These analyses show that in most cases the Internet is vulnerable to selfish behavior of end-nodes. Some algorithms, though, propose methods to achieve fairness in such environments [110].

Another trend in modern network communications is to study incentives that drive malicious users additionally with strategies and objectives,

for example, in DDoS attacks [67]. Based on this knowledge game-based defense mechanisms may be constructed, e.g., [75]. However, in many cases the suggested algorithms are hard to implement as was discussed previously.

Another natural application of the Game Theory came to the overlay system. It is most natural as in overlays the ordinary nodes provide a control plane themselves. Feldman et al. [32] consider P2P systems, where users provide service to each other, under the free-riding and whitewashing set of strategies. In the work, they construct a model which adds penalties to the users who free-ride in the system. Additionally, some restrictions are added for the newcomers in order to prevent whitewashing. Under such restrictions the model shows higher performance compared to a model without incentives. Another work of Mazalov et al. on P2P user communication [77] is based on the auction principle: two players make bids on how much they contribute, the winner is the one closest to some value. The latter analysis again shows some non-trivial Nash equilibrium, which the user may choose.

## 2.4 Backoff protocol

In this section we are going to describe the backoff protocol that is used for congestion resolutions on the MAC layer in more details. The first of the next chapter will be dedicated to optimization problems of the protocol, thus, we support this analysis with previous results and make the comparisons of models and methods for analyzing.

### 2.4.1 Creation and design principle

The backoff scheme was introduced in ALOHAnet [12,59]. It was a network that connects a set of university facilities on Hawaii to a central station and later to the continental part of the USA through a satellite [13, 14]. For example see Figure 2.2. There was introduced a constant backoff protocol (it was referred to as unslotted ALOHA), initially for communication between terminals and then for satellite. For the ground users, it was assumed that many of them are silent most of the time, while some active users perform in a bursty manner, hence it was decided to give possibility for a user to gain as much capacity of the channel as possible, instead of the equal division of the channel among the users. In the satellite case, the antenna on the ground and the satellite used the same medium — a common radio frequency, hence, collisions can happen when both stations send signals simultaneously. As the connection had very high latency, the cost of data loss was huge. After a collision stations cannot start resending

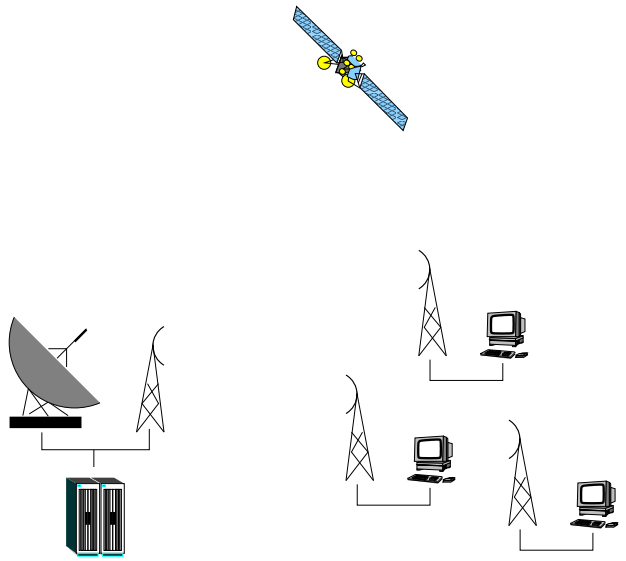


Figure 2.2: ALOHA network design principle. The main station (on the left), communicates to remote university facilities (on the right) and to the satellite through wireless connections.

data immediately or after some deterministic time period, as it will lead to a collision once more with high probability. It has a prefixed number of timeslots $T$ from which stations select one random timeslot and countdown to that time slot till the moment they try to resend data. As the scheme introduces random numbers the probability of collision reduces and depends fully on $T$ and the number of active users.

Later, in 1973, Bob Metcalfe and David Boggs used the same idea of backoff protocol to develop the Ethernet [79], a wired network, where stations can be easily tapped into to start communications (see Figure 2.3). It did not utilize the constant backoff protocol anymore. It was named binary exponential backoff (BEB) protocol or truncated binary exponential backoff protocol. While in the original backoff that was used in ALOHA the time slot had been chosen uniformly from constant value $T$, now the value was sliding and was dependent on a prehistory of the current situation, i.e. the number of uninterrupted collisions.

Let us define the number of collisions that happen in a row as $i$ – we name it *backoff counter*, then the protocol states the following. If $i \geq 16$ discard the message, i.e., say to upper layers that an error has happened. If $i \leq 10$ then set $T_i = 2^i$, otherwise $(10 < i < 16)$ set $T_i = 1024$. Hence,

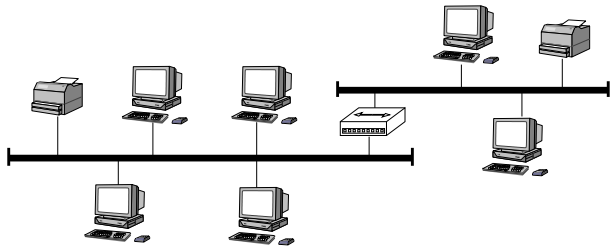


Figure 2.3: An example of an Ethernet network consisting of two segments connected by a hub. All data that a station transmits to the network propagate from one segment of the network to another through this hub.

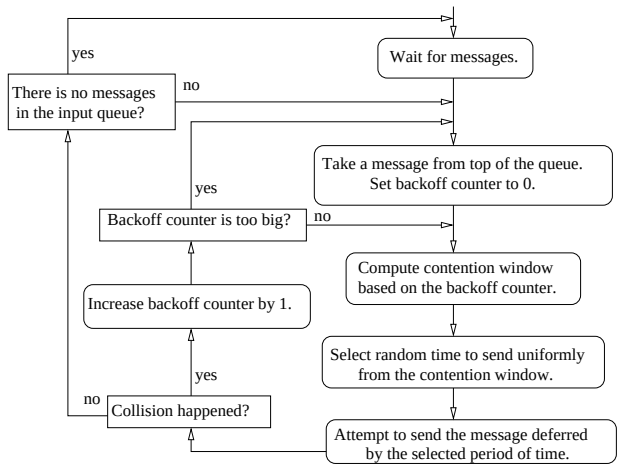



Figure 2.4: Backoff procedure is shown as a block diagram scheme.

every station checks the value $i$ before an attempt to transmit a portion of data (*frame*), based on that value the station decides to discard the packet (and set $i$ to zero) or, otherwise, use value $T_i$. If the station's choice is to use $T_i$ it selects a timeslot which will be used to transmit data uniformly from *contention window (CW):* $[0, 1, \ldots T_i - 1]$ and wait for the timeslot. After that if a collision occurs the station increases $i$ by one, otherwise, if the message was sent successfully the station sets $i$ to zero. The backoff procedure without binding to any values and functions is shown in Figure 2.4.

Previously we said that the CW size is computed by equation $T_i = 2^i$. This backoff protocol version was named binary exponential backoff (BEB),

as it grows with exponential speed and the factor is equal 2. It is widely used, almost all standards that use the backoff protocol utilize it, however, it is worth mentioning that the backoff protocol may be generalized, i.e., CW size is some function of $i$ ($T_i = g(i)$). In literature we additionally meet a linear version of the backoff protocol, where $g(i)$ is a linear function of $i$, and polynomial, when $g(i)$ is a polynomial of $i$. In scheme in Figure 2.4 we omitted the definition of the exact protocol view.

### 2.4.2 IEEE 802.11 modification

Initially, Ethernet used single thick coaxial cable, where collisions could happen, and network hubs to connect together different cables. So it dictated the use of backoff protocol as main part of Carrier Sense Multiple Access with Collision Detection (CSMA/CD) scheme. Later, the technology evolved and Ethernet started to use more sophisticated devices and cables (i.e., twisted pairs and switches). Now, the collisions do not seem to be a great problem for Ethernet. However, recent decade showed a great growth of wireless communications technology, which were predetermined by miniaturization. As by the time Ethernet was very popular and successful standard a similar scheme, namely, Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), was suggested for congestion control in wireless devices. The standard IEEE 802.11 [47] gives the specification for this scheme.

The backoff protocol was also adopted in IEEE 802.11. However, it was slightly different than that was used in Ethernet. Now, if $i$ is the number of successive collisions then $T_i = CW_0 2^i$, where $CW_0$ is initial contention window, which is also used even if no collisions happen before. The number of collisions before discard of the data frame was also changed, although in most parts the backoff scheme remained as in Ethernet and also utilized BEB scheme.

The standard IEEE 802.11 [47] suggests three coordination functions which deal with collisions: Distributed Coordination Function (DCF), Point Coordination Function (PCF), and Hybrid Coordination Function (HCF). While access points utilize PCF for coordination, it in turn works above DCF for contention control. Here we will talk mainly about DCF as the fundamental part of collision resolution procedure. Minutely, DCF states that every station in the IEEE 802.11 network has to wait special periods of time, in order to avoid collisions. In basic scheme, after every transmission, the receiver station has to wait Short Interframe Space (SIFS) before replying with an acknowledgement packet (ACK) or send/receive more advanced frames. SIFS is the minimal gap between two consequent frames required

| | FHSS | DSSS | IR |
|---|---|---|---|
| Slot time | 50 $\mu$s | 20 $\mu$s | 8 $\mu$s |
| SIFS time | 28 $\mu$s | 10 $\mu$s | 10 $\mu$s |
| DIFS time | 128 $\mu$s | 50 $\mu$s | 26 $\mu$s |
| Air propagation time | 1 $\mu$s | 1 $\mu$s | 1 $\mu$s |
| CWmin | 16 | 32 | 64 |
| CWmax | 1024 | 1024 | 1024 |

Table 2.2: Time settings and contention window limits for FHSS, DSSS and IR versions of the IEEE 802.11 standard.

(i) for the receiver to switch from receiving state to sending and (ii) for the sender to separate frames from each another. Other stations before attempt to send should wait at least the DCF Interframe Space (DIFS) period of time during which the network was idle. If a collision happens then the backoff protocol is used, with the specified size for time slots. The minimal size of the contention window ($CW_0$) — CWmin and maximal — CWmax. The standard IEEE 802.11 specifies different values for various modulation principles. In Table 2.2 the values for Frequency-hopping spread spectrum (FHSS), Direct-sequence spread spectrum (DSSS) and Infrared (IR) are given.

In addition to the basic DCF scheme, a RTS/CTS scheme is introduced in the standard. Whenever a station sends a message after waiting DIFS period of idle time it may be collided with another. If the message is a frame with full payload the whole data will be lost and information about the collision will be known only after transmission of such a frame. This is very costly for communication, thus, special "lightweight" frames ready-to-send (RTS) and clear-to-send (CTS) were suggested. Whenever a station wants to send some data, it waits the required DIFS period and sends a RTS frame instead of a full-payload frame. Another station after receiving a RTS frame (addressed to the station) replies with a CTS frame after SIFS period of time. Hence, all collisions happen with RTS/CTS frames, and are almost collision-free in between. Additionally, the RTS/CTS scheme helps to deal with *the hidden station problem* – when two stations not knowing about the existence of each other communicate with one access point (AP). They sense the network is idle, but AP may be in the communication phase with another station, when a CTS frame comes from AP these stations will know that AP will be busy for some period of time.

### 2.4.3 Study of the protocol

Historically, queueing theory was the first way to look at the networking problems from a mathematical perspective. Today it is partly replaced or reconsidered based on game theory, which we are going to talk about at the end of this section.

The study of the ALOHA protocol showed that the unslotted protocol had the throughput $\gamma e^{-2\gamma}$, while the slotted one had $\gamma e^{-\gamma}$, where $\gamma$ is the average total arrival rate at the input [88]. It was considered that the maximum number of active users on the ground may be 324 for the unslotted (i.e. asynchronous) ALOHA protocol [12]. As for the slotted model, for the unslotted model with an infinite user number it was shown that the protocol is unstable. [90] The analysis utilized the study of transient Markov chains and martingales. Other different studies were conducted on the infinite ALOHA model. All of them, however, indicate instability for an infinite number of users.

More interesting was the study of BEB utilized in the Ethernet protocol. For more than the thirty-year-old history of Ethernet it was studied a lot [16, 31, 36, 37, 39, 43, 44, 89, 100], and different models were suggested, starting from infinite models, where the number of stations in the network is infinite [16, 53, 54, 89], ending with a more specific case where only two stations are involved [38].

One of the early studies was suggested by Kelly [53] (later Kelly et al. [54]), a specific infinite model with Poisson incoming rate was used. There was an infinite number of stations in the network, every station sends at most one message, $N_t$ is a random process which indicates the number of backlogged messages. In the first work [53] it was shown that for schemes that grow slower than exponential backoff $N_t \to \infty$, while for exponential backoff there exists a value $v_c$, if the input rate is greater than this value, then $N_t \to \infty$, a lower input rate, however, does not guaranty stability in terms of size of the backlog i.e., $\mathrm{E}(N_t)$. The second work [54] showed the existence of such value a $v_c$ for the general class of random access schemes. On the other hand Aldous [16] used a different infinite station model with backoff scheme working as Poisson processes with different parameters. In his work, Aldous showed that the backoff protocol (and not even backoff protocol, but acknowledgement-based protocols in general) is unstable for any packet arrival rate $\lambda > 0$. Additionally, Rosenkrantz as in the paper earlier for ALOHA protocol proves that under an infinite user model the protocol is unstable [89]. The analysis shows that the corresponding Markov chain is transient using martingales.

However, later Hastad et al. [43] showed using a finite model and Bernoulli

input process that the polynomial backoff protocol is stable for any $\lambda$ in terms of positive recurrence (it contradicts the results of Kelly et al. [53]). They measured stability in terms of the average waiting time $W_{ave}$ and the average number of waiting messages $L_{ave}$ and said that the stability of any of these values leads to stability of another. In contrast they showed that BEB (or any exponential backoff) is unstable if the overall input rate $\lambda \geq .567 + \frac{1}{4N-2}$, where $N$ is the number of stations (it contradicts Aldous' result [16]), and also any linear or sublinear backoff protocol is unstable. Later, the extended version [44] had more details on the analysis. The actual analysis was fully based on the method of potential function (in some sources referred to as Lyapunov [39]), see Theorem 2. The main achievement of Hastad et al. was that they used a model which is closest to a real system, instead of early works with an infinite number of station, and in that model they gained stability conditions in some cases.

