VŠB - Technical University of Ostrava
Faculty of Electrical Engineering
and Computer Science

Using Quantum Key Distribution for Securing Real-Time Applications

Ph.D. THESIS

2017 Miralem Mehić

VŠB - Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Telecommunications

Using Quantum Key Distribution for Securing Real-Time Applications

Ph.D. Thesis

Doctoral Study Branch: 2601V018 Communication Technology
Doctoral Study Programme: P1807 Computer Science, Communication Technology and Applied Mathematics

Supervisor: Assoc. Prof. Miroslav Vozňák, Ph.D

2017

Ing. Miralem Mehić

VŠB - Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Telecommunications
17. listopadu 15/2172, 708 33 Ostrava, Czech Republic

Using Quantum Key Distribution for Securing Real-Time Applications

Ph.D. Thesis; Delivered in April, 2017
Doctoral Study Programme:
P1807 Computer Science, Communication Technology and Applied Mathematics
Doctoral Study Branch:
2601V018 Communication Technology

PhD. Student:
Ing. Miralem Mehić
VŠB – Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Telecommunications
17. listopadu 15/2172, 708 33 Ostrava, Czech Republic
miralem.mehic.st@vsb.cz

Supervisor:
Assoc. Prof. Miroslav Vozňák, Ph.D
VŠB – Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Telecommunications
17. listopadu 15/2172, 708 33 Ostrava, Czech Republic
miroslav.voznak@vsb.cz

This thesis was typeset using LaTeX

I hereby declare that I am the sole author of this PhD thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

In Ostrava, 27.04.2017

....
Miralem Mehić

"Recite in the name of your Lord who created -
Created man from a clinging substance.
Recite, and your Lord is the most Generous -
Who taught writing by the pen.
Taught man that which he knew not."
Al-Alaq 96:1

Acknowledgements

The work presented in this thesis would not have been possible without my close association with many people who were always there when I needed them the most. I take this opportunity to acknowledge them and extend my sincere gratitude for helping me make this thesis a possibility.

The research leading to this thesis was carried out under the supervision of Miroslav Vozňák. I would like to thank him for his encouragement and tremendous support. His door was always open to discuss both scientific and non-scientific problems. His constant effort to cooperate in new scientific fields and explore new ideas was one of the main motivations for this work.

The idea for this study was born during my stay at AGH University of Science and Technology in Krakow (Poland) where Marcin Niemiec introduced me to the field of quantum cryptography. I would like to thank him for investing his time in studying my works and encouraging discussions.

Over the last two years of the study, I had the pleasure to cooperate with Austrian Institute of Technology (AIT) Digital Safety and Security department. I must thank Momtchil Peev and Andreas Poppe who enabled this cooperation. I thank them for long discussions that helped me to get familiar with the problems and the state of the art in the field of quantum cryptography. I had a fruitful collaboration with Oliver Maurhart to whom I hereby thank for all the discussions and debates that emerged from my "short" questions. I thank him for his numerous tips and comments, his time and commitment to the realization of a virtual quantum link between AIT and VŠB. I am very grateful to all the members of AIT business unit Optical Quantum Technology for their support. I would like to thank Cristina, Sandra, Christian, Stefan and Manuel for discussions, socializing and help during my stay in Klagenfurt (Austria) and later.

This thesis has improved substantially thanks to the valuable comments and feedback from Stefan Rass. I'm very grateful to him for time he invested in studying my work. His sharp and precise suggestions as well as long discussions in his office helped me to organize my research.

I have to thank Peppino Fazio from University of Calabria (Italy) for motivating emails, detailed advices, and discussions related to the problems of routing and quality of service.

I want to thank all the members of LIPTEL team of VŠB. Their support has certainly helped me to do my research on time and helped me during my stay in Ostrava. I have to thank Jan, Filip, Martin, Jirka, Pavol, Jaromir, Lukaš, Jaroslav, and Jakub, who were always available for questions, assistance, a cup of coffee and the occasional game of chess.

Finally and most importantly, I would like to thank my girlfriend Lejla for all her love, support, understanding and constant motivation. My parents, Đemila and Muhamed, deserve special thanks for their continued support and encouragement. Without such a team behind me, I doubt that I would be in this place today. This thesis is dedicated to them.

Abstract

Quantum Key Distribution (QKD), based on the laws of physics rather than the computational complexity of mathematical problems, provides a secure way of establishing symmetrical binary keys between two geographically distant users. The keys are secure from eavesdropping during transmission and QKD ensures that any third party's knowledge of the key is reduced to a minimum. In recent years, a noticeable progress in the development of quantum equipment has been reflected through a number of successful demonstrations of QKD technology. While they show the great achievements of QKD, many practical difficulties still need to be resolved, such as to provide better service differentiation. These networks are characterized as being multihop in nature where the consumption key rate is often higher than the charging key rate, which means that the links are available for a limited period of time only. Such features impose several challenges on the effective modeling and evaluation of reliability as well as finding appropriate Quality of Service (QoS) solution. This thesis focuses on research in the field of QKD for securing real-time communication by supporting QoS in QKD networks including a novel QoS model and novel distributed reactive routing protocol to achieve high-level scalability and minimize the consumption of key material used for securing routing data.

As research in QKD networks grows larger and more complex, the need for highly accurate and scalable simulation technologies becomes important to assess the practical feasibility and foresee difficulties in the practical implementation of theoretical achievements. Due to the specificity of QKD link which requires optical/quantum and Internet connection between the network nodes, it is very costly to deploy a complete testbed containing multiple network hosts and links to validate and verify a certain network algorithm or protocol. The network simulators in these circumstances save a lot of money and time in accomplishing such task. A simulation environment offers the creation of complex network topologies, a high degree of control and repeatable experiments, which in turn allows researchers to conduct exactly the same experiments and confirm their results. This thesis describes the design and implementation of QKD network simulation module which was developed in the network simulator of version 3 (NS-3). The module supports simulation of QKD network in overlay mode or in a single TCP/IP mode. Therefore, it can be used for simulation of other network technologies regardless of QKD.

Implemented simulation model was used for verification of proposed QoS solution. A number of simulations were performed. The obtained data have confirmed the primary thesis of this study, that it is possible to use real-time applications in QKD networks.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AES Advanced Encryption Standard. 25

AIT Austrian Institute of Technology. 25, 86

AODV Ad Hoc On-Demand Distance Vector. 52

BGP Border Gateway Protocol. 31

DiffServ Differentiated Service. 75

DSDV Destination-Sequenced Distance-Vector Routing. 52, 99

GG Gabriel Graph. 97

GPSRQ Greedy Perimeter Stateless Routing Protocol for QKD network. 88, 92

IntServ Integrated Services. 74

IPsec Internet Protocol security. 28

ISP Internet Service Provider. 31

ITS Information-Theoretical Secrecy. 23, 30, 79

LSA Link State Announcement. 99

MANET Mobile Ad Hoc Network. 78

NS-2 The Network Simulator NS-2. 41

NS-3 The Network Simulator NS-3. 2, 41

OLSR Optimized Link State Routing Protocol. 52

OSPF Open Shortest Path First. 5, 98

OTP One-Time Pad. 10, 11, 23, 24

P2P Point-to-Point. 22

PKI Public Key Infrastructure. 1

QBER Quantum Bit Error Rate. 14, 15

QKD Quantum Key Distribution. 2, 7, 10, 11, 17–19, 21–23, 27–33, 73, 78, 79

QKDNetSim The QKD Network Simulation Module. 2, 39, 40, 43, 45

QKRA Quantum Key Reservation Approach. 5

QoS Quality of Service. 2, 5

RTT Round Trip Time. 64

SNR Signal-to-Noise Ratio. 23

TTL Time To Live. 64

VANET Vehicular Ad Hoc Network. 78

Chapter 1

Introduction

1.1 Background and research context

One of the central issues in cryptography is a secure distribution of key between geographically distant users, known as secret key agreement problem. The current solutions in most applications are based on the usage of public key infrastructure (PKI) which relies on assumptions about the computation power of eavesdropper and computational complexity of mathematical problems. As results, these solutions fall within the scope of theoretically breakable computational security and with the increase of computational power such solutions are under threat [1]. Quantum information theory suggests the possibility of solving secret key agreement problem by using an information-secure quantum key distribution, known as QKD [2]. Based on the laws of physics, QKD provides a secure way of establishing symmetrical binary keys between two geographically distant users without relying on the hardness of mathematical problems. In contrast to public-key cryptography, the combination of QKD with suitable message authentication scheme has been proven to be secure without intractability hypotheses [3–5].

In previous years, a noticeable progress in the development of quantum equipment has been reflected through a number of successful demonstrations of the QKD networks in several countries around the world [6–10]. In 2015, a record investment of the US Department of Defense to support and develop the prototype of a scalable quantum network with memory has been recorded. The long-awaited launch of the Chinese quantum satellites should be an additional milestone, while the newly opened quantum laboratories in Europe should significantly contribute to the additional development of quantum cryptography.

Therefore, it is reasonable to conclude that quantum technology has grown significantly and is rapidly approaching the level of high maturity. Consequently, we strongly believe that the next natural step in the evolution of quantum systems is to study their performances, suitability, and convergence with the applications used in everyday life.

1.2 Relevance and research objectives

The objectives of the thesis are driven by the need for research describing and analyzing the issues of quantum cryptography from the telecommunication networking point of view, which in turn help improve the efficiency of using QKD in everyday communications.

This thesis focuses on the application of QKD for securing real-time communications by supporting QoS in QKD networks including the QoS model and QoS routing protocol. There is a growing need to provide better service differentiation in QKD networks. These networks are characterized as being multihop in nature where the consumption key rate is often higher than the charging key rate, which means that the links are available only for a limited period of time. Such features impose several challenges for effective modeling and evaluation of reliability, as well as developing QoS solutions for QKD networks.

The second objective of this thesis was fulfilled with the implementation of QKD network simulation module (QKDNetSim) which was developed in the network simulator of version 3 (NS-3). To the best of our knowledge, QKDNetSim is a first application for simulating and testing of QKD networks and gives a significant contribution to the research community in the field of QKD technologies. We assume that QKDNetSim will help in the understanding of the practical use of QKD technology which will allow the usage of a range of applications within the QKD network. Although primarily designed for QKD network, QKDNetSim can be easily utilized to simulate other types of networks.

1.3 Contributions of the thesis

This thesis provides two kinds of results. The first and also the main results are shown as the theoretical solution that is reflected in the novel methods and mechanisms for supporting QoS in QKD networks. Our results should improve the convergence of quantum technology with the user's applications used in everyday life. The practical aspect of this thesis is reflected in the software simulator that should serve as a base substrate for taking the further steps into the research of quantum technology. Considering that currently there are no effective QoS solutions nor available simulators of QKD networks, it is anticipated that our research would generate a great deal of interest, not only among researchers from the field of quantum technologies but also among the general public.

1.4 Thesis Organization

This thesis is organized into two parts. The introduction, State-of-the-Art, aims and the theoretical background for the research are presented in the first part. Chapter 2 provides the State-of-the-Art while chapter 3 outlines the fundamentals of QKD protocols and provides the necessary theoretical background, that is, the methods and procedures for the establishment of a key material which is in the constant focus of QKD network. Chapter 3 is concluded with simplified calculations for key length calculations while the interested reader is referred to the [11] for more details. Chapter 4 specifies aims of the research.

The second part of this thesis presents the main results of the study. Chapter 5 presents the first computing simulation environment of QKD network and deals with the practical realization of QKD networks from a network point of view. The chapter summarizes the limitations and the basic characteristics of QKD network and explains the ways for practical implementation of QKD network. Chapter 6 deals with the mutual dependence of the public and the quantum channel of QKD link and analyzes the impact of changes in the performance of both channels on the overall key material establishment process. The work presents measurement data obtained under laboratory conditions as well as the results obtained by establishing a virtual QKD link. The obtained data clearly show that the public channel has a major impact on the performance of the quantum channel and vice versa. In chapter 7 we discuss a novel flexible QoS solution in QKD network introducing a novel GPSRQ routing protocol. This chapter includes discussions about the existing QoS models, lists the requirements placed on the implementation of the routing protocol for QKD network, defines the metrics for the quantum and the public channel, provides simulation setup of conducted experiments and discusses the obtained result and the broader aspects of used approach. Chapter 8 concludes this study.

Chapter 2

State of the Art

2.1 Quality of Service in Quantum Key Distribution Network

2.1.1 Theoretical Solutions

Considering the common assumption that all nodes along the path in QKD network must be fully trusted, in theory, QKD networks was mainly analyzed from two aspects: security and network performance. The idea of passive eavesdropping, in which the adversary may use eavesdropping not to extract information, but to redirect the data flow towards a node that he controls, have been analyzed in [12]. Following a similar idea, stochastic routing has been proposed in order to avoid deterministic routing which is used in traditional routing protocols [13, 14]. This study examines the possibility of hiding of routing information by aiming to increase security and decreasing the probability of being attacked.The transmission capacity of QKD network has been analyzed in [15], while the queueing model has been reported in [16]. Game-theoretic techniques have been used to find an optimal balance between interdependent service quality criteria with distinct performance indicators [17]. As the result, network provisioning strategies that ensure the promised service level at the optimized performance are obtained. It is worth to note that most of these results have not been (practically) demonstrated due to lack of simulation tools for QKD network.

2.1.2 Practical Solutions

On 1st April 2004, a bank transfer between the bank headquarter and the Vienna city hall has been realized to demonstrate the successful distribution of secret keys in an urban environment [18, 19]. In 2006, a one-time pad (OTP) encrypted surveillance video application was developed to demonstrate the speed, robustness and sustainability of existing QKD system [20]. In October 2007, the Geneva Sate Chancellery installed a QKD system to secure a link transmitting ballot papers to the counting station during the federal elections in Switzerland [21], while in Japan, QKD has been used to secure VoIP communication in metro area network [22]. QKD has been used to secure a communication link at the 2010 FIFA World Cup in Durban, South Africa [23, 24] while in Tokyo, high speed QKD systems have enabled real-time secure video conferencing in a metropolitan area [9]. Although the literature states that a variety of applications have been tested in previously deployed QKD networks, the research in the performance of public channels was generally underestimated and most attention was paid to the quantum channel performances. The research of the public channel is scarce and to the best of our knowledge, reports on network traffic analysis of the communication of QKD post-processing process over the public channel has not been published.

QoS in previously deployed QKD networks has been largely neglected, stating that it is somehow achievable without any difficulties. The prioritization of network traffic and signaling protocols were ignored, and consequently, the solutions from existing conventional networks have been modified for the needs of QKD network. The first such solution, which is based on the modification of the well-known Open Shortest Path First (OSPF) routing protocol [25], has been implemented for the purpose of the DARPA BBN QKD network built in 2004 in the US [26]. Instead of using the routing hop-count metric, a modified OSPF protocol was used to determine the link quality based on the amount of key material in key storage. Although this solution is protected by the patent [27], this modified version of the OSPF does not take into account the status of the public channel.

A similar approach was offered in [28] where the author proposed usage of unencrypted and non-authenticated communication for the dissemination of OSPF routing packets. Obviously, such a network is easy prey for an eavesdropper who is assumed to have unlimited resources at his disposal, especially when the passive eavesdropping is taken into account [12]. Since this solution is based on the usage of the amount of key material in key storages as routing metric, it cannot provide efficient routing due to lack of information about the state of the public channel.

In SECOQC network, another modified version of the OSPF protocol was introduced [29, 30], based on the local load balancing policy which was calculated as a ratio of the number of transmitted bits in the period of time. As the authors note, this solution cannot provide QoS in QKD network [29], since it does not consider the available amount of key material which means that the algorithm may choose the path with insufficient key material for data transmission.

During the development of Chinese HCW QKD network, Quantum Key Reservation Approach (QKRA) which is based on IntServ model has been reported [31, 32]. In these solutions, OSPF is used to find the path from the source to the destination node, and after the path is determined, source node issues a key reservation request to all nodes in the path. After receiving the request, intermediate node responds with a key reservation result message, and finally, destination node determines the possibility of establishing the connection. Since OSPF is focused on finding the shortest path – hence the name, solutions presented in [31, 32] find the shortest path between source and destination and reserve a sufficient amount of a key on a selected path. Note that this path may not be optimal. More specifically, the path is the shortest one but it may not be adequate in terms of QoS. It is known that minimum hop-count (shortest) routing typically finds routes with significantly lower throughput than the best available [33] since it does not consider other parameters of the link. OSPF in its original form does not consider QoS constraints; therefore, it cannot guarantee that the traffic on the selected path will be served adequately. Reservation of resources on the quantum channel, in this case, does not solve the problem of QoS since the path for the public channel may be inappropriate. However, even an extended version of OSPF which includes QoS constraints [34] may not be optimal for QKD networks. Such implementation of OSPF can find the path which has the best characteristics of the public channel, but it does not consider the parameters of the quantum channel. The above shows that the implementation of QoS in the QKD network is not a trivial task.

2.2 Quantum Key Distribution Simulators

Unlike for conventional networks, there are few software applications dealing with QKD. Quantum Cryptography Protocol Simulator [35] developed using C/C++ architecture is able to analyze the quantum bit error rate (QBER) and eavesdropper influence on the performances of the quantum channel when BB84 or B92 QKD protocol is used. A similar application is reported in [36]. Object-oriented simulation for QKD protocols was reported in [37] while an event-by-event simulation model and polarizer as simulated component for QKD protocols with the presence of eavesdropper and misalignment measurement as scenarios were reported in [38]. A simulation framework for QKD protocols using OptiSystem was reported in [39], and a modeling framework designed to support the development and performance analysis of practically oriented QKD system representations was reported in [40].

Yet, all of these applications deal only with the quantum channel performance or QKD protocols and disregard the public channel and the entirety of the protocol stack above the QKD link. To the best of our knowledge, applications for simulating QKD networks with multiple nodes and links are not available.

Chapter 3

Background

3.1 Fundamentals of Quantum Key Distribution

Until recently, cryptography was a meeting point between security engineering and mathematics. The definition has since been extended to include the principles of quantum physics. The concept of quantum cryptography was originally proposed in 1983 by Stephen Wiesner, a student at Columbia University [41], although its real development was achieved in 1984 when Charles Bennett (IBM) and Gilles Brassard (University of Montreal) presented BB84, the first QKD protocol [2, 42]. Five years later they made the first practical demonstration of QKD by establishing a secret key over 32,5 cm through the air. This protocol is the most widely used even today and it will be briefly discussed in this chapter.

3.1.1 Why quantum cryptography matters?

Quantum cryptography focuses on photons (particles of light), using some of their properties to act as an information carrier. Principally, information is encoded in a photon's polarization, and a polarized single photon is referred to as a qubit (quantum bit). A qubit cannot be split, copied or amplified without introducing detectable disturbances and it can be represented as a linear combination of two basic states (horizontal and vertical):

$$|\psi\rangle = \alpha \cdot |\updownarrow\rangle + \beta \cdot |\leftrightarrow\rangle \tag{3.1}$$

Parameters α and β are probability amplitudes: the probability that the outcome of the measurement will be a vertical or a horizontal base, respectively. Unlike the classical bit, which can only have two possible values, 0 or 1, Equation (3.1) shows that the qubit can also have these values, as well as it can potentially have a superposition of both. It means to find out the value carried by a qubit it is necessary to perform measurement of photon's polarization. The special notation $|\rangle$ is called the Dirac Notation and it is the standard notation for states in quantum mechanics [30].

Due to Heisenberg's uncertainty principle [43], which states that certain pairs of physical properties are related in such a way that measuring one property prevents the observer from simultaneously knowing the value of the other with great precision, it is not possible to determine simultaneously the position and the momentum of a particle with an arbitrarily high accuracy. Applying Heisenberg's principle to quantum mechanics means that passive observation (eavesdropping) of photons is not feasible. Since each measurement of the quantum state modifies the quantum state, an eavesdropper would be incapable of monitoring communications without altering the transmitted information, which would be easily detectable by the sender and the receiver [30].

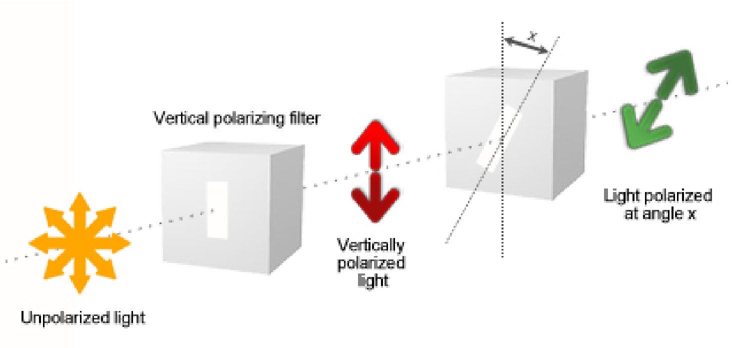
Vertical polarizing filter
x
Light polarized at angle x
Vertically polarized light
Unpolarized light

Figure 3.1. Polarization of light

Let us analyze the example shown in Fig. 3.1 where non-polarized light enters a vertically aligned filter, which absorbs some part of the light and polarizes the remainder in the vertical direction. Then, the light is directed through the second filter oriented at angle x to the vertical one, and there is a certain probability that the photon will pass through the second filter as well where this probability depends on angle x. As x increases, the probability of the photon passing through this filter decreases until it reaches zero at x=90° when we consider this filter a horizontal polarizing filter. But, when x=45°, there is a 50% probability the photon will pass through the filter.

A pair of orthogonal polarization states used to describe the polarization of photons is referred to as the basis so it is easy to distinguish the following bases:

- rectilinear (vertical polarization, where x=0°, and horizontal polarization, where x=90°)
- diagonal (diagonal polarization, where x=45° and x=-45°)

Consider the communication between two users. User A, typically designated Alice in literature, uses a filter in the rectilinear basis (either a horizontal or vertical polarizing filter) to polarize the photon and direct it to user B, typically designated Bob in literature. Bob can reliably determine the polarization of the received photon only if he uses a filter aligned to the same, rectilinear basis, see Fig. 3.2-a. However, if Bob uses a filter in the diagonal basis, he cannot consider the received information to be reliable since the probability of receiving photon is only 50% which is not enough to accurately determine the basis used by Alice, see Fig. 3.2-b. Because of this, additional steps are needed in which users exchange information about the sequences in which the same basis were used.

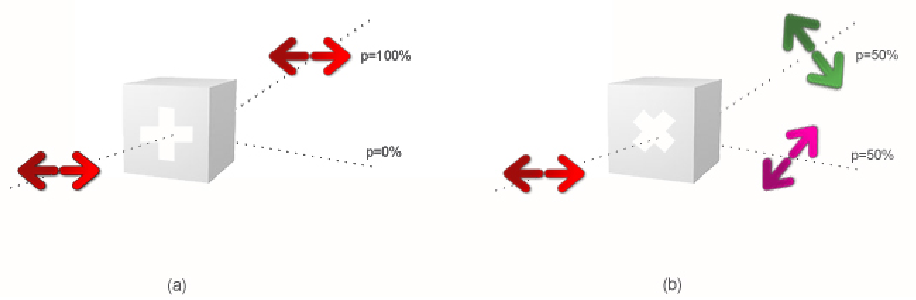

Figure 3.2. Probability of detecting a horizontally polarized photon with a rectilinear (a) and a diagonal (b) basis)

The second quantum physical principle says it is not possible to produce a photon with the same properties once again. To be precise, the no-cloning theorem says it is not possible to duplicate an unknown quantum state while keeping the original intact [44]. Therefore, the man-in-the-middle attack is excluded. If an eavesdropper attempts to perform such an attack, he will be able to produce a photon with the same properties only with a certain probability, while other photons will not have the same properties.

Let us assume the channel is eavesdropped. If eavesdropper Eve uses the same basis as Alice, she can recover the original polarization of the photon. But if she uses a misaligned filter (diagonal basis), she will get no information (like Bob in the previous case) and, by doing so, she will change the initial polarization of the photon, so Bob will receive a "garbled" photon. Finally, when Alice and Bob later exchange details about the used basis via a public channel and uncover a part of obtained measurements, they can detect the presence of Eve. Of course, an eavesdropper is not the only one who is responsible for errors in transmission. There are errors in the measurement (caused by disturbance of the quantum channel, noise in the detectors or an optical misalignment) but from theory, the boundary for these errors is already known. Therefore, if the error rate exceeds this rate, Alice and Bob can be quite sure that eavesdropping was carried out. It should be noted that quantum mechanics does not prevent from eavesdropping. It only allows the detection of the presence of an eavesdropper.

3.1.2 Information-Theoretical Secrecy

A major aspect of quantum cryptography is the methodology named Quantum Key Distribution (QKD) which is used for generating and distributing random encryption keys between two geographically separate users using the principles of quantum physics[1]. The keys are secure from eavesdropping during transmission and QKD ensures that any third party's knowledge of the key is reduced to a minimum [45]. There are other research fields of quantum cryptography like quantum authentication, quantum secret sharing, quantum signature but QKD is a major aspect of quantum cryptography.

The primary goal of QKD is to provide information-theoretic secure (ITS) communication. An information-theoretic secure system means that a system is still secure even if an attacker has unlimited resources available to perform the cryptographic analysis. This definition was presented by Claude Shannon in well-known paper about "Perfect Secrecy" [46] where the perfect secrecy system is defined as a system where a posteriori probabilities of a ciphertext (encrypted message), which are intercepted by an eavesdropper, are identical to a priori probabilities of the same message before the interception. Shannon showed that a One-Time Pad (OTP) is an unbreakable cryptosystem and has perfect secrecy because it does not reveal any details about the original encrypted message since both the original and the encrypted message have equal entropy. A necessary and sufficient condition for perfect secrecy is that the conditional probability of ciphertext E, if message M is chosen, is equal to the probability of obtaining ciphertext E from any cause chosen. That is, $P_M(E)$ needs to be independent of message M. In that case, no matter how much material is intercepted, the eavesdropper can not obtain information about the encrypted message M.

$$P_M(E) = P(E), \text{ for all } M \text{ and all } E \tag{3.2}$$

[1] QKD is also known as Quantum Key Growing, since it needs a small amount of key material pre-shared between two parties to establish a larger amount of the secret key material. The pre-shared secret key serves to guarantee the in-tegrity of the protocol in the first transaction and it should not be used for any other purposes except to establish a new key material.

OTP cipher was presented in 1917 [47] by the American scientist Gilbert S. Vernam (AT&T Bell Labs), who introduced the most important key variant to the Vigenère cipher system. The main feature of this cipher is that the key must be totally random, and it must be used only once. Also, the length of the key must be the same as the length of the message that needs to be encrypted. The sender performs an XOR-operation on the clear text message and secret key to perform encryption and provide ciphertext while the receiver uses the same operation on the ciphertext to perform decryption and obtain the clear text message.

The most significant problem of OTP is the secure distribution of such a large key between remote clients. In World War II, the one-time key material was printed on silk which was concealed inside of agent's clothing and whenever the key had been used, it was torn off and burned [48]. Today, QKD can be used to exchange a secret key securely. The combination of QKD with OTP and an information-theoretically secure message authentication scheme such as Wegman-Carter [49, 50] is often referred to as quantum cryptography. In this sense, quantum cryptography provides encryption which cannot be broken by any analysis, irrespective of the advances made in mathematics or computer science including quantum computation.

3.1.3 BB84 protocol

BB84 is the oldest and most widely used QKD protocol and will be shortly explained here. It uses the two bases of orthogonal polarization: rectilinear and diagonal and it consists of the following six successive stages: secret key exchange, sifting-extraction of the raw key, error rate estimation, error correction, privacy amplification and authentication. Only the first stage is performed over the quantum channel while all other stages are performed over the public channel.

At the beginning of communication, Alice and Bob must agree on the same alphabet. Since BB84 assumes that the polarization of photons is used as a carrier of information, Alice, and Bob need to know which polarization presents which value of the bit. For example, the bit value 1 is presented as vertical or diagonal polarization, while the bit value 0 is presented as horizontal polarization or opposite diagonal polarization. It is important to stress that two polarizations represent the same value of the bit. Then, Alice defines a random key with length Q and uses a randomly selected polarization from the alphabet as a carrier of the key. It is the first phase of the key establishment procedure in BB84 as depicted in Fig. 3.3.

For example, for the bit value 1 Alice can choose either vertical (x=90°) or diagonal polarization (x=45°), and for the bit value 0 she can choose either horizontal polarization (x=0°) or opposite diagonal polarization (x=-45°). On the receiving side of the quantum channel, Bob chooses a randomly selected basis for detection. Since Bob does not know which basis Alice has

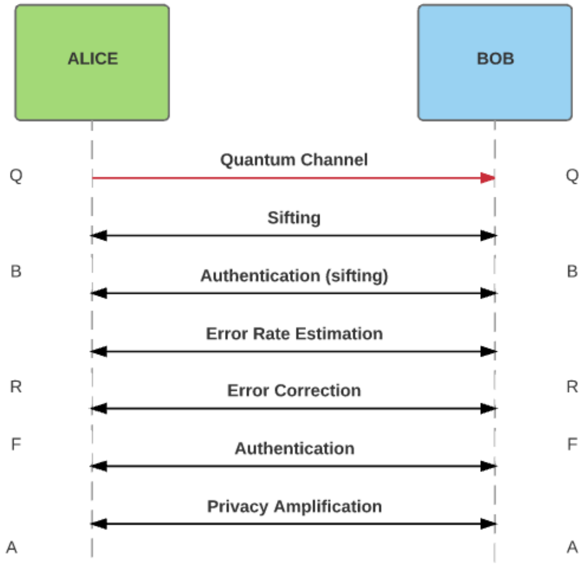

Figure 3.3. BB84 - Key establishment procedure

used, and he uses a randomly selected basis, he will be able to reliably detect only 50% of the sent key, as it was explained in the previous text. Table 3.1 shows an example of this situation.

Table 3.1. An example of key distribution in BB84 protocol

Alice	↘	↔	↗	↙	↘	↕
	0	0	1	1	0	1
Bob	✕	✛	✕	✛	✛	✕
	↘	↔	↙	↕	↕	↘
	0	0	1	1	1	0
Result	OK	OK	OK	Luck	Error	Error

Now let us consider the situation in which Eve is eavesdropping. Since Eve does not know which basis Alice has used, Eve needs to use a random basis for the detection of photon's polarization. If Eve uses the correct basis, she will not change the polarization of the photons;

however, if she uses the incorrect base, the original polarization of the photon will be changed. The situation with Eve's influence is presented in Table 3.2.

Table 3.2. Eavesdropping in BB84 protocol

Alice	⤡	↔	⤢	⤢	⤡	↕
	0	0	1	1	0	1
Eve	×	×	+	+	×	+
	⤡	⤢	↕	↕	⤡	↕
Bob	×	+	×	+	+	×
	⤡	↕	⤢	↕	↔	⤡
	0	1	1	1	0	0
Result	OK	X!	Luck	Luck	Luck	Error

For the first bit, Eve uses the correct basis for detection, and she obtains the correct information without changing the initial polarization of the photon. However, for the second bit, Eve uses the incorrect basis for detection; this changes the original polarization, and, as a result, Bob will be able to detect the incorrect value even if he uses the correct basis for detection.

Sifting - Extraction of a raw key

After exchanging the key values over the quantum channel, all further communication is performed over the public channel. First, Bob informs Alice which polarization basis he has used for each received bit. Second, Alice responds when Bob uses the correct polarization basis and when he uses incompatible bases (these bits should be discarded). It must be stressed that Bob only discloses information about the used basis while the value of the measurement remains secret. After this step, Bob is certain of the sequence of correct polarization he used for detection. The length of Bob's reliably received key is marked with B in Figure 3.3.

Error rate estimation

Eve is not solely responsible for errors in the quantum channel since errors may occur due to disturbance of the quantum channel, noise in the detectors or an optical misalignment. To distinguish the origin of errors, Alice and Bob need to determine the error rate p of the quantum channel. The threshold of bit error rate (p_{max}) for quantum channel without presence of Eve is known in forward, so Alice and Bob need to compare a small portion of their key in order to estimate the quantum bit error rate (QBER). If the error rate is higher than a given threshold ($p > p_{max}$), Alice and Bob revealed the presence of Eve and the key distribution process starts all over again.

Alice and Bob need to decide about the length of the sample block which is going to be used to estimate error rate. If they choose a short sample block (uncover 0 bits), then they will know nothing about the QBER and the possible presence of Eve. On the contrary, if they choose a long sample block (uncover all n bits), they are completely sure about QBER value in the channel. But then they will shorten the key even more since the values of the sample block must be announced publicly and Eve may have access to these values. The length of the sample block used for error rate estimation is defined with a "level of security" parameter S. In [51], two levels of security are defined (basic and advanced) and they are bounded with the percentage of the key assigned to comparison as shown in Table 3.3. Alice and Bob should select the desired level of security S, and use Equation (3.3) to calculate the number of bits k that will be used for the QBER estimation, where n is the total length (Q) of the original key.

$$S(k) = \frac{-\sum_{k=1}^{n} \frac{k}{n} \cdot \log \frac{k}{n}}{n} \tag{3.3}$$

Table 3.3. Values of "level of Security" parameter S

Level	Minimal Value	Part of key(%)	Maximal Value	Part of key (%)
Basic	0.01	8.16	0.1	38.10%
Advanced	0.1	39.10	0.24	85.48%

The value k can be presented as the percentage of the key used for the QBER estimation. However, Alice and Bob need to delete the part of the key which they used to estimate the error rate. It means that the original key will be shortened even more. We use notation R in Figure 3.3 to mark the length of the key after this phase.

Error correction

When Alice and Bob are sure that the key distributed via the quantum channel has a low error rate, they must find and correct or delete all errors in the rest of the key. This phase is known to be highly interactive and time-consuming, since the discussion about the location of errors in the key is performed through the public channel. The cascade protocol [52] is the most widely used reconciliation protocol due to its simplicity and efficiency. It is run iteratively in the given number of iterations where random permutation of the key is performed with the objective to evenly disperse errors throughout the key. Next, the permuted key is divided into equal blocks of k_i bits, and after each iteration permutations are performed again and the block size is doubled: $k_i = 2 \cdot k_{i-1}$. For each block, Alice and Bob exchange the results of the parity test and perform a binary search to find and correct errors. Instead of going through all the iterations continuously, the cascade protocol investigates errors in pairs of iterations. The process is recursive. This means that no bits are discarded during the first iteration. It also means that for any error corrected in the second iteration there must be at least one matching error contained in the same block in the previous iteration, since neither error was found or corrected in that iteration. For this reason, for each correction made in any iteration after the first one, a binary search is rerun on the block containing the bit corrected in all previous iterations, in order to identify any potential matching errors. For any new error detected, it follows that another error in a previous iteration was masked, thus the process is repeated so that the error detection and correction process cascades through all previous iterations. This process is illustrated in Fig. 3.4, where the following notation is used: e_i represent identified errors, e_m represent masked errors, and e_c represent errors that have already been corrected.

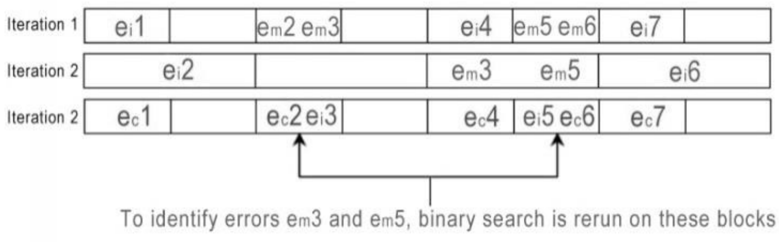

Figure 3.4. Error detection process in cascade protocol

The length of the initial block k_1 is a critical parameter, and should depend on the estimated error rate. An empirical result in [52] indicates that the optimal value of k_1 is 0.73/p, where p is the estimated QBER. The cascade protocol is modified in [53], with the aim of reaching the theoretical limit for protocol efficiency. From these results, it is clear that four iterations are

Table 3.4. Percentage of corrected errors per iteration

Iteration	I	II	III	IV
Percentage of corrected errorrs	54.5223%	45.3478%	0.4517%	0.002%

sufficient for a successful key reconciliation, as suggested in [52]. However, since the initial block length depends on the estimated error rate, it is necessary to perform all iterations. The number of iterations i is increased to the value for which the length of block k_i can be used to split the raw key into two parts ($k_i < \frac{n}{2}$). Now, let us go back to the parity check results. If the parity of a block disagrees between Alice and Bob, they perform a binary search on that block with the aim of identifying the single bit error. The binary search consists of dividing the block in half and comparing the parity check results for the divided block until the error is located. This means a maximum of $1 + \lceil \log_2 k_i \rceil$ parity bits are exchanged for each block with an error bit, since $1 + \lceil \log_2 k_i \rceil$ is the maximum number of times block k_i can be divided, and one parity bit is exchanged for blocks without errors. In order to minimize the amount of information gained by Eve, it is advisable to discard the last bit of each block and sub-block for which the parity bit was exchanged. Now if we define L_i as the maximum number of leaked bits, and k_i as the length of the block in the i^{th} iteration, it is clear that:

$$\sum L_i = \sum_i \Big(\sum_{\substack{initially\\even\\blocks}} 1 + \sum_{\substack{initially\\odd\\blocks}} (1 + \lceil log_2\, k_i \rceil) + \sum_{\substack{other\\errors\\corrected}} \lceil log_2\, k_i \rceil \Big) \tag{3.4}$$

This can be shorted [54] to:

$$L = \sum L_i = \sum_i \Big(\frac{n}{k_i} + \sum_{\substack{errors\\corrected}} \lceil log_2\, k_i \rceil \Big) \tag{3.5}$$

where $k_i = 2 \cdot k_{i-1}, k_i < \frac{n}{2}$ and n is the length of the initial key. The number of leaked bits depends on the initial block size and error rate. However, from the results presented in [53] it is evident the majority of errors are corrected in the first two iterations.

