SDN-Enabled Resiliency in Computer Networks

Jonathan Vestin

Faculty of Health, Science and Technology

Computer Science

LICENTIATE THESIS | Karlstad University Studies | 2018:17

SDN-Enabled Resiliency in Computer Networks

Jonathan Vestin

LICENTIATE THESIS | Karlstad University Studies | 2018:17

SDN-Enabled Resiliency in Computer Networks

Jonathan Vestin

LICENTIATE THESIS

Karlstad University Studies | 2018:17

urn:nbn:se:kau:diva-66992

ISSN 1403-8099

ISBN 978-91-7063-848-0 (tryck)

ISBN 978-91-7063-943-2 (pdf)

Distribution:
Karlstad University
Faculty of Health, Science and Technology
Department of Mathematics and Computer Science
SE-651 88 Karlstad, Sweden
+46 54 700 10 00

Tryck: Universitetstryckeriet, Karlstad 2018

WWW.KAU.SE

SDN-Enabled Resiliency in Computer Networks

Jonathan Vestin
Department of Mathematics and Computer Science

Abstract

In computer networking, failures, such as breaking equipment, cable cuts, power failures and human errors continuously cause communication interruptions. Such failures may result in dissatisfied customers, loss of product reputation, violation of SLAs and even critical failures in industrial systems. Recently, the concept of SDN was introduced. SDN opens up and centralizes the control plane, which allows designing networks more resilient to failures.

In this thesis, we address the usage of SDN in order to provide resiliency in high availability networks. First, we consider how SDN enabled, proactive failure recovery can be used to provide the reliability required in ICNs. We also investigate how the same approach could be applied to mmWave backhaul networks to cope with fast channel attenuation and the resulting outage. Through extensive experiments, we can demonstrate an increase in reliability for both ICNs and mmWave backhaul networks. Second, we look at Split MAC-based Wireless LAN, and how SDN-enabled traffic control algorithms could improve connection reliability. Through our experiments we can show that both discriminatory and non-discriminatory algorithms significantly increase the connection reliability. In combination, these results serve to strengthen the image of SDN as a provider of resilient, high-availability networks.

Keywords: networking,sdn,openflow,resiliency,icn,mmwave,5g,wlan

Acknowledgements

I am grateful for the support of my supervisor and mentor Andreas Kassler, and my co-supervisors Karl-Johan Grinnemo and Johan Åkerberg. Also thanks to Peter Dely and Martin Blom, both for encouraging my continued studies in computer science.

Karlstad University, April 16, 2018 Jonathan Vestin

Contents

List of Appended Papers

I. **Jonathan Vestin** and Andreas Kassler. QoS enabled WiFi MAC layer processing as an example of a NFV service. *2015 1st IEEE Conference on Network Softwarization (NetSoft)*, London, United Kingdom 2015

Comments on Participation: I implemented the QoS extensions to Open vSwitch, which includes support for SFQ, CoDel and FQ_CoDel. I also extended Open vSwitch to support classless (non-discriminatory) traffic control algorithms. I set up and performed the experiments, and analyzed the results. I am also responsible for a significant portion of the written text.

II. **Jonathan Vestin**, Andreas Kassler and Johan Åkerberg. Resilient software defined networking for industrial control networks. *2015 10th International Conference on Information, Communications and Signal Processing (ICICS)*, Singapore 2015

Comments on Participation: I designed the experiments, implemented the required extensions to the CORE emulator. I setup and performed the experiments, and analyzed the results. I am also responsible for a majority of the text.

III. **Jonathan Vestin** and Andreas Kassler. Low frequency assist for mmWave backhaul - the case for SDN resiliency mechanisms. *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, Paris, France 2017

Comments on Participation: I implemented the mmWave LOS/N-LOS/outage model in the CORE emulator. I designed, set up and ran the experiments and also analyzed the results. I am the main author of the text.

Other Papers

Apart from the papers included in this thesis, I have co-authored the following papers.

1. **Jonathan Vestin** and Andreas Kassler. QoS Management for WiFi MAC Layer Processing in the Cloud: Demo Description. *In Proceedings of the 11th ACM Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet)*, Cancun, Mexico 2015, **Best Demo Award**

   Relation to Included Paper: This paper presents the Open Daylight Extensions, and integration with InfluxDB and Grafana to Paper I.

2. **Jonathan Vestin** and Andreas Kassler. Resilient SDN based small cell backhaul networks using mmWave bands. In *IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Coimbra, Portugal 2016

   Relation to Included Paper: This paper includes a presentation of the emulation environment used in Paper III, and also includes preliminary evaluation results.

3. **Jonathan Vestin** and Andreas Kassler. SDN enabled resiliency in LTE assisted small cell mmWave backhaul networks. In *2017 20th Conference*

on Innovations in Clouds, Internet and Networks (ICIN), Paris, France 2017

Relation to Included Paper: This paper is a preliminary paper to Paper III, which focuses more on packet loss and latency.