Additionally, Goodman et al. [38, 39] performed an analysis of a model with a finite number of stations, which is also closer to reality. They analyzed the $N$ station model, however, a more exact result has been reached for a two-station model. The simplified version of it says that the BEB algorithm is stable for $\lambda_1 = \lambda_2 \leq 0.15$. The work of Shenker [92], however, argues with the results of Hastad about the stability of polynomial backoff. It states that polynomial backoff is stable for the whole channel capacity $\lambda < 1$, but exponential and linear are not.

Finally, it is worth to mention a set of papers on traffic analysis of the Ethernet protocol [24, 65, 94]. While the first one [94] gives some early practical results for the protocol work, which are widely used in theoretical analysis, another paper [24] after some period performs analysis of the traffic with additional knowledge of theoretical assumptions, and also gives an answer to which assumptions were adequate and which were not applicable. It also shows that a great impact on the performance has the *capture effect*, an effect when one station is able to send with higher probability than others (so it unfairly captures the channel). The theoretical work of Hastad et al. [44] utilizes it during the proof of the theorem on potential function. The self-similarity of Ethernet traffic was also studied in [65].

The second inspiration in the study of backoff protocols came up with the adoption of the scheme in the IEEE 802.11 protocol [22] and work on the performance of Bianchi [23]. The work was mainly focused on the performance of the protocol instead of the long-term dependencies (such as stability and ergodicity). It used a set of assumptions which later were considered to be reasonable and that simplified the study. Two of them were standard assumptions in theoretical works: stations are identical and

time divided into timeslots, and two were new. The first one is that the stations are saturated with messages. It means that a station always has messages in the queue to send. It is a natural assumption, as if the station has no messages in the sending queue then it is most probably performing better than the station with messages in the queue. In many cases the saturated model shows worse performance than the unsaturated; however, a study without such assumption creates a lot of cases that should be considered separately. Lastly, there was an assumption on the steady state of the network work, i.e. the collision probability at any moment of time is identical and equal to $p_c$ (it tends to be true with increase of the number of stations). This is the most strong restriction of the analysis. Recently a study of such assumption was conducted in [58], where the restriction was named fixed-point equation (FPE). The work of Bianchi suggested a corresponding Markov chain model and studied the standard BEB protocol implemented in IEEE 802.11. An advanced analysis on the same model, with a wide range of exponential backoff schemes was conducted in the work of Kwak et al. [60]. The most optimal value for the factor, which is used in an exponential protocol (in standard BEB protocol factor equals 2) was found and also a model for study of bounded backoff protocol was suggested, when the number of failed retrials before discarding were bounded from above, while a classical analysis (for finite and infinite station models) assumes it to be equal infinity. Based on these analyses an extended schema which differs greatly from the backoff protocol was suggested [95,96]. They have the form of adaptive backoff protocols, which take into account the load of the network at any moment in time. A recent survey and more bibliographical notes can be found in the following work [26].

The development of game theory in the computer science field also affected the study of backoff protocols. The protocols were reconsidered from the game-theoretical point of view, and the analysis indicated that the equilibrium behavior is selfish in nature, i.e., non-standard protocols will be taken [57]. The later work suggested an algorithm for fairness called CRISP (Cooperation via Randomized Inclination to Selfish/Greedy Play), which is logically based on Prisoner's Dilemma. Another algorithm to deal with unnecessary selfish behavior was suggested by Altman et al. [17]; it utilizes the jamming mechanism, to ensure that the misbehaving station will receive necessary punishment.

## 2.5 Future Internet: overlays, confidentiality, techniques

In this section we are going to talk about new trends and technologies that are being innovated for the Internet today. They were studied and proposed a while ago, and mainly are hot study subjects today, however, they are not massively accepted by the industry and pop up as commercial projects one by one (e.g. [9]). Some standards for such technologies are in the process of formation nowadays ( [3, 8]).

We will discuss briefly all the technologies, as in some sense they are all related to our research. First we will list and briefly introduce P2P technologies, and structures on them. After that we give an overview of HIP, finalizing it with the publish subscribe technique.

### 2.5.1 Peer-to-Peer and Distributed Hash Tables

A decade ago a growing interest in Peer-to-Peer (P2P) communications shook the Internet community. It was the communications not based on the well known server-client basis, but on an equal-to-equal basis (or peer-to-peer), where any participant of the communications could play the role of a server or a client for others. Thus, peers distributed the content among themselves, without a central server or with it, but only for some interaction control, instead of being data servers for others. The first P2P systems were later named unstructured, and the most known of them are Gnutella [25] and KazaA [63] (others may be found in any survey). There were some legacy issues on the first P2P systems, but for the community it was shown that the chaotic P2P system may possess strong survival properties.

Later, the structured versions of P2P systems were suggested by the academia. The most known were CAN [87], Tapestry [109], Chord [98] and Pastry [91]. There is a lot more, and also new structured, as well as unstructured P2P systems appear every day. The structured P2P systems were named distributed hash tables (DHT) as their main role is to be a large hash table. Data are presented there as key-value pairs which are distributed among the network: every participant in the network is responsible for a set of keys and keeping the data for these keys. The protocol itself states which node is responsible for which key and how different nodes should interact in order to keep the network consistent. A full overlay survey can be found in [68] with additional comparison on time consumption for search, insert, delete operations for different DHTs and unstructured P2P systems.

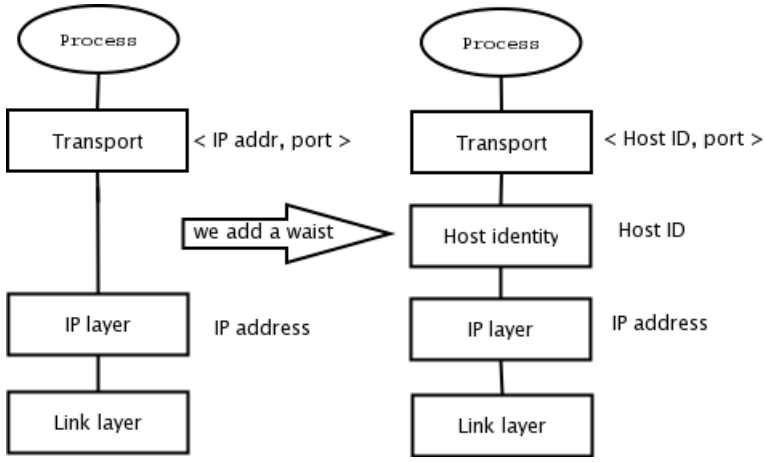



Figure 2.5: Host identity protocol as a new layer in OSI model (image is taken from InfraHIP Project [5]).

### 2.5.2 Host Identity Protocol

Host Identity Protocol [41] is a new layer in the Open System Interconnection Reference model (OSI model), the main purpose of which is to separate the network layer from the transport layer of the OSI model. Because the network layer plays the dual location/identity role and the transport layer is fully based on the network layer, the mobility properties of the nodes were restricted in the Internet. When an application starts some network connections it mainly uses the transport layer or even higher layers (sockets for example) and, thus, unaware of the network layer, from the application position it mainly uses the identity of the data in the Internet, instead of the actual position. The P2P networks above also showed that the communications are mainly based on the identity (for hosts, or a piece of data) rather than on actual locations. The location problem is handled by underlay mechanisms. Thus the question of location/identity separation arose for the current TCP/IP stack. The HIP introduces a new layer in the network stack as shown in Figure 2.5.

Additionally, HIP introduces security on this layer, based on Internet Protocol Security (IPsec) and (Encapsulating Security Payload) ESP encapsulation of the traffic. All nodes communicate between each other through Host Identity Tags (HITs). To communicate, the nodes use HIT (in fact it is a hashed public key) as the identities of stations that they want to connect to, and the HIP, based on the identities, states location where

to send the traffic on the network layer. The HIP uses a four-way handshake mechanism to initialize any connection (to agree on the HITs used and the algorithm), and a two-way handshake for update packets in order to switch location (mobility) or interface (multihoming), i.e., change the IP address from one to another. The initial four-way handshake is called base exchange (BEX) in HIP. For the start of the BEX, the HIP receives HIT for the remote node which it is asked to connect to. However, the communication goes on the standard network layer (IPsec), which requires the IP address of the remote machine, not the HIT. To get the $(HIT, IP)$ pair, the HIP defines a few mechanisms which may replace each other. It is (a) use of DNS, (b) use of local information (hosts file), (c) use of overlay, some DHT (as the HIT is already an identity in a form of a hash), (d) use rendezvous servers, and some more. After the connections are established, the traffic goes to the IP which is in a local database corresponds to the given HIT. The IP in the database may be changed on-the-fly (update messages will inform the remote machine), and thus support mobility.

One of the available implementations of HIP for Linux platforms, is called HIPL [5] (InfraHIP project). There are other available HIP implementations; all of them are realizations of the HIP specification of the IETF working group [3].

### 2.5.3 Publish/Subscribe technique

The Publish/Subscribe technique [29] responds to the recent trends in Internet development. The main purpose of this paradigm is to change the way Internet parties communicate. Instead of the classical point-to-point interactions it suggests publish/subscribe interaction, which in many cases adopts P2P communications principle. When a set of nodes contributes to the other set of nodes, one set of nodes defines the service they provide (or data they contribute) and the other set of nodes defines what services (or pieces of data) they want and subscribe to them. Such a scheme may deal with SPAM (unwanted traffic) by simple logic: if one does not want something from the others, it won't subscribe for that. This thinking also includes the idea that the subscriber hides its location address from others parties, and reveals only some specific address for those publishers from whom it wants to receive the service. Others, possibly even knowing that specific address, should not be able to use it (send data there) or the subscriber can easily change it if needed.

One of the implementations of the publish/subscribe technique is Internet Indirection Infrastructure [97]. It is an application, which is based on the Chord DHT algorithm. It implements a DHT service network which

gives the ability to external stations to publish special information – triggers in the DHT. A trigger has the form $(id, R)$, where the $id$ is the identity number of the trigger (used in DHT); it is used as the address where data may be sent, and $R$ is another address where each packet that comes to the trigger on address $id$ should be redirected. Hence, any node may insert a trigger with some $id$, and set $R$ equal to its own IP address, in that case the data that comes to the trigger $id$ will be redirected to the node IP. Thus, individual stations for communication do not have to reveal their IP addresses. The $(id, R)$ pair may also point to another trigger in the i3 network, i.e., if $id_1$, $id_2$ - triggers $id$, then it is possible to insert chain triggers as $(id1, id2)$, $(id2, R)$. i3 also supports unicast, mobility and anycast, it only restricts that there should not be any cycles. Such indirect infrastructure forms a publish/subscribe scheme, we will show how we use it in the following chapter.

## 2.6 Simulators and network tools

As a part of any new protocol study, simulation produces the necessary connection between theory and practice. It is crucial to test any theoretical ideas on a system that is closer to reality, before starting the actual development process and implementation phase of any algorithm. Some assumptions from theory could miss the confirmation in practice, and the theory may be too far from reality. One should remember that the simulation tools also introduce some assumptions, that, however, are not as important as the theoretical, and different tools are good for different individual tasks. In this section we are going to list most known network simulators. Additionally, in our research we used all of them.

### 2.6.1 NS-2

Network simulator – 2 (NS-2) [10] is one of the most known and accepted simulators nowadays. Recently, with development of the new NS-3 simulator, it started to give up its positions. However, the NS-3 is not a direct continuation of NS-2, thus, it is not such a straightforward modification as the names suggest. During the period of development of NS-3 and inclusion of new models to it, NS-2 may be used for the models that are not supported by NS-3 yet. We utilized it for backoff protocol analysis for different backoff factors (see the next chapter) using built-in Ethernet model. However, later we switched to NS-3 simulator and the IEEE 802.11 model as it is more important for us than Ethernet (IEEE 802.3) model. This

simulator also has a NAM graphical viewer, where anyone may observe the process of the networking.

### 2.6.2 OMNet++/INET

Another very well known simulator tool is OMNeT++ [6], it is a C++ based simulator. In some sense it is a rival simulator to NS-2, however, [101] compares them in the following manner: "...the NS-2 project goal is to build a network simulator, while OMNeT++ intends to provide a simulation platform, on which various research groups can build their own simulation frameworks". OMNeT++ supports different models, including compilation of real-life systems, such as BSD TCP/IP stack, and a lot of models specially created for OMNeT++. One of the most known and accepted is [4], which introduces a wide set of Internet protocols to make the simulation more complete. We used the protocol for our study on defense from DoS and congestion algorithms on the server side analysis; it is supported by BRITE (see next sections) and a close-to-real Internet structure may be easily simulated using the combination BRITE/OMNeT++.

### 2.6.3 NS-3

Network simulator – 3 (NS-3) [11] is a logically newer simulator than NS-2. That means that it is aimed at development of new models, which are demanded by the computer science field. The paper [61] contains thoughts on why it should be a new simulator instead of an upgraded version of NS-2. Additionally, this simulator grew up from a small simulator for wireless networks, and, thus, it has a big advantage on precision of the results in wireless IEEE 802.11 study. Finally, NS-3 is well written, with excellent programming patterns and programming templates, that meet high-level code requirements, with garbage collectors, callback mechanisms, singleton patterns, object factory patterns and others. Thus, programming in it may produce high quality code, with high testability. Unfortunately (or not), it does not support any graphical viewer or editor and thus, all results can be analyzed only by the produced traces. As OMNeT++, it supports real-life system, and even more, the authors say that a real system may be compiled and integrated into it easily. A paper on performance comparison of the NS-2, NS-3 and OMNeT++ [104] affirms that the NS-3 is one of the fastest with least memory consumption.

### 2.6.4 Network generation

As we said earlier, for our analysis of DoS defense mechanisms and congestion resolution algorithms we used OMNeT++ simulator with connection to BRITE network generator [2]. This network generator suggests a set of models for generation of networks that are close to the Internet structure. BRITE also supports a set of additional tools, such as the Otter visualization tool. It may export the generated topology to NS-2, OMNeT++ and others, but, unfortunately, it does not support NS-3. We used a slightly modified generation method for OMNeT++, in order to produce a model that contains some additional information, which lets the simulation process work faster. More details on BRITE may be found in [78].

### 2.6.5 OverSim

Lastly, we want to mention the OverSim [7] simulation tool. It is a framework that supports a set of P2P and DHT protocols. Compared to others, which are mostly special tools for some fixed DHTs (such as Chord, Kademlia, and so on), OverSim supports development of new general protocols, and modification or improvement of the older ones. It is quite fast and simple. One may find a comparison of a set of known tools in [21].