We mark the length of the key after the key reconciliation phase as F in Figure 3.3. More details about the cascade protocol and error correction can be found in [30].

Privacy amplification

Alice and Bob finally have an identical key without errors, but since Eve may have gained significant knowledge of the key, Alice and Bob should strengthen their privacy.

This is done by deleting some of the bits of the final key, so the raw key is shortened even more. The number of rejected bits during the privacy amplification process is defined in Equation 3.6 [55], where G is the number of bits that need to be discarded and n is the length of the key (B). We mark the length of the key after this phase as A in Figure 3.3.

$$\frac{n \cdot 2^{-G}}{log 2} < 1 \tag{3.6}$$

It is important to mention that the quantum channel only allows simplex (one-way) communication while Alice and Bob can send classical messages forth and back over the public channel. The public channel does not rely on the laws of quantum physics, so an eavesdropper can listen without penalty to all the communication that is in progress on this channel. In order to prevent a man-in-the-middle attack over the public channel, situations where an eavesdropper is able to change the messages that are being sent, the authentication of the public channel is required. It is advised to exchange hashed values of received key after sifting and error correction phases, as is depicted in Fig. 3.3.

Authentication

Due to intended level of security of QKD which is ITC secrecy in the presence of an adversary with unlimited computer power and memory, in QKD, no restrictions is put on adversary's computational power and storage capability. Since the communication over the quantum channel relies on the laws of quantum physics, the easiest way for attack is the public channel. The public channel without authentication is like any other channel susceptible to a man-in-the-middle attack. To avoid a man-in-the-middle attack by Eve, communication over the public channel must be authenticated.

The authentication problem was described in the original BB84 paper [2], where the authors proposed a solution based on universal families of hash functions introduced by Wegman and Carter [5]. Nowadays, more effective symmetric authentication methods are known, but the Wegman-Carter authentication is often used in literature since it describes an upper bound for the needed symmetric authentication key [50, 56, 57]. Secure authentication of the public channel requires from both communication parties to pre-share identical random strings[2]. To simplify this, Alice and Bob must pose a small secret key which is used to select a hash function from the family to generate an authentication hash that will be used on the public channel. This secret key should not be reused ever again. Fortunately, a complete authenticated conversation can validate a large number of new shared secret bits from QKD and later, a small portion of these bits can be used for the further authentication process. Therefore, QKD does not create a secret key out of nothing. To be precise, it expands a short secret key into a long one, so – strictly speaking – QKD is a technique of key-growing.

In QKD, there are two types of authentication: immediate[3] authentication and delayed authentication. Immediate authentication implies the authentication of messages immediately after they are received while the delay authentication implies the authentication for all messages exchanged during the session together to be done at the end of the session. There are variations in the details, but all QKD protocols contain authentication. In this chapter, we follow the approach from [30] where authentication is performed only two times. The first time, before error correction phase, where Alice and Bob authenticate the outcome of the measurement. This authentication is necessary to prevent intercept/resend attack. Finally, the authentication is done at the end of the session in order to verify that the key is indeed identical on both sides.

The author in [56] divided authentication schemes into two categories: Information-theoretically secure (ITS) and computationally secure message authentication schemes. Also, comparative analysis of Wegman-Carter, Sinson, Boer, Bierbrauer, Krawczyk and a novel authentication scheme is performed. In [59] it was shown that Wegman-Carter authentication which is based on ASU2 (Almost Strong Universal2) hashing is very well suited for authentication in QKD. To perform authentication it is necessary to sacrifice a certain part of the key and an upper bound for the key needed for authentication is defined with by the following equation:

$$k_{upperbound} = 4 \cdot (b + \log_2 \log_2 a) \cdot \log_2 a \tag{3.7}$$

where a is the length of the message which needs to be authenticated and b is the length of authentication tag.

Finally, it means it is necessary to exchange one authentication message to verify measurement on the quantum channel where the length of the message which needs to be authenticated is the length of $log_2 Q$ [bits], and it is necessary to verify a key of length P. The amount of key which need to be sacrificed for authentication can be calculated using Equation (3.8):

[2] Indeed, there are authentication schemes that do not require pre-shared secrecy, but these schemes are not proven to be ITC secure. The aim of QKD is to establish a secret key and provide ITC security, so the same level of security needs to be guaranteed for all components involved.

[3] Often named as „continuous authentication“. More details can be found in [58]

$$k_{auth} = 4 \cdot (b + \log_2 \log_2 \log_2 Q) \cdot \log_2 \log_2 Q + 4 \cdot (b + \log_2 \log_2 P) \cdot \log_2 P \tag{3.8}$$

Now it is easy to compare the length of the key from each of previous steps as shown with Equation (3.9), while more details about the key reduction through QKD protocol phases can be found in [MPTV15, MNV15]:

$$Q > B > R > F > A \tag{3.9}$$

3.1.4 BB92 protocol

In order to simplify BB84 protocol, Benner developed B92 protocol by catching the essence of non-distinguishable quantum states in the simplest way [60]. In order to establish a secret key using B92 protocol, the sender hereinafter named Alice and the recipient hereinafter named Bob, must follow following communication steps. First, Alice needs to generate a random binary sequence $S_A = [11001011]$ of length Q and use non-orthogonal polarizations to modulate the photons, where i refers to the i^{th} bit of a sequence S_A:

$$|\varphi\rangle = \begin{cases} |0\rangle & \text{if } S_A[i] = 0 \\ \frac{|0\rangle + |1\rangle}{\sqrt{2}} & \text{if } S_A[i] = 1 \end{cases} \tag{3.10}$$

Then, Alice sends modulated photons over the quantum channel to Bob. Since Bob has no information about the sequence S_A which Alice used, he needs to generate his random sequence $S_B = [00101010]$ and apply identical non-orthogonal polarizations rules ((3.10)) to his sequence S_B in order to measure the incoming photons. If Bob chooses the correct basis, he will measure the incoming photon. However, if he chooses the wrong basis, he will not measure anything; a condition in quantum mechanics which is known as an erasure [61]. Finally, after completion of the measurement, Bob will inform Alice about the positions in which he received incoming photons. Let Q be the length of the raw key, then log_2Q-bits are necessary to tell Alice the measured positions. Then, Alice and Bob will discard the bits corresponding to the photons which Bob measured with an incorrect basis. Note that Bob does not reveal anything about the basis he used. However, we can conclude that the original length of the key Q is significantly decreased since Bob reliably receives approximately 25% of the original key.

B92 consists of identical post-processing stages (sifting-extraction of the raw key, error rate estimation, error correction, privacy amplification and authentication) as previously explained for BB84 protocol.

3.1.5 Key Length Calculations

It is difficult to predict the precise length of the final key (A) since it depends on the error rate in the quantum channel and the Eve's influence [MPTV15]. It also depends on the number of leaked bits and the number of iterations needed to discover the errors in the key. A short key cannot be used for strong encryption, while a longer key can be shortened. As such, it is necessary to define the length of the raw key (Q) which will result in a usable length of the final key (A). To summarize, we present the Equation (3.11) for calculating the length of the raw key (Q) from the length of the final key (A), error rate and parameter *level of security* S.

$$Q = T \cdot \left\lceil A + \frac{\log \left\lceil \frac{\frac{Q}{T}}{\log 2} \right\rceil}{\log 2} + S \cdot \frac{Q}{2} + L + k_{auth} \right\rceil \tag{3.11}$$

where:
S is a percentage of raw key (Q) used for calculating QBER,
L represents a number of bits leaked during the key reconciliation phase,
A denotes the length of the final key and

$$T = \begin{cases} 2 & \text{for BB84,} \\ 4 & \text{for B92} \end{cases}$$

Solving Equation (3.11) for Q involves the Lambert W-function, also known as the Product Logarithm function. The value of L for different values of error rate is shown in Table 3.5, while the lengths (Q) of the raw key based on the length of the final key (A) and error rate for different security levels S are shown in Fig. 3.5 and 3.6. Table 3.5 shows how the maximum number of leaked bits (L) increases with error rate. It is easy to see that the values of leaked bits for error rate 0.01 increases more than double for error rate 0.05.

Table 3.5. Maximum number of leaked bits (L)

Key Length (in bits)	QBER			
	0.01	0.05	0.1	0.15
1280	45	182	385	537
640	26	94	194	271
480	22	72	147	204
256	15	41	80	111
192	13	32	61	84
160	13	27	52	71
128	12	23	42	57

The length of the final key increases with the rate and the level of security. The maximal tolerated quantum bit error rate (p_{max}) is defined as 12.9% [30, 62], and for QBER values that are below this threshold ($p \leq p_{max}$) Equation 3.11 provide very accurate results. It is worth noting that the influence of eavesdropping is not included in the Equation (3.11) since the entire QKD process will be repeated if the estimated QBER is higher than maximal tolerated QBER.

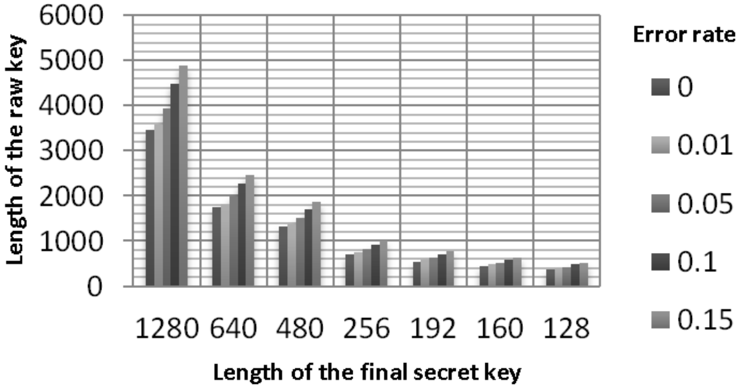

Figure 3.5. Length of raw key (Q) based on final secret key (A) and error rate in BB84; $S(K) = 0.06 => 25.30\%$

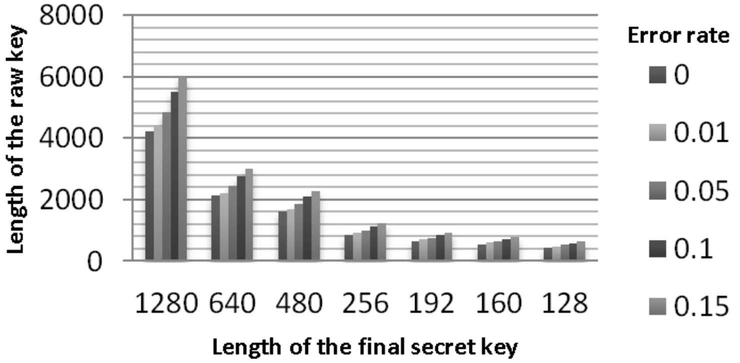

Figure 3.6. Length of raw key (Q) based on final secret key (A) and error rate in BB84; $S(k) = 0.11 => 39.10\%$

Also, it is worthwhile to mention the idea of establishing direct and confidential communication between users by sending the message directly on the quantum channel. This idea is known as "Quantum Secure Direct Communication" [63], but it was soon recognized that such communication will not give desired results. Firstly, it is not possible to make such robust message against losses (previously mentioned the reduction of the key), and secondly, there is no analogy of privacy amplification, so if an eavesdropper Eve intercepts such communication, she will be able to obtain the raw message. More theoretical and practical details about QKD protocols can be found in [24, 58, 64–66].

3.2 QKD Link

QKD employs two distinct channels between communicating parties: the quantum channel, which is used for transmission of quantum key material encoded in certain photon properties such as polarization or phase, and the public channel, which is used for verification of exchanged key material. The combination of these two channels forms a QKD communication link[4], over which QKD allows two remote users to exchange specific type of data, for example, secret keys. The existence of these two channels is both a drawback and an advantage of QKD when compared to conventional connections. The disadvantage is that the quantum channel has a limited key generation rate and cannot be used over arbitrarily long distances while the advantage is that the quantum channel guarantees the ITS secure key distribution based on the laws of quantum physics. Secure links can only be established between two parties that are connected by a direct optical fiber or free line of sight in a point-to-point (P2P) manner for a certain distance due to absorption and scattering of polarized photons [65–68]. This distance depends on the quality of fiber, type of optical source and detector, and on the QKD protocol which is used.

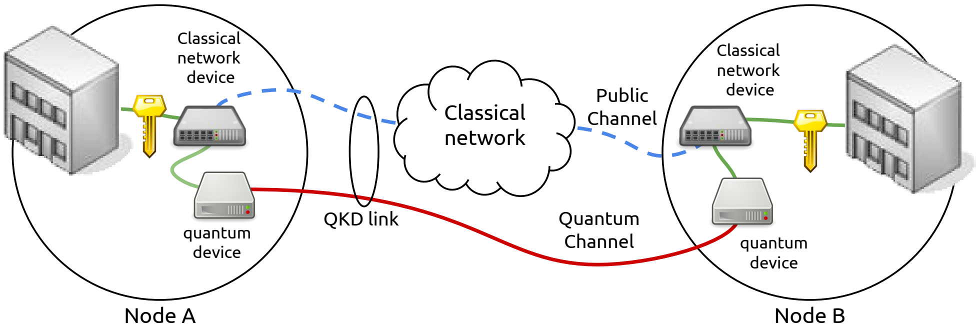

Figure 3.7. Overview of a QKD link which consist of an optical quantum channel (continuous red line) and a public/classical channel (dashed blue line)

[4] From now on referred to as the "QKD link" or simply the "link"

Although fiber is a good and commonly used medium to transmit qubits, the installation of a dedicated optical channel for QKD purposes (frequently known as dark fiber) is not practical for all eventualities[5]. A free space link is highly convenient, although it has its drawbacks since it needs a directly visible light path, good atmospheric conditions and an acceptable Signal-to-Noise ratio (SNR) which severely limits the usage time. Nevertheless, the results obtained from the experiment in Los Alamos [74] and the experiment in Munich, where the link between the ground and an airplane flying at 290 km/h was established [75], are promising towards a connection to a satellite in space [74–77].

The maximum distance of link decreases with increasing losses and increasing optical detector noise. For a given detector and settings, the detector dark-count[6] rate is constant, but the key rate decreases with increasing distance due to cumulative losses. For current systems, the distance at which QKD link is possible is roughly limited to 100 km in optical fibers, while the key rate is currently restricted to a few tens or hundreds of kbps depending on the distance [64, 68]. This is unacceptably low if the keys are to be used with an OTP cipher for high-speed traffic flows. It may be acceptable if network resources are used optimally, which requires effective QoS-aware solutions to manage network resources and provide guarantees for different classes of traffic.

An equally important feature of QKD link is the avarage amount of key material that can be established in a unit of time. This amount may vary due to humidity, temperature, the stability of devices, global radiation, pressure, dust, sunshine duration or other factors [30, 64]. However, it mostly depends on the length of the link and it is often referred to as the *key generation rate* or simply *key rate* (Fig. 3.9). The key rate is still an important aspect of further research since the key rate determines the type of encryption which can be used.

Due to the limited key rate, links are organized in the following way: both endpoints of the corresponding link have key storages which are gradually filled with the new key material and subsequently used for the encryption/decryption of data flow [30]. The type of used encryption algorithm and the amount of network traffic to be encrypted determines the speed of emptying the key storage, often referred as key consumption rate, while the key rate of the link determines the key charging rate [7, 30][MNV15]. If there is no enough key material in the storage, encryption of data flow cannot be performed [79] and QKD link can be characterized as "currently unavailable". Key material storage has a limited capacity [30] and QKD devices constantly generate keys at their maximum rate until key storages are filled [29].

To provide information-theoretically secure (ITS) communication, the key tends to be applied

[5] Although there are studies that analyze the use of bright light for data communication and the quantum signal in a single optical fiber [69, 70], in previously deployed QKD networks only dedicated optical connections (dark-fibers) has been used for QKD purposes [71–73].

[6] A dark count is an event where a single-photon detector clicks although there is no photon [78]

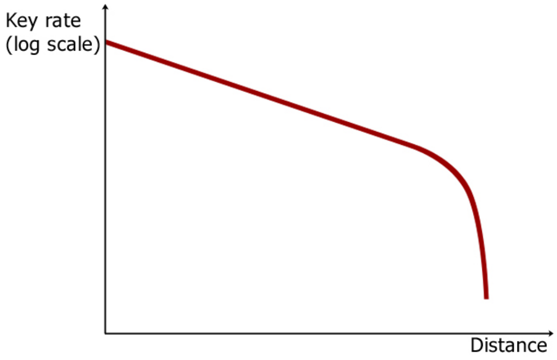

Figure 3.8. Typical key generation rate versus distance for a single QKD link

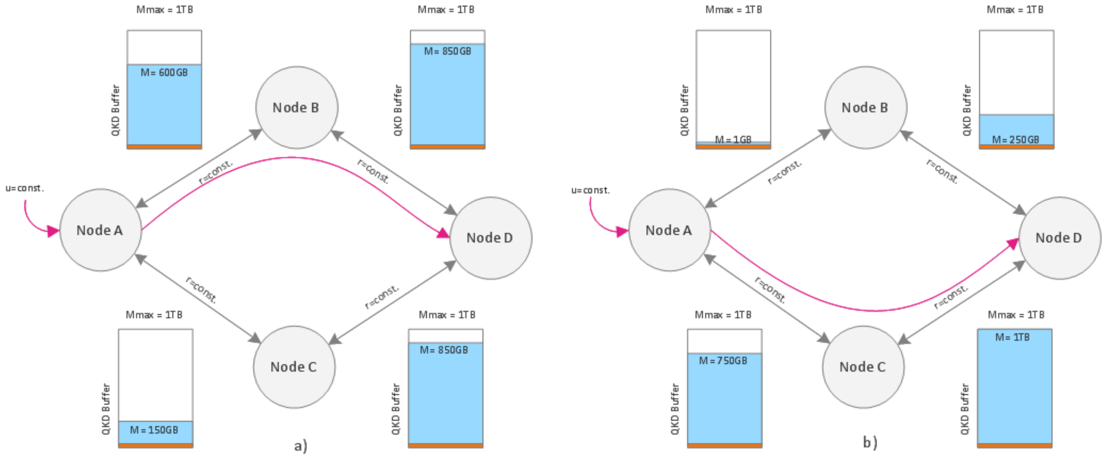

Figure 3.9. A simple example of routing with consideration of the amount of key material in key storages. In the case of communication between nodes A and D, node A needs to make a decision on choosing the best path A-B-D or A-C-D, which depends on the state of network links. Initially node A chooses the path A-B-D (left) and switch to path A-C-D (right) when the path A-B-D become unavailable due to lack of key material.

with a One-Time Pad (OTP) cipher and authenticated using an information-theoretically secure message authentication scheme such as Wegman-Carter when communicating over the public channel [49, 50]. As a result, ITS communication consumes always more bits than the length of the message secured which has an impact of the key consumption rate. However, if the ratio between the charging and consumption rates is not appropriate [15, 16], OTP cannot be used due to the lack of key material, and using less secure algorithms that do not require too much material such as Advanced Encryption Standard (AES) becomes inevitable [12].

3.2.1 QKD Packet Encapsulation

Since there are no defined standards for QKD network packets [80, 81], there are different variants of encapsulation of QKD traffic. In this section, we provide short details about the solution used in AIT R10 QKD Software [82].

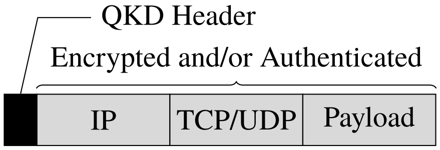

Figure 3.10. QKD Packet Encapsulation

To facilitate the ease of routing operation, in practice, encryption and authentication are performed between data link (L2) and network (L3) ISO/OSI layer as shown in Fig. 3.10. The packet is encapsulated in a QKD header which contains authentication tag and information about used encryption and authentication scheme [MKM^{+}16] as shown in Fig. 3.11. Note that QKD header is not encrypted but it is authenticated. Table 3.6 provides short explanation of QKD header's fields[7].

In the case of realization of QKD network as an overlay network, the encapsulated packet is directly forwarded to the underlying network. However, in the case of realization of QKD network as a network with a single TCP/IP stack, the packet is further encapsulated in MAC header and forwarded to the network interface card.

[7] It should be noted that currently there is no defined standard for QKD header. QKD Header shown here is the one used in version R10 of AIT QKD software which is further analyzed in this document.

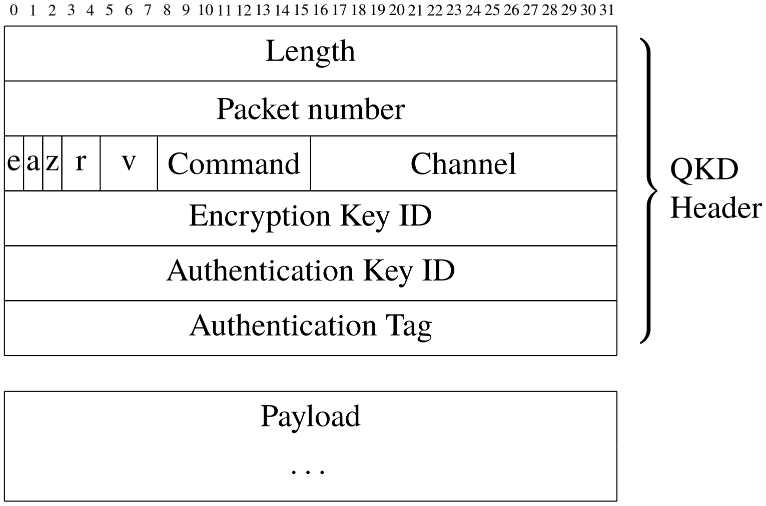

Figure 3.11. QKD Header

Table 3.6 shows that QKD header carries no information on the type of encryption algorithm and the authentication cipher which is used. QKD header contains field "Command" which is used for internal sub-protocol key management operations (LOAD, STORE, ACKNOWLEDGE and other) [82]. In chapter 7 we propose modification and extension of QKD header.

Table 3.6. QKD Header Fields

Field	Length	Description
Length	32 bits	Total packet length in bytes
Packet number	32 bits	The packet number
e	1 bit	The packet needs to be encrypted
a	1 bit	The packet needs to be authenticated
z	1 bits	Compressed (Zipped)
v	3 bits	Version
r	2 bits	Reserved for further use
Command	4 bits	Control Packet Command
Channel	4 bits	The Channel ID
Encryption Key ID	32 bits	ID of Key used for Encryption
Authentication Key ID	32 bits	ID of Key used for Authentication
Authentication-tag	32 bits	Authentication tag
Payload	-	Data payload

3.3 QKD Network

To enable secure communication between multiple clients and extend the range of QKD systems to longer distances, QKD network is introduced. QKD network is composed of usually static nodes representing secure access points that are considered to have the unlimited processing power and power supply. Due to the point-to-point behavior of the links connecting the nodes, previously mentioned testbeds [6, 7, 83] have shown that the secure key in the QKD network can be transmitted from node to node in a hop-by-hop manner [72] or in a key relay concept [7, 78]. Both methods rely on the assumption that all nodes along the path between the sender and the receiver in QKD network must be fully trusted [79, 84]. This restriction can be overcome when multiple path-based communication or Quantum Network Coding [85] is used. In this chapter, previously deployed QKD networks are discussed in short with focus on the methods of communication, routing protocols and network organization.
To facilitate the organization, the QKD network can be divided into three separate layers [86]:

- Quantum layer where a secure symmetrical key is established,
- Key management layer used to verify and manage the previously established key,
- Communication layer where the established key is used to secure data traffic.

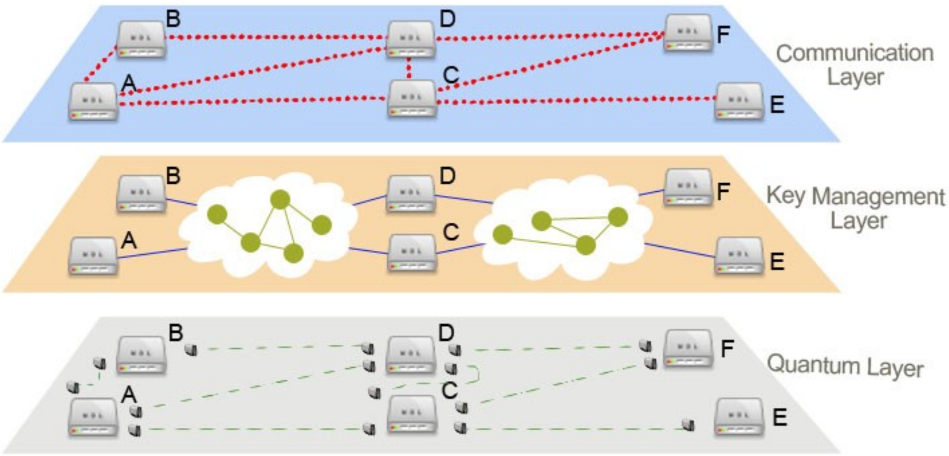

Figure 3.12. QKD network hierarchy

Taking into account the comprehensive and detailed documentation of quantum optical communication [58, 64–66], the focus of this thesis is placed on the two upper layers. These layers can have entirely different and independent network organization since communication between the nodes is realized by the existing standard connection, such as the Internet, which may include an arbitrary number of intermediate devices (Fig. 3.12). The key management layer is in charge of managing key storage resources, routing protocol, QoS and other. The topmost communication layer uses previously established key material for the encryption of data traffic using an existing security protocol suite such as Internet Protocol Security (IPsec) [71, 87]. Such a hierarchy distributes the responsibility for security in all three layers.

3.3.1 QKD Network Attributes

QKD represents a new generation of security solutions that do not rely on mathematical insolvable problems. However, QKD network must be integrated into the existing environment and needs to meet certain criteria and conditions. Some of the most common requirements that are required by QKD networks are listed below.

Key Rate

One of the vital parameters describing QKD network is the average key rate of QKD link. Since encryption and decryption operations cannot be performed without sufficient key material, the competition between the rate at which key material is stored in the key storage, and the rate at which it is consumed for encryption and decryption operations, has a major influence on network performance.

Through chronological comparisons of previously deployed QKD networks and testbeds, it is clear to notice a rapid improvement in the development of quantum equipment. QKD systems implemented in 2002 in DARPA QKD network were able to achieve a key rate of approx. 400 bps over 10 km [7]; In 2007 in SECOQC, the maximal key rate was 3.1 kbps over 33 km [6], while the solutions presented in 2009 in Tokyo achieved a key rate of 304 kbps over 45 km [83]. In 2014, QKD system connecting the cities of Hefei-Chaohu-Wuhu (HCW) in China with a total of nine QKD nodes[8] was reported in [10]. The whole QKD network has been operating for more than 5000 hours starting from 21 December 2011 until 19 July 2012 and it was installed in Anhui provincial telecommunication network of China Mobile Ltd., with over 150 km coverage area. The HCW QKD network consists of two metropolitan networks: the Hefei QKD network, which has five nodes [8, 88, 89], and the Wuhu QKD network [90–92], which has three nodes. These two networks were connected with an intercity QKD link, which combined Hefei and Wuhu metropolitan area QKD networks through a trusted intermediate node in the Chaohu Branch of China Mobile Ltd. [10]. The maximal key rate in the HCW QKD network was 16.15 kbps

[8] In the Hefei-Chaohu-Wuhu wide area QKD network, 13 QKD devices within nine nodes in total were employed to support the two metropolitan area QKD networks and the intercity QKD link [10].

between West Campus and North Campus nodes over a 3.1 km long link connected to an optical switch located in the Campus Library [10].

It should be noted that although key rate results of up to 1 Mbps have been achieved [83, 93–95], such solutions are limited to very short distances. In the future, it is reasonable to expect an optimal solution which will significantly exceed the present key rate and distance values although the race between the generation and consumption of key material will remain.

The Length of the Link

One of the shortcomings of QKD link is a limited length of realizations. Namely, due to absorption and scattering of polarized photons [65–68], the quantum channel can be realized by a direct optical fiber or free line of sight only for a certain distance. It is interesting to compare the lengths of the links of previously built QKD networks[9]. The maximal length in the DARPA QKD network was a 29 km long connection through the optical switch between Harvard and Boston Universities [7]. In SECOQC, the maximal length of the link was 82 km between the BREIT and St.Pölten nodes [6], while in Tokyo the maximal connection between the nodes was a record 90 km between the Koganei-1 and Koganei-2 nodes [96]. In HCW, the maximal distance was 85.1 km via the HCW intercity link between Hefei and Chaohu [10].

Therefore, for current systems, the distance at which a QKD link is possible is roughly limited to 100 km in optical fibers, while the stable key rate is currently restricted to a few tens or hundreds of kbps depending on the distance [64, 68].

Protection of Key Material

The main reason for interest in QKD is the privacy of the established key material. This means that the nodes of the QKD network can be sure with a strong probability that the established key material is unique and inaccessible to third parties [45]. The security of key material is evaluated not only in its establishment but also in its management, storage and finally the use. Therefore, it is very important to ensure the safety of each level of the QKD network architecture.

Key Usage

Due to the scarce resources that link provides in the aspect of establishing a new material (generation key rate), communication in the networks is reduced to a minimum since each additional

[9] Other QKD systems have also been reported, usually with a significantly smaller number of nodes and lower key rates. In October 2007, the Geneva Sate Chancellery installed a QKD system to secure a link transmitting ballot papers to the counting station during the federal elections in Switzerland [21]. QKD has also been used to secure a communication link at the 2010 FIFA World Cup in Durban, South Africa [23]. More details about practical applications of QKD can be found in [24].

packets means spending an additional amount of previously established key materials. Given that the main objective of QKD is to provide ITS communication, and since the communication is usually performed on a hop-by-hop basis where trustworthiness of all nodes in the path is required, it is necessary to choose the shortest routing path in order to minimize the number of nodes which can potentially be abducted or attacked by an eavesdropper. Since longer paths involve a higher consumption of key material, minimizing the number of hops is also preferred.
In the case of congestion or some problems in communication, a key material used is deliberately discarded and new key materials for retransmission is used in order to reduce the risk of possible leaks [30].

Robustness

Because of the cost and manner of implementation, it is expected that the QKD network will be slowly integrated into traditional and everyday environments. Thus, it is important to ensure the robustness of the QKD network, which is reflected in a gradual and seamless addition of new nodes and the establishment of new links in the QKD network. QKD network needs to provide adequate replacement paths that will avoid nodes that are defective or under severe attacks. Regardless of all the techniques of security protection, it is always important to remember that an attacker can easily find a way to terminate the optical link and cease the QKD connection. The QKD network needs to have an adequate response to even those situations.

3.3.2 Lack of Quantum Router in Practice

The basis of every modern networks are devices that have the ability to manage and control network traffic. One of the fundamental network devices is a router that that connects at least two networks and have traffic directing functions.
Although there are theoretical and pioneering results in the field of quantum repeaters and quantum relays [97–99], in practice they remain unachievable with current technology [68, 100]. The idea behind realizing a quantum router is to use quantum entanglement of photons to communicate over different quantum channels. Quantum entanglement is a key element for applications of quantum communications and quantum information. In short, quantum entanglement means that multiple particles are linked together in a way such that the measurement of one particle's quantum state determines the possible quantum states of the other particles. Even when the particles are separated by a large distance they still make up the joint quantum system. A measure which is used to describe how well the entanglement between two subsystems is preserved in a quantum process is defined as entanglement fidelity.
Whereas in theory the application of entangled states and entanglement swapping there are two main drawbacks of these concepts: first, the larger the distance between two entangled systems,

the lower the fidelity will become. In fact, the fidelity of a quantum state decreases exponentially with the distance of its qubits due to noisy quantum channels [68, 101]. In this context, the concept of entanglement purification [102, 103] can be used to increase the fidelity of one single entangled state by using a number of noisy entangled states (as described in [98]). Nevertheless, this will increase the resources (i.e., the number of entangled states) required for transmission of one qubit over a quantum repeater. Second, to realize a quantum repeater following the scheme in [98, 99] quantum memory is required. Since entanglement swapping as well as entanglement purification relies on performing quantum operations on several qubits, these qubits have to be stored in the quantum repeater for a certain amount of time. Although implementations of quantum memories exists, which can store a qubit for several milliseconds up to one second or even more, this is still too short for a practical use.

Therefore, each node in QDK network acts as a router and forwards each others' packets to enable information sharing between QKD hosts.

3.3.3 QKD Overlay Network

An overlay network utilizes the services of an existing underlying network in an attempt to implement better services and one of its most important features is the independence of the path that is offered by Internet service providers (ISP). In simple words, the crux of this technology is to find an alternative route that can avoid routes of poor quality, quickly switch communication via this route or even use multiple paths at the same time. The use of multiple network paths between a pair of hosts has been proposed for a wide variety of network technologies in order to achieve higher bandwidth of data transfers, to select paths with low latency, to balance load in the network, or to protect against failures [104–107]. A measurement study of a large ISP showed that almost 90 percent of point-of-presence pairs have at least four link-disjoint paths between them [108, 109]. In addition, interconnected autonomous systems usually exchange routing information using an exterior routing protocol such as Border Gateway Protocol (BGP) [110], which is known to be slow in reacting and recovering from network events. Previous measurement studies have shown that BGP may take tens of minutes to reach a consistent view of the network topology after a failure [111]. In addition, since BGP advertises only one route, network nodes are prevented from seeing alternative paths, including paths they might prefer.

Due to the possibility to circumvent such problems, overlay networks are typically spawned between end-nodes that share resources with each other in a peer-to-peer fashion which means that the network traffic is encapsulated into the traffic of the underlying network. An example of an overlay network model is shown in Fig. 3.13. Specifically, overlay networks have been attracting attention as a way of adding new functionalities that are hard to implement in a physical network. Due to its ability to easily bypass "untrusted" nodes, QKD network is usually deployed in an overlay point-to-point manner [29, 30].

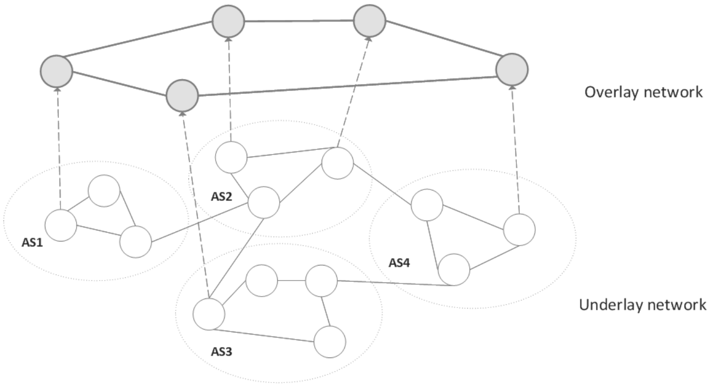

Figure 3.13. Illustration of Overlay Network Model between multiple autonomous systems (AS)

3.3.4 QKD Network Types

Although there have been many proposals and various hybrid realizations, QKD networks can be roughly divided into two distinct categories:

- Switched QKD network
- Trusted repeater QKD network

Switched QKD network

Switched QKD networks consist of nodes which are connected to a dedicated fully optical network. This network contains a switching mechanism that is used to establish a direct optical point-to-point QKD connection between any two nodes in QKD network. Due to the mentioned distance restriction of the point-to-point QKD link, these networks are restricted to a metropolitan or regional scale [100]. Since every optical switch adds, at least, several dB of loss to the photonic path, these networks may significantly reduce the network's reach.
The main drawback of switched QKD networks is the existence of dedicated optical infrastructure for quantum channels, which is often not economically feasible. On the other hand, the major advantage of this class of networks is the reliance on the optical switch which enables to establish a connection between two nodes without the active participation of other network nodes (Fig. 3.14). Another drawback of switched QKD networks is the consistency of the QKD technique used. It is not possible to combine different QKD techniques, such as free-space QKD and QKD over dark-fiber since there is no appropriate device in the path that could perform such transformation. The first switched QKD network was the DARPA QKD Network by implementing an optical crossbar switch [7].