4. **Jonathan Vestin** and Andreas Kassler. QoS Enabled WiFi MAC Layer Processing as an Example of a NFV Service. In: Proceedings of Swedish Communication Technologies Workshop (Swe-CTW), Karlstad, Sweden 2015.

   Relation to Included Paper: Poster version of Paper I

Previous Papers

These are papers published before starting my PhD studies and are listed for reference.

1. Peter Dely, **Jonathan Vestin** and Andreas Kassler. CloudMAC - Using OpenFlow to Process 802.11 MAC Frames in the Cloud. In: PIK - Praxis der Informationsverarbeitung und Kommunikation, Volume 36, Issue 1.

2. Dely, P. and **Vestin, J.** and Kassler, A. and Bayer, N and Einsiedler, H and Peylo, C. CloudMAC - An OpenFlow based architecture for 802.11 MAC layer processing in the cloud. In: Globecom Workshops (GC Wkshps), 2012 IEEE.

3. **Vestin, J.** and Dely, P. and Kassler, A. and Bayer, N. and Einsiedler, H. and Peylo, C. CloudMAC - Towards software defined WLANs. In: Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM 2012.

Introductory Summary

1 Introduction

One of the challenges in computer networking is how to provide resiliency, that is, provide a network which is able to withstand failures. Failures in networking equipment include device failures, cable cuts, power outages, signal fading, software errors, human errors etc. [1]. Some of these failures, such as cable cuts, may cause network packets to be lost. Packet losses can be a significant threat to the Quality of Experience (QoE) for the users of the network. For example in Voice Over IP, lost packets cause interruptions in the voice data, which makes the users perceive the voice service as unreliable [2, 3]. Other examples of situations where packet loss negatively impact the user experience are online video and video gaming [4]. Furthermore, lengthy disruptions in packet transmission, such as when moving between access points in a wireless network with no handovers, can interrupt the service completely. Such interruption may terminate an ongoing voice call or cancel a download midway.

An area where the end-to-end reliability is of an even higher importance is machine-to-machine communication, in particular in an industrial setting. Manufacturing plants, avionics, railway systems, and other industrial processes require a near 100% data transmission and reception reliability [5]. Many of these networks are deployed in extreme environments, much unlike home and office networks. This may include extreme temperatures, altitudes, pressure and acceleration which further increase the probability of a network connectivity failure. Also, these networks typically transmit very time-sensitive data, such as the communication between a temperature sensor and a cooling system in an industrial plant, or the communication of heading data to the course correction system in nautical or avionic environments. Failure to reliably transmit such data can have profound consequences, e.g., resulting in both monetary loss and risk to human health.

Network equipment used in both regular and machine-to-machine communication often use vendor-specific configuration interfaces, which do not allow the addition of custom architectures and protocols. One approach to improve the situation is the usage of Software Defined Networking (SDN) [6], which opens up and logically centralizes the control plane. This allows the implementation of a custom control plane (implemented by a SDN controller), that can manage the forwarding logic of multiple switches. As the communication protocol between the controller and the switch typically is implemented as an open standard, this increases both the flexibility and manageability of the network. With the centralized view provided by SDN, broken links and malfunctioning equipment can be detected the SDN control plane, which in turn, can recalculate new paths through the network, bypassing the faulty equipment.

One of the problems with this failure recovery approach is that the detection and subsequent repair of a broken link can take a significant amount of time [7, 8]. Also, the recovery time is typically not very predictable. However, the most well-researched and developed SDN protocol, OpenFlow [9], has

in later versions included the possibility to configure fast failover, a method for providing local link repair. Fast failover requires some underlying technology for detecting the link failure, such as Ethernet link integrity detection. Failover can be performed as fast as within (3.3 ± 0.8) ms [10], depending on the configuration.

In this thesis, we propose using SDN-enabled technologies, such as SDN-configured Quality of Service (QoS) [11] and fast failover combined with Bi-Directional Forwarding Detection (BFD) [12], to improve the reliability of packet transmission in multiple types of computer networks. We extend the CloudMAC wireless network architecture to include SDN-based QoS configuration, improving the connection establishment. We also investigate how fast failover, combined with packet replication can be used to improve the resiliency of Industrial Control Networks (ICNs). Further, we attempt to utilize fast failover in order to reduce the packet loss in mmWave networks caused by fast channel attenuation. This is further extended with a last-resort LTE backup link, which is configured to be used as a final backup link, should all available mmWave links fail.