## 2.7 Summary

This chapter is dedicated to a review of existing literature and background on the studied topic. First of all, in the first section, we gave resource-sharing problem definitions with historical remarks. There we showed what congestion control and multiple-access schemes are, then we showed what a denial-of-server attack and selfishness of users are; finally we explained the problems of confidentiality. The following two sections covered necessary mathematical background for Markov chain theory and Game Theory. These sections contain basic definitions, theorems and examples of application to communication networks. After that, we summarized the backoff protocol study in a separate chapter. The study itself has a 30-year-history and, thus, it is very complex with many sometimes contradictory results. The next section covered topics of the future Internet view. These include P2P technologies, where users communicate to each other as equal ones instead of the classical client-server paradigm; after that we presented HIP architecture, which introduces security solutions and defense mechanisms of the DoS problem. It also produces the location/identity split for the classical Internet architecture. We conclude the section with the pub/sub

technique that allows to implement a novel scheme, when a user gets only what the user subscribed to previously. Finally, in the last section we presented simulation tools and network tools that allow us to fully evaluate different aspects of new algorithms and protocols.

# Chapter 3

# Summary of results

In this chapter we are going to give a summary of the main results, which were published and peer-reviewed on conferences and in journals. The author's copy of them will be attached to the end of this thesis. The sections in this chapter cover different aspects of a general backoff protocol, i.e., queueing-theoretical and game-theoretical analysis of the protocol, fairness and simulation of the performance. Additionally, we study the defensing mechanism against Denial-of-Service (DoS) attacks and congestion resolution algorithms on TCP level. Next, we will try to cover some topics on the fairness in peer-to-peer (P2P) interactions (overlays) and discuss the reputation metrics which may be adopted. We use these both for DoS and P2P study. In the end we are going to discuss confidentiality issues for today's Internet architecture and suggest techniques to improve it.

The work itself is dedicated to the concurrent resource sharing problems in multi-user environments. For all the problems we are applying our reputation methodology in order to design a system where users performing selfishly still form optimal system behavior. As well for each problem separately we present analyses that cover all sides of the specific problems. We cover three types of situations: exclusive concurrent access to a common resource (backoff protocol), non-exclusive concurrent access to a common resource (TCP congestion control and DoS), concurrent access to multiple resources (overlay). Throughout this chapter, we show how a general reputation system may be adopted to these problems. However, more specific technologies (such as, MKFS algorithm) cannot be applied to another forms of resource sharing problems (i.e., MKFS is not applicable to the backoff scheme). With these methods and algorithms we produce an analysis of the mentioned access type problems and in a sense, they produce a cover of different resource-sharing problems under various assumptions.

## 3.1 Backoff protocol analysis

In the paper on the general backoff algorithm [70], we performed an analysis of the backoff protocol from a queueing theory viewpoint, in terms of average service time in slots, and time units. We suggested an extension [73] for the basic protocol in order to make it more fair for end stations (especially, in case of the IEEE 802.11 protocol). Additionally, we studied the cases of equilibrium with selfish user behavior [69], and point out how a technique of an arbiter can improve the overall performance in case of selfishness of end stations. Additional studies on reputation metric suggest metric functions possible to use in order to control the behavior of the stations.

The following sections describe it all in the publication order. First of all we are going to talk about the basic protocol and its measurements. After that some advanced version of the protocol will be given. Lastly, the game-theoretical study of the protocol will be explained.

### 3.1.1 Basic protocol

As was said in the previous chapter, the backoff protocol has more than a 30-year-old study history. The results of these analyses deviate a lot, and in many cases even contradict each other. Everything is dependent on the adopted model and assumptions produced in the analysis. Based on these studies, the model of Bianchi [23] seems to be most relevant as the practice and the results are also supported by corresponding simulations. The same model, though written in slightly different form, was used in Kwak et al. [60]. The first work studied BEB for IEEE 802.11 and the average time consumed for the protocol work. The second one studied a set of exponential backoffs, where exponent (or *backoff factor*) $r$ is not restricted by 2. It also suggests a model for a bounded backoff model, where the number of failed attempts to send a message is bounded. In our work [70], we include these models, however, we study a general backoff protocol.

The backoff protocol was discussed in the previous chapter. We will formulate it in terms of general backoff function and will give the relevant model. First of all, we have four main assumptions, based on which we may analyze the work of only one station to get the network behavior:

(i) All stations are *identical*.

(ii) Everything is synchronized by *timeslots* (the IEEE 802.11 has the property of synchronization by ends of messages, as during transmission of messages, all remaining stations hold their states.).

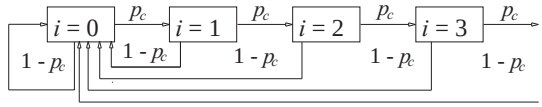



Figure 3.1: Unbounded state model for backoff counter forms transitions between states for corresponding Markov chains.

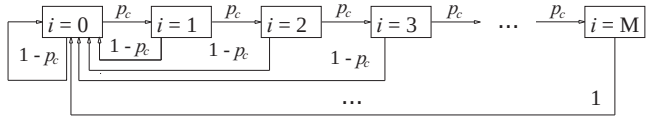



Figure 3.2: Bounded state model for backoff counter forms transitions between states for corresponding Markov chains.

(iii) Model is in *steady state*, i.e., every station at any moment of time sees messages in the network with probability $p_c$ ($p_c \in (0,1)$).

(iv) Model is in *saturated state*, i.e., at any moment of time our station has messages in the queue.

Let us consider one station out of $N$ stations in the network. It has *a backoff counter* $b$ - the number of successive collisions. The station tries to send messages in its input queue. The procedure is as follows: the station takes a message from the top of the queue, and gets value $b$ for the message, if the message is taken the first time then $b$ should be equal to 0. The station has a set of integer-valued (if it is a set of real numbers, it will not affect the analysis a lot) functions $g(i)$, $i \geq 0$. The station tries to send a message to one of the following $g(b)$ timeslots randomly with uniform distribution. In [70] we used function $W_0 f^{-1}(i)$ instead of $g(i)$. We know that any such attempt fails with probability $p_c$. If the message is sent correctly, then the station sets $b = 0$ and takes the following message in the queue, otherwise it increases the backoff counter by one $b \leftarrow b+1$ and repeats the procedure. If the value of $b$ becomes larger than some prefixed $M$, then the message is discarded, the backoff counter is also nullified and the station takes a new message from the queue. We say that the backoff protocol is bounded if $M < \infty$, and unbounded otherwise.

State models for bounded $M < \infty$ and unbounded $M = \infty$ cases are shown in Figure 3.2 and Figure 3.1, respectively.

We name the backoff function $G(p_c) \equiv \sum_{i=0}^{M} g(i) p_c^i$, this function will hold all the properties of the backoff protocol used, depending on the values

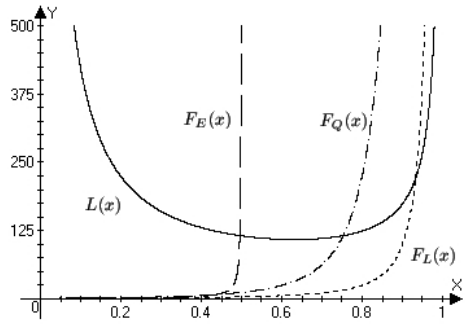



Figure 3.3: Level function and its intersection with different backoff protocols. $G(x) = F_E(x)$ for exponential backoff, $G(x) = F_Q(x)$ for quadratic (polynomial) backoff, $G(x) = F_L(x)$ for linear backoff.

$g(i), i \geq 0$. If we define the "level" function as

$$L(p_c) \equiv \frac{\left(1 - p_c^{M+1}\right)\left(1 + (1-p_c)^{\frac{1}{N-1}}\right)}{W_0\,(1-p_c)\left(1 - (1-p_c)^{\frac{1}{N-1}}\right)},$$

then equation $G(p_c) = L(p_c)$ gives us point $p_c$ as solution. The behavior of the level function is shown in Figure 3.3.

We also prove that a sufficient condition for the equation $G(p_c) = L(p_c)$ to have only one solution in case $M = \infty$ is monotonous increase of functions $g(i)$ with respect to $i$. Later, the same condition was also proved in [58] as Theorem 5.1.

If we define $S$ as the service time for a message, then the average service time for such a model in timeslots has the following form

$$\mathrm{E}S = \frac{\left(1 - p_c^{M+1}\right)}{(1-p_c)\left(1 - (1-p_c)^{\frac{1}{N-1}}\right)},$$

and the necessary condition to have stability in terms of the size of the message queue is

$$\lambda < \frac{N\,(1-p_c)\left(1 - (1-p_c)^{\frac{1}{N-1}}\right)}{\left(1 - p_c^{M+1}\right)}.$$

where $\lambda$ is the average input rate, it may be a Bernoulli process or any other. The saturation condition is also not so important for stability as the

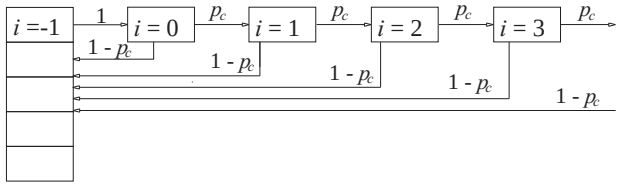



Figure 3.4: Unbounded state model for backoff counter forms transitions between states for corresponding Markov chains with artificial state for the input process, that helps to use negative drift technique.

technique of negative drift may be used. With a slightly modified model (see Figure 3.4) we say that we let a station have $C$ messages (where $C$ is large). After that if the expected number of steps to return to the station with such a number of messages is finite, then it may be shown that the chain is stable and it holds if the bound on $\lambda$ above holds.

In the special case of unbounded protocol (when $M = \infty$) we may minimize the average service time, the minimum point

$$p_c^* = 1 - \left(1 - \frac{1}{N}\right)^{N-1},$$

which tends to $1 - e^{-1}$ as the number of stations goes to infinity. In such a case the supreme of the stable input rate is

$$\lambda^* = \sup\left\{\lambda : \lambda < \left(1 - \frac{1}{N}\right)^{N-1}\right\},$$

which tends, with an increase of number of stations, to the well known limit $e^{-1}$.

The equation $G(p_c^*) = L(p_c^*)$ gives us all protocols that achieve this minimum point, i.e., any optimal protocol should have the following condition fulfilled

$$G(p_c^*) = \frac{2N-1}{\left(1 - \frac{1}{N}\right)^{N-1}}.$$

When the number of stations goes to infinity, the optimal point tends to $1 - e^{-1}$ and optimal backoff tends to exponential backoff with factor $e$, i.e. $g(i) = e^i$.

The case when a backoff protocol is bounded (when $M < \infty$) can be

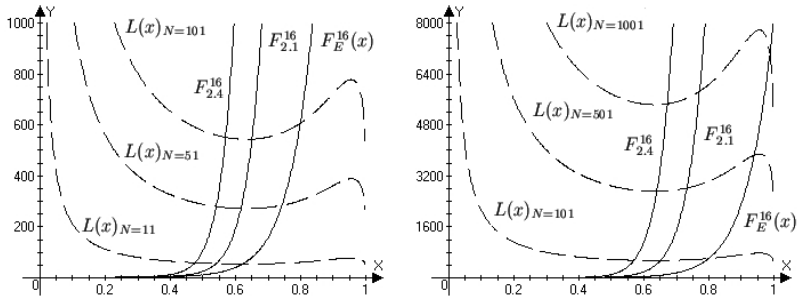



Figure 3.5: The graphs show how level functions (L(x)) for different number of stations $N$ intersect bounded exponential protocols $F_a^{16}(x)$, with factors $a$ equal to 2.1 and 2.4, and intersect truncated BEB protocol from Ethernet $F_E^{16}(x)$. The $X$ axis is the probability of collisions and $Y$ axis almost proportional to average service time (for us most important on the graph are the intersection points).

viewed in Figure 3.5, where

$$F_E^{16}(p_c) = \sum_{i=0}^{10} 2^i p_c^i + \sum_{i=11}^{16} 2^{10} p_c^i$$

$$= \frac{1-(2p_c)^{11}}{1-2p_c} + 2^{10} p_c^{11} \frac{1-p_c^6}{1-p_c}.$$

is the protocol similar to the one used in IEEE 802.11 (although with $W_0 = 1$) and

$$F_a^M(x) = \sum_{i=0}^{M} a^i x^i = \frac{1-(ax)^{M+1}}{1-ax},$$

is just an exponential backoff protocol, with exponent $a$.

The numerical analysis for it is given in Table 3.1.

As said, the optimality task for average service time E$S$ was given in terms of timeslots. It is a natural model as, though the timeslots have different time lengths, the protocol in practice can also be virtually divided in operational slots which do not intersect and cover the whole time line. However, it is also important to see the optimization problem, not in number of timeslots, but in real time. For real-time measurements, we need to define three kinds of slots: (a) idle slot, when no transmission happens, the average time for such a slot is $T_i$, (b) collision slot, when two or more

| Number of stations | Backoff function | $p_c$ | $\frac{ES}{N}$ | $P_{\text{discard}}$ |
|---|---|---|---|---|
| 11 | $F_{2.4}^{16}(x)$ | 0.48 | 2.77 | $3 * 10^{-6}$ |
| 11 | $F_{2.1}^{16}(x)$ | 0.54 | 2.64 | $3 * 10^{-5}$ |
| 11 | $F_{E}^{16}(x)$ | 0.62 | 2.59 | $3 * 10^{-4}$ |
| 51 | $F_{2.4}^{16}(x)$ | 0.54 | 2.76 | $3 * 10^{-5}$ |
| 51 | $F_{2.1}^{16}(x)$ | 0.62 | 2.69 | $3 * 10^{-4}$ |
| 51 | $F_{E}^{16}(x)$ | 0.74 | 2.83 | $6 * 10^{-3}$ |
| 101 | $F_{2.4}^{16}(x)$ | 0.57 | 2.74 | $6 * 10^{-5}$ |
| 101 | $F_{2.1}^{16}(x)$ | 0.65 | 2.71 | $7 * 10^{-4}$ |
| 101 | $F_{E}^{16}(x)$ | 0.80 | 3.02 | 0.022 |
| 501 | $F_{2.4}^{16}(x)$ | 0.64 | 2.72 | $5 * 10^{-4}$ |
| 501 | $F_{2.1}^{16}(x)$ | 0.73 | 2.82 | $5 * 10^{-3}$ |
| 501 | $F_{E}^{16}(x)$ | 0.94 | 3.86 | 0.349 |
| 1001 | $F_{2.4}^{16}(x)$ | 0.67 | 2.73 | $1.2 * 10^{-3}$ |
| 1001 | $F_{2.1}^{16}(x)$ | 0.77 | 2.93 | 0.012 |
| 1001 | $F_{E}^{16}(x)$ | 0.99 | 3.52 | 0.809 |

Table 3.1: Numeric results for bounded exponential protocols $F_a^{16}(x)$, with factors $a$ equal to 2.1 and 2.4, and for truncated BEB protocol from Ethernet $F_E^{16}(x)$, where $p_c$ stands for collision probability, $N$ for the number of stations, $ES$ for the average service time for packets on a station and $P_{discard}$ for the probability to discard a message (failed to send).

stations collide during one slot, the average time for such an event is $T_c$, and (c) successful slot, when some station successfully transmits a message during the timeslot, the average time of it is $T_s$. Finally, the expected average time in time units for such a system can be found by the following equation

$$\begin{aligned}\mathrm{E}_T S =& N(T_s - T_c) + \frac{1}{1-p_c} T_c + \\ & (1-p_c)(N-1)\left((1-p_c)^{1-\frac{1}{N-1}} - 1\right)\Big((1-p_c)T_i + p_c T_c\Big).\end{aligned}$$

It is worth to mention that the value of $T_s$ does not affect the optimization problem and if $T_c = T_i$, then optimization in time units is equivalent to the one in timeslots.