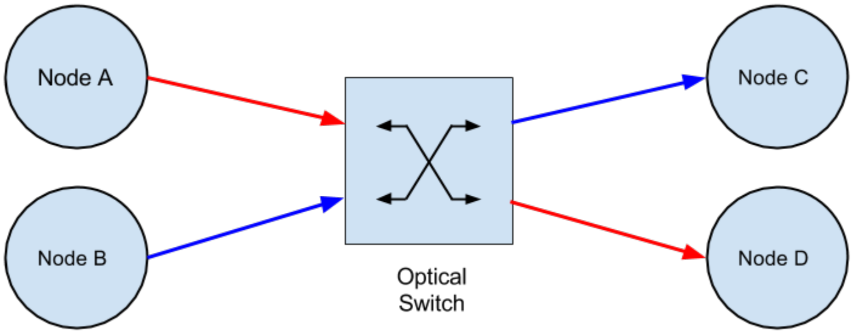

Figure 3.14. Topology of switched QKD network

Trusted repeater QKD network

Trusted repeater QKD network consists of nodes that are specially protected and that are believed to be secure (hence the name). All nodes are equal and the security of each node in the path of data transmission is required for secure transmission of information. The point-to-point communication between two nodes provides identical keys to the nodes and thus enables a secure communication. Taking into account the lack of quantum router, nodes are also responsible for routing and forwarding mechanisms, as is depicted in Fig.3.15. Such organization of network represents its greatest drawback since the safety of transfer depends on the security of all the nodes in the path.

Trusted repeater networks are not limited in distance or number of nodes and these networks can be built out of different QKD devices, implementing different QKD technologies.

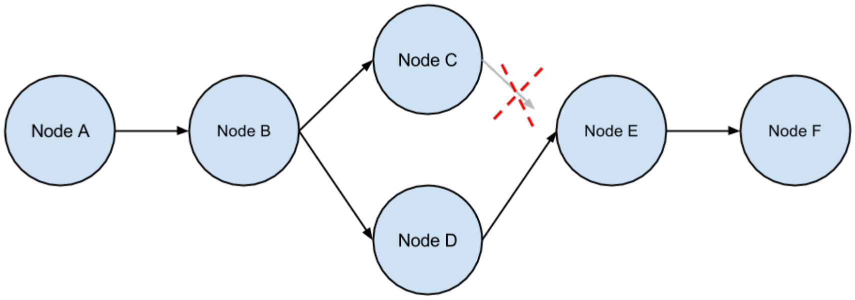

Figure 3.15. QKD Network with Trusted Relays

3.4 Real-Time Communication

Generally speaking, traffic flows can be roughly distinguished into two broad categories:

- Elastic traffic, typically referred to as data traffic where the transmitted information is not time sensitive but requires eventual proper delivery such as email, web browsing, file transfers and similar,

- Inelastic traffic, typically referred to as real-time traffic for which the transmitted information is only useful if it is received within a given amount of time, such as VoIP, IPTV, video conferences, time-dependent interactive applications and similar.

The most important aspect of inelastic traffic is that a message generated by the source device by an application is time sensitive, and it must be received by the destination device within a given amount of time. If the time between the generation of the message at the source device and its reception at the destination device, which is defined as delay, exceeds this time constraint, the message is considered lost, regardless whether it is ever received at the destination. Also, in packet-switching networks, it is possible that the aggregate rate of the input traffic to the network (or a portion of the network) temporarily exceeds the network's capacity, in which cases packets may experience long delays or get dropped by the network nodes. This is referred to as network congestion. Different applications react differently to the starvation of resources. A real-time multimedia stream may be completely unable to decode meaningful content whereas a file download can be delayed slightly. To decide whether there are enough resources, the current network load needs to be known. For predicted services, the current network load is based on measurement; for guaranteed services, it is based on the prespecified characterization of existing connections.
In general, network needs to provide two types of capabilities to provide QoS [112]:

- A packet network needs to be able to differentiate between classes of traffic so that the end users can treat one or more classes of traffic differently than others,

- After distinguishing of traffic classes, network needs to be able to treat these classes distinctly by providing resource assurance and service differentiation within the network.

According to the ITU-T recommendation, Quality of Service (QoS) refers to "a totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service" [113]. QoS is defined as a set of mechanisms able to share fairly various resources offered by the network to each application as needed in order to provide the desired quality if possible to every application. QoS is characterized by a certain number of parameters depending on the application, and it can be defined as the degree of user satisfaction with the services provided in the network [114, 115].

For real-time traffic transmission, the perceived audio and video quality are the most important QoS metrics as they are related directly to the quality experienced by end users (QoE). Factors influencing the perceived quality – delay, delay variation (jitter), bandwidth and packet loss – are the four main parameters which determine the QoS of perceived voice/video. There are other factors affecting end-to-end perceived audio and video quality such as echo, noise, cross-talk, etc. [116, 117]; however, these factors mainly depend on the practical implementation of the network, and they are not considered in this thesis. Summarizing the above, the basic requirements for QoS in QKD networks can be stated as follows:

- Bandwidth on demand,
- Low end-to-end delay,
- Low delay variation – jitter, Acceptable error/loss rate without retransmission as the delay would be unacceptable with retransmission for real-time traffic,
- Some data flows require authentication, while others demand encryption and authentication in the same time and depend on the amount of key material available in key storages (known as short "key bandwidth"),
- The time until QKD link will be available or unavailable, which mainly depends on the ratio between the charging and consumption key rate.

QoS parameters can be roughly classified into additive (e.g., delay) and min-max (e.g., available bandwidth). In the case of an additive parameter, the QoS value of a path is equal to the sum of the corresponding weights of the links along that path. For min-max parameters, the QoS value of a path is the minimum (or maximum) link weight along that path. It should be noted that there are also multiplicative parameters in which the QoS parameter along the path is equal to the product of QoS parameters of the links (e.g. loss probability). However, by taking the logarithm of multiplicative parameters, the problem boils down to additive parameters.

The first four parameters listed above are the most commonly specified requirements on the Internet, and in the case of QKD networks they are related exclusively to the public channel. The other two parameters – the amount of key material available in key storage and the time until the link is available or unavailable –mainly depend on the quantum channel of the link. Note that for high-quality and secure communication in QKD networks, it is necessary to find a compromise between the requirements.

Different applications have different requirements, and in practice the user should be able to specify the QoS level required. For example, for telephone or videophone services, the end-to-end delay is crucial. An upper limit for a one-way delay is 150 ms according to the guidelines of the ITU-T's G.114 recommendation. This end-to-end delay includes the total time taken to capture, digitize and encode/compress audio/video data, encrypt and transport it from the source to the destination, and decrypt, decode and display it to the user. The default G.729 codec requires packet loss to be significantly lower than 1% to avoid audible errors [118]. Additionally, VoIP calls require constant bandwidth which depends on the codec used. Therefore, some of the parameters are more important for communication than others, which means some parameters can tolerate fluctuations without affecting communication quality. However, it is more intuitive and convenient for applications to specify QoS requirements and the required level of guarantee. In general, three QoS levels regarding guarantees are distinguished [119]:

- Hard or deterministic, which implies that the user requirements must be met in full,
- Soft or statistical, where user-specified QoS requirements should be met to a certain specified level,
- Best effort where no guarantee is provided at all.

Today's multimedia and real-time applications consume high network resources and demand high flow rates and short transfer delays. Existing QKD networks, in their original state, are not able to satisfy these QoS requirements. It is known that providing QoS in the network requires the propagation of QoS information within the network; the key to the support of QoS reporting is QoS routing, which provides path QoS information at each source. Given the specificity and purpose of QKD networks, in order to support QoS for real-time traffic in QKD networks we need to know the minimum delay on the path to the destination, the bandwidth available on it, the amount of key material in key storages of the links on the path, and the time until the storages are filled or emptied. For example, with a VoIP call setup, a connection should be accepted only if resources are available, such as enough key material in key storages to enable secure communication, sufficient bandwidth, and acceptable end-to-end delay and jitter. Otherwise existing communication in the network is disrupted. Routing with QoS indications is thus needed to manage network resources efficiently.

Chapter 4

Aims

This chapter lists the specific goals of the research as described in the Ph.D. thesis proposal of the candidate.

4.1 Aim 1: A new proposal for a Quality of Services-aware computational model in QKD networks, with an emphasis on real-time traffic

The primary goal of this thesis is to provide an appropriate Quality of Service (QoS) solution for QKD networks by addressing the following issues:

- Due to the scarce resources that QKD links provide in the aspect of establishing a new material (generation key rate), it is necessary to answer the question whether the reservation of resources in such a network is the best approach. More precisely, it is necessary to provide an answer to the question about which QoS model best suits the needs of QKD network.
- Determining a suitable path for high-quality communication in QKD networks; more precisely, it is necessary to provide solutions which will offer efficient routing in QKD network.
- Differentiating traffic in QKD networks based on flow priorities; it is necessary to provide a solution which will show a clear distinction between traffic of different priorities. In this case, the higher priority traffic needs to be prioritized in the network when network resources are allocated.

4.2 Aim 2: Verification of the proposed model in QKD network

Taking into account the high cost of quantum technology, the second objective of this thesis is the verification of the QoS aware computational model in QKD network simulators. Since QKD simulators are currently unavailable, this section will make a significant contribution to the research community and the implementation will result in the development of the first QKD computing environment simulation.

The ability to simulate QKD networks will improve access to further development of quantum technology such as research in the field of optimal topologies for QKD networks. A simulated model of QKD network will provide an adequate platform for further testing of QKD networks with an unlimited number of QKD nodes and QKD links. Additionally, such an environment is necessary for testing and verifying results obtained from Aim 1 of this thesis.

Chapter 5

A Novel Proposal of Quantum Key Distribution Network Simulation Tool

After the design of a new network solution, a researcher has typically several possibilities to evaluate and validate obtained results. Analytically theoretical quantifying the performance and complex behavior of even simple network protocols in the aggregate is often impractical. It uses mathematical models to evaluate network performance where queuing theory is one of the most common tools in network performance studies. Unfortunately, theoretical analysis of networks containing a large number of nodes and links is a demanding process, since the mathematical constructs get very complex for realistic considerations. A simulation is an essential tool for computer networks research. A simulation environment offers the creation of complex network topologies, a high degree of control and repeatable experiments, which in turn allows researchers to conduct exactly the same experiments and confirm their results. Also, the simulator provides an easy way to manipulate desired parameters while setting other parameters fixed. The simulation represents a simpler and more comfortable way of looking at the problem, with a further reduction of costs in relation to the practical testbed. Taking into account the high cost of deploying of QKD network testbed, we present here the first computing simulation environment of QKD network.

5.1 The QKD Network Simulation Module

In contrast to previously developed simulation tools that are focused on the quantum channel and QKD protocols, the QKD Network Simulation Module (QKDNetSim) described in this chapter focuses on the public channel and simulation of QKD network. Hereof, QKDNetSim seeks for a maximum simplification of QKD link and puts the focus on network performance measurement and estimation, routing protocols, packet encapsulation, traffic management including generation and consumption of key material. More specifically, QKDNetSim is intended to facilitate additional understanding of QKD technology with respect to the existing network solutions. It seeks to serve as the natural playground for taking the further steps into this research direction (even towards practical exploitation in subsequent projects or product design).

5.1.1 Requirements

The aim of QKDNetSim project was not to develop the entire simulator from scratch but to develop the QKD simulation module in some of the already existing well-proven simulators. Therefore, QKDNetSim was developed to meet the four fundamental requirements which are listed below:

Accuracy

Although the primary goal of the simulation is to use abstraction to reduce the complexity of the analyzed problem, a very important aspect is the accuracy and credibility of the obtained results. Essentially, a good simulator is credible if it is validated and reliable network design decisions can be based on it. It is required from the simulator to reflect real-world scenarios as much as possible. However, due to the high degree of abstraction of the reality which carries a significant simplification as well as faster simulating performances, simulators are often unable to map the real problems and they do not fit precisely with real-world measurements. To reduce the level of abstraction and make obtained results closer to the real-world measurements, but still with the aim to retain the benefits of the simulated environment, the concept of emulators is introduced. An important advantage of emulation environments over simulation environments is the possibility of validation against real traffic. On the other hand, the advantage of emulation environments over real-world experiments is the possibility of scaling to larger topologies by multiplexing simulated elements on physical resources such as network interfaces and other.
Therefore, for basic simulation platform, we sought the one that is well-tested and well-proven. We looked for the credible simulator that is already well accepted in the scientific and research community, allows reliable simulations and has the emulation capacity.

Extensibility

Expandability is reflected in terms of ease of upgrade, that is, implementation of new models, new solutions in the field of QKD and other technologies. Without this, the development of solutions that is not modular, nor provides a useful baseline for the upgrade will quickly lead to usefulness. From a software engineering perspective, this implies the development of a common reusable code which can be utilized for various needs.

Usability

Usability is reflected in the aspect of ease of use and integration with existing solutions. Also, a very important aspect is the software support, that is, active cooperation of developers which is followed with discussions and latest updates, error reporting, clear and concise instructions and user manuals.

Availability

The QKD Network simulation module described in this chapter will be distributed as open-source freeware and should leverage and permit inclusion of other free and open-source networking software. Therefore, we looked for simulation platform that is open-source and available for extensions. The aim was to provide an opportunity for integration and testing of QKD technology with a variety of existing network solutions by minimizing the cost of the simulation as low as possible.

5.1.2 The Network Simulator NS-3

Network Simulator of version 3 (NS-3) is an open-source software which is licensed under GNU GPLv2 and welcomes developers in contributing code from across academia, industry, and government. The development of the simulator is followed by actively encouraging community participation by providing an open mailing list for user and active developer discussions, a tracker for error reporting and a wiki with user-contributed instructions. Also, NS-3 comes with instructions and descriptions of all simulator elements in the detailed *doxygen* web edition.

NS-3 is a discrete-event simulation written entirely in C++, with optional Python bindings, architected similar to Linux computers. NS-3 emphasizes emulation capabilities that allow NS-3 to be used on testbeds and with real physical devices and applications. This is achieved by introducing "Emu NetDevice" component which allows NS-3 simulations to send data on a physical network (Fig. 5.1). In addition, a "Tap NetDevice" allows a host from physical network to participate in a NS-3 simulation as if it were one of the simulated nodes. An NS-3 simulation may be constructed with any combination of simulated, Emu, or Tap devices. NS-3 consists of several already developed simulation modules such as Wifi, WiMAX, LTE, point-to-point, UAN and other [120, 121], which can be used to reasonably handle matters of the public channel in QKD network. NS-3 has widespread use reported in the scientific literature and it has been proved to be a quality successor to the previously popular NS-2 simulator [121]. Taking this into account, we decided for NS-3 as the basic simulation platform.

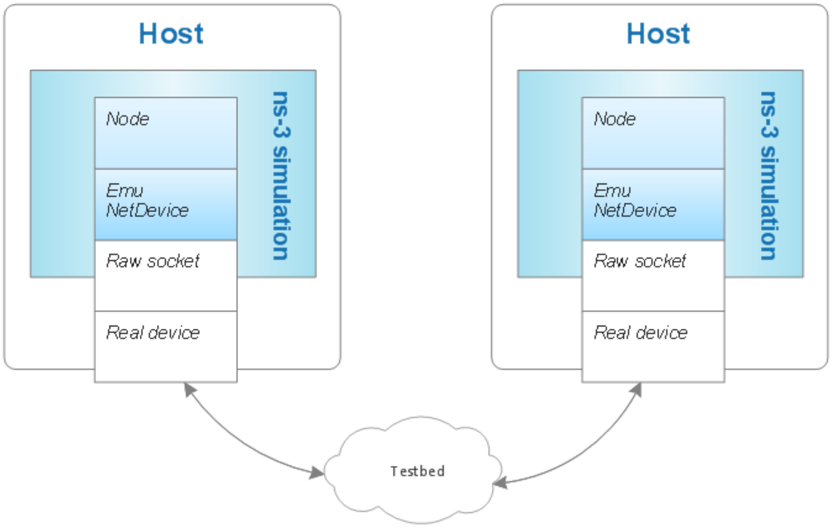

Figure 5.1. Example implementation of testbed consists of two separate hosts, each running a subset of a "global" simulation. Real hardware is used to connect the hosts instead of virtual NS-3 channel which allows NS-3 applications and protocol stacks attached to a simulation node to communicate over real hardware

5.1.3 Design

Due to the nature of QKD link, QKD network is mainly limited to metropolitan scale [10, 122, 123] in which the network can be geographically divided into multiple domains (autonomous systems) or in the simplest case it can be a simple network in a single domain. Clearly, the geographic distribution of the network dictates the addressing and routing between nodes. Taking the assumption that these networks are not limited only to QKD traffic (i.e. the network is utilized for various type of applications which can lead to dynamic and unpredictable link performance), the implementation of a routing protocol between multiple domains may not be a trivial task. In a simplified scenario, in which the network is implemented in a single domain and is exclusively intended for QKD traffic, the problem of addressing and routing is much simplified. To support different simulation scenarios in various situations, our network simulation model allows the simulation of QKD network in both cases. In the following sections, we discuss overlay QKD network, while the implementation of QKD network with a single TCP/IP stack is discussed in section 5.1.3.

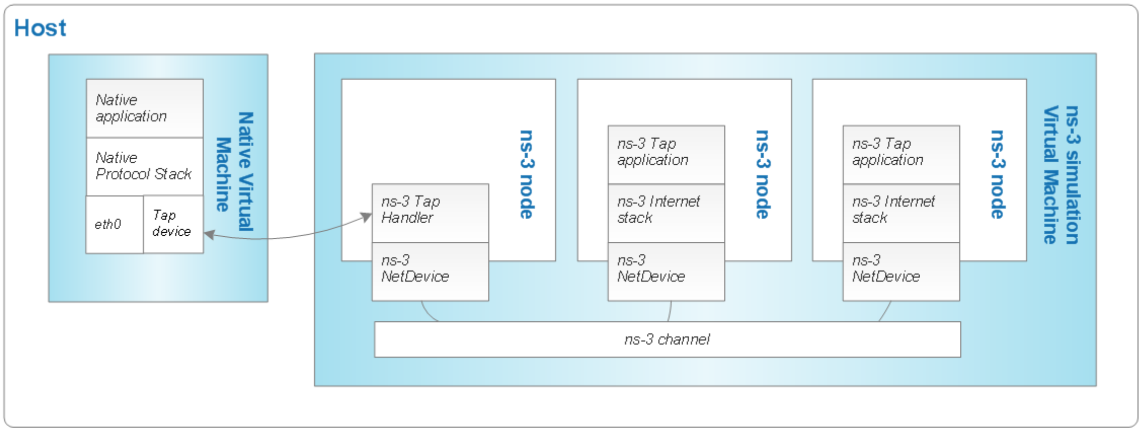

Figure 5.2. Implementation overview of emulated channel. This figure shows host which runs several virtual machines (VM) including virtual machine which runs NS-3 simulator (on the right) and virtual machine running native (Linux) application and protocol stack (on the left). This machine is connected with simulation by a TAP network device and NS-3 tap handler which allows the tap devices on the native VMs to behave as if they were NS-3 net devices in the simulation VM. This allows the native software and protocol suites in the native VMs to believe that they are connected to the simulated NS-3 channel.

Overlay TCP/IP Stack

QKDNetSim allows realization of overlay network which can be used for various purposes regardless of QKD network. To ensure the independence of the underlying network, each QKD node implements an overlay TCP/IP stack with an independent overlay routing protocol as shown in Fig. 5.3. During the development of QKDNetSim, we aimed to minimize changes to the existing core code of NS-3 simulator. But, still, to provide independent overlay networking, QKDNetSim implements following classes in the *internet* module of NS-3 simulator:

- virtual-ipv4-protocol
- virtual-tcp-l4-protocol
- virtual-tcp-socket-base
- virtual-tcp-socket-factory
- virtual-tcp-socket-factory-impl
- virtual-udp-l4-protocol

- virtual-udp-socket
- virtual-udp-socket-factory
- virtual-udp-socket-factory-impl
- virtual-udp-socket-impl

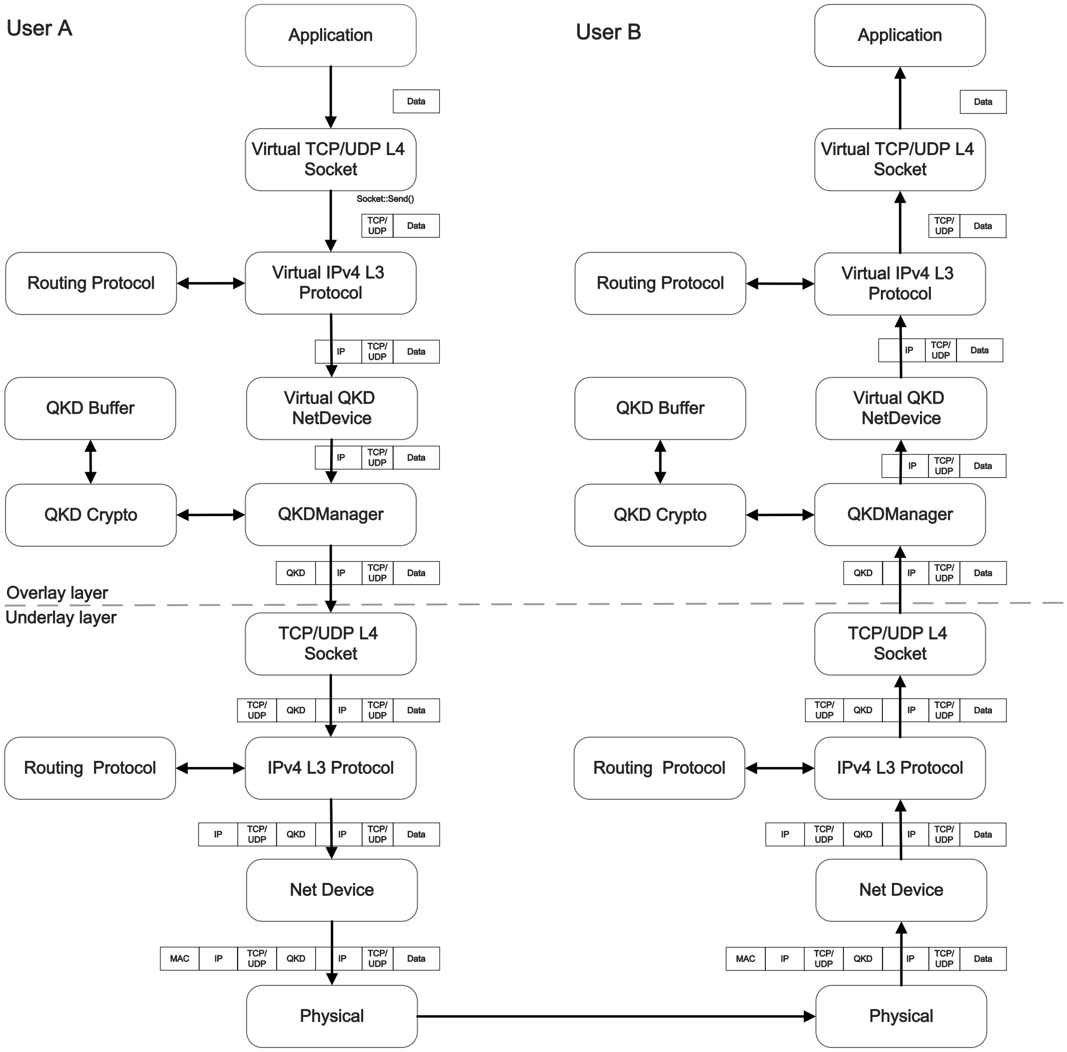

Figure 5.3. Packet Encapsulation in QKD overlay network

Class *ipv4-protocol* keeps a list of all IPv4 address associated with IPv4 interfaces. Thus, to distinguish IP addresses of underlying and overlying network *virtual-ipv4-protocol* class is introduced. This implies the implementation of independent overlay TCP and UDP L4 protocol classes which pass packets to *virtual-ipv4-protocol* instead of original *ipv4-protocol*. Additionally, realization of independent TCP and UDP L4 protocol classes allows simple modification of overlying L4 communication without affecting the underlying TCP/IP stack.

QKD Key

QKD key is an elementary class of QKDNetSim. It is used to describe the key that is established in QKD process. QKD key is characterized by several meta-key parameters, of which the most important are the following:

- key identification (ID)
- key size
- key value in *std::string* or *byte* format
- key generation timestamp

QKD Buffer

QKD keys are stored in QKD buffers which are characterized by following parameters:

- QKD is also known as Quantum Key Growing since it needs a small amount of key material pre-shared between parties to establish a larger amount of the secret key material. The pre-shared secret key serves to guarantee the integrity of the protocol in the first transaction and it should not be used for any other purposes except to establish a new key material [MNV15]. The amount of pre-shared key material for that purpose is denoted with M_{min},
- The key material storage depth M_{max}, used to denote the maximal amount of keys that can be stored in QKD buffer,
- The current value $M_{cur}(t)$, representing the amount of key material in QKD buffer at the time of measurement *t*, where it holds that $M_{cur}(t) \leq M_{max}$,
- The threshold value $M_{thr}(t)$ at the time of measurement *t* is used to indicate the state of QKD buffer where it holds that $M_{thr}(t) \leq M_{max}$. During the simulation, $M_{thr}(t)$ can be static or variable depending on the state of the network. An algorithm for calculation of M_{thr} is considered in Section 7.3.1 .

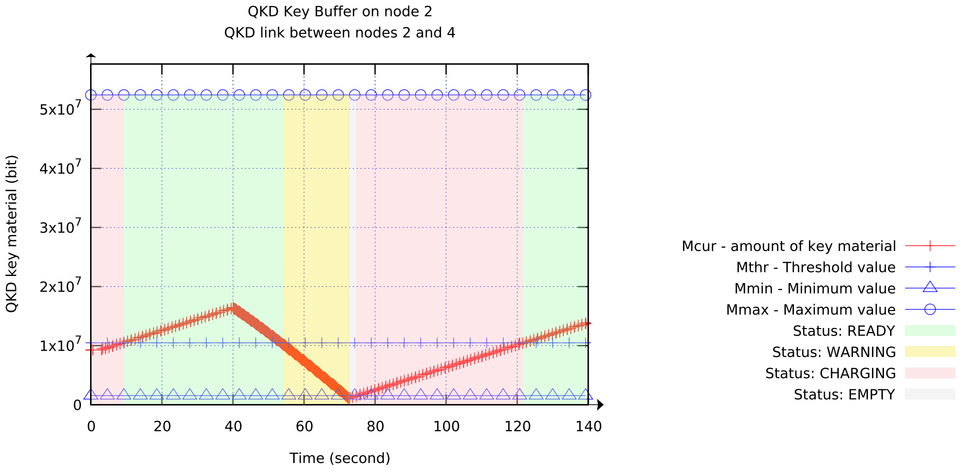

Figure 5.4. Graphical representation of QKD Buffer status generated using QKD Graph

As shown in Fig. 5.4, QKD Buffer can be in one of the following states:

- READY - when $M_{cur}(t) \geq M_{thr}$,
- WARNING - when $M_{thr} > M_{cur}(t) > M_{min}$ and the previous state was READY,
- CHARGING - when $M_{thr} > M_{cur}(t)$ and the previous state was EMPTY,
- EMTPY - when $M_{min} \geq M_{cur}(t)$ and the previous state was WARNING or CHARGING.

The states of QKD buffer do not directly affect the communication, but it can be used for easier prioritization of traffic depending on the state of the buffer. For example, in EMPTY state, QKD post-processing application used to establish a new key material should have the highest priority in traffic processing. QKD post-processing applications are discussed in section 5.1.3.
Without loss of generality, QKDNetSim allows storage of the *virtual* key material in QKD buffers instead of the actual occupation of memory by establishing and storing the symmetrical key material in QKD buffers. Simply put, key material is not generated, nor it takes up computer memory, it is only represented by a number that indicates the amount of key material in QKD buffer. In network simulators, such operations are common since they reduce the duration of simulation and save computational resources. For example, instead of generating packets with real or random information in packet payload field, network simulators often generate empty packets. However, QKDNetSim allows the user to turn off this level of abstraction in a way to choose the real key material generation and storage in QKD buffers.

QKD Crypto

QKD crypto is a class used to perform encryption, decryption, authentication, atuhentication-check operations and reassembly of previously fragmented packets. QKD crypto uses cryptographic algorithms and schemes from *Crypto++* open-source C++ class cryptographic library. Currently, QKD crypto supports several cryptographic algorithms and cryptographic hashes including One-Time Pad (OTP) cipher, Advanced Encryption Standard (AES) block cipher, VMAC message authentication code (MAC) algorithm and other.
Also, QKD crypto implements functions for serialization and deserialization of the packet into a byte array which is used as the input in cryptographic algorithms and schemes.

QKD Virtual Network Device

To facilitate the ease of operation of the overlay routing protocol, encryption, and authentication of the incoming frame is performed on the data link layer prior leaving QKD network device (NetDevice). QKD NetDevice implements a sniffer trace source which allows recording of the overlay traffic in *pcap* trace files. Also, QKD NetDevices allows fine tuning of the MAC-level Maximum Transmission Unit (MTU) parameter which limits the size of the frame in the overlying network [MKM+16].
In practice, in overlay network, these devices are usually registered as Virtual QKD NetDevice since in the overlay network they do not add MAC header to the frame as shown in Fig. 5.3. Instead, MAC header is replaced with a QKD header which contains authentication tag and information about used encryption. In QKDNetSim, bond with QKD crypto is realized via QKD manager. Packets leaving QKD NetDevice are passed to QKD manager which is in charge of control of the cryptographic process. Similarly, the reception packet is processed (authentication check and/or decryption) before passing into the QKD NetDevice. Such implementation allows the use of other types of devices instead of QKD NetDevices, that is, use of different network technologies such as Point-to-Point, WiFi, WiMAX, LTE, UAN and other.
In the case of the network with a single TCP/IP stack, QKD manager is called from Traffic Control Layer that sits between NetDevice (L2) and IP protocol (L3). Placing connection to QKD manager in the same layer with waiting queues of Traffic Control Layer allows simple usage of various types of NetDevices and it follows previous work on encryption of the packet content above OSI data link layer (Fig. 5.3) [MKM+16] [82].

QKD Manager

QKD manager is installed at each QKD node and it represents the backbone of QKDNetSim. It contains a list of QKD Virtual NetDevices from the overlying network, a list of active sockets in the underlying network, a list of IP addresses of interfaces in the overlying and underlying network and a list of associated QKD buffers and QKD cryptos. Therefore, QKD manager serves as a bond between the overlying NetDevices and sockets in the underlying network. Since the QKD link is always realized in a point-to-point manner between exactly two nodes [30], QKD manager stores information about NetDevices of the corresponding link and associated QKD buffers. Using the MAC address of NetDevice, QKD manager unambiguously distinguishes QKD crypto and QKD buffer which are utilized for packet processing. Finally, QKD manager delivers the processed packet to the underlying network. Receiving and processing of incoming packets follows identical procedure but in reverse order.

QKD Post-processing Application

Key material establishment process is an inevitable part of the QKD network. It is performed using QKD protocols to provide a key to the participant of symmetric system transmission in a safe manner. Although there are several types of QKD protocols, they consist of nearly identical steps at a high level, but differ, among others, in the way the quantum particles or photons are prepared and transmitted over quantum channel [MNV15, MPTV15]. Communication via the public channel is referred as *post-processing* and it used for extraction of the secret key from the raw key which is generated over the quantum channel. Although there are differences in implementations, almost every post-processing application needs to implement following steps: extraction of the raw key, error rate estimation, key reconciliation, privacy amplification and authentication. Given that the focus of the QKDNetSim is placed on network traffic and considering that there are different variations of post-processing applications, QKDNetSim provides a simple application which seeks to imitate traffic that is generated by the real-world post-processing applications such as AIT R10 QKD software [82]. The goal was to build an application that credibly imitates traffic from existing post-processing applications to reduce the simulation time and computational resources. Fig. 5.5 shows the comparison of the traffic generated using QKDNetSim post-processing application and AIT R10 QKD Software. The Pearson correlation coefficients between the measurement and modeling results is 0.732.

It is important to emphasize that the impact of post-processing applications cannot be ignored in QKD network. Considering that node in a QKD network constantly generate keys at their maximum rate until their key storages are filled [29], in some cases, the traffic generated by the post-processing application can have a significant impact on the quality of communication over the public channel, especially when it comes to public channels of weaker network performance.

QKD Graph

QKD graphs are implemented to allow easier access to the state of QKD buffers and easier monitoring of key material consumption. QKD graph is associated with QKD buffer which

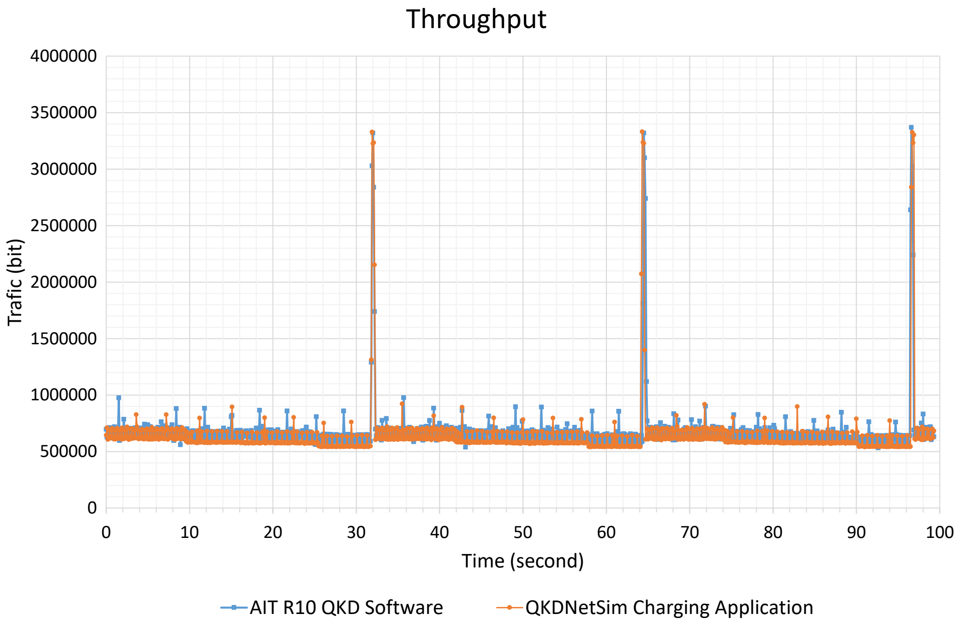

Figure 5.5. Comparison of the traffic generated using QKDNetSim post-processing (charging) application and AIT R10 QKD Software that was used for post-processing of keys in AIT quantum laboratory in April 2016. The traffic shown here was recorded in a local network, that is, without of the traffic generated by any other application.

allows plotting of graphs on each node with associated QKD link and QKD buffer. QKD Graph creates separate PLT and DAT files which are suitable for plotting using popular *Gnuplot* tool in PNG, SVG or EPSLATEX format. An example QKD Graph is shown in Fig. 5.4. QKDNetSim supports plotting of QKD Total Graph which is used to show the overall consumption of key material in QKD Network. QKD Total Graph is updated each time when key material is generated or consumed on a network link. An example graph is shown in Fig. 5.4.