The rest of the thesis is structured as follows. Section 2 introduces the reader to SDN-enabled resiliency, ICN, mmWave networks and CloudMAC, which serves as background material for the main contributions. Section 3 describes the research objectives, while Section 4 briefly describes the scientific methods used. Section 5 presents the main contributions of this work. Section 6 summarizes the appended papers. Finally, Section 7 provides concluding remarks and discussion of potential future work. The thesis concludes with an appendix containing the full text of the papers which this work was primarily based on. The appended papers have been slightly modified for formatting.

2 Background

This section will introduce the background of the thesis. First, the background behind and the functionality of various resiliency improving technologies will be described. Then, SDN and the options for network resiliency it provides will be described. Finally, the application areas which we focus on in this thesis, Wireless Local Area Network (WLAN), ICN and mmWave networks in 5G, are described.

2.1 Resiliency in Computer Networks

Ethernet, the most commonly used link layer protocol, was not designed with high reliability and rapid failure detection in mind. However, in certain application areas, such as in industrial networking, there is a demand for technologies which provide high reliability. Such technologies can be divided into two groups: alternate path protocols and parallel operation protocols [5]. Alternate path protocols provide one or more alternate traffic paths, which are used if the primary network path fails. One example of a technology which provides reliability through alternate path operation is the Spanning Tree

Protocol (STP). STP builds a spanning tree of the network, and disables links which are not part of the tree. This prevents loops in the network, and when a failure is detected, disabled links can be returned to service [13]. In practice, this typically takes around 30 s. Failures in STP are detected through information frames called Bridge Protocol Data Units (BPDUs), which switches periodically send containing details of the switch, such as port status, detected loops, etc. An improvement of STP is Rapid Spanning Tree Protocol (RSTP) [14]. RSTP improves STP by adding a specialized failure detection and recovery mechanism. Recovery in RSTP is much faster and failure detection to repair times are typically around 1 s, even though in some specific topologies and configurations it can be considerably faster [5]. Another protocol which provides alternate path resiliency is Media Redundancy Protocol (MRP) [15]. MRP is used in ring topologies and initially puts one of the links into a disabled state. Traffic is run across the enabled ring path, until a link in that paths fails. When the failure is detected, the previously disabled link is enabled and traffic can traverse it. MRP provides varying recovery times depending on the amount of devices, but can reach as low as 10 ms in smaller networks.

The other type of high availability mechanism is parallel path operation. In parallel path operation, data is replicated over multiple paths, providing recovery times as low as 0 ms. However, this requires concurrently running network devices, which can increase both installation and operational costs. Furthermore, these protocols often require specialized hardware and/or software support in the end nodes. One of these protocols is the Parallel Redundancy Protocol (PRP) [16]. In PRP, the sender duplicates packets and sends them over two completely separate networks. The destination node then proceeds to discard any duplicate packets which it receives. This provides increased reliability in controlled environments, however, larger networks require routers to connect LAN segments, which PRP does not support, making it less useful. Further, PRP requires duplicate MAC addresses, which makes network management more difficult, and duplicates traffic indiscriminately, potentially duplicating traffic which does not need high availability. The IP Parallel Redundancy Protocol (iPRP) [17] is an improvement to PRP, which solves these issues and allows reliability over IP-based networks. Another type of parallel path operation protocol is High-Availability Seamless Redundancy (HSR) [18], where the sender sends all traffic across two separate paths. The recipient detects duplicate packets and discards them before sending the packet to the application layer.

2.2 Software Defined Networking

Computer networking devices can be divided into three logical planes: the data plane, the control plane and the management plane. The management plane performs management and orchestration of the network, typically spanning both the control plane and the data plane. The management plane defines the policies from which the control plane is built from [19], and can also provide monitoring and management of the control and data plane through a

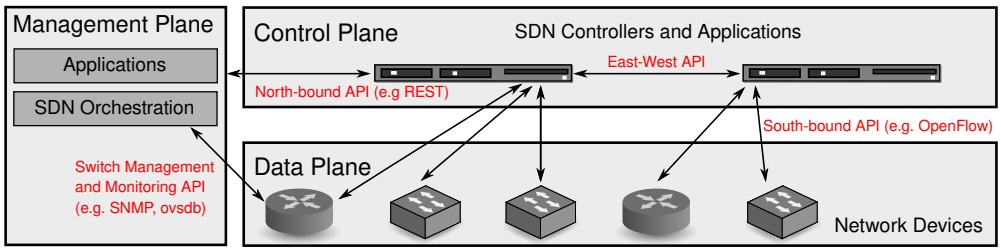


Figure 1: Overview of typical SDN architecture.

Network Management System (NMS), using protocols like Simple Network Management Protocol (SNMP) [13, 20]. The control layer in turn maintains the policy defined on the network management plane. This involves deciding where to send traffic, communicating to other network devices through routing protocols (such as Border Gateway Protocol (BGP) and Open Shortest Path First (OSPF)), traffic control configuration and configuring the match-action tables of the data plane. Finally, the data plane (also known as the forwarding plane) performs the time-critical packet forwarding. Due to early design considerations of the Internet, the control and data planes are traditionally tightly coupled, and the configuration is distributed [21].