If we consider a particular case of the equation above, when the BEB protocol is considered instead of a general backoff protocol, then we will

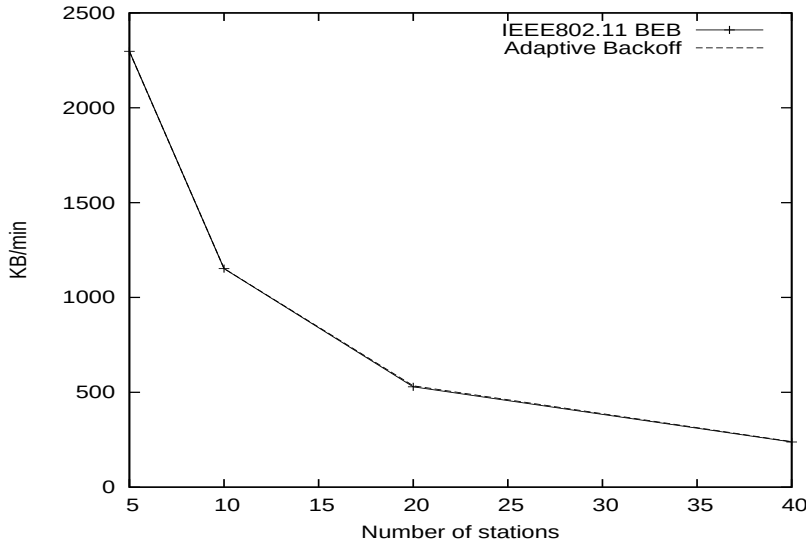



Figure 3.6: Average throughput for a saturation during 1 min

get the same result as in Bianchi [23].

### 3.1.2 Extensions

Our previous work studied the backoff protocol under the assumption of identity of the stations and the steadiness of the network (i.e., existence of $p_c$). The latter is called fixed-point analysis, and is studied in more detail in Kumar et al. [58]. However, these assumptions are not always true, and in this section we are going to talk about the case when they break. First of all, the steadiness of the network clearly hides the presence of the capture effect from us. In [24] it was shown how the capture effect affects the protocol work. Thus, the analysis above is more applicable for networks with a large number of stations, while it has a set of unnecessary assumptions for small networks. Additionally, the condition on the identity of the stations breaks for wireless protocols, where stations may be scattered around the AP. The ones closer to the AP have higher probability to successfully transmit than others, because the signal strength is stronger, and it is being reduced greatly with distance. Hence, even collisions could not be sensed by the closest station, while more distant ones are not able to get consistent signals to the AP.

Based on that, we present and study [73] a modified class of backoff protocols in order to reduce the capture effect. To define the class, we will

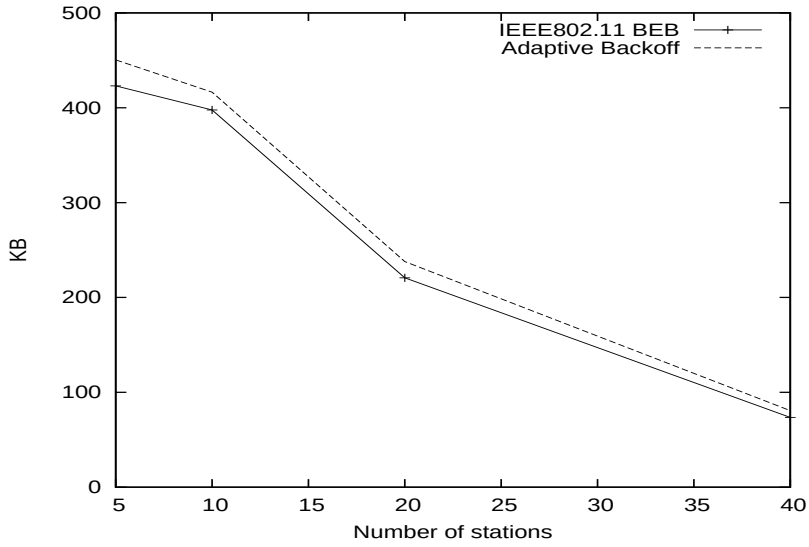



Figure 3.7: Standard deviation for stations during 1 min

consider a model from the previous section. Let us have for any station $M$ states, for every state we wait randomly with uniform distribution on interval $g(i)$, where $i$ is the number of states. The probability of collision in the network is still $p_c$. However, the states are connected in a different manner. If a station collides when it is in state $j$, then

(a) if $j = M$ then the message is discarded, and the message goes to a specified state (not necessarily 0).

(b) if $j < M$ then the state changes to state $j + 1$.

It has little difference to the protocol above, however, in case of success the state returns not to initial state 0, but according to a matrix

$$\hat{P} = \begin{pmatrix} p_0 & p_1 & \dots & p_i & \dots & P_M \\ 1 & 0 & 0 & 0 & \dots & 0 \\ p_{2,0} & p_{2,1} & 0 & 0 & \dots & 0 \\ p_{3,0} & p_{3,1} & p_{3,2} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ p_{M,0} & p_{M,1} & p_{M,2} & p_{M,3} & \dots & 0 \end{pmatrix},$$

where the sum of every row is equal to 1. As we can see the matrix is defined based on the states above, and returns back (in some cases forward) with some probability. We name the whole class of such protocols Reverse Matrix

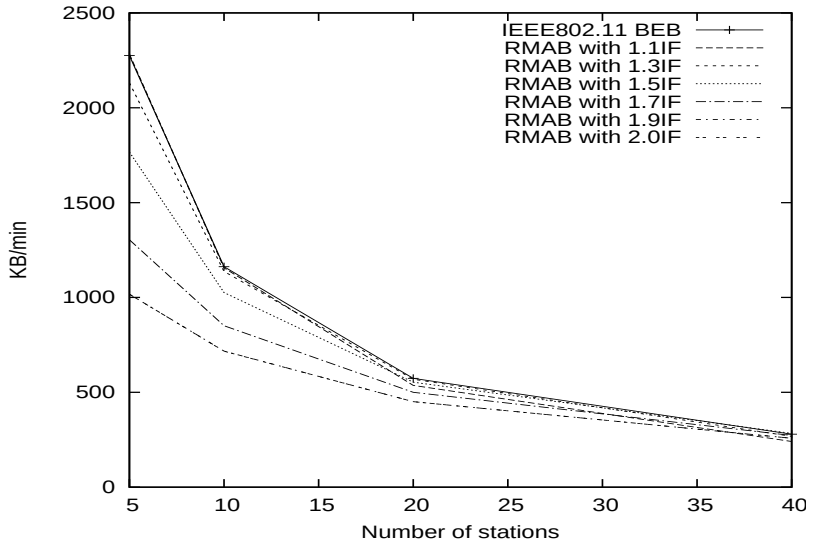



Figure 3.8: Average throughput for RMAB during 1 min

Adaptive Backoff (RMAB) algorithms. If $p_0 = 1$ and, hence, the remaining $p_i = 0$ then it is a special case and we name it, simply, Matrix Adaptive Backoff (MAB) as there is no "reverse" jumps in the corresponding matrix:

$$\hat{P} = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ p_{2,0} & p_{2,1} & 0 & 0 & \dots & 0 \\ p_{3,0} & p_{3,1} & p_{3,2} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ p_{M,0} & p_{M,1} & p_{M,2} & p_{M,3} & \dots & 0 \end{pmatrix}.$$

For such sets of protocol we want to find a stationary distribution for the corresponding Markov chain. We need it in order to get the average service time for this adaptive backoff protocol which may be found as in the previous section based on the state distribution (in previous chapter states were simply geometrically distributed). Much of the work is dedicated to study of RMAB, in the end, however, we explicitly solve only one special case - the MAB algorithm.

The resulting probability to send successfully (or discard unsuccessfully)

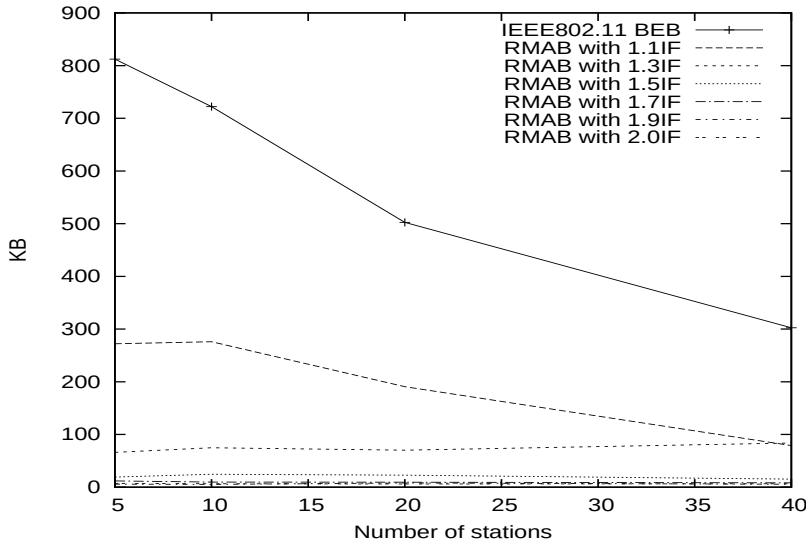



Figure 3.9: Standard deviation for RMAB during 1 min

in state $i$ $\pi_i^*$ for RMAB can be found by the following system:

$$\begin{cases} \pi_0^* = (1-p_c)\sum_{k=0}^{M} p_{k,0}\pi_k^*, \\ \pi_i^* = p_c\pi_{i-1}^* + (1-p_c)\sum_{k=0}^{M} p_{k,i}\pi_k^*, & 1 \le i \le M-1, \\ \pi_M^* = p_c\pi_{M-1}^* + p_c\pi_M^* + (1-p_c)\sum_{k=0}^{M} p_{k,M}\pi_k^*. \end{cases}$$

In a special case of MAB it is

$$\begin{cases} \pi_0^* = (1-p_c)\sum_{k=0}^{M} p_{k,0}\pi_k^*, \\ \pi_i^* = p_c\pi_{i-1}^* + (1-p_c)\sum_{k=i+1}^{M} p_{k,i}\pi_k^*, & 1 \le i \le M-1, \\ \pi_M^* = p_c\pi_{M-1}^* + p_c\pi_M^*. \end{cases}$$

The importance of RMAB is proved in the simulation part of the work. Using NS-3 for the wireless IEEE 802.11 standard we simulated a set of stations ($N = 5, 10, 20, 40$) under the RTS/CTS scheme, in order to neglect the influence of frame size on the results. The simulation runs for 60 seconds over 11 Mbps channels and every station always has messages to send. Thus all stations are active, without the hidden station problem. As was discussed in the previous chapter, the RTS/CTS scheme helps to deal with the problem. We simulated the MAB protocol, where $g(i) = 15 * (1.1)^i$ and

the MAB matrix is

$$\hat{P} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The difference as one can see in Figure 3.6 and in Figure 3.7 is not so big to the standard BEB protocol. However a RMAB protocol of the form

$$\hat{P} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

showed a great increase in fairness in the transmission, the overall transmission of bytes does not deviate a lot, as one can see in Figure 3.8. A simple MAB scheme does not show the increase, e.g., in Figure 3.9.

### 3.1.3 Selfishness

The previous paragraphs touched on the topic of backoff protocol in terms of optimality of service time and fairness. They suggested a mathematical basis for analysis of this optimality based on a variation of backoff function forms $g(i)$ for all $i$ and the change of the state model. Based on that, stations can gain the most optimal behavior in the system of stations, if they behave identically. An extension of the backoff protocol, which was mentioned in the previous section, partly considers the problems when stations do not behave identically. It was said that some stations may be closer to the AP and that there exists a capture effect for such networks.

In this section we are going to talk about the case when stations intentionally do not behave identically, but instead every station adjusts its backoff function, or more generally the time to send any message at any moment of time is based on individual choice. When the station gains some strategy space to make a selfish choice, the problem becomes game-theoretical, compared to the problem, that is based on a pre-fixed protocol.

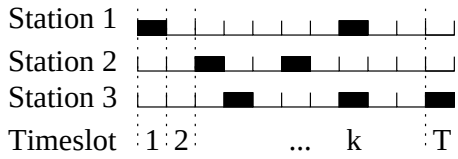



Figure 3.10: Three stations make a decision to occupy 2, 2 and 3 timeslots from $T$ given (occupation is a random process). Stations 1 and 3 occupy one common timeslot — $k$ and, hence, the messages that they send will collide.

In the work [69], we consider a simplified model, where every station decides on a strategy only at the start of a round. The round consists of $T$ timeslots. Every station decides only the number of timeslots it will take during the round. Hence, the strategy for station $i$ consists of a choice of the number $n_i$, where $0 \leq n_i \leq T$. After choosing the value $n_i$ the timeslots which will be occupied will be selected uniformly (one of $\binom{T}{n_i}$, where $\binom{n}{m}$ is a binomial coefficient and $\binom{n}{m} = \frac{n!}{m!(n-m)!}$). An example of such an allocation and choice is shown in Figure 3.10.