QKD Helper

QKDNetSim comes with a helper class (QKD helper) which provides easier installation of QKD managers at network nodes, setting the parameters of QKD links and QKD buffers, and drawing of QKD graphs. Given that cryptographic operations are called from QKD managers, QKD module can be easily used in overlay networks with two TCP/IP stacks or in simple networks with a single TCP/IP stack. In the case of usage of the overlay network, it is necessary for applications to use

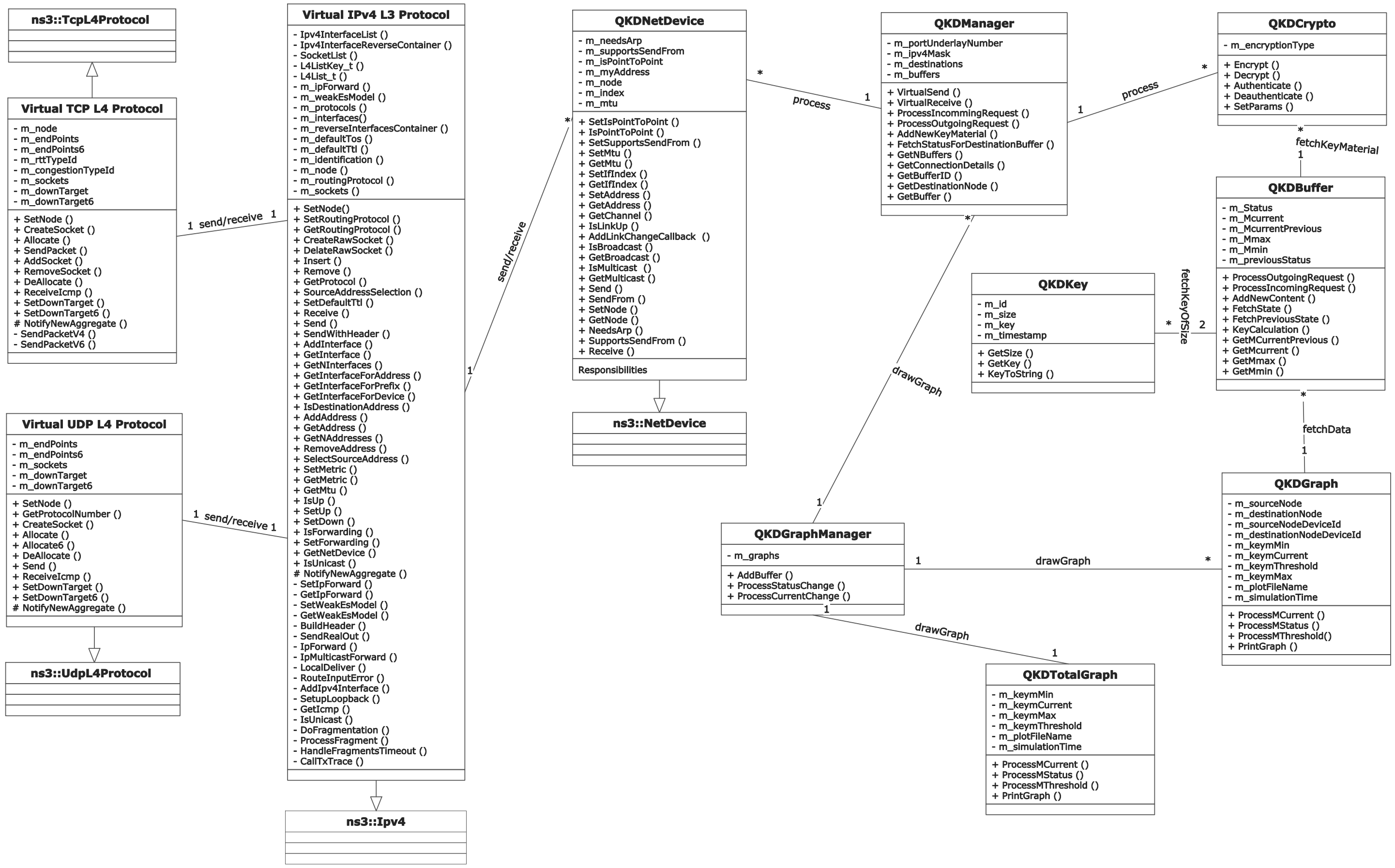

Figure 5.6. QKDNetSim class diagram

virtual-TCP-L4 or *virtual-UDP-L4* protocols which pass the packet to virtual-ipv4-l3 protocol. The packet is further passed to corresponding NetDevice and QKD manager. After performing cryptographic operations, the packet is finally delivered to the underlying network. In the case of a simple network when a single TCP/IP stack is used, standard *ns3::tcp-l4*, *ns3::udp-l4* and *ns3::ipv4-l3* protocols are utilized, while QKD helper sets sending and receiving ns3::callback from NetDevice to QKD manager to perform cryptographic operations. QKDHelper is realized to facilitate the procedure for the establishment of these configurations (Fig. 5.6).

5.1.4 Example of Use

QKDNetSim enables simple performance testing of routing protocols in QKD Network. Therefore, we set up a simulation with 7 fixed nodes forming the topology shown in Fig. 5.7 using single TCP/IP network stack. Table 5.1 presents the model parameters including the key generation rate, charging key rate, packet size, and data traffic parameters. The parameters not given here are the default parameters of the NS-3 simulator and QKDNetSim module.

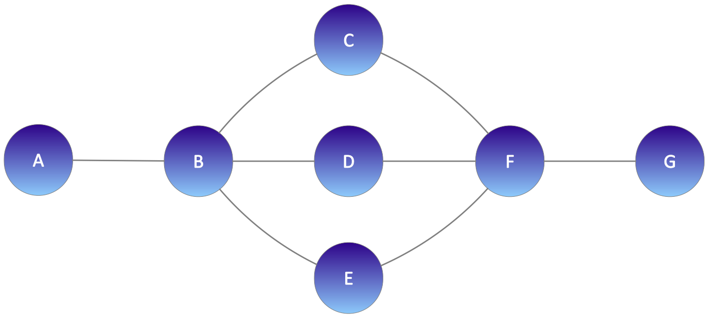

Figure 5.7. Topology of simulated single TCP/IP stack network in which encrypted data UDP flow of rate 300 kbps between nodes A and G is established.

Parameters in Table 5.1 indicate the links C-F, D-F, and E-F have the least amount of initial key material. As tested routing protocols have no information about the state of links, they need to choose between one of the three possible routes: A-B-C-F-G, A-B-D-F-G or A-B-E-F-G. Yet, after a while, the usage of any of these paths results in a disruption of communication since the available key material is quickly consumed and QKD links become "currently unavailable". Then, the routing protocol in use needs to choose an alternative route while the depleted link is charged and recovered. When the used link is depleted again, the routing protocol should switch to an alternate path and so on. It is important to stress that links without available key material are "currently unavailable" only for communication that requires the usage of key material but

Table 5.1. Parameter values of the simulation

Parameter	Value
Total number of nodes	6
Packet Size	512 Bytes
Packet Traffic Type	UDP; CBR
Packet Traffic Rate	300 kbps
Encryption type	OTP
Authentication type	VMAC
Authentication tag lenght	32 bits
Maximal amount of key material (all links)	6400 kBytes
Minimal amount of key material (all links)	128 kBytes
Threshold value (all links)	2000 kBytes
Traffic Queue Capacity (per device)	10000 Packets
Initial amount of key material (link A-B)	6400 kBytes
Initial amount of key material (link B-C)	6400 kBytes
Initial amount of key material (link B-D)	6400 kBytes
Initial amount of key material (link B-E)	6400 kBytes
Initial amount of key material (link C-F)	1600 kBytes
Initial amount of key material (link D-E)	1032 kBytes
Initial amount of key material (link E-F)	750 kBytes
Initial amount of key material (link F-G)	6400 kBytes
Charging key rate for all links (per second)	12.8 kBytes
Total Simulation time	300 seconds

IPv4 interfaces remain active for unencrypted communication. It means that routing protocol which does not measure the state of QKD buffers is not able to instantly register link availability changes.

On the other hand, given that the main objective of QKD is to provide ITS communication, routing packet needs to be encrypted and authenticated [30]. Therefore, in our simulation, each routing packet is encrypted and authenticated, and in the case, when there is no enough key material, routing packets are not transmitted. Consequently, the only way for routing protocol to detect the unavailability of the link is the detection of lack of routing packets and routing information. In our simulations, we used AODV, DSDV and OLSR routing protocols which are embedded in the NS-3 simulator by default.

It is important to note that when QKD buffer is in EMPTY state, that is when the amount of key material is below minimal threshold, the residual key material should be used only for the establishment of new key material [MNV15]. To realize such behavior, we implemented priority waiting queues which perform classification of network traffic using Differentiated Services Code Point (DSCP)/Type of Service (ToS) bits in the IP header. The traffic generated by QKDNetSim charging application has the highest priority while the traffic originated from routing protocol as well as user's traffic has the lower priority.

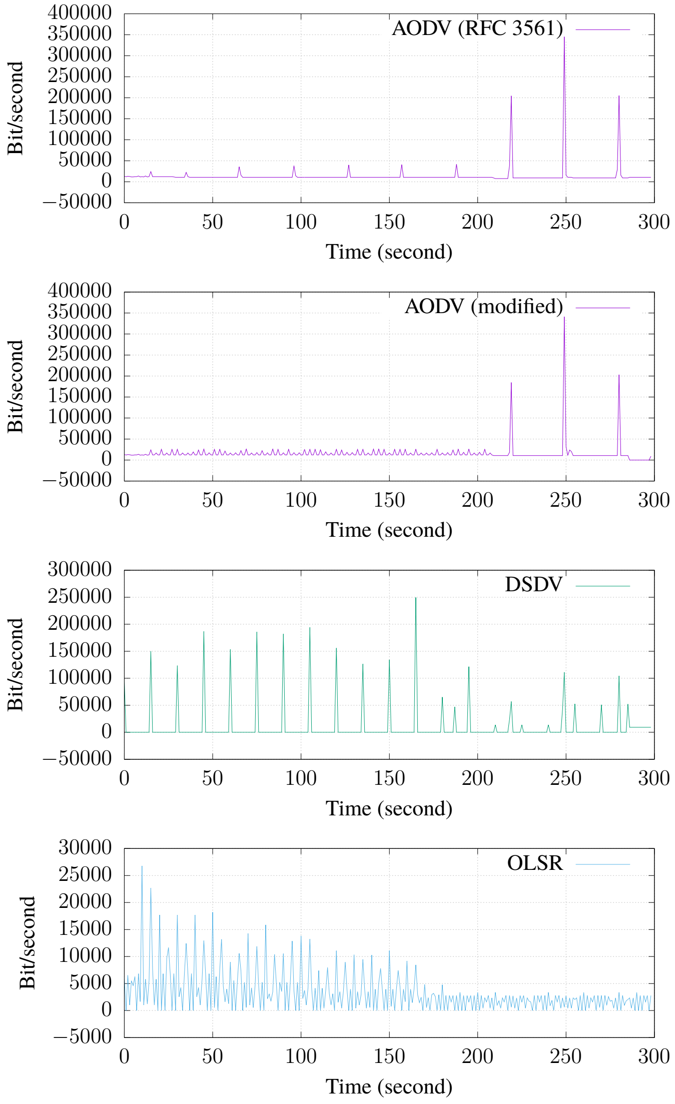

Figure 5.8. Comparison of traffic generated by routing protocols

Table 5.2 compares the obtained values based on the number of sent routing data and Packet Delivery Ratio (PDR) which is calculated as the ratio of received and sent application packets. These two values are used to assess the effectiveness of the routing protocol within the specified simulation environment.

Table 5.2. Comparison of the obtained values. The number of routing packets and Packet Delivery Ratio (PDR - which is calculated as the ratio of received and sent application packets) are used to assess the effectiveness of the routing protocol

	Routing Data		Application Data
Routing Protocol	Packets	Bytes	PDR (%)
AODV (RFC 3561)	10839	510952	33.7772
AODV (modified)	12887	611480	68.9868
DSDV	4786	338464	73.4228
OLSR	1620	125328	49.9641

Table 5.2 shows that AODV reactive routing protocol sends the largest number of routing packets. AODV is based on RFC 3561 and it floods the network with route request and hello messages which results in a large key material consumption. In our simulation, AODV uses only route A-B-E-F-G while two other routes are ignored. Given that AODV has no method for measuring the state of links (no QoS support), it assumes A-B-E-F-G is the best route toward the destination. When the key material on link E-F is depleted, path A-B-E-F-G is unsustainable but no alternative route toward destination is found. According to RFC 3561, AODV needs to update Active Route Life of the route which is used for packet forwarding to be no less than the current time plus ACTIVE_ROUTE_TIMEOUT which is set to 3 seconds by default. Therefore, forwarding node is in charge to extend the validity of route toward destination regardless of whether that route is feasible at all. In our example, it means that node B assumes that route toward destination G is valid over node E even in the case when there is no enough key material to forward packet over link E-F. Also, each time the link E-F is recovered (QKD buffers are charged with the new key material), AODV on node E floods for establishing of the route toward destination resulting in periodic peaks shown in Fig. 5.8 in a period of 0-220 second of simulation time. To emphasize the impact of policy in AODV forwarding mode, we modified the forwarding policy to obey updating of the route toward the destination in forwarding mode. In this case, AODV sends routing packets to refresh the route each time when the route expires and communication with the destination is a request, but it does not update the route toward the destination in forwarding mode at all. As shown in Fig. 5.8, our modification resulted in additional routing traffic but it improved the PDR for more than double. Also, our modification prevents futile consumption of key material on links A-B and B-E in the case when no route toward destination is feasible. In the period 220-300 of simulation time, the key material on links A-B, B-E and E-F is depleted and each time the new key material is generated, AODV requests for route toward destination G resulting in large peaks as shown in Fig. 5.8. It is important to note that AODV maintains the obtained route as long as that route is used. Since QKD devices constantly generate keys at their maximum key rate until the key storages are filled [29], routes to neighboring nodes are constantly maintained.

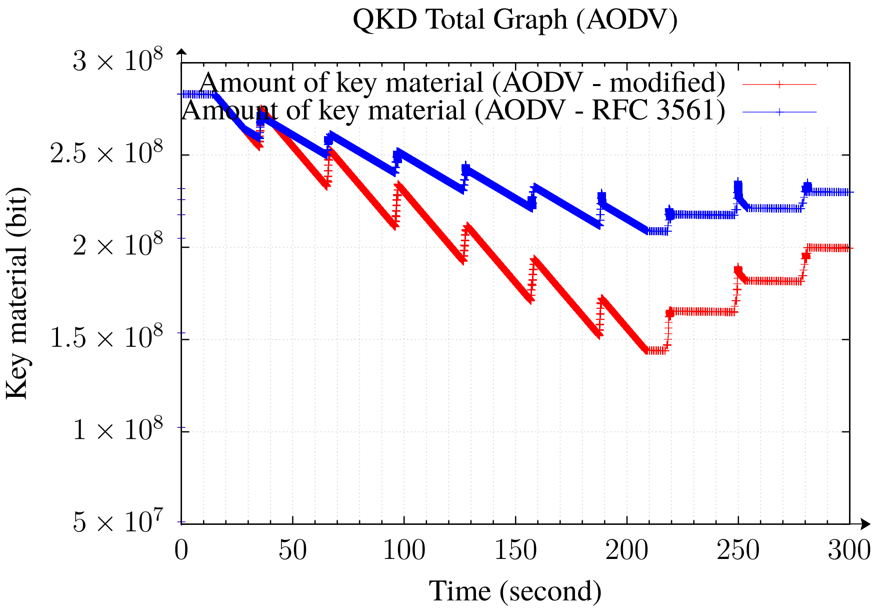

Figure 5.9. QKD Total Graph shows the total consumption and generation of key material in the network when AODV routing protocol is used. AODV is known as a reactive routing protocol where routing paths are searched only when needed, mainly by flooding the network. The discovery procedure terminates when either a route has been found, or no route is available after all route permutations have been checked. AODV selects route A-B-E-F-G as the best route to the destination and stores it in the cache to reduce flooding the network. Once the link E-F is available the communication between source and destination is established consuming the available key material (visible on the graph on second 250). Otherwise, no communication between source and destination is available (visible on the graph in period from 220-250 second).

DSDV is the most popular proactive routing protocol based on the distributed Bellman-Ford algorithm. Each node is in charge to periodically broadcast its routing table to first neighbor nodes (one-hope away) by using periodic update packets based on *periodic route update interval* which is set to 15 seconds by default (Fig. 5.8). After receiving the update packet, the neighbor node updates its routing table by incrementing the number of hops by one and forwards the packet further in the network. The process is repeated until all the nodes in the network receive a copy of the update packet with a corresponding value. In addition to regular periodic updates, DSDV uses triggered updates when the network topology suddenly changes. The purpose of triggered updates is to advertise the information that has recorded since the last periodic update. However, if a periodic and triggered update occurs in a short period of time, the values may be merged and only the periodic update will be performed [MFV+16].

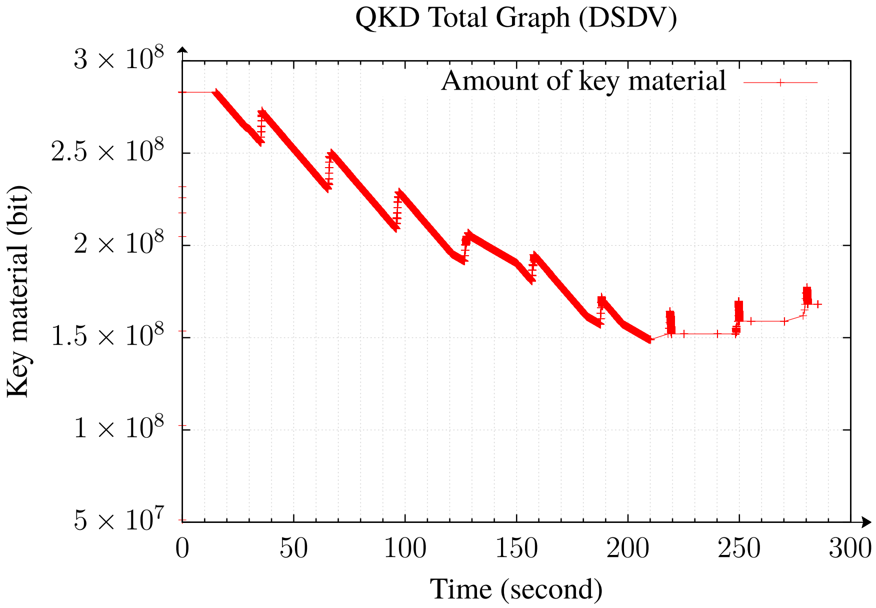

Figure 5.10. QKD Total Graph shows the total consumption and generation of key material in the network when DSDV routing protocol is used. A higher consumption of key material in the network as the available route toward the destination is found is shown with a steeper curve on the graph. The gentle slope of the curve reflects key material consumption only on the source node which assumes that the route to the destination is available. However, the traffic is discarded on intermediate nodes due to the lack of key material for further forwarding toward the destination.

In the beginning of the simulation, DSDV uses route A-B-E-F-G and switches to route A-B-D-F-G when key material on the link E-F is depleted. Thereafter, DSDV uses routes A-B-C-F-G and combines all three route toward destination. DSDV does not have QoS methods to measure the state of links and since all links have the same performances in the view of delay, DSDV favors the latest received information assuming the freshest information from the network. Fig. 5.10 shows the total consumption and generation of key material in the network when DSDV routing protocol is used. This figure shows a higher consumption of key material in the network as the available route to the destination is found which is shown with a steeper curve on the graph in simulation time periods: 15-122, 128-130, 150-183, 187-197, 218-220, 248-250, 280-282. Given that DSDV detects a lack of routing information only after 15 seconds (no triggered updates since the IPv4 interfaces remain active), the source node assumes that the route to the destination is available and sends the encrypted traffic which is silently discarded on intermediate nodes due to the lack of available key material for further packet forwarding. Therefore, less key material consumption in the network is noticeable and shown with gentle slope of the curve on the graph in simulation time periods: 122-128, 130-150, 183-187 and 197-218.

It is important to underline that the key material establishment process takes about 30 seconds as shown in Fig. 5.5 while the periodic update interval of DSDV is set to 15 seconds by default. Hereof, it is likely that in the process of key material establishment DSDV will notice the lack of periodic update from the node which is connected with a link with the depleted key material. Then, DSDV deletes the entry in the routing table toward that node which interrupts key material establishment process and leaves the link in the locked position. Due to the lack of a route it is not possible to generate new key material and due to the lack of the key material, it is not possible to exchange routing packets and update routing tables. Such a scenario is registered for link C-F which remains locked after second 160 and disables route A-B-C-F-G.

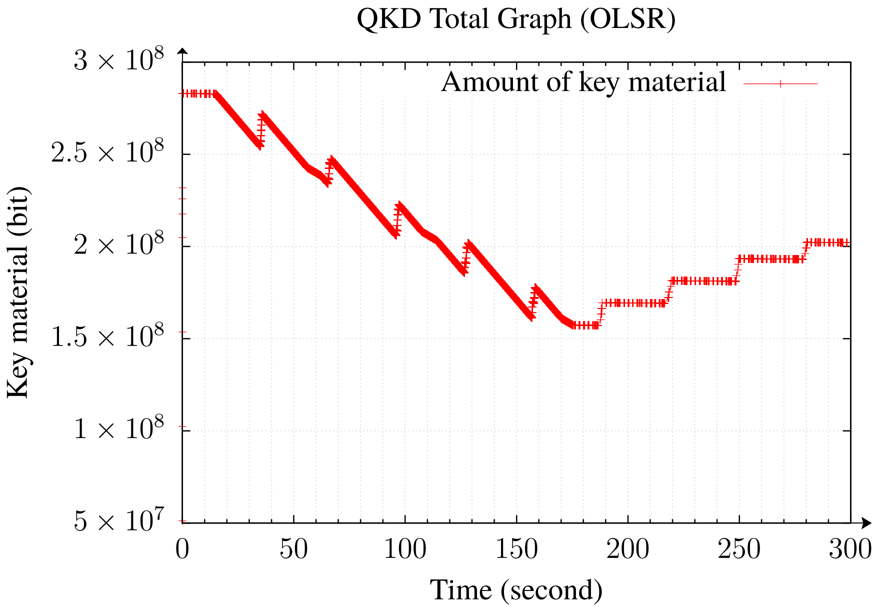

Figure 5.11. QKD Total Graph shows the total consumption and generation of key material in the network when OLSR routing protocol is used. All links are utilized until the available key material is depleted. After second 170, links B-F, C-F, and D-F are in a locked state and communication between the source node A and the destination node G is disabled. Due to the lack of a route, it is not possible to generate new key material and due to the lack of key material, it is not possible to exchange routing packets and update routing tables.

OLSR is based on a proactive link-state approach which uses Hello and Topology Control (TC) routing messages to discover and disseminate link-state information through the network. OLSR reduces the control traffic overhead by using Multipoint Relays (MPR), which is the key idea behind OLSR. The MPR node is a node's one-hop neighbor which has been chosen to forward packets. Instead of pure flooding of the network, packets are forwarded by node's MPRs. This reduces the network overhead, thus being more efficient than pure link state routing protocols. In our simulation, OLSR uses all three possible routes toward destination. First, OLSR favors route A-B-C-F-G until second 56 when key material on link C-F is depleted. But, OLSR is not able to detect this change until second 63 when node B notes the lack of information about the route toward node G in routing packet which is received from node C. Then, OLSR on node B switches to alternative route A-B-D-F-G. The similar situation occurs on second 114 when OLSR on node B switches to route A-B-E-E-F-G which is utilized until second 172 when node B announce lack of any route toward destination node G.

By default, OLSR exchanges Hello messages each 2 seconds and defines "holding time" as three times the Hello message period. Holding time indicates a waiting time prior deleting route due to the lack of fresh Hello message [MFVC16]. Therefore, OLSR is able to detect a link failure after 6 seconds which significantly increases the likelihood that during the key material establishment when QKD buffers are in EMPTY state, the link in question will become locked. In our simulation, link B-F is locked in second 56, link C-F is locked in second 104 while link D-F is locked in second 170. Therefore, after second 170 there is no available route from source node A toward destination node G and only key material applications on non-locked links are available as shown in Fig. 5.11.

5.2 Summary

This chapter addressed the practical realization of QKD networks from a network point of view. The work summarized the limitations and the basic characteristics of QKD network and described the ways of implementation of QKD networks, which can be overlay mode or network with a single TCP/IP stack. The main part of this chapter dealt with QKDNetSim simulation environment which is primarily intended for testing of existing and implementation of new solutions in the field of QKD network.

Considering that currently there are no available simulators of QKD networks, QKDNetSim gives a significant contribution to the research community in the field of QKD technologies. We assume that QKDNetSim will help in the understanding of the practical use of QKD technology which will allow the usage of a range of applications within the QKD network. An example of use of QKDNetSim listed in this document indicates that traffic originated from routing protocol needs to have higher priority in the allocation of network resources to avoid bringing QKD links in a locked position. Also, the obtained results confirmed previously published results in [MFVC16] stating that DSDV routing protocol provides significantly better performance in terms of PDR and the number of transmitted routing packets when compared to other tested routing protocols.

Although primarily designed for QKD network, QKDNetSim can be easily utilized to simulate other types of networks. Implementation of a virtual TCP/IP stack allows simple simulation of overlay networks while QKD Buffers and QKD Cryptos simplify simulations that consider the use of a symmetric cryptographic key. The QKDNetSim source code is free for download from *git* repository "https://bitbucket.org/liptel/qkdnetsim".

Chapter 6

Analysis of the Public Channel of Quantum Key Distribution Link

QKD networks have been studied extensively by research teams and laboratories in recent years focusing mainly on basic characteristics of the network such as the greatest possible distance of the quantum channel [94], the maximum key generation rate [83, 124] or research in the field of optimal network topologies [67, 125]. However, the public channel was mainly neglected, and to best of our knowledge, no investigations on the traffic over the public channels were made or published beyond studying confidentiality, authenticity and the correct QKD protocol functionality. So, it is not known how QKD link behaves in the presence of congestion on the public channels and what are the values of the usual performance of these channels such as throughput or delay. Therefore, this chapter addresses the question of traffic analysis of communication over the public channel of QKD link in QKD post-processing process.

The key material establishment process is performed using QKD protocol to provide a key to remote user in a safe manner. Generally speaking, QKD protocols can be roughly distinguished into three broad categories: discrete-variable protocols (BB84, B92, E91, SARG04), continuous-variable protocols and distributed-phase-reference coding (COW, DPS) [66, 126]. It is important to note that QKD protocols consist of nearly identical steps at a high level, but differ, among others, in the way the quantum particles or photons are prepared and transmitted over the quantum channel [MNV15, MPTV15]. After exchange of raw key material over the quantum channel, all further communication is performed over the public channel. The aim is to extract the secret final key from the raw key in QKD post-processing process. Although there are differences in implementations, almost every QKD post-processing process includes following steps: extraction of the raw key (sifting), quantum bit error rate (QBER) estimation, key reconciliation, privacy amplification and authentication [66].

QKD starts with the transmission of photons over the quantum channel. The obtained raw key material is pushed to sifting module to the first analysis and preminary processing. The sifted key may contain errors due to background photons, detector noise, polarization imperfections or eavesdropping influence. To correct or delete those errors, the sifted key is further forwarded to QBER estimator and key reconciliation module [64, 65]. To reduce potential information leakage in the quantum transmission phase and to strengthen key's privacy, some bits of the key are discarded in privacy amplification stage. Finally, the processed key is authenticated using a hash algorithm to verify its symmetry on both sides and it is stored in key material storages. The key is subsequently used to secure user's data communications [30].

The Austrian Institute of Technology (AIT) R10 QKD software is one of the most popular QKD post processing application [82]. The software is designed to be hardware agnostic and therefore not bound to the QKD hardware. The QKD stack handles the necessary steps for a complete QKD negotiation including hardware pickup and presifing, sifting, QBER estimation, key reconciliation (Cascade and LDPC), privacy amplification and confirmation.

In the experiments we performed, we have focused on pure naive setups of QKD post-processing pipelines with a single error correction protocol. This has been Cascade, since it is the most prominent one in the context of used BB84. Therefore, our QKD post-processing pipeline consisted of sifting, cascade, privacy amplification and confirmation modules which were serial connected.

6.1 Laboratory Measurements of QKD Link

The traffic analysis of communication over the public channel was realized in AIT Quantum Laboratory in April 2016. The channel was established between two computers connected in a simple local network and AIT R10 QKD software was utilized for post-processing of raw key which was generated using AIT R10 qkd-simulator module. QKD simulator was used for continuous generation of raw key material independent of environmental and laboratory conditions. The parameters of qkd-simulator which were used in our experiment are listed in Table 6.1 while the parameters not given here are the default parameters of the AIT R10 QKD software[1]. The average QBER of the generated raw key was 3%.

It is important to underline that the traffic was generated and recorded on a local network, that is, without any additional traffic originating from any other application. Since the machines were connected directly to a local network switch in between and given that there was no additional traffic, total communication resulted in large amounts of data. The average network throughput was 6.57 Mbps with periodic peaks which were repeated with a period of 30 seconds as a result

[1] The AIT QKD R10 is free for download w/o registration via git with "git clone http://sqt.ait.ac.at/git/qkd-public.git" [82]

Table 6.1. The used parameters of the qkd-simulator module of AIT R10 QKD software

Parameter	Value
Source photon rate	100000 Hz
Fiber length	1 km
Fiber absorption coefficient	1 db/km
Source signal error probability	3 %
Sync detection time standard deviation	1.0
Detection efficiency	50 %
Dark count rate	100 Hz
Time slot width	30 ns
Detection time standard deviation	1
Detection time delay	5 ns
Detector down time	10 ns

of sifting operations on the newly generated raw key material. In those moments the highest throughput of 13.9 Mbps was recorded. Fig. 6.1 shows the network throughput indicating that the traffic generated by the BB84 sifting modules and cascade modules had a dominant part of the overall traffic.

In practice, QKD network is usually deployed as overlay point-to-point network, but it can be realized as a local network with a single TCP/IP stack. Since QKD tends to have a reliable communication, in practice, TCP transport protocol is favored instead of UDP. In addition, in the case when the overlay QKD network does not possess information about the underlay communication, TCP over TCP connections are quite common [MKM$^+$16] [29]. Therefore, AIT R10 QKD post-processing software use TCP transport protocol whereas, in this experiment in a local network, the communication with a single TCP/IP stack has been realized. Table 6.2 lists the packet length statistics which show that the packets of sizes from 80 to 159 bytes dominate in the traffic flow.

Table 6.2. Packet lengths in bytes of the traffic generated using AIT R10 QKD post-processing application in a local network

Packet Lengths	Average Length	Min length	Max length	Count	Count(%)
40-79	66,04	66	78	20067	2,01
80-159	96,23	82	126	966078	96,61
640-1279	796,43	646	1018	2728	0,27
1280-2559	1559,52	1320	2141	11076	1,11

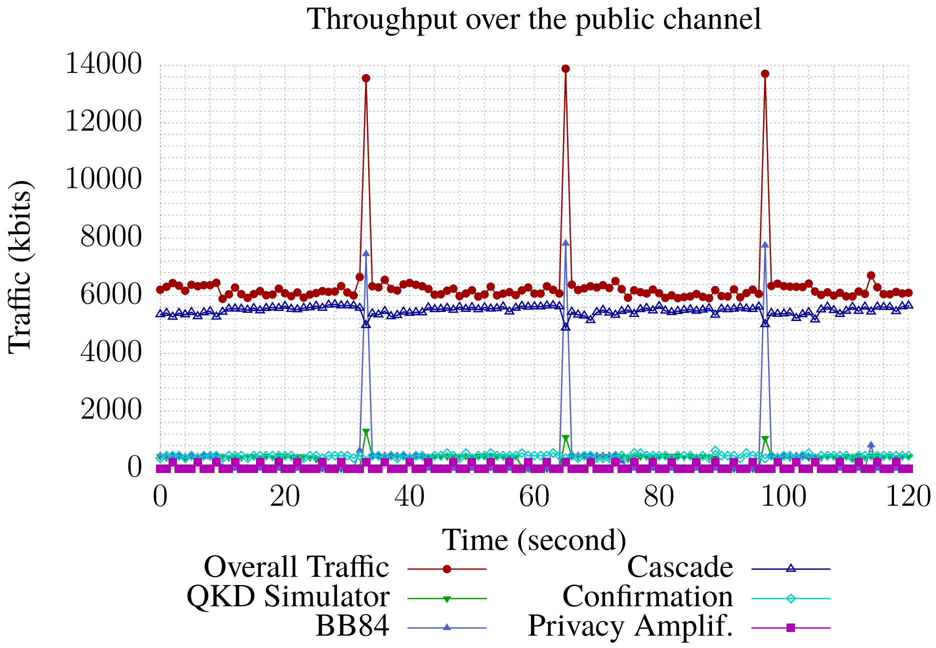

Figure 6.1. The throughput of the public channel of QKD link established in Austrian Institute of Technology (AIT) Quantum Laboratory in April 2016. The graph shows the amount of traffic generated by different modules of AIT R10 QKD post-processing application.

6.2 Virtual QKD link

Having analyzed the traffic on the local network, in August 2016, *Virtual QKD link* has been established between AIT headquarter in Vienna and VŠB Technical University of Ostrava. The link is called "virtual" because only the public channel was physically established between the machines located in Vienna and Ostrava. The machines on both sides of the corresponding link were 64bit with Debian 8.3 installed, 16GB of RAM memory. The raw key that is usually established by the quantum channel was generated in advance using qkd-key-gen AIT R10 module and deployed manually to the remote machine. The reason for using of this tool is a simple key generation process with an emphasis on QBER and the public channel performances. The aim was to minimize delays that may occur in communication between quantum devices and machines used for post-processing. qkd-key-gen declares no delays or other assumptions on the QKD hardware and with qkd-key-gen, we state that the QKD hardware delivers raw key material instantly.

To get a deeper understanding of virtual QKD link, we used the traceroute tool which exploits the usage of Time To Live (TTL) value in packet's IP header. Traceroute was activated periodically every 60 seconds in the period between 30th September 2016 and 1st November 2016 resulting in 45,414 traces. The average hop counts encountered on the path was 13 hops. IP localization services[2] were used to locate the geographical position of the routers and plot the most representative path as shown in Fig. 6.2[3]. The results show that the most often utilized path was through Germany, Netherlands and United Kingdom. Also, we can state that several routes exist between AIT and VŠB and that they are at least partly disjoint. Table 6.3 lists the geographical locations of the nodes in the path and number of occurrences. The shown locations are only approximate as IP geolocation service may be inaccurate [128].

Table 6.3. Geolocation details of hops on the path between VŠB and AIT

ID	IP	Occurrence	Latitude	Longitude	Location
1	195.113.113.190	45412	49.89794921875	18.191959381104	Hlucin (CZ)
2	195.113.209.53	45411	49.834648132324	18.282039642334	Ostrava (CZ)
3	83.97.88.41	45412	50.088039398193	14.420760154724	Prague (CZ)
4	62.40.98.75	45410	50.115520477295	8.6841697692871	Frankfurt(DE)
5	62.40.98.69	130	53.57532119751	10.015339851379	Hamburg (DE)
6	80.249.210.137	45409	52.374031066895	4.8896899223328	Amsterdam (NL)
7	83.97.89.33	45411	51.733329772949	-2.3666698932648	Cambridge (UK)
8	130.244.82.54	45367	52.374031066895	4.8896899223328	Amsterdam (NL)
9	130.244.82.63	43845	48.208488464355	16.372079849243	Vienna (AT)
10	212.152.189.65	45413	48.208488464355	16.372079849243	Vienna (AT)
11	212.152.193.54	45413	48.208488464355	16.372079849243	Vienna (AT)
12	62.218.144.2	45413	48.208488464355	16.372079849243	Vienna (AT)
13	195.113.113.129	45413	49.89794921875	18.191959381104	Hlucin (CZ)
14	195.113.113.201	2	49.89794921875	18.191959381104	Hlucin (CZ)
15	195.113.235.89	4	50.088039398193	14.420760154724	Prague (CZ)
16	212.151.176.245	5	59.403160095215	17.944789886475	Kista (SE)
17	130.244.38.232	5	52.374031066895	4.8896899223328	Amsterdam (NL)
18	130.244.82.56	5	52.374031066895	4.8896899223328	Amsterdam (NL)

In addition to finding intermediate nodes, traceroute tool reports the Round Trip Time (RTT) it takes for a packet to get to a hop and back to the originating node. A statistical evaluation of results is listed as a BoxPlot graph in Table 6.4. The highest RTT was 401 ms, the minimal value was 0.25 ms while the typical (median) response from all hops comes within 30 ms.