In mathematical literature the problem is known as the problem of random allocations [56]. The solution of the problem suggests taking everything for every station — be fully selfish. However, the problem itself is formulated in such a manner that everything is decided beforehand, thus the idea of repeating Prisoners' Dilemma is not applicable here. However, it may be changed if the number of rounds ($T$ timeslots each) is not equal to 1 (i.e., repeating rounds).

The problem stated above gets optimal value with fair division of the slots among participants only if everyone will take "its" own amount of the channel $\frac{T}{N}$. But it is not the individual choice of stations, thus, a technique of arbiter may be applied here. I.e., a station (maybe AP) or a set of stations checks that everyone sends its partition of data and not more than fixed amount, otherwise the arbiter jams the channel making it unavailable (in case of AP it simply does not accept excess traffic). This strategy, however, does not guarantee the existence of only one Nash equilibrium coincident to the system optimum as we wanted to design. Additionally, not all stations may want to send some data at a given round. Thus, this technique has restrictions on applicability.

We may apply our studies on reputation metrics in that case. Every station receives some starting reputation at an initial moment of time, say $k_0$. After that every station decides how much it needs compared to

suggested amount $T/N$, let it be a function $u(t) : \forall t\ 0 \leq u(t) \leq 2$. The value $u(t) = 0$ says that a station does not ask anything from the channel at moment $t$, while $u(t) = 1$ says that it demands its amount $T/N$. A greater value says that a station asks more than its fair amount, and thus will "pay" for it in the future. In mathematical analysis we restricted the variable by 2, but it is not so important in practice. We may neglect the influence of the remaining stations, in that case the problem may be formulated in terms of control theory; we studied it in [71]. The problem may be formulated in the following form

$$\begin{cases} \int_0^T \frac{k_i(t)}{X(t)} u(t)dt \to \max \\ \dot{k_i}(t) = k_i(t)(1 - k_i(t))(1 - u(t)) \\ 0 \leq u(t) \leq 2 \\ k_i(0) = k_i^0 \\ 0 < k_0 < 1. \end{cases}$$

where $k_i(t)$ plays the role of history, i.e., the result of past behavior of a player $i$ and $X(t) = \sum_i k_i(t)$. The solution of the system in terms of control theory (it may be considered, as a particular case of game theory) suggests that if a station stays in the system long enough then it will tend to request its own fair proportion of channel. Otherwise, if a station does not stay in the system more than one round, for example, it will try to take as much as possible ($u(t) = 2$). This model is true for a large amount of players. For two players, a similar model is suggested in [72]. The optimal trajectories are also found there, however they are not found explicitly.

Some additional papers on the jammers in wireless networks and fairness may be found [17].

## 3.2 Defense against Denial-of-Service attacks and congestion control algorithms

The Denial-of-Service (DoS or DDoS, for distributed) attack is one of the major Internet problems. In this section we are going to summarize the defense techniques that we suggested in works [71] and [74]. These techniques do not distinguish malicious users from the benign ones (by assumption attacking packets by themselves are not distinguishable), but they use some properties that differentiate the first group of users from the second one. Thus the techniques themselves are also suitable for congestion control on

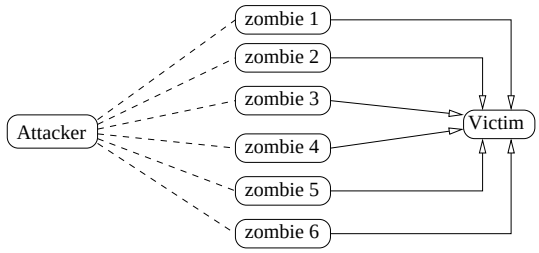



Figure 3.11: Scheme of basic DDoS attack: an attacker gives an order to zombie nodes (infected) to attack/flood the victim server. The attacker may interact with the zombie nodes indirectly.

the server side or a link, although they require some central control for that.

Schematically DDoS attacks can be considered as shown in Figure 3.11. An attacker initiating the attack sends commands to the zombie machines, possibly through some intermediate machines, and the zombie stations start the attack on the server. From the server's point of view, a lot of new requests from new clients start to circulate in the network. If the packets are simple and distinguishable then the server can start to filter them out, but by our assumption they are not. The biggest problem for the server may cause spoofed IP addresses, when a zombie machine generates a new identity for a new packet. Thus the server discovering a new IP address cannot say whether the address belongs to a new benign client or forged by a zombie. Based on that we suggest two algorithms, one for new clients — it deals with new identities and ensures the work of the server by attacking from the spoofed addresses (we name it Most Knocked First Served (MKFS)) and another for addresses that were already checked and "entered" the system. The second algorithm is based on reputation of the users, while the first one is mainly based on a specific queueing policy. The scheme of the algorithm interaction may be found in Figure 3.12.

### 3.2.1 Spoofed addresses

For dealing with the spoofed IP addresses, we suggest a novel MKFS algorithm (MKFS for Most Knocked First Served) [74]. The algorithm uses the property of forged IP address. Whenever a message that came from a spoofed address will be sent back to the address it will never get to the zombie, which sent it (Of course, it does not apply when middle boxes are poisoned, the middleboxes may speak for a subset of clients, the middle-

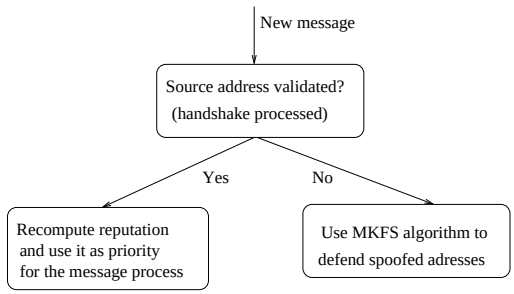



Figure 3.12: The relation of DDoS defense mechanisms against spoofed addresses and misbehavior of indistinguishable users.

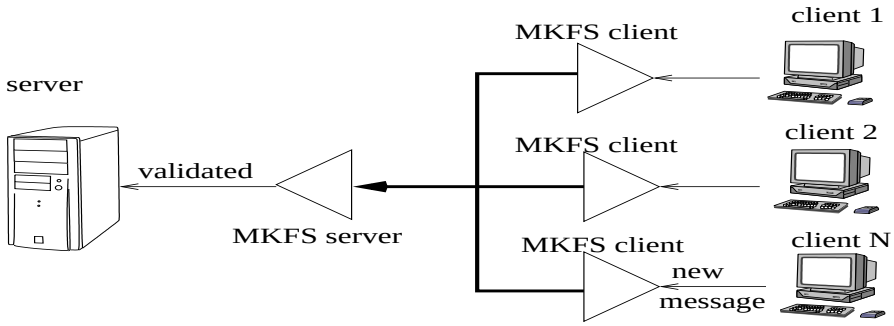



Figure 3.13: Network view for MKFS algorithm work.

boxes, however, probably are more secured than the normal users' systems). So the server just says all clients "knock me with as many packets as you can, I will reply to the ones who knocked most, i.e., who is on the top of the corresponding priority queue". The benign clients will stop knocking whenever they receive a reply, and spoofed zombie stations will send some number of request before changing the identity. However, the zombie stations lose if they send too few messages (thus they do not get to the top of the queue), or send too many messages, thus, sending superfluous amount of messages. On top of everything, to get one reply from a server zombies spend a number of requests (say $K$); in the classical FIFO scheme, they would generate a new packet for every identity; in this scheme the power of the attack is reduced by factor $K$.

The network principle of the algorithm is shown in Figure 3.13. It may be implemented as some tree network of supporting routers if the attack goes on server link capacity instead of the actual server CPU cycle

**Algorithm 1** MKFS algorithm.

**Require:** $Q$ is a valid priority queue.

```
1: loop
2:   if server terminates its work then
3:     return
4:   end if
5:   if server is ready to process I and I has a message then
6:     Get id next from I.
7:     if id is a new and is not ans(id) then
8:       pr(id) ← 1
9:       add id to Q with priority pr(id)
10:    else
11:      pr(id) ← pr(id) + 1
12:      update Q with priority pr(id) for id
13:    end if
14:  end if
15:  if server is ready to process one message then
16:    take id from top of Q
17:    update Q
18:    send reply to id
19:    ans(id) ← true
20:  end if
21: end loop
```

consumption. The tree may separate traffic and just send forward a number of requests from identities. Algorithms that enforce the use of the server-side routers (filters, or middleboxes) are commonly suggested in literature, we, however, try to make the usage of these devices as minor as possible.

The analysis of this queueing policy suggests that for an attacker the optimal number of retries (number of packets with the same identity before forging a new one) grows until the attacks become insignificant (as the growth of the retry number $K$ reduces the strength of attack). We produced a simulation of the algorithm for two cases, when the attackers are able to send traffic at the same speed as benign users and a case when a zombie produces the traffic ten times slower than benign users. The graphs are shown in Figures **??**. For the analysis we used the BRITE generator (with our own modifications) to produce the network for the OMNeT++ simulator.

In the figures one can see that independently of the time when the attack started, the delay for benign users to enter the system is relatively

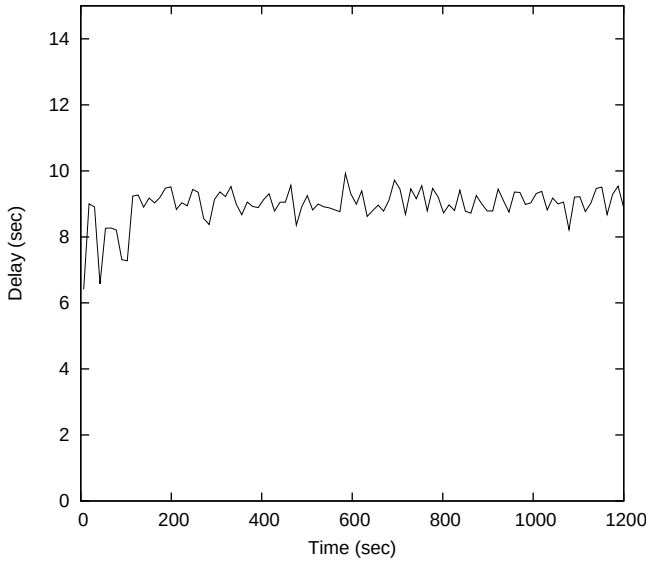



Figure 3.14: The entrance time for new benign users when they may produce traffic at the same speed as zombie stations.

constant.

### 3.2.2 Indistinguishable users with true identities

The previous algorithm suggested the defense mechanism against users, whose identities are forged and hence the server does not know, who actually sends the messages. In this section we will talk about the users, who "entered" the system and, thus, their identities were approved. As an attacker may implement a DDoS attack with the help of spoofing, the large botnets may produce the same kind of attack without spoofing, simply creating a mob randomly "wandering" on the server, i.e., sending some messages that have some sense for the server. If the server is not able to produce a filter for such clients or actions based on separate packets, then the server should observe all historical sequences of behavior. As we suggested in [71], the server based on the identity produces some measure of the last action of the client $i$ at time $t$: $a_i(t) \in (-1, 1)$. The value $a_i(t) = -1$ for the worst behavior, $a_i(t) = 1$ for the best behavior and $a_i(t) = 0$ for neutral. We do not fix any method which may produce the measure, it may be Bayesian networks or some heuristic, or what an engineer of an actual system suggests. The main rule is that it should somehow adequately

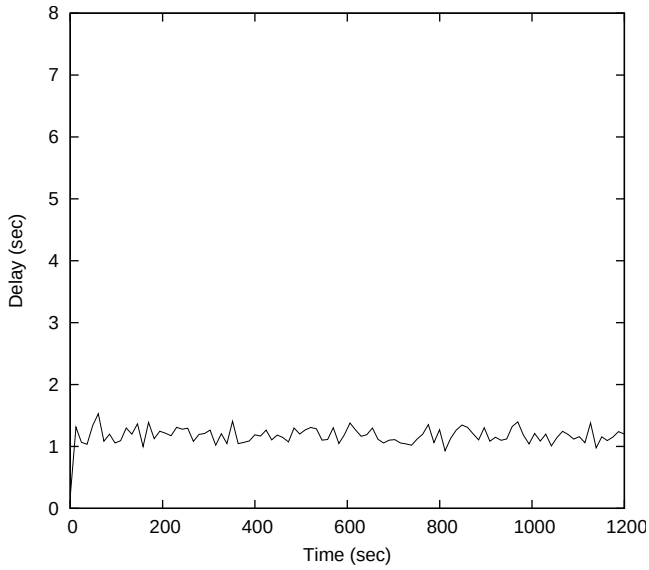



Figure 3.15: The entrance time for new benign users when they may produce traffic ten times faster than zombie stations.

represent the measure of the client's action.

Based on that we suggest client reputation recomputation and the server capacity separation mechanism. It was already introduced in the back-off protocol section. If for user $i$, $k_i(t)$ — is the reputation at moment $t$, then the server for every requested data may answer only proportion $\frac{k_i(t)}{\sum_i k_i(t)}$. Additionally, the metric evolves with change of $a_i(t)$ as following $\dot{k}_i(t) = k_i(t)(1 - k_i(t))a_i(t)$. Hence, a bad sequence of behavior reduces the reputation metric, while a good increases it; thus the metric's grow increases the proportion above. In the paper [71] we used the property that the number of users is huge, i.e. $N$ is a large number, where $i \in [1, N]$ and thus for any station the behavior of remaining stations is not so relevant. Own behavior becomes the most important, and thus, the formulated multi-user game problem may be swapped with the control theory problem like this one (replacing $u(t) = 1 + a_i(t)$ and assuming $k_0$ to be the initial

reputation):

$$\begin{cases} J = \int_0^T x(t)u(t)dt \to \max \\ \dot{x}(t) = x(t)(1 - x(t))(1 - u(t)) \\ 0 \le u(t) \le 2 \\ x(0) = k_0 \\ 0 < k_0 < 1, \end{cases}$$

The analysis is based on the maximum principle of Pontryagin [84], which searches the optimal solution in the open loop manner (controls depending on the time and trajectory depending on time). The solution of the suggested control problem states that moving along the optimal trajectory the player may gain $J = 2\ln\left(1 + \sqrt{e^T \frac{k_0}{1-k_0} + 1}\right)$, which tends to $J/T \to 1$ while $T \to \infty$, i.e. the player selects neutral behavior on the average. The analytical results are partly supported by simulation in the OMNeT++ framework (with BRITE for network generation). However, additional simulation, where more sophisticated strategies are employed, could be required and may strengthen the system.