The sifting phase of QKD is responsible for ensuring that both parties of the corresponding link have the same knowledge about the qubits which were exchanged over the quantum channel. Sifting includes the exchange of information about the polarization bases which were used for the measurement of the qubits, but without disclosing the results of the measurements. As can

[2] https://www.iplocation.net/ and http://www.ip2location.com/demo

[3] The map was plotted using OpenStreetMap software [127]

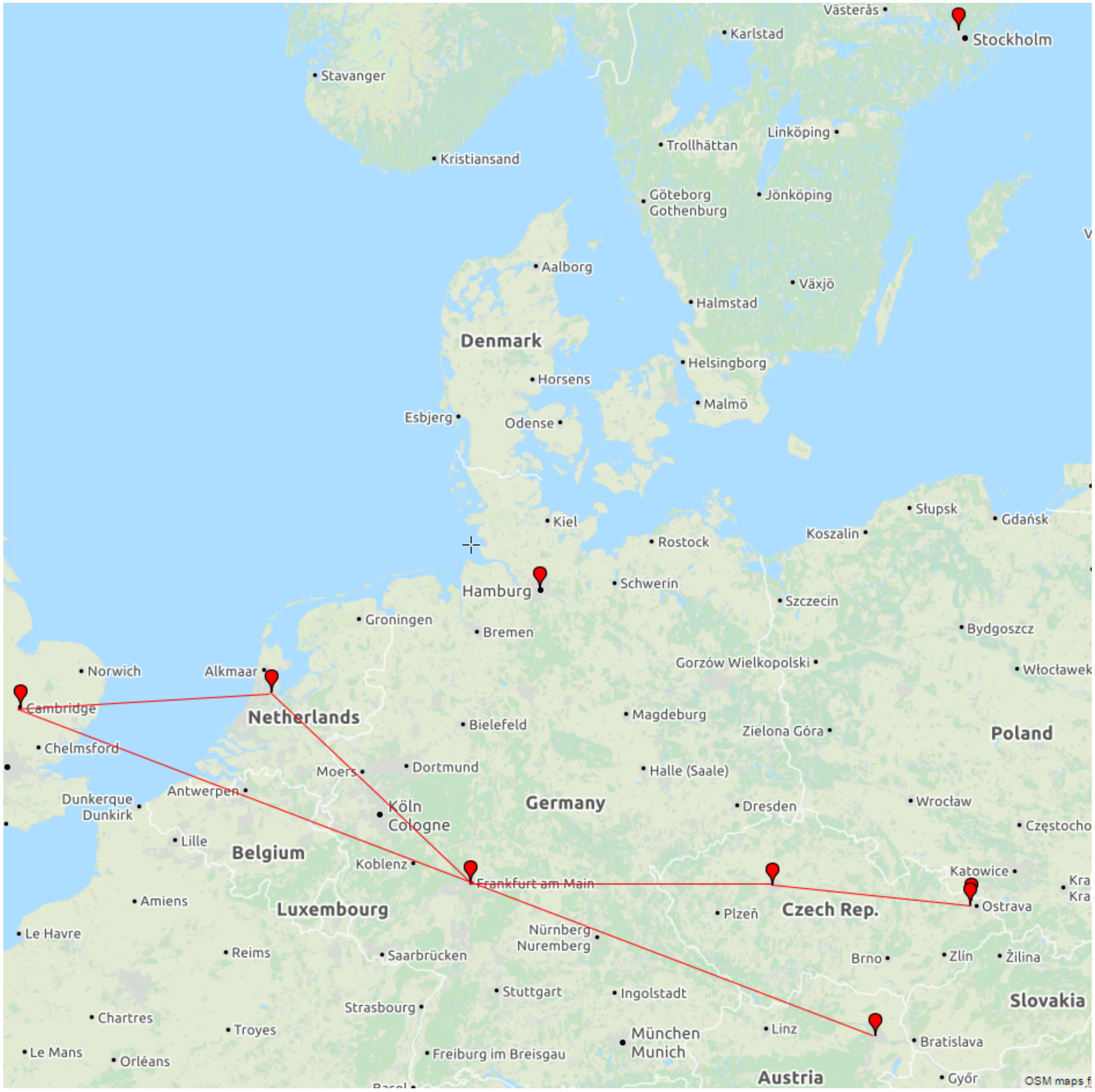

Figure 6.2. Geographical representation of communication between VŠB and AIT. The red line denotes the most often used path which was Ostrava-Hlucin-Prague-Frankfurt-Amsterdam-Cambridge-Vienna.

be seen from Table 6.5, sifting module generates packets of various sizes where we emphasize those in the range 640-1279 bytes. The sifted key is forwarded to key reconciliation module for error correction.

Table 6.4. Round Trip Time to nodes on the path between VŠB and AIT

ID	IP	Mean	Median	Minimal	Maximal	Standard Deviation
1	195.113.113.190	2.728	0.389	0.256	401.867	20.699
2	195.113.209.53	1.570	0.519	0.345	141.173	5.871
3	83.97.88.41	7.287	7.037	6.909	32.922	1.733
4	62.40.98.75	19.764	19.504	19.246	41.584	1.259
5	62.40.98.69	27.742	27.648	27.485	31.432	0.573
6	80.249.210.137	20.556	20.38	20.16	102.763	1.06
7	83.97.89.33	20.821	20.274	20.096	44.728	1.745
8	130.244.82.54	30.548	30.528	24.159	37.208	0.404
9	130.244.82.63	25.305	25.185	24.693	31.784	0.555
10	212.152.189.65	29.168	25.453	24.841	234.613	12.472
11	212.152.193.54	28.359	25.15	24.609	171.643	9.889
12	62.218.144.2	28.496	27.417	27.11	269.961	10.845
13	195.113.113.129	2.209	1.897	0.528	129.913	3.306
14	195.113.113.201	0.580	0.5805	0.547	0.6140	0.047
15	195.113.235.89	7.350	6.8495	6.087	9.61600	1.577
16	212.151.176.245	29.937	29.936	29.881	29.993	0.042
17	130.244.38.232	24.454	24.524	24.203	24.678	0.204
18	130.244.82.56	24.968	24.912	24.835	25.134	0.123

The Cascade protocol is well-known key reconciliation protocol which is able to operate at improved efficiencies and also at high Mbit/s throughput rates [129, 130]. Cascade begins with the random permutation of the key with the objective to evenly disperse errors throughout the key. The permuted key is divided into equal blocks, and cascade continues to run iteratively in the given number of iterations. After the each iteration, permutations are performed again and the block size is doubled. For each block, cascade will compare the results of the parity-check test and perform a binary search to find and correct errors. The process is recursive and instead of going through all the iterations continuously, cascade investigates and corrects errors in pairs of iterations (hence the name). Cascade protocol is known to be highly interactive and time-consuming. Despite this limitation, Cascade is one of widely used reconciliation protocol in practice due to its relative simplicity and efficiency for relatively low-rate discrete-variables QKD setups [26, 30, 131].

AIT R10 QKD post-processing software implements qkd-confirmation module which ensures that the keys are indeed equal on both sides of the corresponding channel by applying a binary "AND" on the whole key with a random number and publishing the parity of the result. Confirmation module runs after cascade, and in a case when the keys are not identical, they are silently discarded. Otherwise, identical keys without errors are pushed to privacy amplification (PA) module which primary function is to strengthen key's privacy and minimize information leakage in the quantum transmission phase. Therefore, PA discards some of the bits of the key where the number of

Table 6.5. Packet lengths in bytes of the traffic originating from sifting module

	Number of packets				
Packet length	QBER 1%	QBER 2%	QBER 3%	QBER 5%	QBER 7%
40-79	10037	7015	5277	3894	2767
80-159	357804	225023	177145	125141	101883
640-1279	118504	74501	58618	41385	33775
TOTAL	486345	306539	241040	170420	138425

Table 6.6. Packet lengths in bytes of the traffic originating from cascade module

	Number of packets				
Packet length	QBER 1%	QBER 2%	QBER 3%	QBER 5%	QBER 7%
40-79	1751567	2075216	1837502	2437968	2339890
80-159	8606389	8852591	9143509	9092778	9291960
640-1279	0	0	0	0	834
TOTAL	10357956	10927819	10981040	11530813	11632684

discarded bits depends on the configuration and measured parameters [51] [MNV15].
To simulate the impact of congestion of the public channel on the overall QKD link performances, the "Wondershaper" tool was employed to gradually decrease the upload and download traffic rate of network interface card of the machine located at VŠB according to values listed in Table 6.7 [132]. Every three hours, the rates were reduced by double with the aim to limit the capacity of the public channel.

Table 6.7. Settings of Wondershapper tool

Hour	Rate	Hour	Rate
0-3	16.384 Mbps	18-21	256 kbps
3-6	8.192 Mbps	21-24	128 kbps
6-9	4.096 Mbps	24-27	64 kbps
9-12	2.048 Mbps	27-30	32 kbps
12-15	1.024 Mbps	30-33	16 kbps
15-18	512 kbps	33-36	1 kbps

QKD modules traffic analysis

Fig. 6.3 and Fig. 6.4 show the throughput of core modules of AIT R10 QKD post-processing software. It is evident that cascade takes the largest part of the overall traffic, while plots for other modules have similar curves. Although cascade needs more time to locate and eliminate errors in the process of establishing the new key material for higher values of QBER, the obtained values are different to the extent as is the case with other modules. As listed in Table 6.6, number of generated packets grows with QBER. But, since the cascade generates and uses only information about the results of the parity-check operations, generated packets are small and do not impact greatly on overall throughput. On the other hand, sifting, confirmation and PA of the key with lower QBER results in a greater amount of traffic.

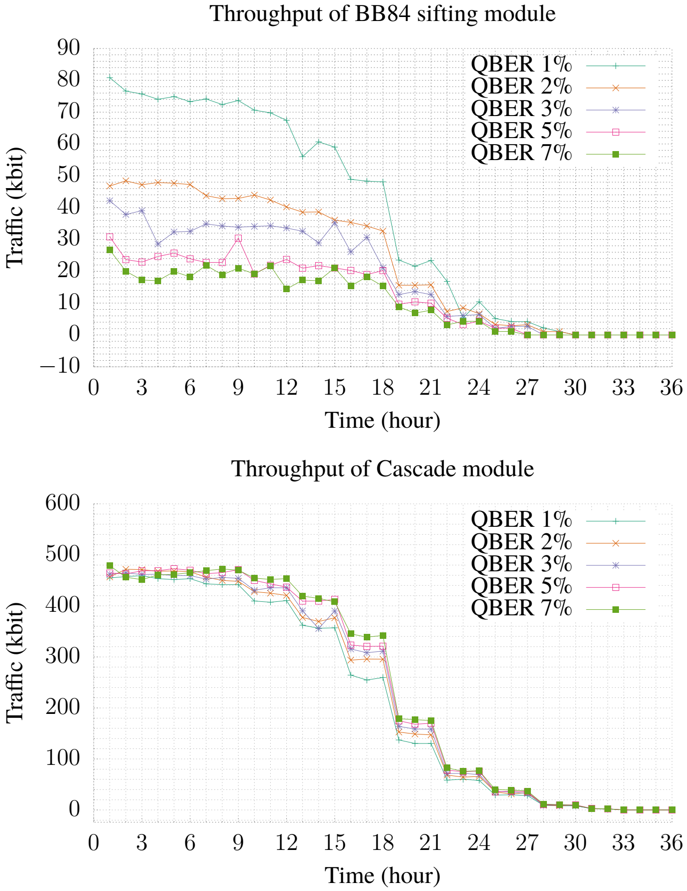

Figure 6.3. Throughput of BB84 sifting and cascade modules of AIT R10 QKD post-processing software measured on the machine located at VŠB which was utilized on the public channel of the virtual QKD link between VŠB and AIT

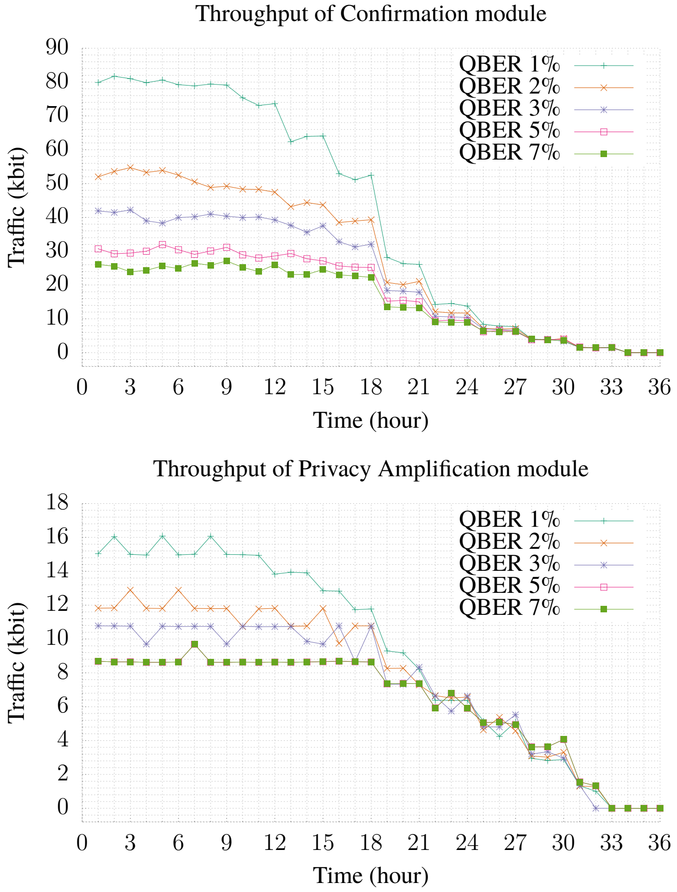

Figure 6.4. Throughput of confirmation and privacy amplification modules of AIT R10 QKD post-processing software measured on the machine located at VŠB which was utilized on the public channel of the virtual QKD link between VŠB and AIT

As the QBER increases, the number of packets that cascade generates grows since more information is needed for detection and correction of errors. It means that cascade needs more time to complete the key reconciliation process of the key with higher QBER. On the other hand, given that the machines between which the public channel was established have been configured in a way that AIT R10 QKD software modules were connected in series, time delay in cascade processing was reflected to other modules. For example, cascade cannot process new key material from sifting module until it finish processing of the previously received material. Since the increased QBER increases the processing time and the number of error bits that cascade needs to discard, the number of packets that is generated from the following modules in the series

(confirmation, PA) is reduced. Simply said, other modules need to wait until cascade completes key reconciliation.
Fig. 6.5 shows the overall delay over the public channel, and it shows that the number of samples (final established keys) reduces as QBER grows due to the reasons listed above. Also, it is evident that the overall delay in QKD post-processing is larger for higher values of the QBER.
Fig. 6.6 show the overall throughput of the public channel and it shows that the overall throughput decreases as the value of QBER grow. Although the number of packets that cascade protocol generates increases with higher values of QBER, these packets are small and do not affect too much on the overall throughput. On the other hand, for smaller values of QBER, there are more packets originating from sifting, confirmation and PA module. Since these packets are significantly larger than the packets that are generated by cascade, they significantly affect the overall throughput.

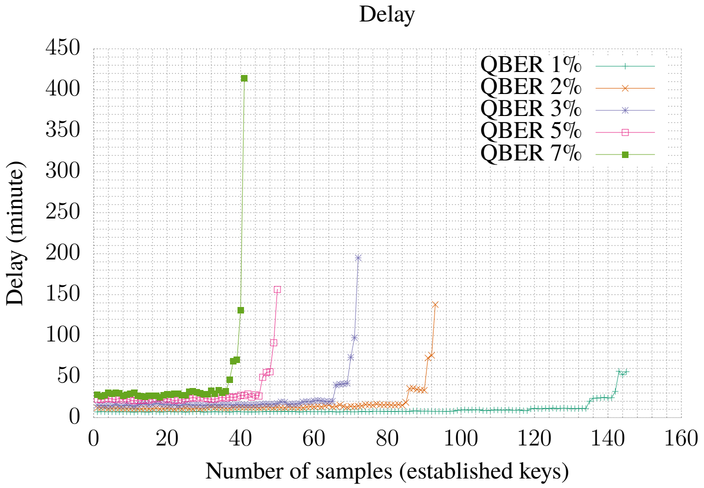

Figure 6.5. Overall delay of QKD post-processing for various values of QBER measured on the virtual QKD link between VŠB and AIT

When considering the impact of congestion on performances of QKD post-processing over a public channel, it is important to note the differences in the values obtained from laboratory testing and the values derived from the virtual QKD link between VŠB and AIT. In laboratory measurements, the values for the overall throughput reached 13.9 Mbps as shown in Fig 6.1, while the measurement of the virtual QKD link are with significantly lower values (Fig. 6.6).

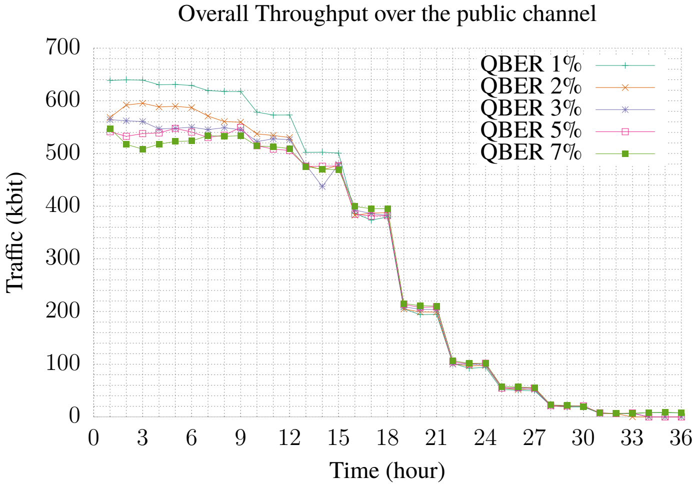

Figure 6.6. Overall throughput of QKD post-processing for different values of QBER measured on the public channel of the virtual QKD link between VŠB and AIT

The reason lies in the overall delay in the exchange of information between the machines. Given that in the laboratory testing, machines were connected to a local area network with minimal latency and without the presence of traffic from any other application, it has allowed a large flow of packets originating from AIT R10 QKD post-processing software. The overall delay and presence of third-party applications that shared communication resources of the utilized routes between VŠB and AIT, limited the maximal overall throughput to 639.89 kbps even in the time period of the first three hours of the experiment when the tool Wondershaper set the maximum upload and download rate of the network interface of VŠB machine to 16.384 Mbps. By the 9th hour, the impact of Wonderhshapper tool is imperceptible. But the measurement data from the 9th up to the 15th hour reveal the slight influence while from the 15th hour Wondershaper clearly limits the capacity of the public channel. This experiment shows that QKD post-processing application aims to use maximums of available communication resources, but the capacity of the established public channels significantly depends on the parameters of the route through which communication is performed, such as network load and delay.

6.3 Summary

This chapter addressed the question of the impact of the performance of the public channel on the overall performance of the QKD link, such as the maximal key establishment rate. The work summarized the measurements that were performed in the laboratory and on virtual QKD link which was established between AIT and VŠB. Our measurements clearly showed that the performance of the quantum channel significantly affects the performance of the public channel. For small values of QBER, there is a larger amount of traffic data on the public channel. As QBER grows, it increases the time it takes to process the raw key by QKD post-processing application. Furthermore, the increased delay means a smaller key generation rate and a smaller amount of traffic data over the public channel. Conversely, a decrease in QBER leads to a higher network load which can finally lead to the congestion of the public channel. It is apparent that more time is needed for communication over the congested public channel which is reflected in the reduction of key generation rate or increased retransmission of network packets over the public channel. Therefore, the performance of the public channel directly affects the performance of the quantum channel and vice versa!

Also, our measurements showed that cascade plays a key role in the overall QKD post-processing process, in view that its operation directly affects on the duration of the entire process.

Chapter 7

A Novel Proposal of Quality of Service in Quantum Key Distribution Network

During the 30 years since the discovery of the first quantum protocol [2], quantum technology has grown significantly and is rapidly approaching the level of high maturity. The next natural step in the evolution of quantum systems is to study their performances, suitability and convergence with the applications used in everyday life. A noticeable progress in the development of quantum equipment has been reflected through a number of successful demonstrations of QKD network [6–10], but without showing the clear suitability to assess how such a network competes with its classical counterpart under real life enterprise and real-time traffic. The traffic in these networks was mainly considered with equal importance and it was treated with the same priority. While such approach may be acceptable for some applications, it is not acceptable for voice, video and today's broadly used collaborative applications. Because not all network traffic is equal, it should not be treated equally since different applications may have different service requirements with respect to quality of service. Accordingly, methods of traffic management, congestion control and QoS approaches become an important issue. Following the above arguments, this chapter provides a QoS solution which allows the usage of a range of applications within QKD networks.

7.1 Quality of Service in QKD Networks

A lot of work has been done in supporting QoS in the conventional networks, but unfortunately, none of it can be directly used in QKD due to the specificity of QKD link. In contrast to conventional networks where the link state information such as delay, bandwidth, cost, loss rate are necessary, QKD network requires additional information of the quantum channel. To obtain and manage the link state information in QKD network is additionally difficult due to rapid changes in the availability of these channels. The challenge we face is to implement complex QoS functionality with limited available resources in a dynamic environment.

7.1.1 QoS Models

QKD characteristics generally lead to the conclusion that this type of network provides a weak support to QoS. The quantum channels provide low charging rate which with the lack of control over underlying physical infrastructure of the public channel leads to a dynamic nature of a network. This results in frequent and unpredictable changes of network topology, adding difficulty and complexity to routing among the QKD nodes. From this, it is clear that classical QoS models in its original form cannot entirely cope with the needs of QoS provisioning in the QKD networks.

Integrated Services and QKD Networks

The basic concept of the Integrated Services (IntServ) model is per-flow reservation of resources in advance, where a flow is defined as an application session between a pair of end users [133]. A flow-specific state should include information about bandwidth requirement, delay, jitter, cost or other. The main component of this model is the Signaling Resource ReSerVation Protocol (RSVP) which is in charge of reserving resources before transmission of data. Still, the IntServ model is not suitable for QKD networks because of the following:

- *Inability to guarantee the reservation:* When the QoS is provided in the IP network, an IP router has total control over its packet buffers and the output link bandwidth, and can directly schedule these resources. In contrast, in the overlay network, the node cannot directly access the available resources in the overlay path. It can only rely on the measurement techniques where the high accuracy cannot be guaranteed and it cannot directly control or reserve the resources in the underlying network. Besides, other non-QKD traffic can pass through the links anytime since the underlying medium is shared, which results in variations of the overlay paths service resources. Thus, it is possible that the parts of an overlay path may not satisfy the application requirements during the data transfer process. The only thing that node in QKD network is able to do is to guarantee the resources of the quantum channel by performing reservation of key material in key storages. However, without control of the public channel, such reservation does not pose any gain.

- *Signaling:* RSVP is an "out-of-band" signalization protocol, that is, signalization is not included in data packets. Therefore, the RSVP signaling packets contend for network resources with the data packets and consume a substantial amount of scarce key material and network resources.

DiffServ and QKD Networks

Differentiated Service (DiffServ) uses Differentiated Services Code Point (DSCP) bits in the IP header and a base set of packet forwarding rules known as Per-Hop-Behavior. DiffServ is known as edge-provisioning model and it does not provide any QoS guarantees *per se*. At the network boundary, the edge router posses the knowledge of allowed traffic volumes that can be admitted into privileged traffic classes and it controls the traffic entering the network with classification, marking, policing and shaping mechanisms. When a data packet enters a DiffServ-enabled domain, the edge router marks the packet's DSCP field and the interior core routers along the path forward the packet based on its DSCP field. Since the DSCP field defines very limited service classes, the processing of interior core routers is simple and fast. The application of DiffServ in its original form is limited in QKD network due to the following:

- *Edge router selection*: The current QKD technology limits the deployment of QKD network to metropolitan scale [10, 122, 123]. In such a network, it is necessary to clearly define what are the edge routers which play a key role in the processing of traffic.
- *Lack of Service Level Agreement*: Each network node needs to comply with the rules for the classification and processing of traffic of different priorities. However, since the concept Service Level Agreement (SLA) is not defined for QKD networks, it is questionable how nodes of possibly different domains can negotiate the traffic rules.

In section 7.3 we describe a flexible QoS network model for QKD network which uses some of the features of the DiffServ models for easier operation in a dynamic network environment.

7.1.2 QoS Signalization

QoS signaling is directly dependent on the used QoS model and it is usually the most complex component in a network. Since signaling needs to support complex network services and has strict performances and reliability requirements, defining optimal signaling solutions represents a challenging task.

As mentioned in Chapter 6, QKD protocol which establishes a new key material consists of six successive stages: secret key exchange, extraction of the raw key (sifting), error rate estimation, error correction, privacy amplification and authentication [134] [MPTV15]. Only the first stage is performed over the quantum channel while all other stages are performed over the public channel. It is important to note that the authentication module generate packets that are authenticated by default (Fig. 7.1).

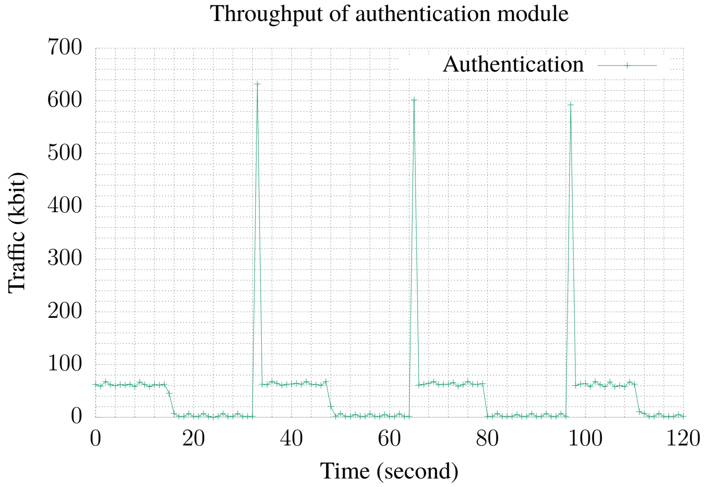

Figure 7.1. The throughput of authentication module of AIT R10 QKD post-processing application.

In addition, a detailed analysis of traffic generated by AIT R10 QKD Software [82], which is used for post-procession of the raw key material showed that at least 14 kbps of traffic generated by this software is always present at QKD link. This traffic is generated by the software key management sub-protocols in order to synchronize key material in key storages (Fig. 6.1). Since some of these packets are authenticated, we propose extension of the payload of the authenticated packets with routing or signaling information. This provides an elegant way to tackle the problem of distribution of routing and signaling packets without introducing additional traffic overhead.

7.1.3 QoS Routing

QoS Routing is the process of finding a feasible path (if one exists) by selecting a path such that QoS metrics of interest stay within specific bounds. In our view, a routing protocol well-suited for operation in dynamic QKD network should fulfill the main design objectives listed by priority as follows:

- It is necessary to reduce the consumption of scarce key material by choosing the shortest optimal path [30]. When choosing a path it is necessary to consider both channels (public and quantum) of QKD link since these channels are mutually dependent (See Chapter 6 for more details). Routing algorithm needs to find a balance between the requirements, since the path that meet the requirements of the public channel may not be suitable for quantum channel and vice versa,

- Given that the primary objective of QKD is to provide ITS communication, routing packet needs to be encrypted and authenticated [30]. This entails that the number of routing packets need to be minimized in order to preserve scarce key material,
- Due to the nature of QKD, an eavesdropper is not able to gain information about the key which is transported via link, and in the best, a denial of service can be performed in order to disable the communication. To prevent such an attack, it is necessary to minimize number of nodes that possess information about the utilized routing path. Thus, number of broadcast routing packets should to be minimized [12],
- Since the number of nodes may vary for various realizations of the QKD network, routing protocol should be scalable to different network sizes,
- Due to dynamic overlay environment and low key charging rate, link interruptions are common in QKD network. Hence, routing protocol should be robust enough to find an adequate replacement path in such situations.

In general, routing solutions can be divided into three broad categories: source routing, hierarchical routing, and distributed routing. In source routing, each node maintains an overall view of the network state, which is used at the source node to centrally calculate the route which is written in the data packet header. Then, intermediate nodes relay packet based on the information which is obtained from the packet header. Still, the performance of source routing algorithms relies on the availability of precise state information, while the dynamic nature of QKD network makes the available state information inherently imprecise. Given that the constant maintenance of state information is mostly done by periodic flooding, this solution is inadequate for QKD network.

In hierarchical routing, nodes are organized into groups, which are recursively merged into higher-levels, creating a multilevel hierarchy. In every level of the hierarchy, independent routing algorithms may be used. Although there are several examples of the use of hierarchical network organization [8–10], in our opinion such network organization is not suitable for QKD since nodes of upper hierarchical level represent a potentially easy attack target to disassemble the network. Disabling of high-level hierarchical nodes would simply disconnect the entire network region and bring into question the purpose of QKD network.

In distributed routing, the computation of the path is shared among network nodes, which exchange the necessary routing information. This exchange can be periodic in case of proactive (table-driven) protocol, or only when routing path is requested in case of reactive (on-demand) protocol. Proactive routing protocols mainly use static update period time for keeping routes up-to-date, which is against the dynamic nature of QKD network. Therefore, in overlay network reactive routing perfoms better in terms of efficiency and stability than proactive routing [135, 136].

7.2 The Similarities Between QKD and MANET and VANET Technologies

The specific QKD issues and constraints described above pose significant challenges in QKD network design. However, by analyzing the characteristics of QKD network, we note similarities with Mobile Ad Hoc Networks (MANET) and Vehicular Ad Hoc Networks (VANET) [137–139]. First, we specify the main characteristics of QKD technology from a simple point of view:

- QKD links, described above, are always implemented in a point-to-point behavior, and they can be roughly characterized by two features: limited distance and the key rate inversely proportional to the distance [30]. Also, QKD links may become unavailable when there is no enough key material or when the public channel is congested. Such behaviour is similar to Wi-Fi links which are limited in length and where the communication speed depends on the user's distance from the transmitter antenna.

- One of the main features of QKD networks is the lack of a quantum repeater or quantum router in practice, so communication is usually performed on a hop-by-hop basis [100] (more details in section 3.3.2).

In MANET networks communication takes place on a hop-by-hop basis and mobile nodes are typically powered by batteries placing special attention to energy-aware solutions. The nodes connect themselves in a decentralized, self-organizing manner with no authority in charge of managing and controlling the network. The main characteristic of MANET networks is the unpredictable mobility of nodes, which can often lead to unstable routing paths [140]. The energy amount of battery power and mobility of MANET nodes can be easily linked to the amount of key material in QKD key storages. The limitations in range of wireless link can be easily mapped to the limitations in the length of QKD link, while the lack of a dedicated network infrastructure (such as a router) increases the similarity between these two technologies. On the other hand, the poor mobility of QKD nodes increases the similarity with VANET technology in which communication takes place via a predefined path. We believe that some of the solutions from MANET/VANET technology can be effectively applied in the field of QKD networks.
Although at first glance MANET and QKD networks have nothing in common, a simple analysis of the features of these networks reveals their similarity. However, what clearly distinguishes these two networks is their purpose. MANET networks are designed for fast and simple communication in situations where installed pre-existing infrastructure is not available (such as search and rescue operations in the case of natural disasters, earthquakes, fires, battlefield scenarios and etc.). In turn, the primary goal of QKD is to provide ITS secure communication. This may have a significant impact when choosing network solutions, since solutions which are required in one

situation may not be appropriate in another. For example, consider routing solutions based on flooding the network. QKD networks rely on the assumption that all nodes are trusted when communication is performed in hop-by-hop or key-relay manner [79, 84], and by following this assumption strictly, the eavesdropper is restricted to attacking QKD links only. Due to the nature of QKD, the eavesdropper is not able to gain any information about the key being transported via the link, and service may be denied in order to disable the communication. Although results have been obtained by combining multiple paths to establish a secure key material [68, 125, 141, 142], it is believed that the amount of routing information being sent to the nodes should be reduced to a minimum. In order to prevent a denial of service attack, all nodes in the network should not know the routing request, therefore the number of broadcast packets should be minimized. Also, given that the main objective of QKD is to provide ITS communication, routing packets need to be encrypted and authenticated [143]. This means that the number of routing packets in the network needs to be minimized (routing overhead) with respect to the material to be preserved for the protection of data, which is the primary goal of secure communication. From this, it follows that protocols based on flooding are not preferred in QKD networks.

7.3 FQKD: A Flexible Quality of Service Model for QKD Network

To cope with the problem of providing quality of services in the dynamic environment, we present a flexible QoS model for QKD networks (FQKD). Our model avoids a centralized resource management scheme or reservation of resources mechanism. Instead, we turn to a distributed approach to control traffic load by providing soft-QoS constraints. There is no flow or session state information maintained in support of end-to-end communication.

FQKD defines three roles for nodes in a QKD network: ingress, interior and egress. Each node can take any of these roles depending on the position in network flow. The source node that sends data is referred as ingress node. Interior nodes are nodes that forward data toward the final destination node which is referred as egress node.

Although there are cases in which public channels of multiple QKD links are shared, in this thesis, we consider only independent QKD links, that is, every QKD link implements dedicated quantum and public channel. From the standpoint of the practical realization it means that each node implements as many network interfaces as there are links to that node. Thus, the routing problem boils down to the choice of the network interface that leads to the next hop. To make the right decision, routing protocol needs to have an accurate insight into the state of links.

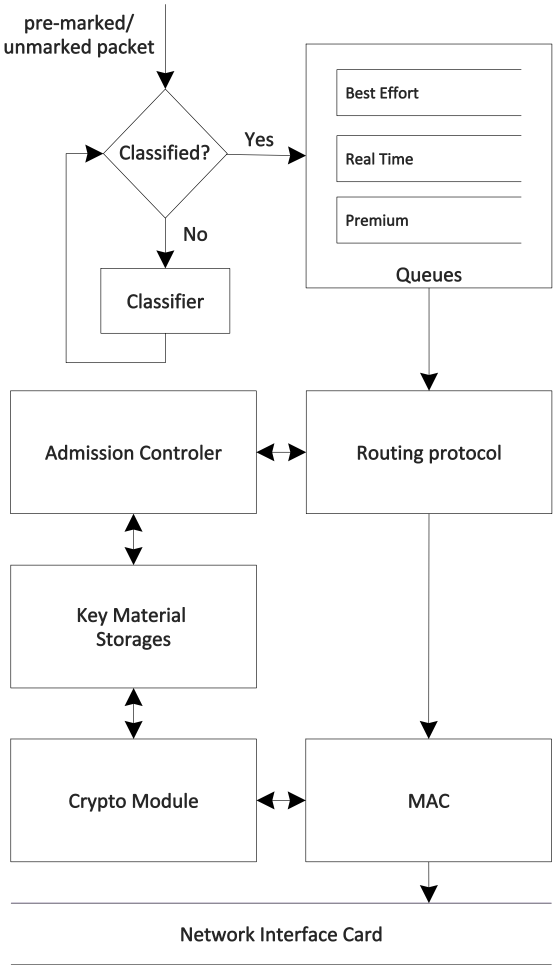

Figure 7.2. FQKD Model

7.3.1 Provisioning and Conditioning

As shown in Fig 7.2, FQKD Model consists of following elements: sender-based classifier, waiting queues, local node based admission controller, crypto module and dynamic regulation of admitted session at MAC layer. The classifier is put at the ingress node where the traffic originates to distinguish between traffic classes by marking the DiffServ (DSCP) field in the IP packet header. DSCP value is used to indicate the importance of the delivery of the packet in terms of delay. Therefore, FQKD distinguishes between three traffic classes with corresponding DSCP values: best-effort, real-time and premium class. FQKD implements waiting queues for each class which are processed by priority such that the packets from the premium class queue are served first, and in the case when this queue is empty, the packets from real-time class queue are served. Finally, the packets from the best-effort queue are served if other queues are empty. The packets are forwarded by interior nodes in per-hop behaviour associated with the assigned DSCP value.

Considering that nodes in a QKD network constantly generate new keys at their maximum key rate until their key storages are filled [29], it means that each node is able to measure and reliably known the state of links to its first neighboring nodes, that is, nodes that can be reached in one hop (e.g. direct communication), that are hereinafter referred to as neighbors. Thus, before setting the route, routing protocol contact admission controller to selects those links to its neighbors that have sufficient resources to serve the classified network packet. If such link does not exists, the packet waits in the queue for reprocessing. Otherwise, routing protocol calculates the path and the packet is forwarded to MAC layer for further processing. In section 7.3.2, we propose QKD link metric for evaluation of QKD link state while in section 7.4 we discuss routing path calculation algorithm.