Both techniques (MKFS and reputation based) may work separately as well as in tandem.

## 3.3 Overlays: confidentiality and misbehavior

In this section we are going to talk about methods for how the overlays, more precisely DHTs, help to defend against DDoS attacks if they are enforced with HIP, and how they help get independence for the HIT-to-IP resolution problem. The publish/subscribe technique helps to achieve better algorithms for the secured traffic in the Internet. We touch on the topic of misbehavior in overlays and mechanisms to control user behavior in the structured or unstructured P2P networks. The mechanism was partly already used in the backoff protocol and in the DDoS defense mechanism in previous sections.

### 3.3.1 Separation of data/control plane

In the previous chapter, we mentioned HIP, which separates the network and transport layers of the OSI model. We also showed that for actual communications through HITs, the protocol needs to receive the corresponding IP addresses from some outer source, be it DNS, overlay, which maps HITs to IP addresses. The most critical for HIP is the BEX phase because both

machines do not know the IP-addresses of each other (one of them does not even know that another wants to connect to it). After the initial phase, if one of the machines changes its IP address then it may itself give necessary information to another machine about the new IP address (through update messages). Thus, the mechanism is mainly used only during the initialization of the connection (BEX), and rarely in case both stations changed their IP addresses simultaneously.

In the paper on the hi3 [42], we suggested mechanisms to send and receive the BEX messages without knowing the IP addresses of each other. The i3 network itself is based on the triggers, thus any station may register a trigger in the i3 network, which has the identity equal to the HIT of the station, and the second destination part to own IP address, thus any message that will be sent to the corresponding HIT will be delivered to the IP address. Any station that knows only the HIT of another station in that network, simply sends information on the trigger with the HIT identity and it will be delivered. Thus, the station does not need to resolve HIT to IP through some DNS server or whatever mechanism.

After the handshake is done through the i3 network, both stations know the IP of each other, and thus they may start to use the underlaying network for communications. They use i3 only for the control messages, it is not optimal to send the data messages through the overlay, because it adds latency to the data delivery process and the protocol becomes slower.

This technique additionally has some defense against the DDoS attacks. As any station reveals its own IP address on the later stage of the BEX, it may select the stations which it wants to give the information to, and stations which it does not want to connect. The second message from the initiator should solve a puzzle that the responder station sends in the first message. Thus, a mechanism against the spoofing of the address is present in the hi3 protocol. The triggers themselves may be switched if some trigger is under a DDoS attack.

We produced a real-life realization of the hi3 protocol for the HIPL version of HIP. It uses a part of the i3 protocol as a library, and open world-wide i3 servers as i3 communication nodes, where triggers may be inserted. The hi3 realization showed that the scheme is fully functioning, however, the i3 library still has version 0.3 and some problems with communication are present.

### 3.3.2 Reputation-based communications

We already said that the past decade showed the growth of different overlays, and mainly the growth was due to the file-sharing networks, when

users distribute some data among themselves. One of the mostly used tools for that is BitTorrent [1]. In most implementations a BitTorrent protocol user may select the upload and download stream speed. The selection of upload and download stream is necessary to restrict the software to use the whole available bandwidth, and jam other user's activity (such as classical HTTPs).

In a sense the selection of the speeds is a strategy space for a user. In that sense it presents as a player in a system. If there were no restrictions and the player was fully aware about it, the player would use a strategy to take all and not give anything back. Such selfish behavior is called free-riding and some thoughts on it may found in [32]. On the other hand, a user should be able to get something before others start to ask it for contributions (where otherwise would the user get a piece of information to contribute before consuming anything, if, of course, it is not own file?). Thus, the system gives data to a user (especially a newcomer), which did not contribute, however this leads to the possibility of whitewashing: a user enters a system, gets all the system gives and leaves the system, changing the identity and reenters with a new identity as a newcomer again. Hence, a system (by a system we mean different sets of P2P users) should be able to contribute to a newcomer, however, it should be restricted to make the whitewashing as unprofitable as possible, keeping in mind control over free-riding.

The discussion above leads to the idea of reputation, when we distinguish participants by the history of their actions. A good taxonomy on the reputations in P2P systems is available in [76]. A game-theoretic approach suggests the ways to deal with incentives for contribution, one of them is in an auction among players [77].

However, for such a problem, we suggest and study a reputation metric as before, and we previously showed that if the number of communicating users is large then the problem may be formulated in terms of control theory [71]. Indeed, with the growth of the number of users, one's personal influence on the whole stream is insignificant. However, the P2P protocols are designed to uniformly distribute load (it is called *load balancing*), and thus, the number of competing nodes which ask for some resource from the same source are low, the mutual influence of the concurrent users is high. Hence, it is more important to study the case when there are only two nodes asking some data from the same source – another bound problem formulation, opposite to the DDoS formulation. In that case it is clearly a game-theoretical problem, which we address in [72]. In general we may

formulate it as follows:

$$\left.\begin{array}{l} \int_0^T \left[\dfrac{f(x_1)r_1(t)}{f(x_1)r_1(t)+f(x_2)r_2(t)+\epsilon} - lc_1(t)\right] dt \to \max \\ \int_0^T \left[\dfrac{f(x_2)r_2(t)}{f(x_1)r_1(t)+f(x_2)r_2(t)+\epsilon} - lc_2(t)\right] dt \to \max \\ \dot{x_1}(t) = c_1(t) - r_1(t) \\ \dot{x_2}(t) = c_2(t) - r_2(t) \\ 0 \le c_1(t), c_2(t), r_1(t), r_2(t) \le 1 \\ x_1(0) = x_2(0) = 0, \end{array}\right\}$$

where functions $r_i(t)$ and $c_i(t)$ play the role of how much user $i$ requests and contributes at moment $i$. Requests are not the same as consumption, it only shows how much one is asking. The amount of received data is the proportion $\frac{f(x_1)r_1(t)}{f(x_1)r_1(t)+f(x_2)r_2(t)+\epsilon}$, where $x_i(t)$ is the history of player's actions (in some sense reputation), or the full sum of all contributed data and all requested data $\dot{x_1}(t) = c_1(t) - r_1(t)$, and $f(x(t))$ is a function that weighs it (or maps on $[0, \infty]$ space). The variable $\epsilon$ is technical, needed only for the adequacy of the solutions (we do not want to have zero divided by zero problem in some cases). We formulate it for the same specific metric as before (i.e., it is the same if we put $\dot{x_1}(t) = c_1(t) - r_1(t)$), but for a two-player game in the following form:

$$\left.\begin{array}{l} \int_0^T \left[\dfrac{k_1 r_1(t)}{k_1 r_1(t)+k_2 r_2(t)+\epsilon} - lc_1(t)\right] dt \to \max \\ \int_0^T \left[\dfrac{k_2 r_2(t)}{k_1 r_1(t)+k_2 r_2(t)+\epsilon} - lc_2(t)\right] dt \to \max \\ \dot{k_1}(t) = k_1(1-k_1)(c_1(t) - r_1(t)) \\ \dot{k_2}(t) = k_2(1-k_2)(c_2(t) - r_2(t)) \\ 0 \le c_1(t), c_2(t), r_1(t), r_2(t) \le 1 \\ k_1(0) = k_2(0) = 0.5. \end{array}\right\}$$

Now $k_i$ plays the role of history, reputation and the weight itself. In the paper, we solve the game problem in terms of the rules of finding the optimal trajectory (it is not solved in the explicit form of trajectory as in the control theory case). For such a problem, the Hamiltonians have the following form:

$$H_1 = \left(lc_1 - \frac{k_1 r_1}{k_1 r_1 + k_2 r_2 + \epsilon}\right) + \lambda_{11} k_1 (1 - k_1)(c_1 - r_1) + \lambda_{12} k_2 (1 - k_2)(c_2 - r_2)$$

$$H_2 = \left(lc_2 - \frac{k_2 r_2}{k_1 r_1 + k_2 r_2 + \epsilon}\right) + \lambda_{21} k_1 (1 - k_1)(c_1 - r_1) + \lambda_{22} k_2 (1 - k_2)(c_2 - r_2)$$

And the co-state variables are:

$$\lambda_{11}(t) = \frac{-1}{k_1(1-k_1)} \int_t^T \frac{(k_2 r_2 + \epsilon) k_1 r_1 (1 - k_1)}{(k_1 r_1 + k_2 r_2 + \epsilon)^2} dt.$$
$$\lambda_{12}(t) = \frac{1}{k_2(1-k_2)} \int_t^T \frac{k_1 r_1 k_2 r_2 (1 - k_2)}{(k_1 r_1 + k_2 r_2 + \epsilon)^2} dt.$$
$$\lambda_{22}(t) = \frac{-1}{k_2(1-k_2)} \int_t^T \frac{(k_1 r_1 + \epsilon) k_2 r_2 (1 - k_2)}{(k_1 r_1 + k_2 r_2 + \epsilon)^2} dt.$$
$$\lambda_{21}(t) = \frac{1}{k_1(1-k_1)} \int_t^T \frac{k_1 r_1 k_2 r_2 (1 - k_1)}{(k_1 r_1 + k_2 r_2 + \epsilon)^2} dt.$$

Now, only for convenience, we define new functions:

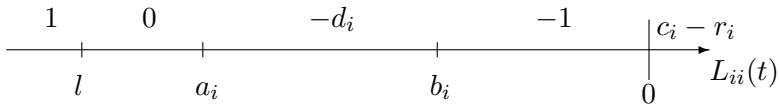
$$L_{11}(t) = -\lambda_{11} k_1 (1 - k_1) \quad \text{i.e. } L_{11}(t) = \int_t^T \frac{(k_2 r_2 + \epsilon) k_1 r_1 (1 - k_1)}{(k_1 r_1 + k_2 r_2 + \epsilon)^2} dt.$$
$$L_{12}(t) = \lambda_{12} k_2 (1 - k_2) \quad \text{i.e. } L_{12}(t) = \int_t^T \frac{k_1 r_1 k_2 r_2 (1 - k_2)}{(k_1 r_1 + k_2 r_2 + \epsilon)^2} dt.$$
$$L_{21}(t) = \lambda_{21} k_1 (1 - k_1) \quad \text{i.e. } L_{21}(t) = \int_t^T \frac{k_1 r_1 k_2 r_2 (1 - k_1)}{(k_1 r_1 + k_2 r_2 + \epsilon)^2} dt.$$
$$L_{22}(t) = -\lambda_{22} k_2 (1 - k_2) \quad \text{i.e. } L_{22}(t) = \int_t^T \frac{(k_1 r_1 + \epsilon) k_2 r_2 (1 - k_2)}{(k_1 r_1 + k_2 r_2 + \epsilon)^2} dt.$$

Based on these functions, which contain "information on the remaining tail" of the optimal path, we define the optimal control as follows:

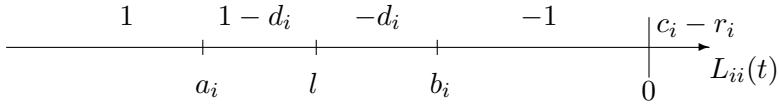
$$c_i = \begin{cases} 0, & \text{if } L_{ii}(t) \leq l, \\ 1, & \text{if } L_{ii}(t) > l. \end{cases} \tag{3.1}$$

$$r_i = \begin{cases} 0, & \text{if } L_{ii}(t) > a_i(t), \\ 1, & \text{if } L_{ii}(t) \leq b_i(t), \\ d_i(t), & \text{otherwise .} \end{cases}$$

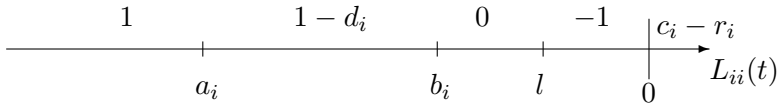
where $d_i(t) = -\frac{1}{a} + \sqrt{\frac{1}{aL_{ii}(t)}}$. The optimal control tells that the optimal trajectory is to follow five possible tendencies, depending on the case, when $c_i - r_i$ is equal to $-1$, $-d_i$, 0, $1 - d_i$ or 1. (See Figure 3.16).




(a) $l > a_i(t)$.




(b) $a_i(t) \geq l \geq b_i(t)$




(c) $l < b_i$

Figure 3.16: Three possible cases for optimal controls for reputation-based communications.

Thus, we get the solution on the optimal decision for a player, at any moment of time (though not in explicit form). The solution states what to optimally do at any given point. However, the reputation here is not global, it may be computed by all individual nodes based on their interaction. For example, if we have four nodes and nodes 1 and 2 connect to node 3, then node number 4 does not know anything about the interconnection of

nodes 1, 2 and 3. For such a problem the algorithm of EigenTrust [52] is suggested. The algorithm spread the local information to global, thus all local reputations may be summed up and form a global reputation for a node. Otherwise, a player may produce the whitewashing, by changing the node it requests for data every time (if local interaction data do not propagate).

# Chapter 4

# Conclusion and future work

### Conclusion

In this thesis we have studied a series of multi-user resource-sharing problems for the Internet. The analysis was made from different perspectives: optimality of protocols, selfishness of end users, fairness of the protocol for end users. This multifaceted analysis allows us to select the most suited protocols among a set of various available based on trade-offs of optima criteria, on one hand. On the other hand, the future Internet predictions dictate new rules for optimality we should take into account and new properties of the networks that cannot be neglected anymore. In the thesis we have studied new protocols for such resource-sharing problems as:

(a) Optimality of backoff protocol.

(b) DDoS defense mechanisms.

(c) Fairness and confidentiality in P2P networks.

For the backoff protocols based on the criteria of overall optimality for, selfishness of and fairness among end-users we suggested different schemes for the protocol, which in turn may be connected in a more general scheme. Mainly, we constructed methodology on the analysis of optimality for such schemes and partly produced the simulation work in order to reconcile theory and practice. However, for further development of the optimal backoff protocol (from different perspectives) the real implementation is the best development of the presented achievements. Among the suggested analyses, the study of fairness seem to be the most important for today's Internet, as the mobility of users grows and with it the amount of wireless stations that have to interact in a common area grows. Thus, the problem that the stations closest to the resources gain more access than distant ones is

the heaviest. It leads to the problem of selfishness as users with modified protocols may gain access to the resources with higher probability than others, however, with the growth of the number of selfish users the access to the resource may be jammed fully as we showed in our previous work.

For the DDoS defense mechanisms we suggested two algorithms: one for dealing with the problem of forged addresses (MKFS) and one for indistinguishable users, based on reputations. The first one utilizes the idea that stations with spoofed addresses do not receives feedback, and thus have less knowledge about the server work than benign users. Reputations, on the other hand, work for unspoofed zombie stations, which using a large amount of packets attempting to stifle server performance. The reputation mechanism based on the large amount of such packets compute characteristics of how much every user may gain from the server (the server only needs to be able to weigh such packets in an adequate manner on the average). Both algorithms are analyzed and show gain in performance compared to the classical FIFO scheme.

Lastly, we suggest techniques for confidentiality and fairness of the peers in overlay networks. The confidentiality is based on the indirection scheme of the i3 network, when all the traffic goes through overlay indirectly (using triggers). The hi3 scheme strengthens the idea with additional security for BEX, while the payload traffic (ESP) goes directly through underlay (Internet) in order to make communication faster and overlay less loaded. Additionally, for interactions in overlays we suggest the metrics that may achieve fairness for end-user resource sharing. The metric may help to deal with such overlay problems as free-riding and whitewashing. We formulate the problem in game theoretical form and for specific reputation metrics we found the optimal control solution in a decision-making form.

## Future work

The research covered wide set of problems from many perspectives, which, of course, still leaves a lot questions for further study. We are going to address extensions that are possible to produce in the theory or/and a set of implementations that can give the study results in practice.

For the backoff scheme, the work produces a set of optimality criteria and shows how to achieve those separately (as trade-offs). Here, a more sophisticated algorithm may be introduced, which is based on the current network situation: it may decide to choose one optimal protocol over another in order to adapt to the current network load and behavior, instead of a prefixed one. Also, the NS-3 simulator is a good tool for the different backoff protocol analyses, however it does not support changes of

backoff protocols, and thus, these modifications have to be hard-coded inside the simulator. A new flexible model for evaluation of general backoff schemes is another direction of future work. Lastly, the new schemes themselves were studied mainly theoretically and were evaluated on simulators (widely acknowledged, however), thus, it is an important step in future work to produce real-life implementation of new backoff schemes in order to strengthen the produced analysis.

For the DoS problems, the promising MKFS algorithm should be studied further and extended. A few possible extensions of the algorithms include addition of a puzzle-scheme to the initial packets that the attacker and benign clients sent to the server. The server, on the other hand, checks the puzzle of an incoming message with some probability (in order to reduce the load). When the puzzle is incorrect the server stops serving the packets of the corresponding sender. This scheme allows us to add some computational difficulty to the sender and more severe punishment for misbehaving. As another extension of future work, both DoS defense mechanisms may be implemented in HIP, thus making it transparent for the application and transport layer of the OSI model. This is very important, because it allows us to use all currently existing transport protocols and applications without any changes. Finally, these schemes may be implemented as part of HIP or intermediate routers in order to reduce the load of the server.

For the P2P and overlays, a useful reputation scheme was introduced. As future work it may be implemented in (on top of) today's P2P networks such as BitTorrent in order to produce behavioral control. Additionally, this scheme is good to implement under the DHT algorithm (while currently all reputation schemes are produced above), in the sense that DHT itself is fully based on reputations of communicating nodes, i.e., send less control packets to misbehaving nodes and so on. As the next intermediate step in the protocol, implementation of reputation scheme for a P2P network in the OverSim simulator is required. It will allow us to see more details and behavior of such a scheme in reality – which is still missing in the work. The hi3 scheme is also a good realization and proof of concept, however for the HIP it should be extended and evaluated additionally.

# References


[1] BitTorrent. http://www.bittorrent.com.

[2] BRITE: Boston University Representative Internet Topology Generator. http://www.cs.bu.edu/brite/.

[3] Host Identity Protocol (HIP) IETF working group. http://www.ietf.org/dyn/wg/charter/hip-charter.html.

[4] INET Framework for OMNeT++. http://inet.omnetpp.org.

[5] Infrastucture for HIP (InfraHIP) project. http://infrahip.net.

[6] OMNeT++. http://www.omnetpp.org.

[7] OverSim: The Overlay Simulation Framework. http://www.oversim.org.

[8] Peer-to-Peer Session Initiation Protocol IETF working group. http://www.ietf.org/dyn/wg/charter/p2psip-charter.html.

[9] Skype. http://www.skype.com.

[10] The Network Simulator — NS-2. http://www.isi.edu/nsnam/ns.

[11] The NS-3 network simulator. http://www.nsnam.org.

[12] N. Abramson. The ALOHA System: Another alternative for computer communications. In *AFIPS '70 (Fall): Proceedings of the November 17-19, 1970*, volume 37, pages 281–285, New York, NY, USA, 1970. ACM.

[13] N. Abramson. Packet switching with satellites. In *AFIPS '73: Proceedings of the June 4-8, 1973*, pages 695–702, New York, NY, USA, 1973. ACM.

[14] N. Abramson. Development of the ALOHANET. *IEEE Transactions on Information Theory*, IT-31(2):119–123, Mar 1985.

[15] A. Akella, S. Seshan, R. Karp, S. Shenker, and C. Papadimitriou. Selfish behavior and stability of the Internet: a game-theoretic analysis of TCP. *SIGCOMM Comput. Commun. Rev.*, 32(4):117–130, 2002.

[16] D. Aldous. Ultimate instability of exponential back-off protocol for acknowledgment-based transmission control of random access communication channels. *IEEE Transactions on Information Theory*, 33(2):219–223, Mar 1987.

[17] E. Altman, K. Avrachenkov, and A. Garnaev. Fair resource allocation in wireless networks in the presence of a jammer. In *ValueTools '08: Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools*, pages 1–7, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[18] Amazon. `http://www.amazon.com`.

[19] S. Asmussen. *Applied Probability and Queues*. John Wiley and Sons, New York, 1987.

[20] T. Basar and G. J. Olsder. *Dynamic Non-cooperative Game Theory*. SIAM, 2nd edition, 1999.

[21] I. Baumgart, B. Heep, and S. Krause. OverSim: A Flexible Overlay Network Simulation Framework. In *Proceedings of 10th IEEE Global Internet Symposium (GI '07) in conjunction with IEEE INFOCOM 2007, Anchorage, AK, USA*, pages 79–84, May 2007.

[22] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: a media access protocol for wireless LANs. In *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, pages 212–225, New York, NY, USA, 1994. ACM.

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

[24] D. R. Boggs, J. C. Mogul, and C. A. Kent. Measured capacity of an Ethernet: myths and reality. In *SIGCOMM '88: Symposium proceedings on Communications architectures and protocols*, pages 222–234, New York, NY, USA, 1988. ACM.

[25] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making gnutella-like P2P systems scalable. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 407–418, New York, NY, USA, 2003. ACM.

[26] J.-W. Cho and Y. Jiang. Fundamentals of the Backoff Process in 802.11. *CoRR*, abs/0904.4155, 2009.

[27] L. F. Cranor and B. A. LaMacchia. Spam! *Commun. ACM*, 41(8):74–83, 1998.

[28] eBay. `http://www.ebay.com`.

[29] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. *ACM Comput. Surv.*, 35(2):114–131, 2003.

[30] K. Fall and S. Floyd. Simulation-based comparisons of Tahoe, Reno and SACK TCP. *SIGCOMM Comput. Commun. Rev.*, 26(3):5–21, 1996.

[31] G. Fayolle, P. Flajolet, and M. Hofri. On a functional equation arising in the analysis of a protocol for a multi-access broadcast channel. *Advances in applied probability*, 18(2):441–472, 1986.

[32] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica. Free-riding and whitewashing in peer-to-peer systems. In *PINS '04: Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems*, pages 228–236, New York, NY, USA, 2004. ACM.

[33] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, Cambridge, Massachusetts, 1991.

[34] V. Fuller, T. Li, J. Yu, and K. Varadhan. Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy. United States, 1993. RFC 1519.

[35] R. Garg, A. Kamra, and V. Khurana. A game-theoretic approach towards congestion control in communication networks. *SIGCOMM Comput. Commun. Rev.*, 32(3):47–61, 2002.

[36] L. A. Goldberg and P. D. MacKenzie. Analysis of practical backoff protocols for contention resolution with multiple servers. *J. Comput. Syst. Sci.*, 58(1):232–258, 1999.

[37] L. A. Goldberg, P. D. Mackenzie, M. Paterson, and A. Srinivasan. Contention resolution with constant expected delay. *J. ACM*, 47(6):1048–1096, 2000.

[38] J. Goodman, A. G. Greenberg, N. Madras, and P. March. On the stability of the Ethernet. In *STOC '85: Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 379–387, New York, NY, USA, 1985. ACM.

[39] J. Goodman, A. G. Greenberg, N. Madras, and P. March. Stability of binary exponential backoff. *J. of the ACM*, 35(3):579–602, 1988.

[40] M. T. Goodrich. Efficient packet marking for large-scale IP traceback. In *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*, pages 117–126, New York, NY, USA, 2002. ACM.

[41] A. Gurtov. *Host Identity Protocol (HIP): Towards the Secure Mobile Internet*. Wiley Publishing, 2008.

[42] A. Gurtov, D. Korzun, A. Lukyanenko, and P. Nikander. Hi3: An efficient and secure networking architecture for mobile hosts. *Comput. Commun.*, 31(10):2457–2467, 2008.

[43] J. Hastad, T. Leighton, and B. Rogoff. Analysis of backoff protocols for multiple access channels. In *STOC '87: Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 241–253, New York, NY, USA, 1987. ACM.

[44] J. Hastad, T. Leighton, and B. Rogoff. Analysis of backoff protocols for multiple access channels. *SIAM journal on computing*, 25(4):740–774, 1996.

[45] A. Hussain, J. Heidemann, and C. Papadopoulos. A framework for classifying denial of service attacks. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures,*

*and protocols for computer communications*, pages 99–110, New York, NY, USA, 2003. ACM.

[46] IEEE. IEEE 802.3 LAN/MAN CSMA/CD (Ethernet) Access Method. `http://standards.ieee.org/getieee802/802.3.html`, 2008.

[47] IEEE. IEEE 802.11 LAN/MAN Wireless LANs. `http://standards.ieee.org/getieee802/802.11.html`, 2009.

[48] J. Ioannidis and S. M. Bellovin. Implementing Pushback: Router-Based Defense Against DDoS Attacks. In *Proceedings of Network and Distributed System Security Symposium, Catamaran Resort Hotel San Diego, California 6-8 February 2002*, 1775 Wiehle Ave., Suite 102, Reston, VA 20190, February 2002. The Internet Society.

[49] V. Jacobson. Congestion avoidance and control. *SIGCOMM Comput. Commun. Rev.*, 25(1):157–187, 1995.

[50] C. Jin, H. Wang, and K. G. Shin. Hop-count filtering: an effective defense against spoofed DDoS traffic. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 30–41, New York, NY, USA, 2003. ACM.

[51] A. P. Jordan Robertson. Authorities bust 3 in infection of 13m computers. `http://www.usatoday.com/tech/news/computersecurity/2010-03-02-botnet-arrest_N.htm`.

[52] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The Eigentrust algorithm for reputation management in P2P networks. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 640–651, New York, NY, USA, 2003. ACM.

[53] F. P. Kelly. Stochastic Models of Computer Communication Systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 47(3):379–395, 1985.

[54] F. P. Kelly and I. M. MacPhee. The Number of Packets Transmitted by Collision Detect Random Access Schemes. *The Annals of Probability*, 15(4):1557–1568, 1987.

[55] A. Y. Khinchin. *Mathematical methods of the theory of mass service*, volume 49 of *Trudy Mat. Inst. Steklov.* RAN USSR, Moscow, 1955.

[56] V. Kolchin, B. Sevastianov, and V. Chistiakov. *Random allocations.* Scripta series in mathematics. Washington, USA, 1978.

[57] J. Konorski. A game-theoretic study of CSMA/CA under a backoff attack. *IEEE/ACM Trans. Netw.*, 14(6):1167–1178, 2006.

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

[59] F. F. Kuo. The ALOHA System. *SIGCOMM Comput. Commun. Rev.*, 25(1):41–44, 1995.

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

[61] M. Lacage and T. R. Henderson. Yet another network simulator. In *WNS2 '06: Proceeding from the 2006 workshop on ns-2: the IP network simulator*, page 12, New York, NY, USA, 2006. ACM.

[62] T. K. Law, J. C. Lui, and D. K. Yau. You Can Run, But You Can't Hide: An Effective Statistical Methodology to Trace Back DDoS Attackers. *IEEE Transactions on Parallel and Distributed Systems*, 16(9):799–813, 2005.

[63] N. Leibowitz, M. Ripeanu, and A. Wierzbicki. Deconstructing the Kazaa Network. In *WIAPP '03: Proceedings of the The Third IEEE Workshop on Internet Applications*, page 112, Washington, DC, USA, 2003. IEEE Computer Society.

[64] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, and S. Wolff. A brief history of the Internet. *SIGCOMM Comput. Commun. Rev.*, 39(5):22–31, 2009.

[65] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Trans. Netw.*, 2(1):1–15, 1994.

[66] J. Li, J. Mirkovic, M. Wang, P. Reiher, and L. Zhang. SAVE: Source Address Validity Enforcement Protocol. Technical report, 2001.

[67] P. Liu, W. Zang, and M. Yu. Incentive-based modeling and inference of attacker intent, objectives, and strategies. *ACM Trans. Inf. Syst. Secur.*, 8(1):78–118, 2005.

[68] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. *IEEE Communications Surveys and Tutorials*, 7:72–93, 2005.

[69] A. Lukyanenko, I. Falko, and A. Gurtov. N-Player Game in a Multiple Access Channel is Selfish. In *Proc. of Workshop on Networking Games and Management*, pages 51–56, June 2009.

[70] A. Lukyanenko and A. Gurtov. Performance analysis of general backoff protocols. *Journal of Communications Software and Systems*, 4(1), March 2008.

[71] A. Lukyanenko and A. Gurtov. Towards behavioral control in multiplayer network games. In *GameNets'09: Proceedings of the First ICST international conference on Game Theory for Networks*, pages 683–690, Piscataway, NJ, USA, 2009. IEEE Press.