Waiting queues are usually implemented on data link layer (L2), but in FQKD additional waiting queues sit between L3 and L4 ISO/OSI layer. The reason for such implementation is the avoidance of conflicts in decision making which might lead to inaccurate routing. Suppose the queues are implemented only on data link layer (L2) and suppose that they are half filled with packets. Since the routing protocol used the routing metric that at the time t_1 of calculating the route had a different value from the time t_2 when the packet came on line in the queue to be served, significantly changes of the state of links in the time interval $\Delta t = t_2 - t_1$ might occur which can lead to inaccurate and incorrect routing. Instead, having waiting queues implemented between L3 and L4 layer means that the packet for which routing protocol calculated the route will be directly forwarded to lower layers and immediately sent to the network. This implies usage of one set of waiting queues (set of three waiting queues for best-effort, real-time and premium traffic classes) for all network devices. Using queuing at L2 layer is not excluded, but the additional attention is given to queues at a higher level due to the dynamic nature of the network.

Assuming the key rate is constant in time when the quantum channel has the fixed length (Fig. 3.9), it is evident that the key storage can be identified with the Token Bucket traffic shaping mechanism as shown in Fig. 7.3. This simplifies the view of the admission controller which in this view behaves as the traffic conditioner. The amount of traffic over QKD link is limited by the amount of key material in key storage which is used for encryption or authentication of data over that link. As listed in section 5.1.3, the key material storage (QKD buffer) of link k between nodes a and b can be represented using following parameters:

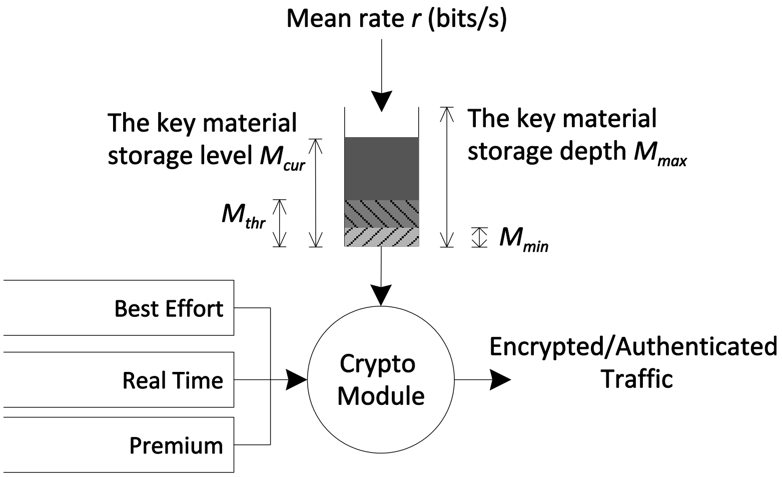
Mean rate r (bits/s)
The key material storage level M_cur
M_thr
The key material storage depth M_max
M_min
Best Effort
Real Time
Premium
Crypto Module
Encrypted/Authenticated Traffic

Figure 7.3. The processing of traffic in FQKD Model

- Time measurement interval *t*, measured in seconds,
- Mean key generation rate r_k, measured in bits per seconds used to indicate the charging rate of the storage,
- The key material storage depth $M_{max,k}$, used to indicate the capacity of the storage,
- The current value $M_{cur,k}(t)$, representing the amount of key material in the storage at the time of measurement *t*, where it holds that $M_{cur,k}(t) \leq M_{max,k}$.
- The threshold value $M_{thr,a,b}(t)$ or simply $M_{thr,k}(t)$,
- The minimal amount of pre-shared key material denoted with $M_{min,k}$.

The amount of key material in the storage at the measurement time *t* can be calculated using Equation (7.1), while the average operational rate can be calculated using Equation (7.2).

$$D_k(t) \leq r_k \cdot t + M_{cur,k}(t) - M_{min,k}(t) \tag{7.1}$$

$$A_k(t) = \frac{D_k(t)}{t} = r_k + \frac{M_{cur,k}(t) - M_{min,k}(t)}{t}. \tag{7.2}$$

The overall amount of traffic data that can be transmitted over links can be calculated multiplying the length of the key used to encrypt or authenticate the data traffic and the value obtained from Equation (7.1). An incoming packet is served from the queue if there is enough key material in the storage. Otherwise, the packet remains in the queue waiting for storage to be charged. The length of queues is limited and traffic shaping algorithms are in charge for packet management operations.

To avoid blocking of work due to the lack of the key material used to generate new key material, special attention is placed on the categorization of the traffic. If storage stops charging, the purpose of the link loses functionality. Therefore, traffic generated by the post-processing application that is used to establish new key material has the highest priority and it is sorted in the premium queue. Only traffic from post-processing applications has the right to use key material in the situation when $M_{cur,k}(t) \leq M_{min,k}$, while the traffic from the other two queues is served only when $M_{cur,k}(t) > M_{min,k}$.

The threshold value M_{thr} is proposed to increase the stability of QKD links, where it holds that $M_{thr,k}(t) \leq M_{max,k}$. The meaning of this parameter is best to explain by considering simple topology shown in Fig. 7.4 where node *a* needs to establish a VoIP connection with remote node *e*. Suppose the routing protocol uses only information about the state of links to its neighbors. Then, assuming that all network links have the same performances of public channels, we consider only the status of key material storages which are marked next to links as shown in Fig. 7.4-a. Upon receipt of the packet from node *a*, the routing protocol on node *b* selects path *b*-*c* toward destination *e* since the link *b*-*d* has a lower performance. However, since the node *b* does not consider the state of links which are away more than one hop, the traffic may stuck on the link between nodes *b* and *c* as shown in Fig. 7.4-b. To avoid such behavior, we propose usage of M_{thr} value which is calculated for each link as follows:

- Each node *a* calculates value L_a summarizing the M_{cur} values of links to its neighbors *j* and dividing it with the number of its neighbors N_a, that is:

$$L_a = \frac{\sum_j^N M_{cur,a,j}}{N_a}, \forall j \in N_a \tag{7.3}$$

- Then, each node exchanges calculated value L_a with its neighbors. The minimum value is accepted as the threshold value of the link, that is:

$$M_{thr,a,b} = min\{L_a, L_b\} \tag{7.4}$$

As shown in Fig. 7.4-c, node *b* calculates $L_b = 36.66$, while node *c* calculates $L_c = 25$. The threshold value of the link *b*-*c* is set to $M_{thr,a,b} = 25$ and it is included in link metric calculation

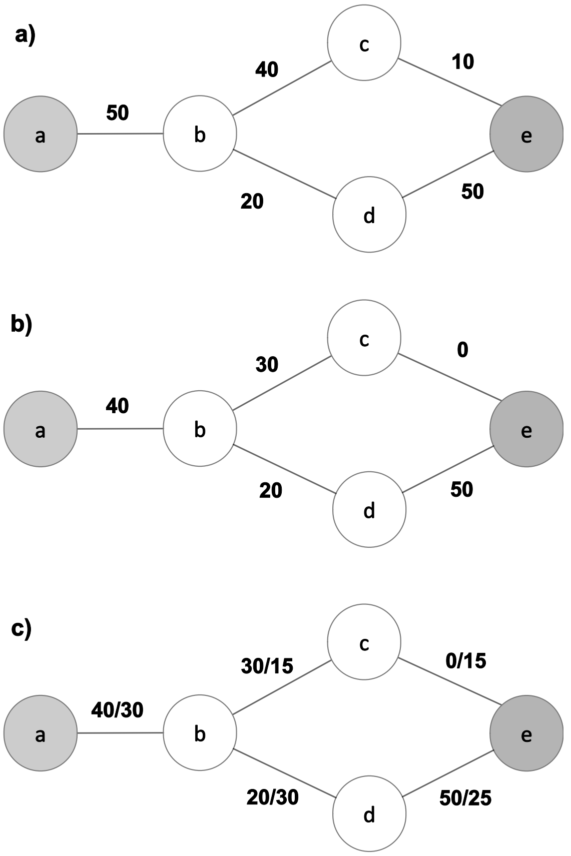

Figure 7.4. Simple topology that shows the calculation of M_{thr}: a) The traffic is routed along the route *a*-*b*-*c*-*e*; b) The traffic is routed along the path *a*-*b*-*c* regardless of key material depletion of link *c*-*e*; c) Calculated M_{thr} values are marked next to M_{thr} values

as described in the section 7.3.2. By using M_{thr}, node gains information about the statuses of network links. The higher the value, the better the state of links that are more than one hop away. In addition, we propose usage of ECN-based regulation of the traffic when $M_{cur,k}(t) \leq M_{thr,k}$. When a node detects such status of key material storage, it starts marking the ECN bits in the packet's IP header. The destination node monitors the ECN bits and informs user's application about the upcoming rerouting or termination of the connection in case when there is no alternative route.

7.3.2 QKD Link Metric

As mentioned in Chapter 6, it is important to note that in QKD network, public and quantum channel of QKD link are mutually interdependent. Secure communication can be realized only if there is enough key material, which is used for encryption or authentication of data flow, and if the public channel is not congested. For proper evaluation of the link status, it is necessary to consider both channels. The popular metrics from conventional networks that describe the state of the communication link (such as delay or bandwidth), can not be adequately used in QKD network since they describe only the public channel. Therefore, it is necessary to define new metrics that clearly define the state of QKD link taking into account its most important features.

Quantum Channel Status Metric

The remaining key material level of the key storage is the main factor contributing to the link's availability. We use Equation (7.6) to express the state of the quantum channel between nodes s and *i*, where $Q_{frac,s,i}$ is the ratio of the squared amount of key material at the time of measurement ($M_{cur,s,i}{}^2$) multiplied by the threshold value ($M_{thr,s,i}$) and the cubed capacity of the key storage ($M_{max,s,i}{}^3$) as defined by Equation (7.5). $Q_{frac,s,i}$ is in the range [0,1] and it highlights the current amount of key material on direct links in relation to the amount of key material of links that are further away since those links are unreachable when direct links do not have enough key material.

$$Q_{frac,s,i} = \frac{M_{cur,s,i}{}^2 \cdot M_{thr,s,i}}{M_{max,s,i}{}^3} \tag{7.5}$$

$$Q_{m,s,i} = 1 - \frac{Q_{frac,s,i}}{e^{(1-Q_{frac,s,i})}} \tag{7.6}$$

$Q_{m,s,i}$ is the utility function associated with the key material level of the link. $Q_{m,s,i}$ uses an exponential formula to address the fact that the less key material is in the storage, the more critical the situation is. This is, when the less key material is residual, the less time is left for the routing protocol to react.

The value of $Q_{m,s,i}$ is normalized as a grade ranging from 0 to 1 as shown in Fig. 7.5, where lower value means better quantum channel state. In example show in Fig. 7.4-c, the routing protocol should favor the link *b*-*d* since $Q_{m,b,d} < Q_{m,b,c}$.

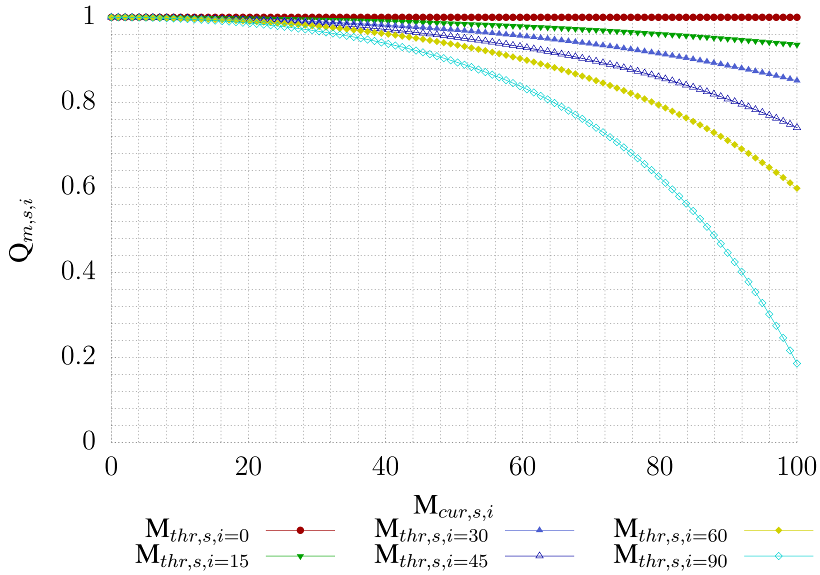

Figure 7.5. Q_m of QKD link between nodes s and i for different values of M_{thr}; $M_{max,s,i} = 100$

Public Channel Status Metric

Instead of using popular approach from conventional or overlay networks which is based on sending of probe packets to evaluate the state of links [144] [MFV$^+$16], we turn to the fundamentals of QKD. Since QKD devices constantly generate keys at their maximum key rate until the key storages are filled [29], there is almost permanent communication between nodes.

Therefore, we propose usage of meta-data of keys such as the time duration of the key establishment process and the number of IP packet retransmissions in order to effectively assess the state of the public channel. The time duration or retransmission of packets in key post-processing process is very likely to be caused by congestion or interruptions of the public channel and this information can be a clear indicator of the public channel status. Fig. 7.6 shows the changes of the throughput of the public channel of the virtual QKD link established between AIT and VSB Technical University of Ostrava in September 2016 (more details in Chapter 6) .

Consequently, we use $P_{m,s,i}$ defined with Equation (7.7) to evaluate the state of the public channel between nodes s and i. $T_{last,s,i}$ is the amount of time spent on the establishment of the key at the time of measurement, while the $T_{maximal,s,i}$ is the maximum time that can be tolerated for the establishment of the key. $T_{maximal,s,i}$ is calculated as double value of the average duration of key material establishment process in the long run, denoted as $T_{average}$ in Equation (7.8).

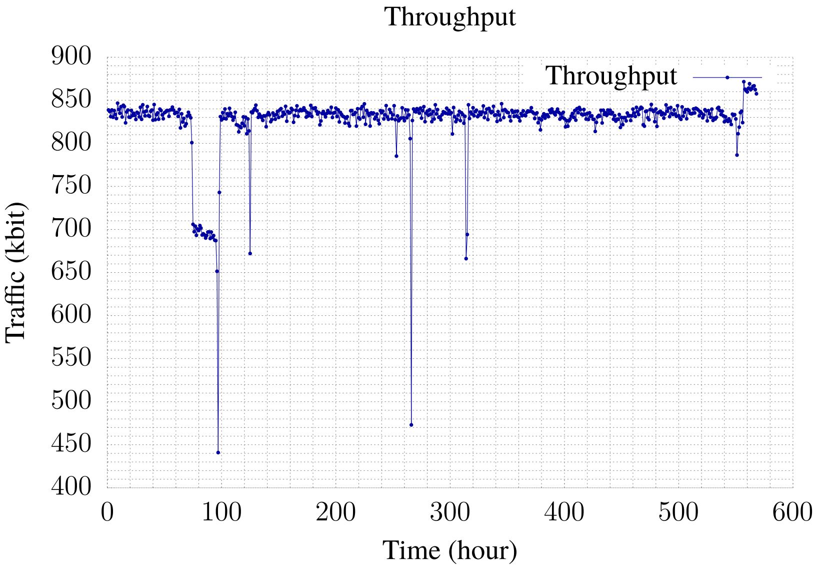

Figure 7.6. Throughput of Virtual QKD link between AIT Vienna and VŠB-TUO Ostrava for a period of 568 hours recorded in September 2016.

$$P_{m,s,i} = \frac{T_{last,s,i} + \Delta t}{T_{maximal,s,i}} \tag{7.7}$$

$$T_{maximal,s,i} = 2 \cdot T_{average} \tag{7.8}$$

Δt is used to describe the freshness of the information and is defined as the difference between the current time of measurement and the time when the $T_{last,s,i}$ is recorded. Note that $T_{average}$ is not equal for all links of the network, since it depends on the load of the network, type of quantum and network devices, QKD post-processing application and the performances of the public channel. The value of $P_{m,s,i}$ is mainly located in range [0,1] where the lower value means better public channel state. Values greater than 1 indicate that the link has a problem with the establishment of new key material.

Overall QKD Link Status Metric

The key material depletion is not the same for all links since it depends on the type of used encryption algorithm and the amount of network traffic to be encrypted. Consequently, it is necessary to distinguish the network flows that can meet the more stringent or relaxed values of the metric. In example, QKD link between nodes s and i having a small value of $Q_{m,s,i}$ may be suitable for the network flow encrypted with less secure algorithms that do not require too much material such as AES cipher, but it may not correspond with the flow encrypted using OTP cipher which require a lot more key material. Thus, a factor α that reflects the balance between the requirements is introduced in Equation (7.9) to compensate this effect. Equation (7.9) puts together the utility functions of the quantum and public channel in a normalized value in the [0,1] range where lower value means better link state.

$$R_{m,s,i} = \alpha \cdot Q_{m,s,i} + (1 - \alpha) \cdot P_{m,s,i} \tag{7.9}$$

The parameter α takes the value from the [0,1] range and in case of usage of OTP cipher we suggest the value of $\alpha = 0.5$. That is, both channels of QKD link are considered with equal importance. In the case of usage of AES cipher, we suggest the value of $\alpha = 0.25$ that puts more emphasis on the public channel due to lower requirements for key material.

7.4 Greedy Perimeter Stateless Routing Protocol for QKD network

Motivated by the similarities between MANET and QKD networks, we present Greedy Perimeter Stateless Routing Protocol for QKD networks (GPSRQ). The main motivation for designing of GPSRQ is to minimize the number of routing packets with regard to the requirement for minimizing key material consumption. Due to the dynamic nature of a QKD networks which are characterized by frequent changes of link state, the propagation of the link state packets and routing information across a network would be inherently imprecise considering the short time validity of such information. Therefore, we propose usage of distributed geography reactive routing to achieve high-level scalability. GPSRQ is based on the GPSR routing protocol [145] with several significant changes.

We assume that all nodes know the geographical locations of all other network nodes they wish to communicate. Therefore, there is no periodic flooding of node's location details and we assume a location registration and lookup service that maps node's address to a location. By using such service, each node is able to quickly determine the geographic position of every other node based on its address. This thesis does not deal with the implementation details of such service, but we assume it can be realized using internal or other communication channels [146]. As indicated in section 7.1.3, authenticated packets in key material post-processing can be used to effectively exchange information about the geographical position of nodes.

Although there are several experiments regarding the possibility of mobile or ad-hoc QKD networking [74, 147–149], we assume that QKD network is composed of static nodes representing secure access point. Therefore, there is no need to inform nodes of location changes in remote parts of the network since the location information accuracy decreases with the distance from the node. This shortcoming is balanced by the distance effect: "*the greater the distance separating two nodes, the slower they appear to be moving with respect to each other*" [150]. This is used to implement effective caching in GPSRQ which is discussed further below.

GPSRQ sets network without hierarchical organization, which means that all nodes in the network are of equal importance. This avoids marking the "hierarchical critical points" that might be subject to attacks to disassemble the network. Nodes do not exchange routing tables which significantly minimize consumption of scarce key material and reduces the probability of passive eavesdropping [12]. An eavesdropper is not able to intercept routing packets and find out the exact route to the destination since until the last moment, it is not known at which node's network interface the packet will be forwarded.

Route selection, that is, the decision about the next hop is made in per-hop behavior (PHB) such that the packet is moved closer to the destination based on the status of links in the local environment and on a geographical distance from the node. GPSRQ uses DSCP value from the packet's header to determine the path that best suits the traffic which is processed.

As noted in section 7.3.1, the premium traffic class is reserved for the traffic generated in the process of key material establishment and it is performed only on links that lead to node's neighbors. Consequently, the problem of QoS routing boils to route calculation for best-effort and real-time traffic class. GPSRQ uses the algorithm which consists of two methods for packet forwarding: *greedy forwarding* and *recovery-mode forwarding*.

7.4.1 Greedy Forwarding

Each node determines its own geographic location which consists of latitude and longitude and it maintains a table of locations of nodes in the network. The data for filling this table, GPSRQ takes from the location service which provides location information.

By definition, greedy forwarding entails forwarding to the neighbor geographically closest to the destination. An example of greedy forwarding is shown in Fig. 7.7, where an ingress node a which is surrounded by three adjacent nodes b, d, and k, needs to communicate with the egress node g. Ingress node a forwards the packet to b, as the Euclidean distance between b and g is less than the distance between g and any of a's other neighbors. Such greedy forwarding is repeated on interior nodes and it stops when the packet reaches its destination. Still, GPSRQ aims to maximize the network utilization by using different paths for different traffic classes. It uses Equation (7.10) to calculate the optimal path to forward the packet:

$$F_{s,d,i} = \beta \cdot G_{s,i} + \big(1 - \beta\big) \cdot R_{i,d} \tag{7.10}$$

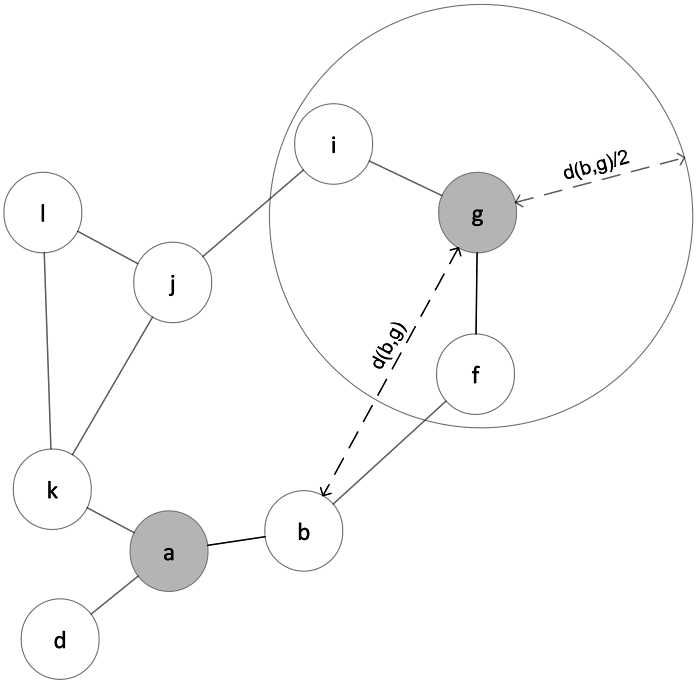

Figure 7.7. The simple topology used to describe GPSRQ protocol. Ingress node *a* which is surrounded by three adjacent nodes *b*, *d*, and *k*, aims to communicate with the egress node *g*. In the absence of paths through the node *b*, the node *a* writes in the internal cache that along the path *a*-*b* it is not possible to route packet toward the region marked with a circle of radius *d(b,g)/2* with the center in node *g*. Any further request for routing toward any node which is placed in the defined circle region will be ignored over the route via node *b*, and an alternative route is to be found.

where $R_{s,i}$ denotes the state of link between source node *s* and neighboring node *i* using Equation (7.9) and $G_{i,d}$ represents the Euclidean distance between neighboring node *i* and destination node *d*, for each node *i* which belongs to the set N_s of all neighbors of source node *s*, $\forall i \in N_s$. All routes toward destination are sorted in descending order using Equation (7.10) and the route with the lowest value is used.

GPSRQ uses parameter β which takes the value from the [0,1] range to manage network utilization by choosing between forwarding along the "geographically shortest" route or route that have the most available resources. Apparently, value $\beta = 0$ implies usage of links with the best performances. On the other hand, assigning a value $\beta = 1$ excludes consideration of link performance and leads to no-QoS routing. We discuss the simulation results for different values

of β in section 7.4.5.
Greedy forwarding relies on knowledge of the geographical position and state of links to neighbors, which enables a high level of network scalability. But still, there are cases where the only route to the destination requires forwarding packet over a neighboring node that is geographically further from the destination compared to the node that forwards the packet. Suppose the public channel of the link *b-f* of topology shown in Fig. 7.7 is disconnected. Since there is no alternative route from node *b*, the packet is returned back to node *a* which need to choose between nodes *d* and *k* as the next hop toward the destination *g*. In such case when a local maximum occurs (note that node *b* is geographically closest to node *g*), an alternative recovery-mode forwarding is used.
To increase scalability and exclude routes that do not lead toward the destination, we propose a robust caching mechanism to preserve key material consumption. When node *b* realizes that there is only one interface available, that is, the same interface which was used to receive the packet from node *a* (link *b-f* is unavailable due to the lack of key material), it marks "loop" field in GPSRQ packet header and returns the packet back to node *a*. When node *a* detects the packet with "loop" flag in GPSRQ header from node *b*, it calculates the Euclidean distance *d(b,g)* between node *b* and the packet destination node *g*. Then, node *a* writes in its internal cache that along the path *a-b* it is not possible to route packet toward region which is marked with a circle of radius *d(b,g)/2* and the center in node *g*. Upon receiving further requests for routing toward any node which is placed in the defined circle region, node *a* will ignore route over node *b* and look for an alternative route. The validity of cached record is set to time interval defined using Equation (7.11) after which cached record expires and the node is allowed to try establishing the connection once again:

$$T_{cache} = T_{maximal,b,g}/2 \tag{7.11}$$

where $T_{maximal,b,g}$ is defined with Equation (7.8). In the case when GPSRQ detects there is no neighbor closer to the destination, it marks flag "inRec" in GPSRQ header and switches to recovery-mode forwarding.

7.4.2 Recovery-mode Forwarding

Recovery-mode involves the usage of a well-known *right-hand rule* that states that the next edge from node *a* upon arriving from node *b* is the edge *(a,k)* which is sequentially counterclockwise to edge *(a,b)* [145]. The packet stays in recovery-mode until it reaches node on which greedy forwarding can be applied again. However, recovery-mode by itself cannot guarantee delivery of the packet since it can result in the network loop due to the unavailability of links that lead to the destination. To avoid routing loops, GPSRQ header contains information about the IP address of node on which the packet entered recovery-mode and the outgoing interface which was used for first forwarding. In the case of detection of the loop by analyzing the packet header, the packet is returned back to the previous node for re-routing and adding new entry to node's internal cache memory. Suppose the public channel of link *j-i* is unavailable due to the lack of key material which means there is no available path to the destination *g*. Source node *a* forwards the packet to node *k* which forwards the packet to node *j* in greedy forwarding mode. Since link *j-i* is unavailable, node *j* is not able to find any neighbor closer to the destination so it enters recovery-mode, sets the value of the field "inRec" to 1, writes its IP address to the header field "recPosition", writes the interface number that leads to node *l* to the "recIF" header field and forwards the packet to node *l* since it is on the first edge counterclockwise about *j* from the line $\overline{jg}$ as required by the right-hand rule. Upon receipt of the packet, node *l* inspects the header and remains in recovery mode since the note *j* on which the packet entered recovery mode is closer to the destination *g*. Then, the packet is forwarded to node *k* which forwards the packet back to node *j*. When node *j* detects its IP address from the header field "recPosition", it adds in internal cache memory record stating that it is not possible to reach destination node *g* over the interface which leads toward node *l* by adding a record consisting of the triple: IP address of the hop *l*, the radius of circle region *d(l,g)/2* and value of the circle center which is set to the location of node *g*. Given that there is no any other interface available except the interface which was used to receive the packet, node *j* marks GPSRQ header field "loop" to 1 and returns the packet to node *k* which adds to its cache memory a record stating that it is not possible to reach destination node *g* over the interface which leads toward node *j*. Then, node *k* sets the field "loop" to 2 and tries again with greedy-forwarding where node *j* is excluded as next hop. Therefore, the packet will be forwarded to node *l* which will forward the packet to node *j*. Since node *j* does not have any other interface available except the interface which was used to receive the packet, it will set the value of header field "loop" to 1, and return the packet to node *l* which will add to its cache memory a record stating that it is not possible to reach destination node *g* over the interface which leads toward node *j*. The packet is returned to node *k* and afterward, it is returned to source node *a*. This procedure is repeated until a feasible path to the destination is found. Otherwise, if there is no any path available, the packet is discarded on the source node keeping updated records in the cache memory of neighboring nodes.

7.4.3 GPSRQ Protocol Implementation

The packet header fields that GPSRQ uses in recovery-mode are listed in Table 7.1. GPSRQ packet headers include a flag field indicating whether the packet is in greedy mode or recovery-mode and a flag field indicating whether the packet is in a returning loop. Packet headers also include an IP address of node at which packet entered recovery-mode and the outgoing interface which was used for the first forwarding.

The major feature of QKD network is a constant struggle for the preservation of key material

Table 7.1. GPSRQ packet header fields used in recovery-mode forwarding

Field	Function
inRec	Forwarding mode: Greedy or Recovery-mode
loop	Returning Loop indicator
recPosition	IP address of node on which packet entered Recovery-mode
rec IF	OutgoingInterface used for first forwarding in Recovery-mode

resources due to the low key generation rate. A link running out of key material does not offer the service expected, so, in practice, special emphasis is placed on minimizing the packets that need to be encrypted or authenticated.

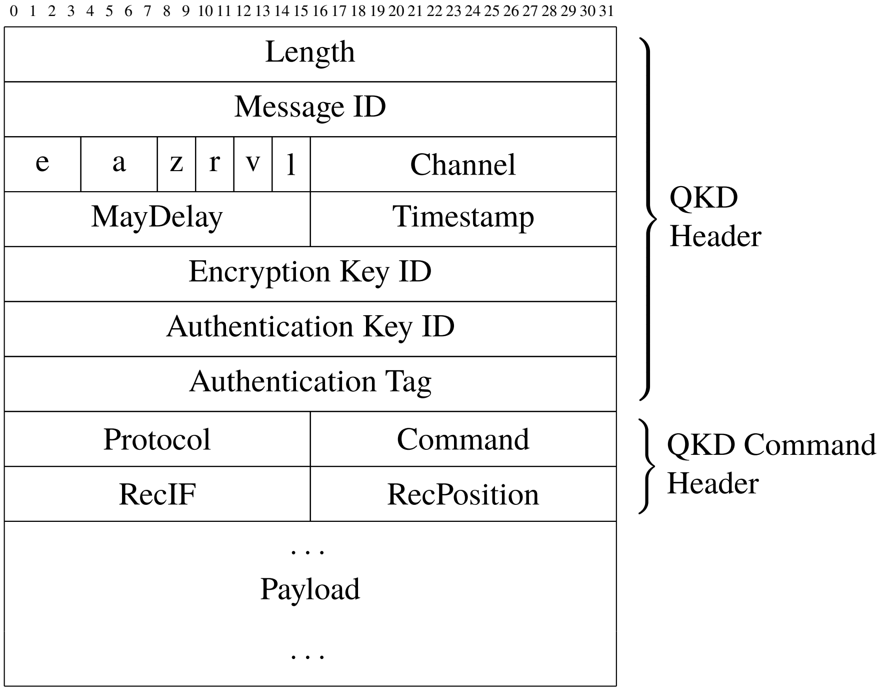

Figure 7.8. QKD Header and QKD Command Header

To facilitate the ease of routing operation, encryption and authentication are performed between data link (L2) and network (L3) ISO/OSI layer [30] [MKM^{+}16]. But, it is important to the note that QKD header in previously deployed QKD networks was transported unencrypted [82]. A denial-of-service attack is possible, if an eavesdropper is able to sniff the traffic over the public channel and blocks the packets of high importance. For example, blocking of routing packets or packets that are used in sub-protocol key management operations (LOAD, STORE, ACKNOWLEDGE or other) will easily disable QKD link due to routing information (section 5.1.4) or due to the lack of information about key management operations. To prevent such scenarios, we propose usage of QKD Command header which is to be encrypted together with the packet payload. Additionaly, instead of adding GPSRQ header to the packet which would increase the amount of key used for encryption, the values from GPSRQ header are rearranged to QKD Header and QKD Command Header prior transmission has shown in Fig 7.9 and Fig. 7.8. Upon receiving the packet, the values are simply moved from QKD Header and QKD Command header to GPSRQ header that is passed to the GPSRQ routing protocol for further processing.

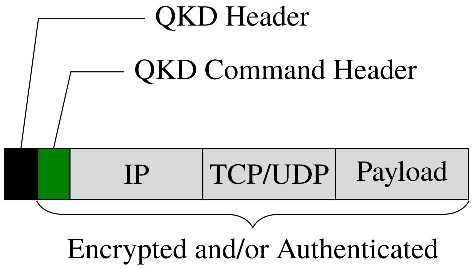

Figure 7.9. Packet encapsulation with QKD Header and QKD Command Header

In typical multimedia communication protocols, excessively delayed packets are not used for the reconstruction of transmitted data at the receiver. That is, delayed packets are considered useless and discarded regardless of whether they ever reach their destination [151]. Thus, there is no need to transfer delayed packets which will be ignored at the destination. Hereof, we propose an extension of QKD header to include two additional fields *Timestamp* and *MaxDelay* which include values of packet's timestamp and maximum tolerated delay, respectively. The values of these fields are written at the ingress node, while the GPSRQ at the interior node checks the values before forwarding of the packet, including the previously mentioned field "loop". If the value of the field "loop" is equal to 2, it means that the packet during the route was in the loop that is now avoided and marked in the internal cache on the previous nodes. In that case, GPSRQ does not take any action against the packet in consideration regardless of the delay. Otherwise, if the value of the field "loop" is equal to 0, it means that the packet was not previously in the loop. Then, GPSRQ checks whether the delay of the packet, which is calculated as the difference between the current timestamp and the *Timestamp* value, is greater than the *MaxDelay*. If this is true, GPSRQ sets the field "loop" to 1 and returns the packet to the previous hop which will result in adding a new entry to the internal cache on previous nodes as mentioned earlier. The function of *Timestamp* and *MaxDelay* fields is similar to the Time-to-live (TTL) field in the IP header of conventional networks, but with the aim to minimize the consumption of scarce key material in QKD network.

Table 7.2 and Table 7.3 provide a short explanation of all fields in modified QKD header and QKD Command Header.

Table 7.2. Extended QKD Header - Description of Fields

Field	Length	Short Description
Length	32 bits	Total packet length in bytes
Message ID	32 bits	Message ID
e	4 bits	Type of used encryption cipher
a	4 bits	Type of used authentication algorithm
z	2 bits	Type of used compression algorithm
v	2 bits	Version
r	2 bits	GPSRQ inRec indicator
l	2 bits	GPSRQ loop indicator
Channel	16 bits	QKD public channel ID
MaxDelay	16 bits	The maximum tolerated time delay
Timestamp	16 bits	The timestamp of packet's generation at the ingress node
Encryption Key ID	32 bits	ID of key used for Encryption
Authentication Key ID	32 bits	ID of key used for Authentication
Authentication-tag	32 bits	Authentication tag
Payload	-	Data payload

Table 7.3. Extended QKD Command Header - Description of Fields

Field	Length	Short Description
Protocol	16 bits	Type of next header in the packet headers chain
Command	16 bits	Key management sub-protocol operation command (LOAD NEW KEY and other.)
RecIF	16 bit	GPSRQ recIf field
RecPosition	16 bit	GPSRQ recPosition field

7.4.4 Simulation Setup

To ensure that simulations are independent of topology or specifics of any specific network, random graphs were constructed. These were developed so their connectivity characteristics approximated those observed in QKD networks which were previously deployed in practice. Since the distance at which direct QKD link is possible is roughly limited to 100 km in optical fibers for current systems [68], random network topologies were created using Waxman model [152] which is a recommended model for small networks that include locality aspects. Waxman model generates short links that fit limits of QKD link length and it corresponds to the requirement for the realization of QKD networks without hierarchical multi-plane organization [153]. It spreads nodes randomly on a grid and adds links randomly, such that the probability P_e for interconnecting two nodes in a single plane is parameterized by the Euclidean distance that separates them as defined by Equation (7.12):

$$P_e(u,v) = \Theta \cdot exp^{-\frac{d(u,v)}{\Omega \cdot \Lambda}} \tag{7.12}$$

where $d(u,v)$ is the Euclidean distance between the nodes d and v, Λ denotes the maximum possible distance between two nodes where $0 < \Omega, \Theta \leq 1$. A large value of Ω increases the number of connections to nodes that are further away while a large value of Θ increases the number of edges from each network node. Additionally, the parameter choices are constrained to assure $P_e(u,v) \in [0,1]$

Our simulations include random static networks with 10, 20, 30, 40, 50, 60 and 70 nodes which are randomly placed in a rectangular region and which are connected with QKD links of parameters listed in Table 7.4 according to Waxman model. Each simulation was repeated four times for random network topologies (but with the same number of nodes) and with random values of the initial amount of key material in key storages. We evaluated GPSRQ against OSPF which was used in previously deployed QKD networks [28–32], and against DSDV which has been used in our previous work [MFVC16].

The simulation was performed using the QKD Network Simulation Model (QKDNetSim) of NS-3 Simulator version 3.26 to deploy GPSRQ and DSDV while NS-3-DCE of version 1.9 was used to deploy OSPF routing protocol. We used the BRITE topology generator to generate random topologies according to Waxman model because it is supported under NS-3 and the source code is freely available [154]. NS-3-DCE and QKDNetSim were set to share the same seed file for generation of random values. Such setup has enabled the use of the same random topology with the identical configuration values in QKDNetSim and NS-3-DCE simulator. Table 7.4 presents the simulation parameters including the key generation rate, charging key rate, packet size, and data traffic parameters. The parameters not given here are the default parameters of the NS-3, NS-3-dce simulator and QKDNetSim module.