[72] A. Lukyanenko, A. Gurtov, and V. Mazalov. Applying a reputation metric in a two-player resource sharing game. In *Proc. of The Third International Conference on Game Theory and Management (GTM'09)*, 2009.

[73] A. Lukyanenko, A. Gurtov, and E. Morozov. An adaptive backoff protocol with Markovian contention window control. In *Proc. of the 6th SPB Workshop on Simulation*, pages 851–856, June 2009.

[74] A. Lukyanenko, V. Mazalov, A. Gurtov, and I. Falko. Playing Defense by Offense: Equilibrium in the DoS-attack Problem. In *Proc. of IEEE ISCC'10*, Piscataway, NJ, USA, 2010. IEEE Press.

[75] A. Mahimkar and V. Shmatikov. Game-Based Analysis of Denial-of-Service Prevention Protocols. In *CSFW '05: Proceedings of the 18th IEEE workshop on Computer Security Foundations*, pages 287–301, Washington, DC, USA, 2005. IEEE Computer Society.

[76] S. Marti and H. Garcia-Molina. Taxonomy of trust: categorizing P2P reputation systems. *Comput. Netw.*, 50(4):472–484, 2006.

[77] V. Mazalov, I. Falko, A. Gurtov, and A. Pechnikov. Equilibrium in a P2P-system. In *Proc. of AMICT'07*, June 2007.

[78] A. Medina, A. Lakhina, I. Matta, and J. Byers. BRITE: An Approach to Universal Topology Generation. In *MASCOTS '01: Proceedings of the Ninth International Symposium in Modeling, Analysis*

*and Simulation of Computer and Telecommunication Systems*, page 346, Washington, DC, USA, 2001. IEEE Computer Society.

[79] R. M. Metcalfe and D. R. Boggs. Ethernet: distributed packet switching for local computer networks. *Commun. ACM*, 19(7):395–404, 1976.

[80] S. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. London, UK, 1993.

[81] J. Mirkovic and P. Reiher. A taxonomy of DDoS attack and DDoS defense mechanisms. *SIGCOMM Comput. Commun. Rev.*, 34(2):39–53, 2004.

[82] J. Nagle. Congestion control in IP/TCP internetworks. *SIGCOMM Comput. Commun. Rev.*, 14(4):11–17, 1984.

[83] T. Peng, C. Leckie, and K. Ramamohanarao. Adjusted Probabilistic Packet Marking for IP Traceback. In *NETWORKING '02: Proceedings of the Second International IFIP-TC6 Networking Conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; and Mobile and Wireless Communications*, pages 697–708, London, UK, 2002. Springer-Verlag.

[84] L. Pontryagin, V. Boltyanskii, R. Gamkrelidze, and E. Mishchenko. *The Mathematical Theory of Optimal Processes*. Interscience Publishers, 1962.

[85] L. Qiu, Y. R. Yang, Y. Zhang, and S. Shenker. On selfish routing in internet-like environments. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 151–162, New York, NY, USA, 2003. ACM.

[86] L. Qiu, Y. R. Yang, Y. Zhang, and S. Shenker. On selfish routing in internet-like environments. *IEEE/ACM Trans. Netw.*, 14(4):725–738, 2006.

[87] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A scalable content-addressable network. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172, New York, NY, USA, 2001. ACM.

[88] L. G. Roberts. ALOHA packet system with and without slots and capture. *SIGCOMM Comput. Commun. Rev.*, 5(2):28–42, 1975.

[89] W. A. Rosenkrantz. Some Theorems on the Instability of the Exponential Back-Off Protocol. In *Performance '84: Proceedings of the Tenth International Symposium on Computer Performance Modelling, Measurement and Evaluation*, pages 199–205, Amsterdam, The Netherlands, The Netherlands, 1985. North-Holland Publishing Co.

[90] W. A. Rosenkrantz and D. Towsley. On the instability of the slotted ALOHA multiaccess algorithm. *IEEE transactions on automatic control*, 28(10):994–996, 1983.

[91] A. I. T. Rowstron and P. Druschel. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In *Middleware '01: Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, pages 329–350, London, UK, 2001. Springer-Verlag.

[92] S. Shenker. Some conjectures on the behavior of acknowledgement-based transmission control of random access communication channels. *SIGMETRICS Perform. Eval. Rev.*, 15(1):245–255, 1987.

[93] S. J. Shenker. Making greed work in networks: a game-theoretic analysis of switch service disciplines. *IEEE/ACM Trans. Netw.*, 3(6):819–831, 1995.

[94] J. F. Shoch and J. A. Hupp. Measured performance of an Ethernet local network. *Commun. ACM*, 23(12):711–721, 1980.

[95] N.-O. Song, B.-J. Kwak, J. Song, and L. E. Miller. Enhancement of IEEE 802.11 distributed coordination function with exponential increase exponential decrease backoff algorithm. *IEEE VTC*, 2003:2775–2778, 2003.

[96] N.-O. Song, B.-J. Kwak, J. Song, and L. E. Miller. Analysis of EIED backoff algorithm for the IEEE 802.11 DCF. In *Proc. of Vehicular Technology Conference, VTC2005-Fall*, volume 4, pages 2182–2186. IEEE, 2005.

[97] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet indirection infrastructure. In *SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols*

*for computer communications*, pages 73–86, New York, NY, USA, 2002. ACM.

[98] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 149–160, New York, NY, USA, 2001. ACM.

[99] M. Sung and J. Xu. IP Traceback-Based Intelligent Packet Filtering: A Novel Technique for Defending against Internet DDoS Attacks. In *ICNP '02: Proceedings of the 10th IEEE International Conference on Network Protocols*, pages 302–311, Washington, DC, USA, 2002. IEEE Computer Society.

[100] B. S. Tsybakov. Survey of USSR contributions to random multiple-access communications. *IEEE transactions on information theory*, 31(2):143–165, 1985.

[101] A. Varga and R. Hornig. An overview of the OMNeT++ simulation environment. In *Simutools '08: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, pages 1–10, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[102] M. Waldvogel. GOSSIB vs. IP Traceback Rumors. In *ACSAC '02: Proceedings of the 18th Annual Computer Security Applications Conference*, pages 5–13, Washington, DC, USA, 2002. IEEE Computer Society.

[103] M. Walfish, M. Vutukuru, H. Balakrishnan, D. Karger, and S. Shenker. DDoS defense by offense. In *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 303–314, New York, NY, USA, 2006. ACM.

[104] E. Weingärtner, H. vom Lehn, and K. Wehrle. A performance comparison of recent network simulators. In *ICC 2009: IEEE International Conference on Communications*, 2009.

[105] J. Wu, G. Ren, and X. Li. Source Address Validation: Architecture and Protocol Design. In *IEEE International Conference on Network Protocols*, pages 276–283. IEEE Computer Society Press, 20072.

[106] J. Xu and W. Lee. Sustaining Availability of Web Services under Distributed Denial of Service Attacks. *IEEE Transactions on Computers*, 52(2):195–208, 2003.

[107] D. K. Y. Yau, J. C. S. Lui, F. Liang, and Y. Yam. Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles. *IEEE/ACM Trans. Netw.*, 13(1):29–42, 2005.

[108] H. Zhang, D. Towsley, and W. Gong. TCP Connection Game: A Study on the Selfish Behavior of TCP Users. In *ICNP '05: Proceedings of the 13TH IEEE International Conference on Network Protocols*, pages 301–310, Washington, DC, USA, 2005. IEEE Computer Society.

[109] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph. Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and routing. Technical report, 2001.

[110] Y. Zhou and H. Sethu. On achieving fairness in the joint allocation of processing and bandwidth resources: principles and algorithms. *IEEE/ACM Trans. Netw.*, 13(5):1054–1067, 2005.

TIETOJENKÄSITTELYTIETEEN LAITOS
PL 68 (Gustaf Hällströmin katu 2 b)
00014 Helsingin yliopisto

DEPARTMENT OF COMPUTER SCIENCE
P.O. Box 68 (Gustaf Hällströmin katu 2 b)
FIN-00014 University of Helsinki, FINLAND

JULKAISUSARJA **A**

SERIES OF PUBLICATIONS **A**

Reports may be ordered from: Kumpula Science Library, P.O. Box 64, FIN-00014 University of Helsinki, FINLAND.

A-2004-1 M. Koivisto: Sum-product algorithms for the analysis of genetic risks. 155 pp. (Ph.D. Thesis)

A-2004-2 A. Gurtov: Efficient data transport in wireless overlay networks. 141 pp. (Ph.D. Thesis)

A-2004-3 K. Vasko: Computational methods and models for paleoecology. 176 pp. (Ph.D. Thesis)

A-2004-4 P. Sevon: Algorithms for Association-Based Gene Mapping. 101 pp. (Ph.D. Thesis)

A-2004-5 J. Viljamaa: Applying Formal Concept Analysis to Extract Framework Reuse Interface Specifications from Source Code. 206 pp. (Ph.D. Thesis)

A-2004-6 J. Ravantti: Computational Methods for Reconstructing Macromolecular Complexes from Cryo-Electron Microscopy Images. 100 pp. (Ph.D. Thesis)

A-2004-7 M. Kääriäinen: Learning Small Trees and Graphs that Generalize. 45+49 pp. (Ph.D. Thesis)

A-2004-8 T. Kivioja: Computational Tools for a Novel Transcriptional Profiling Method. 98 pp. (Ph.D. Thesis)

A-2004-9 H. Tamm: On Minimality and Size Reduction of One-Tape and Multitape Finite Automata. 80 pp. (Ph.D. Thesis)

A-2005-1 T. Mielikäinen: Summarization Techniques for Pattern Collections in Data Mining. 201 pp. (Ph.D. Thesis)

A-2005-2 A. Doucet: Advanced Document Description, a Sequential Approach. 161 pp. (Ph.D. Thesis)

A-2006-1 A. Viljamaa: Specifying Reuse Interfaces for Task-Oriented Framework Specialization. 285 pp. (Ph.D. Thesis)

A-2006-2 S. Tarkoma: Efficient Content-based Routing, Mobility-aware Topologies, and Temporal Subspace Matching. 198 pp. (Ph.D. Thesis)

A-2006-3 M. Lehtonen: Indexing Heterogeneous XML for Full-Text Search. 185+3 pp. (Ph.D. Thesis)

A-2006-4 A. Rantanen: Algorithms for $^{13}C$ Metabolic Flux Analysis. 92+73 pp. (Ph.D. Thesis)

A-2006-5 E. Terzi: Problems and Algorithms for Sequence Segmentations. 141 pp. (Ph.D. Thesis)

A-2007-1 P. Sarolahti: TCP Performance in Heterogeneous Wireless Networks. (Ph.D. Thesis)

A-2007-2 M. Raento: Exploring privacy for ubiquitous computing: Tools, methods and experiments. (Ph.D. Thesis)

A-2007-3 L. Aunimo: Methods for Answer Extraction in Textual Question Answering. 127+18 pp. (Ph.D. Thesis)

A-2007-4 T. Roos: Statistical and Information-Theoretic Methods for Data Analysis. 82+75 pp. (Ph.D. Thesis)

A-2007-5 S. Leggio: A Decentralized Session Management Framework for Heterogeneous Ad-Hoc and Fixed Networks. 230 pp. (Ph.D. Thesis)

A-2007-6 O. Riva: Middleware for Mobile Sensing Applications in Urban Environments. 195 pp. (Ph.D. Thesis)

A-2007-7 K. Palin: Computational Methods for Locating and Analyzing Conserved Gene Regulatory DNA Elements. 130 pp. (Ph.D. Thesis)

A-2008-1 I. Autio: Modeling Efficient Classification as a Process of Confidence Assessment and Delegation. 212 pp. (Ph.D. Thesis)

A-2008-2 J. Kangasharju: XML Messaging for Mobile Devices. 24+255 pp. (Ph.D. Thesis).

A-2008-3 N. Haiminen: Mining Sequential Data – in Search of Segmental Structures. 60+78 pp. (Ph.D. Thesis)

A-2008-4 J. Korhonen: IP Mobility in Wireless Operator Networks. (Ph.D. Thesis)

A-2008-5 J.T. Lindgren: Learning nonlinear visual processing from natural images. 100+64 pp. (Ph.D. Thesis)

A-2009-1 K. Hätönen: Data mining for telecommunications network log analysis. 153 pp. (Ph.D. Thesis)

A-2009-2 T. Silander: The Most Probable Bayesian Network and Beyond. (Ph.D. Thesis)

A-2009-3 K. Laasonen: Mining Cell Transition Data. 148 pp. (Ph.D. Thesis)

A-2009-4 P. Miettinen: Matrix Decomposition Methods for Data Mining: Computational Complexity and Algorithms. 164+6 pp. (Ph.D. Thesis)

A-2009-5 J. Suomela: Optimisation Problems in Wireless Sensor Networks: Local Algorithms and Local Graphs. 106+96 pp. (Ph.D. Thesis)

A-2009-6 U. Köster: A Probabilistic Approach to the Primary Visual Cortex. 168 pp. (Ph.D. Thesis)

A-2009-7 P. Nurmi: Identifying Meaningful Places. 83 pp. (Ph.D. Thesis)

A-2009-8 J. Makkonen: Semantic Classes in Topic Detection and Tracking. 155 pp. (Ph.D. Thesis)

A-2009-9 P. Rastas: Computational Techniques for Haplotype Inference and for Local Alignment Significance. 64+50 pp. (Ph.D. Thesis)

A-2009-10 T. Mononen: Computing the Stochastic Complexity of Simple Probabilistic Graphical Models. 60+46 pp. (Ph.D. Thesis)

A-2009-11 P. Kontkanen: Computationally Effcient Methods for MDL-Optimal Density Estimation and Data Clustering. 75+64 pp. (Ph.D. Thesis)

A-2010-1 M. Lukk: Construction of a global map of human gene expression - the process, tools and analysis. 120 pp. (Ph.D. Thesis)

A-2010-2 W. Hämäläinen: Efficient search for statistically significant dependency rules in binary data. 163 pp. (Ph.D. Thesis)

A-2010-3 J. Kollin: Computational Methods for Detecting Large-Scale Chromosome Rearrangements in SNP Data. 197 pp. (Ph.D. Thesis)

A-2010-4 E. Pitkänen: Computational Methods for Reconstruction and Analysis of Genome-Scale Metabolic Networks. 115+88 pp. (Ph.D. Thesis)