Table 7.4. Parameter values of the simulation

Parameter	Value
Brite Model	Waxman Router
Node Placement	Random
GrowthType	All and Planar (GG)
Neighboring nodes (m - Waxman Parameter)	2
Λ (HS - Waxman Parameter)	100
Ω (Waxman Parameter)	0.4
Θ (Waxman Parameter)	0.4
GPSRQ default β value	0.6
GPSRQ default $T_{average}$	5
Bandwidth distribution	Constant
Bandwidth	10 Mbps
Packet Size	512 Bytes
Packet Traffic Type	UDP; CBR
Packet Traffic Rate	1 Mbps
Encryption type	OTP
Authentication type	VMAC
Authentication tag lenght	32 bits
Minimal amount of key material (all links)	1 MByte
Maximal amount of key material (all links)	100 MByte
Initial amount of key material (all links)	Randomly generated in range (0.5,25) MByte
Charging key rate (all links)	100 kbps
Charging key period (all links)	7 seconds
FQKD L4 Traffic Queue Capacity (GPSRQ)	1000 Packets
L2 Traffic Queue Capacity (per device)	1000 Packets
Total Simulation time	150 seconds

7.4.5 Simulation Results and Evaluation

It is important to emphasize that geographical routing meets the full potential on planar graphs, that is, graphs without crossing edges [155]. Although GPSRQ can achieve valuable results on non-planar graphs (Fig. 7.13), one of the drawbacks of GPSRQ is the large consumption of key material in the full random network such as the one shown in Fig. 7.10. Geographical routing in non-planar graphs cannot quickly determine the shortest path toward the destination which leads to unnecessary forwarding and consumption of scarce key material. It is, therefore, advisable to convert non-planar graphs into planar graphs on which geographic routing protocols can be effectively used. To exclude the intersecting edges, usage of the *no-crossing heuristic* with perimeter probing and mapping is proposed in [155]. This heuristic is used for mapping of the location of each node that forwards the packet in the right-hand rule. If during traversal of a graph by the right-hand rule, the candidate next edge crosses an edge taken earlier, that candidate next edge is ignored, and the next edge in counterclockwise order is taken instead [155].

Instead of using the heuristic to exclude the intersecting edges, we modified the BRITE random topology generator to generate random planar Gabriel Graphs (GG). The GG is proposed in [156] and defined as follows: "*An edge* (u, v) *exists between vertices* u *and* v *if no other vertex* w *is present within the circle whose diameter is the Euclidean distance d(u,v)*" [157], or in mathematical form:

$$\forall w \neq u, v : (d(u, v))^2 < [(d(u, w))^2 + (d(v, w))^2] \tag{7.13}$$

Fig. 7.10 shows a random graph with 50 nodes uniformly placed on a square of 100x100 region and random planar GG graph generated using the BRITE topology generator.

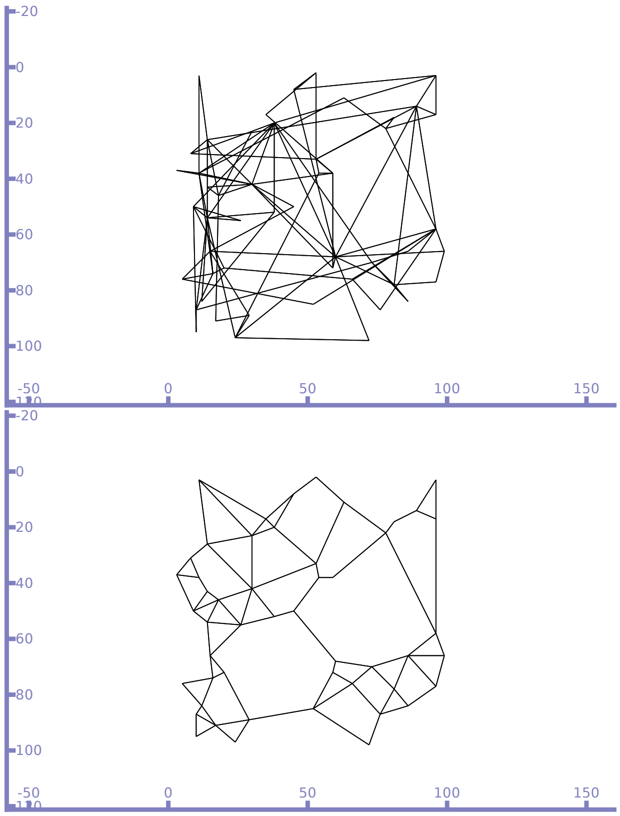

Figure 7.10. Top: The non-planar graph; Bottom: The random planar Gabriel Graph; 50 nodes uniformly placed on a 100 x 100 square region. Both graphs were generated using the BRITE topology generator

Routing Protocol Overhead

Table 7.5. The number and sizes of generated routing packets

	GPSRQ			DSDV			OSPF		
Nodes	Packets	Sum of Packet Sizes	Average Packet Size in Bytes	Packets	Sum of Packet Sizes	Average Packet Size in Bytes	Packets	Sum of Packet Sizes	Average Packet Size in Bytes
10	1740	0.097 MB	56,5	4832	0.417 MB	86,32	1594	0.149 MB	93,57
20	4256	0.238 MB	56,5	41 k	3.208 MB	77,04	5522	0.628 MB	113,88
30	6638	0.372 MB	56,5	115 k	8.582 MB	74,42	11 k	1.422 MB	123,43
40	7896	0.442 MB	56,5	142 k	11 MB	77,96	13 k	1.829 MB	133,45
50	9996	0.56 MB	56,5	283 k	20 MB	72,98	18 k	2.618 MB	138,72
60	12198	0.684 MB	56,5	294 k	23 MB	81,40	27 k	3.866 MB	139,32
70	14804	0.831 MB	56,5	604 k	44 MB	73,61	36 k	5.468 MB	151,63

OSPF routing protocol is a widely deployed link-state routing protocol which means that each node maintains a link-state database describing the network topology and it uses periodic Link State Announcement (LSA) flooding mechanism to update database records [25]. From the link-state database, each node constructs a tree of shortest paths with itself as the root. By default, OSPF floods LSA update information each 30 minutes and it exchanges Hello packets to establish and maintain a neighbor relationship for each 10 seconds. If a node does not receive a Hello message from a neighbor within a fixed amount of time, OSPF modifies its topology database entries to indicate that the neighbor is unavailable. The "dead interval" specifies the time interval that OSPF waits before declaring the neighbor node to be unavailable. By default, this interval is set to four times the default hello interval, which is 40 seconds in case of point-to-point networks. As mentioned in section 5.1.4, DSDV is the most popular proactive routing protocol based on the distributed Bellman-Ford algorithm. It periodically broadcast its routing table to neighbor nodes based on periodic route update interval which is set to 15 seconds by default. In addition to regular periodic updates, DSDV uses triggered updates when the network topology suddenly changes. The main purpose of triggered updates is to advertise the information that has changed since the last periodic update. However, if a periodic and triggered updates occur in a short period of time, the values may be merged and only the periodic update will be performed. To limit the propagation of unstable information, the transmission of triggered updates is delayed using settling time which is recorded in the second DSDV table for each destination node. By default, the settling time is set to 5 seconds [158] [MFV$^+$16].

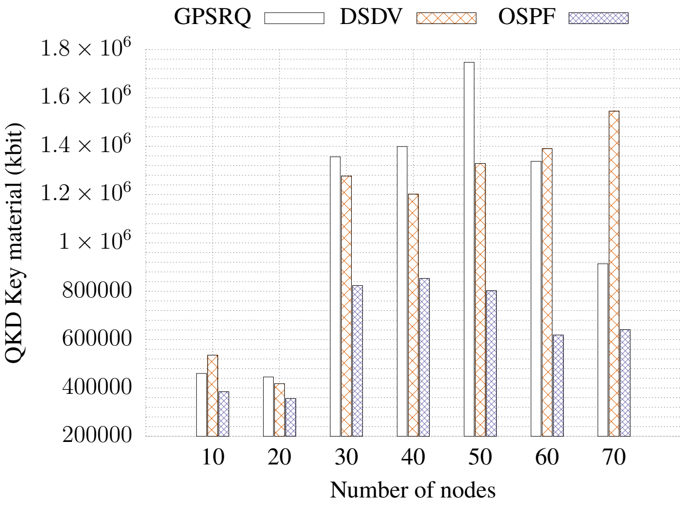

Figure 7.11. Key material consumption. GG graphs

Table 7.5 shows the routing protocol overhead, measured in number and summed sizes of routing protocol packets sent network-wide during the entire simulation for GPSRQ, DSDV and OSPF routing protocols. OSPF exchange its Hello packets each 10 seconds and floods periodic LSA update each 30 minutes which results in a small number of generated packets. On the other hand, DSDV exchanges whole routing tables in the almost regular period of 15 seconds plus it generates additional triggered update packets, which results in a large number of generated packets. GPSRQ relies on knowledge of the geographical position and state of links to neighbors, which enables a high level of network scalability. As such, it exchanges only M_{thr} packets defined in Section 7.3.1 in a period of 15 seconds by default. Also, GPRSQ exchanges M_{thr} each time when a new key material is stored in key material storages which was set to 7 seconds in our simulations (Table 7.4). It is important to note that DSDV and OSPF exchange its routing packets using UDP, while GPSRQ exchanges M_{thr} values using TCP, so table 7.5 includes into calculation all packets including TCP ACK, TCP SYN, TCP FIN and other. In addition, we performed several simulations where M_{thr} has been set to 5,15,20 and 30 seconds besides the exchange of M_{thr} each time when the new key material is added to key material storage. The data showed identical values, from which we can conclude that a single exchange of M_{thr} value in a post-processing period is adequate for the smooth operation of GPSRQ.

In our simulations, values from GPSRQ packets were moved into QKD Command header as described in section 7.4.3 while DSDV and OSPF packets were sent using standard QKD Header as described in section 3.2.1. Table 7.5 shows that GPSRQ consumes the least key material for cryptographic operations on routing packets, while Fig. 7.11 shows the overall key material

consumption in the network.
When taking into account the results of the PDR ratio which is discussed in section below, it is obvious that GPSRQ spends at least key material for securing routing packets in contrast to DSDV and OSPF.

Packet Delivery Ratio

Packet Delivery Success Ratio (PDR) which is calculated as the ratio of received and sent application packets, is used to assess the effectiveness of the routing protocol within the specified simulation environment. Fig. 7.13 shows that GPSRQ in planar GG topologies is able to successfully find an available route to a destination when compared to OSPF and DSDV. Due to the big value of "dead interval", OSPF is not able to react quickly to the changes in network topology which occur due to the lack of key material on discharged links, which finally results in reduced PDR value. DSDV exchanges routing date more often which provides higher PDR value but still insufficiently good as GPSRQ.

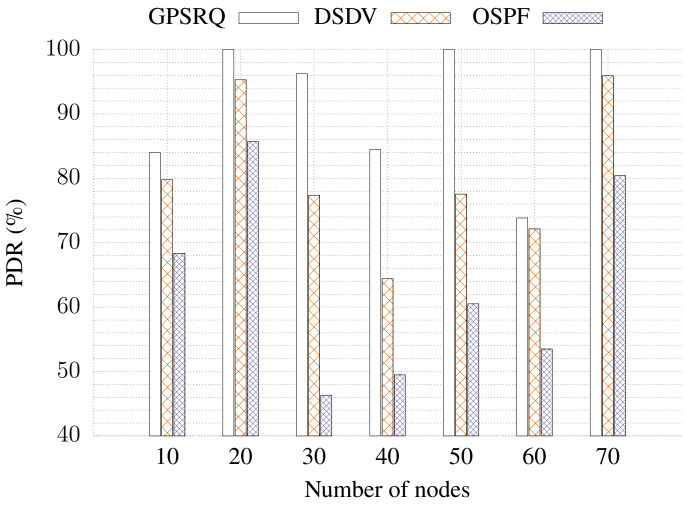

Figure 7.12. Packet Delivery Ratio values. Planar GG graphs

Note that the values for an initial key amount in key storages were generated randomly based on the values from shared seed files in accordance with the settings specified in the table 7.4. It means that there might not be enough resources in the network for maximal PDR value, indicating that all packets sent reached their destination.

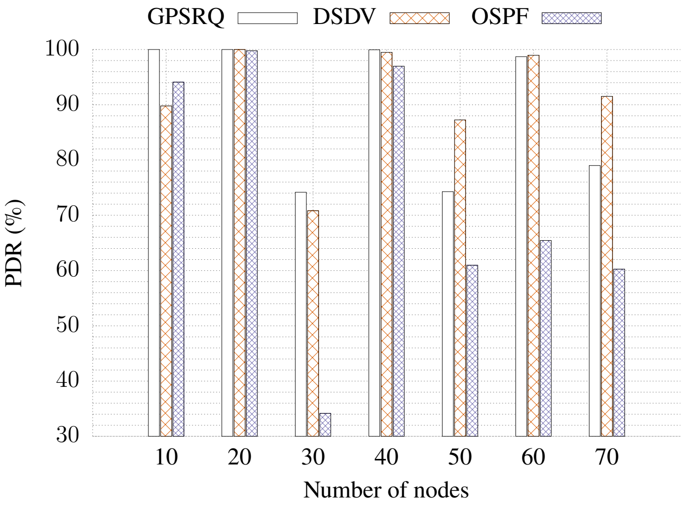

Figure 7.13. Packet Delivery Ratio values. Random non-planar full graphs

Path Length

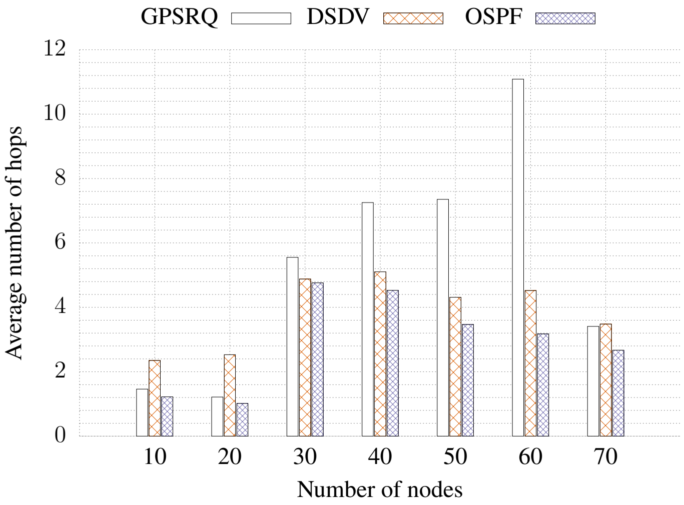

Figure 7.14. Average number of hops. GG graphs

Fig. 7.14 shows the average number of hops between the source and the destination. The data show that GPSRQ recognizes paths to the destination which were not visible to DSDV and OSPF.

More specifically, GPSRQ with the detection mechanism of returning loop increases the number of forwarding, but due to the scalable caching, GPSRQ is able to find the shortest feasible path to the destination.

Time Delay

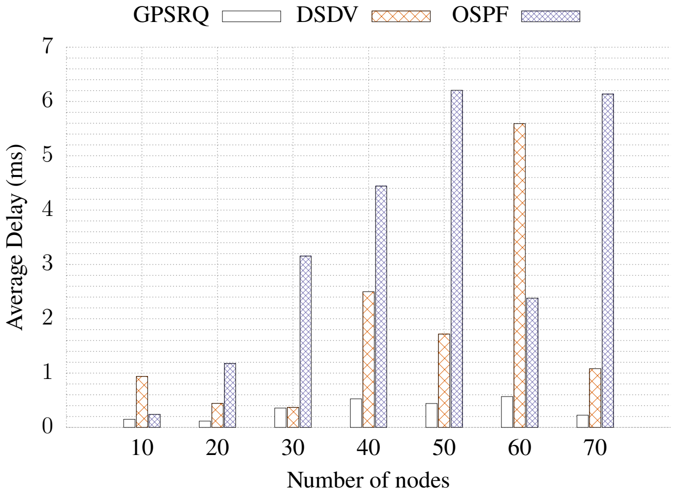

Figure 7.15. Average Delay. GG graphs

The most important aspect of real-time traffic is that a message generated by the source device by an application is time sensitive, and it must be received by the destination device within a given amount of time. Our results show that the impact of scalable caching in GPSRQ has major consequences on the average delay. As shown in Fig. 7.15 these values are significantly lower when compared to other routing protocols due to the active measurements of the state of links and returning loops used to exclude those links that do not lead to the destination. Also, it is important to underline that average the delay of GPSRQ does not increase with the number of nodes, which indicates that GPSRQ supports network scalability and robustness.

Effect of GPSRQ β parameter

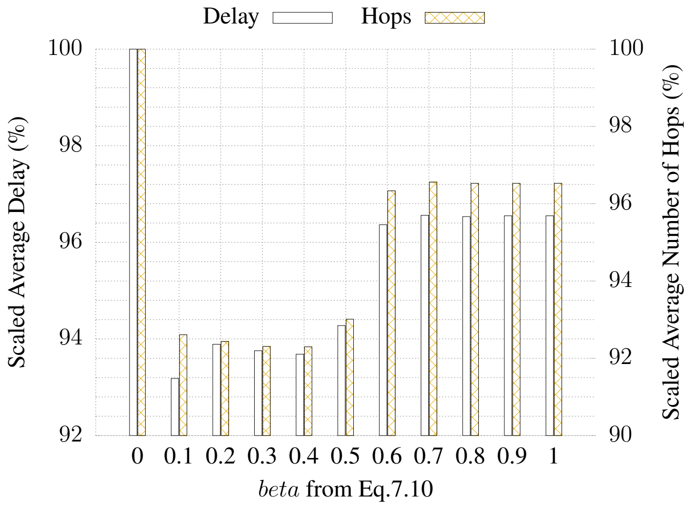

Figure 7.16. Scaled average number of hops and scaled average delay for various values of β (beta) from Eq. (7.10). GG graphs

GPSRO is based on finding a balance between the "geographically shortest path" calculated using Euclidean distance and the path with the best performance which is calculated based on the measurements of public and quantum channel states as defined by Equation (7.10). To show the impact of this parameter, we performed simulations on random networks with 30, 40 and 50 nodes, changing the values of β while other parameters remained fixed as listed in table 7.4.
Fig. 7.16 shows that parameter β which is used in greedy forwarding has a significant impact on the number of hops on the route to the destination. For $\beta = 1$, only geographical distance is considered while $\beta = 0$ means that only states of links are used as the primary routing metric in greedy forwarding mode. Considering that GPSRQ implements recovery-mode as a side algorithm that constantly seeks to forward packets to the destination using right-hand rule, there is no significant differences in the PDR value. However, β directly affects the number of hops to the destination in greedy forwarding which results in increased delay and overall consumption of key material. Although neighboring node may be geographically closer to the destination, without taking into consideration link performance ($\beta = 1$), such as the amount of key material on links which are more than one hop away as described in the example in Fig. 7.4, routing will result in a returning loop and increased delay. On the other hand, routing without taking into account the geographical distance will forward the packet further from the destination ($\beta = 0$), but with the support of recovery-mode, the packet will reach it's destination in the increased

number of hops and a significantly greater delay. Therefore, it is necessary to find a balance which results in the suitable performance. Intuitively, different classes of traffic should be served with different values of β depending on the priority of delivery, which provides GPSRQ flexible tool for traffic management and allows effective utilization of network resources.

Effect of the Public Channel Memory

One of the key parameters of GPSRQ is the number of samples taken for calculating the average duration of key material establishment process in the long run, denoted as $T_{average}$ in Equation (7.8). A large number of samples means that GPSRQ is slow to react to variations in the performance of the public channel. On the contrary, using a few values means that GPSRQ will respond quickly to sudden changes in the public channel. This is the reason for using of $T_{average}$ to determine the validity of entries in the GPSRQ cache T_{cache} as defined by Equation (7.11). Therefore, this value affects the rate of emptying the internal node's cache which has an important impact on the average delay.

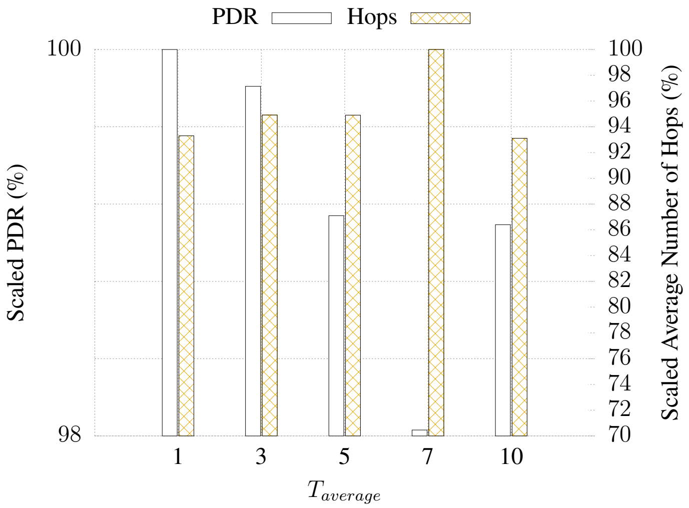

Figure 7.17. Scaled average number of hops and scaled PDR for various values of $T_{average}$. GG graphs

To show the impact of this parameter, we performed simulations on random networks with 30,40 and 50 nodes, changing the values of $T_{average}$ while other parameters remained fixed as listed in table 7.4. Fig.7.17 shows that the average number of hops grows with the $T_{average}$ since GPSRQ tries to alternative routes to the destination. That results in a selection of longer routes toward the destination that can lead to links that do not have enough resources to reach the destination.

Therefore, the value of PDR reduces as the validity of entries in internal caches of nodes stays longer. In addition, we note that setting of $T_{average}$ to the value of timer interval it takes to establish new key material (which was 7 seconds in our simulations), resulting in a significant change of obtained values. Specifically, the addition of new key material changes the performances of links, which can lead to sudden changes in the network topology from the point of view of links availability. This is reflected in the way that GPSRQ with such settings cannot memorize state of the network links, which leads to the increased number of returning loops.

7.5 Summary

The first part of this chapter addressed the issue of providing QoS in QKD network. After examining the existing QoS model from conventional networks such as IntServ and DifServ, we discussed the deficiencies in the application of those models in QKD network. Therefore, we presented a new flexible model for dynamic QKD networks (FQKD). FQKD is based on a distributed approach to control traffic load by providing soft-QoS constraints without using flow or session state information maintained in support of end-to-end communication. It involves the classification of traffic at the ingress node depending on priority of the traffic into the appropriate queues. Although FQKD uses the DSCP field in the IP packet header for classification of the traffic which is one of the primary features of the DiffServ model, what clearly differentiates these two models is the implementation of additional waiting queues at higher ISO/OSI levels to adapt to the dynamic nature of QKD network. FQKD approach is simple since it allows flexible use of network resources depending on the priorities of delivery. It defines guidelines for effective learning of QKD links states using specific metrics for the public and quantum channel, as well as exchanging M_{thr} values which allows node to gain information about the state of links that are more than one hop away. Guided by the results of an analysis of the public channel of QKD links discussed in Chapter 6, we recognized the possibility of exchanging the signaling and routing packets within the traffic that is authenticated as part of key material post-processing process.

In the second part of this chapter, we presented a routing protocol which is designed to match the dynamic nature of QKD network. Due to the dynamic nature of a QKD networks which are characterized by frequent changes of link state, the propagation of the link state packets and routing information across a network would be inherently imprecise considering the short time validity of such information. We provided Greedy Perimeter Stateless Routing Protocol for QKD network (GPSRQ) which uses distributed geography reactive routing to achieve high-level scalability and to minimize the number of routing packets with regard to the requirement for minimizing key material consumption. Further, GPSRQ is equipped with robust caching and detection of returning loops which enables effective forwarding to a destination while minimizing the key material consumption. GPSRQ therefore enables a high level of network scalability and robustness. However, GPSRQ comes with a limitation on the use on planar topologies with the respect that the geographic routing in non-planar topologies may not quickly determine the shortest path toward a destination which leads to unnecessary forwarding and consumption of scarce key material. Taking into account the limits on the geographical distance in the realization of QKD links that limits the deployment of QKD network to metropolitan scale [10, 122, 123] and taking into account that the previously deployed QKD networks were deployed on planar topologies [6–10], we do not consider this restriction as a critical deficiency. Our simulation results clearly showed that GPSRQ achieves significantly better performance when compared to the OSPF and

DSDV routing protocols that were used in previously realized QKD networks. This is particularly reflected in the aspect of minimizing average delay and increasing the Packet Delivery Ratio (PDR) which are the key parameters for efficient use of the real-time application [151, 159].

Chapter 8

Conclusion

Due to noticeable progress in the development of quantum equipment, an attention for convergence of QKD technology with the applications used in everyday life constantly increases. A number of successful demonstrations of QKD networks were performed with the aim to test the implementation and interoperability of different practical solutions. But these demonstrations have not addressed the use of QKD technology for applications of real life enterprise and real-time traffic. The traffic in these networks was mainly considered with equal importance and it was treated with same priority. While such approach may be acceptable for some applications, it can not be considered as a complete solution since different applications may have different service requirements with respect to QoS. Accordingly, methods of traffic management, congestion control, and QoS approaches become an important issue. We believe that the lack of QoS solutions was one of the factors limiting the wider application of QKD cryptography in everyday life.

By analyzing in detail previously deployed QKD networks, a novel simulation model that allows easier testing of QKD network was implemented. This model is implemented in a way that supports both forms of practical realization of QKD networks such as network with a single TCP/IP stack or overlay network. This simulation model is described in detail in Chapter 5 which fulfills the second aim of the research presented in this thesis.

To demonstrate the dependence of the performance of the public channel on the performance of the quantum channel of QKD link and vice versa, the in-depth analysis of the traffic over the public channel in the laboratory conditions as well as on virtual QKD link was performed as described in Chapter 6. The analysis showed a clear interdependence between the two channels, which was used for defining and considering effective QoS solution.

Following considerations dedicated to the requirements and motivation, a novel approach for providing QoS in QKD network was proposed as described in Chapter 7. Analysis of the application of existing QoS model from the conventional networks has been performed which led to the definition of the new QoS model for QKD networks. The new routing metrics are defined by taking into account performance of both channels of QKD link as well as their previously specified interdependence. An important similarity between MANET and QKD technology was observed, which opened the way for the implementation of the novel scalable routing protocol for usage in QKD networks. The proposed theoretical solutions were verified by the performed simulations which showed the efficiency compared to the previously used solutions, and enabled the use of traffic in real-time in QKD networks, thus fulfilling the primary aim of the research presented in this thesis.

8.1 Achievements and contributions

The achievements and contributions of this dissertation can be summarized as follows:

1. An in-depth analysis of the current state of theoretical and practical solutions in the field QKD was carried out.
2. A novel QKD Network Simulation Tool was implemented.
3. An in-depth analysis of the traffic over the public channel of QKD link was performed in the laboratory conditions as well as on virtual QKD link.
4. The analysis of the application of the existing QoS model from conventional networks in QKD networks was carried out.
5. The similarity between QKD and MANET technology is recognized and explained.
6. The new QoS model that corresponds to the dynamic nature of QKD network is defined.
7. New metrics for analyzing the state of public and quantum channels of QKD link are defined.
8. The new metric M_{thr} for measurement of state links that are more than one hop away is defined.
9. The embedment of signaling or routing packets within the authenticated packets that are exchanged in key material post-processing is recognized as an effective way to tackle the problem of distribution of routing and signaling information without introducing additional traffic overhead.
10. The new routing protocol designed for QKD networks is specified (GPSRQ). The protocol was explained in detail. Alongside the main functionality, some protocol extensions were proposal such as: detection of returning loop and scalable caching behaviour.

11. The functionality of the new GPSRQ protocol was implemented and verified in a simulator.

12. Comparison of GPSRQ, DSDV and OSPF routing protocols for different network topologies with a number of simulations was performed. The obtained data confirmed the theoretical considerations.

In light of the presented achievements it can be stated that the hypothesis of this thesis:

It is possible to use real-time applications in QKD networks.

It has been verified as being true.

References

[1] P. P. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science*. IEEE Comput. Soc. Press, 1994, pp. 124–134.

[2] C. H. Bennett and G. Brassard, "Quantum Cryptography: Public Key Distribution and Coin Tossing," in *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, vol. 175, no. 150. New York, 1984, p. 8.

[3] D. Mayers, "Unconditional Security in Quantum Cryptography," *Journal of the ACM*, vol. 48, no. 3, pp. 351–406, 2001.

[4] P. W. Shor and J. Preskill, "Simple proof of security of the BB84 quantum key distribution protocol," *Physical Review Letters*, vol. 85, no. 2, pp. 441–444, 2000.

[5] M. N. Wegman and J. L. Carter, "New hash functions and their use in authentication and set equality," *Journal of Computer and System Sciences*, vol. 22, no. 3, pp. 265–279, 1981.

[6] R. Alleaume, J. Bouda, C. Branciard, T. Debuisschert, M. Dianati, N. Gisin, M. Godfrey, P. Grangier, T. Langer, A. Leverrier, N. Lutkenhaus, P. Painchault, M. Peev, A. Poppe, T. Pornin, J. Rarity, R. Renner, G. Ribordy, M. Riguidel, L. Salvail, A. Shields, H. Weinfurter, and A. Zeilinger, "SECOQC White Paper on Quantum Key Distribution and Cryptography," *arXiv preprint quant-ph/0701168*, p. 28, 2007.

[7] C. Elliott and H. Yeh, "DARPA Quantum Network Testbed," BBN Technologies Cambridge, New York, USA, BBN Technologies Cambridge, New York, USA, Tech. Rep. July, 2007.

[8] F. X. Xu, W. Chen, S. Wang, Z. Q. Yin, Y. Zhang, Y. Liu, Z. Zhou, Y. B. Zhao, H. W. Li, D. Liu, Z. F. Han, and G. C. Guo, "Field Experiment on a Robust Hierarchical Metropolitan Quantum Cryptography Network," *Chinese Science Bulletin*, vol. 54, no. 17, pp. 2991–2997, 2009.

[9] M. Sasaki, M. Fujiwara, H. Ishizuka, W. Klaus, K. Wakui, M. Takeoka, and S. Miki, “Field Test of Quantum Key Distribution in the Tokyo QKD Network,” *Optics Express*, vol. 19, no. 2011, aug 2011.

[10] S. Wang, W. Chen, Z.-Q. Yin, H.-W. Li, D.-Y. He, Y.-H. Li, Z. Zhou, X.-T. Song, F.-Y. Li, D. Wang, H. Chen, Y.-G. Han, J.-Z. Huang, J.-F. Guo, P.-L. Hao, M. Li, C.-M. Zhang, D. Liu, W.-Y. Liang, C.-H. Miao, P. Wu, G.-C. Guo, and Z.-F. Han, “Field and Long-term Demonstration of a Wide Area Quantum Key Distribution Network,” *Optics Express*, vol. 22, no. 18, p. 21739, sep 2014.

[11] M. Tomamichel, C. C. W. Lim, N. Gisin, and R. Renner, “Tight finite-key analysis for quantum cryptography.” *Nature communications*, vol. 3, no. May 2011, p. 634, jan 2012.

[12] S. Rass and S. König, “Turning Quantum Cryptography against itself: How to avoid indirect eavesdropping in quantum networks by passive and active adversaries,” *International Journal on Advances in Systems and Measurements*, vol. 5, no. 1, pp. 22–33, 2012.

[13] Q. Le, P. Bellot, A. Demaille, L. Quoc-Cuong, P. Bellot, and A. Demaille, “Towards the World-Wide Quantum Network,” *Information Security Practice and Experience*, vol. 2, no. iii, pp. 218–232, 2008.

[14] C. Le Quoc, P. Bellot, and A. Demaille, “Stochastic routing in large grid-shaped quantum networks,” *2007 IEEE International Conference on Research, Innovation and Vision for the Future, RIVF 2007*, no. APRIL 2007, pp. 166–174, 2007.

[15] S. Konig and S. Rass, “On the Transmission Capacity of Quantum Networks,” *International Journal of Advanced Computer Science and Applications(IJACSA)*, vol. 2, no. 11, pp. 9–16, 2011.

[16] W. Hao, H. Zheng-Fu, G. Guang-Can, and H. Pei-Lin, “The Queueing Model for Quantum Key Distribution Network,” *J.Phys.G*, vol. G36, no. 7, p. 25006, jan 2009.

[17] S. Rass, “On Game-Theoretic Network Security Provisioning,” *Journal of Network and Systems Management*, vol. 21, no. 1, pp. 47–64, 2013.

[18] A. Fedrizzi, A. Poppe, R. Ursin, T. Lorünser, M. Peev, T. Länger, and A. Zeilinger, “Practical quantum key distribution with polarization entangled photons,” *2005 European Quantum Electronics Conference, EQEC ’05*, vol. 2005, no. 16, p. 303, 2005.

[19] P. Stavroulakis and M. Stamp, *Handbook of Information and Communication Security*. Heidelberg: Springer, 2010.

[20] A. Mink, X. Tang, L. Ma, T. Nakassis, B. Hershman, J. C. Bienfang, D. Su, R. Boisvert, C. W. Clark, and C. J. Williams, “High Speed Quantum Key Distribution System Supports One-Time Pad Encryption of Real-Time Video,” *Proceedings of SPIE*, vol. 6244, pp. 62 440M–1–7, 2006.

[21] P. Morgen, “Geneva Vote Will Use Quantum Cryptography,” 2007.

[22] A. Tajima, A. Tanaka, W. Maeda, S. Takahashi, and A. Tomita, “Practical quantum cryptosystem for metro area applications,” *IEEE Journal on Selected Topics in Quantum Electronics*, vol. 13, no. 4, pp. 1031–1037, 2007.

[23] A. Mirza and F. Petruccione, “Realizing Long-Term Quantum Cryptography,” *Journal of the Optical Society of America B*, vol. 27, no. 6, p. A185, jun 2010.

[24] T. Langer, “The Practical Application of Quantum Key Distribution,” Ph.D. Thesis, University of Lausanne, 2013.

[25] J. T. Moy, “OSPF Version 2,” *Internet Requests for Comment*, vol. RFC 1247, pp. 1–124, 1991.

[26] C. Elliott, A. Colvin, D. Pearson, O. Pikalo, J. Schlafer, and H. Yeh, “Current status of the DARPA quantum network (Invited Paper),” in *Proc. SPIE 5815, Quantum Information and Computation III,*, E. J. Donkor, A. R. Pirich, and H. E. Brandt, Eds., vol. 5815, may 2005, pp. 138–149.

[27] E. Pearson Brig Barnum and David Spencer, “Systems and methods for implementing routing protocols and algorithms for quantum cryptographic key transport,” apr 2010.

[28] Y. Tanizawa, R. Takahashi, and A. R. Dixon, “A routing method designed for a Quantum Key Distribution network,” *International Conference on Ubiquitous and Future Networks, ICUFN*, vol. 2016-Augus, no. 1, pp. 208–214, 2016.

[29] M. Dianati, R. All, R. Alléaume, M. Gagnaire, and X. S. Shen, “Architecture and Protocols of the Future European Quantum Key Distribution Network,” *Security and Communication Networks*, vol. 1, no. 1, pp. 57–74, jan 2008.

[30] C. Kollmitzer and M. Pivk, *Applied Quantum Cryptography*. Springer Science & Business Media, 2010, vol. 797.

[31] J.-y. Sun, J. Lang, C. Miao, N. Yang, and S. Wang, “A digital watermarking algorithm based on hyperchaos and discrete fractional Fourier transform,” *2012 5th International Congress on Image and Signal Processing*, pp. 552–556, oct 2012.

[32] Xianzhu Cheng, Yongmei Sun, and Yuefeng Ji, "A QoS-supported Scheme for Quantum Key Distribution," in *2011 International Conference on Advanced Intelligence and Awareness Internet (AIAI 2011)*, no. 2009. IET, 2011, pp. 220–224.

[33] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing," *Wireless Networks*, vol. 11, no. 4, pp. 419–434, 2005.

[34] G. Apostolopoulos, R. Guerin, and S. Kamat, "Implementation and Performance Measurements of QoS Routing Extensions to OSPF," *IEEE INFOCOM '99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No.99CH36320)*, vol. 2, pp. 1–25, 1999.

[35] M. Niemiec, L. Romanski, and M. Swiety, "Quantum Cryptography Protocol Simulator," in *Communications in Computer and Information Science*, 2011, vol. 149 CCIS, pp. 286–292.

[36] A. Pereszlenyi, "Simulation of Quantum Key Distribution with Noisy Channels," in *Proceedings of the 8th International Conference on Telecommunications, 2005. ConTEL 2005.*, vol. 1. IEEE, jun 2005, pp. 203–210.

[37] X. Zhang and Q. Wen, "Object-Oriented Quantum Cryptography Simulation Model," in *hird International Conference on Natural Computation*, no. ICNC. IEEE, 2007, pp. 7–10.

[38] S. Zhao, H. De Raedt, B. Liu, and Y. Huang, "Event-by-event Simulation of Quantum Cryptography Protocols," *Journal of Computational and Theoretical Nanoscience*, vol. 5, no. 7, pp. 1251–1254, 2008.

[39] A. Buhari, "An efficient modeling and simulation of quantum key distribution protocols using OptiSystem™," in *IEEE Symposium on Industrial Electronics and Applications (ISIEA)*, Bandung, 2012, pp. 84–89.

[40] L. O. Mailloux, J. D. Morris, M. R. Grimaila, D. D. Hodson, D. R. Jacques, J. M. Colombi, C. V. Mclaughlin, and J. A. Holes, "A Modeling Framework for Studying Quantum Key Distribution System Implementation Nonidealities," *IEEE Access*, vol. 3, pp. 110–130, 2015.

[41] S. Wiesner, "Conjugate coding," *ACM Sigact News*, vol. 15, no. 1, pp. 78–88, 1983.

[42] G. Brassard, "Brief history of quantum cryptography: a personal perspective," in *IEEE Information Theory Workshop on Theory and Practice in Information-Theoretic Security, 2005.* IEEE, 2005, pp. 19–23.

[43] W. Heisenberg, “Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik,” *Zeitschrift für Physik*, vol. 43, no. 3-4, pp. 172–198, mar 1927.

[44] W. K. Wootters and W. H. Zurek, “A single quantum cannot be cloned,” *Nature*, vol. 299, no. 5886, pp. 802–803, oct 1982.

[45] R. Renner, “Security of Quantum Key Distribution,” Ph.D. dissertation, PhD Thesis, SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH, dec 2005.

[46] C. E. Shannon, “Communication Theory of Secrecy Systems,” *Bell system technical journal*, vol. 15, no. July, pp. 57–64, 1948.

[47] G. Vernam, “Cipher printing telegraph systems: For secret wire and radio telegraphic communications,” *AIEE, Journal of the*, 1926.

[48] L. Marks, *Between Silk and Cyanide: A Codemaker’s War, 1941-1945*. Simon and Schuster, 2001.

[49] A. Abidin and J.-Å. Larsson, “Security of Authentication with a Fixed Key in Quantum Key Distribution,” p. 14, 2011.

[50] C. Portmann, “Key Recycling in Authentication,” *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 4383–4396, 2014.

[51] M. Niemiec and A. R. Pach, “The Measure of Security in Quantum Cryptography,” *2012 IEEE Global Communications Conference (GLOBECOM)*, pp. 967–972, dec 2012.

[52] G. Brassard and L. Salvail, “Secret-key reconciliation by public discussion,” *advances in Cryptology—EUROCRYPT’93*, pp. 410–423, 1994.

[53] T. Sugimoto and K. Yamazaki, “A Study on Secret Key Reconciliation Protocol,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E83-A, no. 10, 2000.

[54] N. RUTH II-YUNG, “A Probabilistic Analysis of Binary and Cascade,” *math.uchicago.edu*, 2013.

[55] M. Niemiec, “Design , Construction and Verification of a High-Level Security Protocol Allowing to Apply the Quantum Cryptography in Communication Networks,” Ph.D. dissertation, AGH University of Science and Technology, Krakow, Poland, 2011.

[56] A. Abidin, “Authentication in Quantum Key Distribution : Security Proof and Universal Hash Functions,” PhD Thesis, PhD Thesis, Linköping University, 2013.

[57] J. Cederlöf and J. a. Larsson, “Security Aspects of the Authentication used in Quantum Cryptography,” *IEEE Transactions on Information Theory*, vol. 54, no. 4, pp. 1735–1741, 2008.

[58] G. Gilbert and M. Hamrick, “Practical Quantum Cryptography: A Comprehensive Analysis (Part One),” *arxiv:quant-ph*, pp. 1–194, 2000.

[59] J. L. Carter and M. N. Wegman, “Universal classes of hash functions,” *Journal Of Computer And System Sciences*, vol. 18, no. 2, pp. 143–154, 1979.

[60] C. Bennett, “Quantum cryptography using any two nonorthogonal states,” *Physical Review Letters*, 1992.

[61] S. J. Lomonaco, “A Quick Glance at Quantum Cryptography,” *Cryptologia*, vol. 23, no. 1, pp. 1–41, nov 1999.

[62] G. Smith, J. Renes, and J. Smolin, “Structured Codes Improve the Bennett-Brassard-84 Quantum Key Rate,” *Physical Review Letters*, vol. 100, no. 17, p. 170502, apr 2008.

[63] A. Beige, B.-G. Englert, C. Kurtsiefer, and H. Weinfurter, “Secure communication with a publicly known key,” nov 2001.

[64] M. Dusek, N. Lutkenhaus, and M. Hendrych, “Quantum Cryptography,” in *Progress in Optics*. Elsevier, jan 2006, vol. 49, pp. 381–454.

[65] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, “Quantum Cryptography,” *Reviews of Modern Physics*, vol. 74, no. 1, pp. 145–195, jan 2002.

[66] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, “The security of practical quantum key distribution,” *Reviews of Modern Physics*, vol. 81, no. 3, pp. 1301–1350, sep 2009.

[67] R. Alleaume, F. Roueff, E. Diamanti, and N. Lutkenhaus, “Topological optimization of quantum key distribution networks,” *New Journal of Physics*, vol. 11, no. 7, p. 075002, jul 2009.

[68] L. Salvail, M. Peev, E. Diamanti, R. Alléaume, N. Lütkenhaus, and T. Länger, “Security of trusted repeater quantum key distribution networks,” *Journal of Computer Security*, vol. 18, no. 1, pp. 61–87, jan 2010.

[69] S. Aleksic, D. Winkler, G. Franzl, A. Poppe, B. Schrenk, and F. Hipp, "Quantum Key Distribution Over Optical Access Networks," *Proceedings of the 2013 18th European Conference on Network and Optical Communications & 2013 8th Conference on Optical Cabling and Infrastructure (NOC-OC&I)*, pp. 11–18, 2013.

[70] K. Patel, J. Dynes, I. Choi, A. Sharpe, A. R. Dixon, Z. Yuan, R. Penty, and A. Shields, "Coexistence of High-Bit-Rate Quantum Key Distribution and Data on Optical Fiber," *Physical Review X*, vol. 2, no. 4, p. 041010, nov 2012.

[71] C. Elliott, D. Pearson, and G. Troxel, "Quantum Cryptography in Practice," *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '03*, p. 227, 2003.

[72] A. Poppe, M. Peev, and O. Maurhart, "Outline of the SECOQC Quantum-Key-Distribution Network in Vienna," *Journal Of Quantum Information*, vol. 6, no. 2, p. 10, apr 2008.

[73] R. J. Runser, T. Chapuran, P. Toliver, N. A. Peters, M. S. Goodman, J. T. Kosloski, N. Nweke, S. R. McNown, R. J. Hughes, D. Rosenberg, C. G. Peterson, K. P. McCabe, J. E. Nordholt, K. Tyagi, P. A. Hiskett, and N. Dallmann, "Progress Toward Quantum Communications Networks: Opportunities and Challenges," in *Optoelectronic Integrated Circuits IX*, no. 240, 2007, pp. 64 760I–64 760I–15.

[74] T. Schmitt-Manderbach, H. Weier, and M. Fürst, "Experimental Demonstration of Free-Space Decoy-State Quantum Key Distribution over 144 km," *Physical Review Letters*, 2007.

[75] S. Nauerth, F. Moll, M. Rau, C. Fuchs, J. Horwath, S. Frick, and H. Weinfurter, "Air-to-ground quantum communication," *Nature Photonics*, vol. 7, no. 5, pp. 382–386, mar 2013.

[76] C. Bonato, A. Tomaello, V. D. Deppo, G. Naletto, and P. Villoresi, "Some Aspects on the Feasibility of Satellite Quantum Key Distribution," *New Journal of Physics*, vol. 11, no. 4, p. 045017, 2009.

[77] J. G. Rarity, P. R. Tapster, P. M. Gorman, and P. Knight, "Ground to Satellite Secure Key Exchange Using Quantum Cryptography," *New Journal of Physics*, vol. 4, no. 1, pp. 82–82, 2002.

[78] A. V. A. Sergienko, *Quantum Communications and Cryptography*. CRC Press, 2005, vol. 2005.

[79] C. Elliott, “Building the Quantum Network,” *New Journal of Physics*, vol. 4, p. 346, jul 2002.

[80] T. Länger and G. Lenhart, “Standardization of quantum key distribution and the ETSI standardization initiative ISG-QKD,” *New Journal of Physics*, vol. 11, no. November, 2009.

[81] G. Lenhart, “QKD standardization in ETSI,” *Qcw 2010*, vol. 57, pp. 1–36, 2010.

[82] O. Maurhart, C. Pacher, A. Happe, T. Lor, C. Tamas, A. Poppe, and M. Peev, “New release of an open source QKD software : design and implementation of new algorithms , modularization and integration with IPSec,” in *Qcrypt 2013*, 2013.

[83] M. Sasaki, “Tokyo QKD Network and the evolution to Secure Photonic Network,” in *CLEO:2011 - Laser Applications to Photonic Applications*, vol. 1. Washington, D.C.: OSA, 2011, p. JTuC1.

[84] M. Marhoefer, I. Wimberger, and A. Poppe, “Applicability of quantum cryptography for securing mobile communication networks,” *Long-Term and Dynamical Aspects of Information Security: Emerging Trends in Information and Communication Security*, pp. 97–111, 2007.

[85] M. Hayashi, K. Iwama, H. Nishimura, R. Raymond, and S. Yamashita, “Quantum Network Coding,” in *Lecture Notes in Computer Science*, ser. Lecture Notes in Computer Science, W. Thomas and P. Weil, Eds., vol. 4393. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 610–621.

[86] R. Alléaume, F. Roueff, O. Maurhart, N. Lütkenhaus, E. Nationale, R. Alleaume, F. Roueff, O. Maurhart, and N. Lutkenhaus, “Architecture, security and topology of a global Quantum Key Distribution network,” *2006 Digest of the LEOS Summer Topical Meetings*, pp. 38–39, 2006.

[87] S. Marksteiner, “An Approach to Securing IPsec with Quantum Key Distribution (QKD) Using the AIT QKD software,” Ph.D. dissertation, CAMPUS 02 University of Applied Sciences, Graz, Austria, 2014.

[88] T.-Y. Chen, H. Liang, Y. Liu, W.-Q. Cai, L. Ju, W.-Y. Liu, J. Wang, H. Yin, K. Chen, Z.-B. Chen, C.-Z. Peng, and J.-W. Pan, “Field Test of a Practical Secure Communication Network With Decoy-State Quantum Cryptography,” *Optics Express*, vol. 17, no. 8, p. 6540, apr 2009.

[89] T.-Y. Chen, J. Wang, H. Liang, W.-Y. Liu, Y. Liu, X. Jiang, Y. Wang, X. Wan, W.-Q. Cai, L. Ju, L.-K. Chen, L.-J. Wang, Y. Gao, K. Chen, C.-Z. Peng, Z.-B. Chen, and J.-W. Pan, "Metropolitan All-Pass and Inter-City Quantum Communication Network," *Optics express*, vol. 18, no. 26, pp. 27 217–27 225, 2010.

[90] Z.-f. Han, F.-X. Xu, W. Chen, S. Wang, Z.-Q. Yin, Y. Zhang, Y. Liu, Z. Zhou, H.-W. Li, D. Liu, and G.-C. Guo, "An Application-Oriented Hierarchical Quantum Cryptography Network Test Bed," *Optical Fiber Communication Conference*, p. OTuK4, 2010.

[91] S. Wang, W. Chen, Z.-Q. Yin, Y. Zhang, T. Zhang, H.-W. Li, F.-x. Xu, Z. Zhou, Y. Yang, D.-J. Huang, L.-J. Zhang, F.-Y. Li, D. Liu, Y.-G. Wang, G.-C. Guo, and Z.-F. Han, "Field Test of Wavelength-Saving Quantum Key Distribution Network," *Optics letters*, vol. 35, no. 14, pp. 2454–6, 2010.

[92] F. Xu, B. Qi, and H. K. Lo, "Experimental Demonstration of Phase-Remapping Attack in a Practical Quantum Key Distribution System," *New Journal of Physics*, vol. 12, pp. 1–16, 2010.

[93] A. R. Dixon, Z. L. Yuan, J. F. Dynes, A. W. Sharpe, and A. J. Shields, "Continuous operation of high bit rate quantum key distribution," *Applied Physics Letters*, vol. 96, no. 2010, pp. 2008–2011, 2010.

[94] B. Korzh, C. C. W. Lim, R. Houlmann, N. Gisin, M. J. Li, D. Nolan, B. Sanguinetti, R. Thew, and H. Zbinden, "Provably secure and practical quantum key distribution over 307 km of optical fibre," *Nature Photonics*, vol. 9, no. 3, pp. 163–168, feb 2015.

[95] S. Wang, W. Chen, J.-F. Guo, Z.-Q. Yin, H.-W. Li, Z. Zhou, G.-C. Guo, and Z.-F. Han, "2 GHz clock quantum key distribution over 260 km of standard telecom fiber," *Optics Letters*, vol. 37, no. 6, p. 1008, 2012.

[96] K. Shimizu, T. Honjo, M. Fujiwara, T. Ito, K. Tamaki, S. Miki, T. Yamashita, H. Terai, Z. Wang, and M. Sasaki, "Performance of Long-Distance Quantum Key Distribution Over 90-km Optical Links Installed in a Field Environment of Tokyo Metropolitan Area," *Journal of Lightwave Technology*, vol. 32, no. 1, pp. 141–151, jan 2014.

[97] D. Collins, N. Gisin, and H. de Riedmatten, "Quantum Relays for Long-distance Quantum Cryptography," *Journal of Modern Optics*, vol. 52, pp. 735–753, 2005.

[98] W. Dur, H.-J. Briegel, J. I. Cirac, and P. Zoller, "Quantum Repeaters Based on Entanglement Purification," *Physical Review A*, vol. 59, no. 1, pp. 169–181, 1999.

[99] Z.-S. Yuan, Y.-A. Chen, B. Zhao, S. Chen, J. Schmiedmayer, and J.-W. Pan, "Experimental demonstration of a BDCZ quantum repeater node." *Nature*, vol. 454, no. 7208, pp. 1098–1101, 2008.

[100] R. Alleaume, C. Branciard, J. Bouda, T. Debuisschert, M. Dianati, N. Gisin, M. Godfrey, P. Grangier, T. Langer, N. Lutkenhaus, C. Monyk, P. Painchault, M. Peev, A. Poppe, T. Pornin, J. Rarity, R. Renner, G. Ribordy, M. Riguidel, L. Salvail, A. Shields, H. Weinfurter, and A. Zeilinger, "Using Quantum Key Distribution for Cryptographic Purposes: A Survey," *Theoretical Computer Science*, vol. 560, no. P1, pp. 62–81, dec 2014.

[101] S. J. van Enk, "Photonic Channels for Quantum Communication," *Science*, vol. 279, no. 5348, pp. 205–208, jan 1998.

[102] D. Bouwmeester, A. K. Ekert, and A. Zeilinger, *The Physics of Quantum Information: Quantum Cryptography, Quantum Teleportation, Quantum Computation*. Springer Science & Business Media, 2010.

[103] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. a. Smolin, and W. K. Wootters, "Purification of Noisy Entanglement and Faithful Teleportation via Noisy Channels," *Physical Review Letters*, vol. 76, no. 5, pp. 722–725, jan 1996.

[104] A. C. Begen, Y. Altunbasak, O. Ergun, and M. H. Ammar, "Multi-path Selection for Multiple Description Video Streaming over Overlay Networks," *EURASIP Journal of Signal Processing: Image Communications*, vol. 20, no. 1, pp. 39–60, 2005.

[105] Zheng Ma, Huai-Rong Shao, and Chia Shen, "A New Multi-path Selection Scheme for Video Streaming on Overlay Networks," in *2004 IEEE International Conference on Communications (IEEE Cat. No.04CH37577)*, vol. 3, no. 1. IEEE, 2004, pp. 1330–1334.

[106] C. Tang and P. McKinley, "Improving Multipath Reliability in Topology-Aware Overlay Networks," *25th IEEE International Conference on Distributed Computing Systems*, pp. 82–88, 2005.

[107] S. Tao, K. Xu, A. Estepa, T. Fei, L. Gao, R. Guerin, J. Kurose, D. Towsley, and Z. L. Zhang, "Improving VoIP Quality Through Path Switching," *Proceedings - IEEE INFOCOM*, vol. 4, no. C, pp. 2268–2278, 2005.

[108] B. Augustin, T. Friedman, and R. Teixeira, "Measuring Multipath Routing in the Internet," *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 830–840, 2011.

[109] R. Teixeira, K. Marzullo, S. Savage, and G. M. Voelker, "Characterizing and Measuring Path Diversity of Internet Topologies," *SIGMETRICS Perform. Eval. Rev.*, vol. 31, no. 1, pp. 304–305, 2003.

[110] S. Hares, Yakov Rekhter, and Tony Li, "A Border Gateway Protocol 4 (BGP-4)," *RFC 4271*, 2006.

[111] C. Labovitz, A. Ahuja, R. Wattenhofer, and S. Venkatachary, "The impact of Internet policy and topology on delayed routing convergence," *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, vol. 1, pp. 537–546, 2001.

[112] K. I. Park, *QoS in packet networks*, 2005.

[113] ITU-T, "Recommendation E. 800, Terms and definitions related to quality of service and network performance including dependability," Tech. Rep., 1994.

[114] B. Kyrbashov, I. Baroňák, M. Kováčik, and V. Janata, "Evaluation and investigation of the delay in voip networks," *Radioengineering*, vol. 20, no. 2, pp. 540–547, 2011.

[115] J. Frnda, M. Voznak, J. Rozhon, and M. Mehic, "Prediction Model of QoS for Triple Play Services," in *2013 21st Telecommunications Forum Telfor (TELFOR)*. IEEE, nov 2013, pp. 733–736.

[116] J. Davidson and J. Peters, *Voice Over IP Fundamentals*, 2000.

[117] L. Sun, I.-H. Mkwawa, E. Jammeh, and E. Ifeachor, *Guide to Voice and Video over IP*, ser. Computer Communications and Networks. London: Springer London, 2013.

[118] Cisco, "Quality of Service for Voice over IP," no. 1, p. 40, 2001.

[119] G. Lu, "Issues and Technologies for Supporting Multimedia Communications over the Internet," *Computer Communications*, vol. 23, no. 14-15, pp. 1323–1335, 2000.

[120] K. Fall and K. Varadhan, "The Network Simulator (ns-2)," *URL: http://www. isi. edu/nsnam/ns*, 2007.

[121] E. Altman and T. Jimenez, *NS Simulator for Beginners*. Morgan & Claypool Publishers, 2012, vol. 5.

[122] A. Ciurana, J. Martinez-Mateo, M. Peev, A. Poppe, N. Walenta, H. Zbinden, and V. Martin, “Quantum Metropolitan Optical Network Based on Wavelength Division Multiplexing,” *Optics express*, vol. 22, no. 2, pp. 1576–93, 2014.

[123] M. Peev, C. Pacher, R. Alléaume, C. Barreiro, J. Bouda, W. Boxleitner, T. Debuisschert, E. Diamanti, M. Dianati, J. F. Dynes, S. Fasel, S. Fossier, M. Fürst, J. D. Gautier, O. Gay, N. Gisin, P. Grangier, A. Happe, Y. Hasani, M. Hentschel, H. Hübel, G. Humer, T. Länger, M. Legré, R. Lieger, J. Lodewyck, T. Lorünser, N. Lütkenhaus, A. Marhold, T. Matyus, O. Maurhart, L. Monat, S. Nauerth, J. B. Page, A. Poppe, E. Querasser, G. Ribordy, S. Robyr, L. Salvail, A. W. Sharpe, A. J. Shields, D. Stucki, M. Suda, C. Tamas, T. Themel, R. T. Thew, Y. Thoma, A. Treiber, P. Trinkler, R. Tualle-Brouri, F. Vannel, N. Walenta, H. Weier, H. Weinfurter, I. Wimberger, Z. L. Yuan, H. Zbinden, and A. Zeilinger, “The SECOQC quantum key distribution network in Vienna,” *New Journal of Physics*, vol. 11, no. 7, p. 075001, jul 2009.

[124] A. R. Dixon, Z. L. Yuan, J. F. Dynes, A. W. Sharpe, and A. J. Shields, “Gigahertz decoy quantum key distribution with 1 Mbit/s secure key rate,” *Optics Express*, vol. 16, no. 23, p. 18790, 2008.

[125] S. Rass, A. Wiegele, and P. Schartner, “Building a Quantum Network: How to Optimize Security and Expenses,” *Journal of Network and Systems Management*, vol. 18, no. 3, pp. 283–299, 2010.

[126] G.-V. Assche, *Quantum Cryptography and Secret-Key Distribution*. Cambridge University Press, 2006.

[127] O. contributors, “OpenStreetMap. Planet dump,” https://www.openstreetmap.org, 2017.

[128] Y. Shavitt and N. Zilberman, “A geolocation databases study,” *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 10, pp. 2044–2056, 2011.

[129] T. Pedersen and M. Toyran, “High Performance Information Reconciliation for QKD with CASCADE,” *Quantum Information & Computation*, vol. 15, no. 5-6, pp. 419–434, 2015.

[130] A. Mink and J. Bienfang, “QKD on a Board Limited by Detector Rates in a Free-Space Environment,” *ICQNM 2013 : The Seventh International Conference on Quantum, Nano and Micro Technologies*, no. c, pp. 28–33, 2013.

[131] D. Elkouss, A. Leverrier, R. Alleaume, and J. J. Boutros, “Efficient Reconciliation Protocol for Discrete-Variable Quantum Key Distribution,” in *2009 IEEE International Symposium on Information Theory*. IEEE, jun 2009, pp. 1879–1883.

[132] B. Hubert, “The Wonder Shaper,” 2002.

[133] L. Zhang, S. Deering, and D. Estrin, “RSVP: A new resource reservation protocol,” *Network*, vol. 7, no. 5, pp. 8–18, 1993.

[134] C. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin, “Experimental quantum cryptography,” *Journal of Cryptology*, vol. 5, no. 1, 1992.

[135] Y. Zhu, C. Dovrolis, and M. Ammar, “Dynamic Overlay Routing Based on Available Bandwidth Estimation: A Simulation Study,” *Computer Networks*, vol. 50, no. 6, pp. 742–762, apr 2006.

[136] D. G. Andersen, A. C. Snoeren, and H. Balakrishnan, “Best-path vs. multi-path overlay routing,” *Proceedings of the 2003 ACM SIGCOMM conference on Internet measurement - IMC ’03*, p. 91, 2003.

[137] P. Fazio, F. De Rango, and C. Sottile, “A Predictive Cross-Layered Interference Management in a Multichannel MAC with Reactive Routing in VANET,” *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 1850–1862, aug 2016.

[138] P. Fazio, M. Tropea, F. De Rango, and M. Voznak, “Pattern Prediction and Passive Bandwidth Management for Hand-Over Optimization in QoS Cellular Networks with Vehicular Mobility,” *IEEE Transactions on Mobile Computing*, vol. 1233, no. c, pp. 1–1, 2016.

[139] F. D. Rango and P. Fazio, “A New Distributed Application and Network Layer Protocol for VoIP in Mobile Ad Hoc Networks,” *IEEE Transactions on Mobile Computing*, vol. 13, no. 10, pp. 2185–2198, 2014.

[140] K. Sarkar, T. Basavaraju, and C. Puttamadappa, *Ad Hoc Mobile Wireless Networks*. CRC Press, 2008, vol. 1.

[141] P. Schartner, S. Rass, and M. Schaffer, *Quantum Key Management*. InTech, 2012.

[142] Y. Wang and Y. Desmedt, “Perfectly secure message transmission revisited,” *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2582–2595, 2008.

[143] O. Maurhart, T. Lorunser, T. Langer, C. Pacher, M. Peev, and a. Poppe, “Node modules and protocols for the Quantum-Back-Bone of a quantum-key-distribution network,” *2009 35th European Conference on Optical Communication*, pp. 3–4, 2009.

[144] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," *Sosp*, vol. 32, no. 1, p. 66, 2001.

[145] B. Karp and H. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," 2000.

[146] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," *Proceedings of the 6th annual international conference on Mobile computing and networking - MobiCom '00*, pp. 120–130, 2000.

[147] S. Wijesekera, "Quantum Cryptography for Secure Communication in IEEE 802 . 11 Wireless Networks," Ph.D. dissertation, 2011.

[148] K. H. Sheikh, S. S. Hyder, and M. M. Khan, "An overview of quantum cryptography for wireless networking infrastructure," *Proceedings of the 2006 International Symposium on Collaborative Technologies and Systems, CTS 2006*, vol. 2006, pp. 379–385, 2006.

[149] G. Vallone, D. Bacco, D. Dequal, S. Gaiarin, V. Luceri, G. Bianco, and P. Villoresi, "Experimental Satellite Quantum Communications," *Physical Review Letters*, vol. 115, no. 4, p. 040502, jul 2015.

[150] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. a. Woodward, "A distance routing effect algorithm for mobility (DREAM)," *Proceedings of the 4th annual ACMIEEE international conference on Mobile computing and networking MobiCom 98*, pp. 76–84, 1998.

[151] J. Frnda, M. Voznak, and L. Sevcik, "Impact of Packet Loss and Delay Variation on the Quality of Real-time Video Streaming," *Telecommunication Systems*, pp. 1–11, 2015.

[152] B. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1617–1622, 1988.

[153] K. Pussep, C. Leng, and S. Kaune, *Modeling and Tools for Network Simulation*, 2010.

[154] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: an approach to universal topology generation," in *MASCOTS 2001, Proceedings Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. IEEE Comput. Soc, 2001, pp. 346–353.

[155] B. N. Karp, "Geographic Routing for Wireless Networks," Doctoral dissertation, Harvard University Cambridge, Massachusetts, 2000.

[156] K. R. Gabriel and R. R. Sokal, “A New Statistical Approach to Geographic Variation Analysis,” *Systematic Zoology*, vol. 18, no. 3, pp. 259–278, 1969.

[157] K. M. Chandy and J. Misra, “Distributed computation on graphs: shortest path algorithms,” *Communications of the ACM*, vol. 25, no. 11, pp. 833–837, 1982.

[158] C. E. Perkins and P. Bhagwat, “Highly dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers,” *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 4, pp. 234–244, 1994.

[159] M. Voznak, A. Kovac, and M. Halas, “Effective Packet Loss Estimation on VoIP Jitter Buffer,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7291 LNCS, pp. 157–162, 2012.

Candidate's research cited in the thesis

[MFV+16] M. Mehic, P. Fazio, M. Voznak, P. Partila, D. Komosny, J. Tovarek, and Z. Chmelikova. On using multiple routing metrics with destination sequenced distance vector protocol for MultiHop wireless ad hoc networks. page 98480F. International Society for Optics and Photonics, may 2016.

[MFVC16] Miralem Mehic, Peppino Fazio, Miroslav Voznak, and Erik Chromy. Toward Designing a Quantum Key Distrubution Network. *Advances in Electrical and Electronic Engineering*, 14(4):413–420, 2016.

[MKM+16] Miralem Mehic, Dan Komosny, Oliver Mauhart, Miroslav Voznak, and Jan Rozhon. Impact of Packet Size Variation in Overlay Quantum Key Distribution Network. In *Telecommunications (BIHTEL), 2016 XI International Symposium on*, pages 1–6, Sarajevo, Bosnia and Herzegovina, 2016. IEEE.

[MNV15] Miralem Mehic, Marcin Niemiec, and Miroslav Voznak. Calculation of the Key Length for Quantum Key Distribution. *Elektronika ir Elektrotechnika*, 21(6):81–85, dec 2015.

[MPTV15] Miralem Mehic, Pavol Partila, Jaromir Tovarek, and Miroslav Voznak. Calculation of key reduction for B92 QKD protocol. In Eric Donkor, Andrew R. Pirich, and Michael Hayduk, editors, *SPIE Sensing Technology + Applications*, page 95001J. International Society for Optics and Photonics, may 2015.

APPENDICES

Appendix A

List of candidate's research results and activities

ORCID iD: orcid.org/0000-0003-2697-1756 Scopus Author ID: 56043054700

- records in Elsevier Scopus: 17 (14 conference papers, 3 articles in journals)
- records in ISI Web of Knowledge: 15 (14 conference papers, 1 article in journal)
- records in IEEE-Xplore: 3
- h-index according to ISI/WoS: 2 (6 citations)
- h-index according to Scopus: 2 (16 citations)

A.1 Results with wider relation to the topic of dissertation indexed on Web of Science or in Elsevier Scopus

1. M. Mehic, M. Mikulec, M. Voznak, L. Kapicak, "Creating Covert Channel Using SIP," In SPRINGER Communications in Computer and Information Science CCIS, Volume 429, 2014, pp.182-192, doi: 10.1007/978-3-319-07569-3_15. SJR 0.148 (2014/Q4), DOCUMENT TYPE: Conference Paper
2. M. Mehic, M. Voznak, J. Safarik, P. Partila, M. Mikulec, "Using DNS amplification DDoS attack for hiding data," In Proc. SPIE. 9120, Mobile Multimedia/Image Processing, Security, and Applications 2014, 91200R. (Baltimore, Maryland, USA, May 22, 2014) doi: 10.1117/12.2050700. SJR 0.22 (2014), DOCUMENT TYPE: Conference Paper

3. M. Mehic, J. Slachta, M. Voznak, Hiding Data in SIP Session, In Proc. 37th International Conference on Telecommunication and Signal Processing, Berlin, July 1-3, 2014, ISBN 978-80-214-4983-1, ISSN 1805-5435, pp. 18-22. DOCUMENT TYPE: Conference Paper

4. M. Voznak, I. Zbranek, M. Mehic, D. Komosny, H. Toral-Cruz, J. W. Lin, Covert Channel in RTP Payload Using a Pointer in SIP Header, Communications, Volume 18, Issue 1, 2016, pp. 40-47, SJR 0.281 (2014/Q3), DOCUMENT TYPE: Article

5. M. Voznak, J. Rozhon, P. Partila, J. Safarik, M. Mikulec, M. Mehic, "Predictive model for determining the quality of a call," In Proc. SPIE. 9118, Independent Component Analyses, Compressive Sampling, Wavelets, Neural Net, Biosystems, and Nanoengineering XII, 91180Y. (Baltimore, Maryland, USA, May 22, 2014) doi: 10.1117/12.2050661. SJR 0.22 (2014), DOCUMENT TYPE: Conference Paper

6. J. Frnda, M. Voznak, J. Rozhon, M. Mehic, Prediction Model of QoS for Triple Play Services, 21st Telecommunications Forum Telfor, TELFOR 2013, 2013, Article number 6716334, pp. 733-736. DOCUMENT TYPE: Conference Paper

A.2 Other results of research within PhD. study indexed on Web of Science or in Elsevier Scopus

1. M. Voznak, M. Prokes, L. Sevcik, J. Frnda, H. Toral-Cruz, S. Jakovlev, P. Fazio, M. Mehic, M. Mikulec, Vulnerabilities in GSM technology and feasibility of selected attacks, In Proc. SPIE 9456, Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security, Defense, and Law Enforcement XIV, 94560T, Baltimore, Maryland, USA. (May 23, 2015) doi: 10.1117/12.2177111.0.216 (2015), DOCUMENT TYPE: Conference Paper

2. J. Tovarek, P. Partila, M. Voznak, M. Mikulec, M. Mehic, Detection of cardiac activity changes from human speech, In Proc. SPIE. 9496, Independent Component Analyses, Compressive Sampling, Large Data Analyses (LDA), Neural Networks, Biosystems, and Nanoengineering XIII, 94960V, Baltimore, Maryland, USA. (May 21, 2015) doi: 10.1117/12.2177282. SJR 0.216 (2015), DOCUMENT TYPE: Conference Paper

3. M. Mikulec, M. Voznak, J. Safarik, P. Partila, J. Rozhon, M. Mehic, "Interactive video audio system: communication server for INDECT portal," In Proc. SPIE. 9120, Mobile Multimedia/Image Processing, Security, and Applications 2014, 91200U. (Baltimore,

Maryland, USA, May 22, 2014) doi: 10.1117/12.2050690. SJR 0.22 (2014), DOCUMENT TYPE: Conference Paper

4. P. Partila, M. Voznak, T. Peterek, M. Penhaker, V. Novak, J. Tovarek, M. Mehic, L. Vojtech, "Impact of human emotions on physiological characteristics," In Proc. SPIE. 9118, Independent Component Analyses, Compressive Sampling, Wavelets, Neural Net, Biosystems, and Nanoengineering XII, 91180W. (Baltimore, Maryland, USA, May 22, 2014) doi: 10.1117/12.2050679. SJR 0.22 (2014), DOCUMENT TYPE: Conference Paper

5. J. Safarik, M. Voznak, M. Mehic, P. Partila, M. Mikulec, "Neural network classifier of attacks in IP telephony," In Proc. SPIE. 9118, Independent Component Analyses, Compressive Sampling, Wavelets, Neural Net, Biosystems, and Nanoengineering XII, 91180X. (Baltimore, Maryland, USA, May 22, 2014) doi: 10.1117/12.2050671. SJR 0.22 (2014), DOCUMENT TYPE: Conference Paper

6. M. Voznak, P. Partila, M. Mehic, S. Jakovlev, Recognizing Emotions from Human Speech Using 2-D Neural Classifier and Influence the Selection of Input Parameters on its Accuracy, 21st Telecommunications Forum Telfor, TELFOR 2013, 2013, Article number 6716272, pp. 482-485. DOCUMENT TYPE: Conference Paper

A.3 Workshops

1. "Approaches to Routing in QKD networks" ,Mehic M., Workshop on Advances Communication Technologies, Ostrava, Czech Republic in April 2015

2. "Analysis of DDoS attack", Mehic M. Workshop on Selected Issues in Communication Technologies, Ostrava, Czech Republic in March 2014

A.4 Teaching

1. Introduction to Telecommunication Technologies - Lectures and Exercises (Summer semester 2014, VSB-TUO, Ostrava, Czech Republic)

2. Introduction to Telecommunication Technologies - Lectures and Exercises (Summer semester 2015, VSB-TUO, Ostrava, Czech Republic)

A.5 Research Stays

1. Erasmus Study Exchange in Krakow, Poland from 3.3.2014 to 8.7.2014.
   Supervisor: Dr. ing. Marcin Niemiec

2. Erasmus+ Internship in Klagenfurt, Austria from 1.10.2015 to 31.03.2016.
   Supervisor: Dr. ing. ing. Stefan Rass

3. Traineeship in Austrian Institute of Technology (AIT) from 1.10.2015 to 30.11.2015.

A.6 Participating in projects of specific research

1. Research on impact of atmospheric phenomenas on communication in radio channel. SGS SP2014/72, 2014.

2. Knowledge retrieval from communication networks, modelling and simulation - I. SP2015/82, 2015.

3. Knowledge retrieval from communication networks, modelling and simulation - II. SP2016/170, 2016

A.7 Mentoring Experience

1. Filip Rajnec (Bachelor Thesis), 2015, VSB-TUO Ostrava, Czech Republic

A.8 Awards

1. September 2015 - Dean's appreciation award for the results achieved during Ph.D. study, Faculty of Electrical Engineering and Computer Science, VŠB-TUO

Appendix B

Supplementary material

As additional material of this thesis, a flash drive is attached that contains the following:

- This Ph.D. Thesis in .pdf format
- BRITE source code - also publicly available on https://bitbucket.org/mickeyze/brite-planar-graph
- GPSRQ and DSDV source code
- Simulation scripts used in experiments
- ReadMe instructions

Note: QKDNetSim is publicly available on https://bitbucket.org/liptel/qkdnetsim