To cope with the proprietary nature of computer networking devices, a new architecture called Software Defined Networking (SDN) [6] was introduced. SDN breaks up the traditional network device, separating the control and data plane. The control plane is then implemented through one or more logically centralized controllers (also called Network Operating Systems (NOSs)). Each of these SDN controllers can manage multiple network forwarding devices, through a south-bound API (e.g. OpenFlow). The effect can be seen in Figure 1, which shows a basic overview of a typical SDN architecture.

Much like in traditional networking, the controllers manage the forwarding logic of these forwarding devices, along with other control plane functions such as path calculation and traffic control configuration. This greatly simplifies forwarding devices, as the control plane functions can be relocated to a separate, more powerful device. It also increases the flexibility of the control plane which can obtain a centralized view of the network, using it to calculate more optimal routing strategies. Furthermore, the SDN Controller can be written in a high-level language and run on any device. This allows network device manufacturers to keep the low-level implementation details of their devices hidden from competitors, while it also allows network administrators and researchers to perform experiments introducing new protocols and architectures using SDN-enabled devices.

Finally, just like in traditional networking, the management plane interacts with both the control and data plane. The layer typically consists of multiple SDN applications, whose functionality range from network provisioning to accounting. These application communicate with the control plane using a north-bound interface (e.g. REST). One of the major responsibilities of the management layer applications is SDN Orchestration, which involves manag-

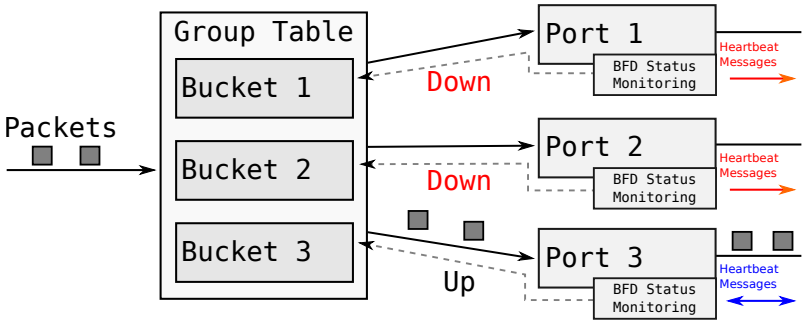


Figure 2: BFD and Fast Failover

ing multiple controllers, ensuring Service Level Agreement (SLA) adherence, providing network administrators with monitoring information and other high-level functions.

In SDN, there are two main approaches for providing link recovery: *proactive* and *reactive* recovery. Using the reactive approach, when a link fails, the SDN controller is informed, re-calculates new forwarding rules and installs them in the device where the link failed. Other devices may also have their forwarding tables updated as required. As the controller has to receive information of the failed link, calculate new forwarding rules, and perform the rule installation, reactive recovery typically takes comparatively long time. The proactive approach improves the reaction time by pre-installing the available backup paths in the forwarding devices. The forwarding devices continuously monitor link liveness, and should the link go down, one of the live alternate backup links are used instead. This allows *local* recovery. The downside is an increase in the size of the flow tables. Further, non-local failure requires additional consideration, such as using Crankback forwarding [10] or OpenState [22].

One of the most popular SDN protocols for the south-bound interface is OpenFlow [9]. OpenFlow has seen much development recently, and can be considered one of the standard SDN south-bound protocols, with support from both hardware (e.g. Pica8 [23]) and software (e.g. Open vSwitch [24]) switches. With the introduction of OpenFlow 1.1 [25], support for group tables was added. A group table is a group of output ports combined with a group type. Instead of forwarding an incoming packet to a specific port, the controller can forward it to a group table. The group table will then, depending on the group type, select one or more of its ports, and forward the packet. For example, the `select` group type uses a selection algorithm to decide which port to forward the packet to, providing simple load-balancing. Fast failover is another group type, which sends packets to the first live port. If no ports are considered live, the packets are dropped. This is considered a type of *proactive* link recovery. Fast failover requires that the ports in the group table have some liveness detection method configured. There are multiple methods for detecting the liveness of the link, such as relying on Ethernet Normal Link Pulses (NLPs) or using another protocol like BFD.

Bi-Directional Forwarding Detection (BFD) [12] is a platform agnostic

method for rapid detection of communication failures, which can be used over multiple underlying protocols (network layer, link layer etc.). BFD can also be used over both single hop links and multihop routed paths. Initially BFD establishes a BFD session, through a three-way handshake, between the link endpoints. When the session is established, BFD enters one of two modes: the *asynchronous mode* or the *demand mode*. Using the asynchronous mode, the endpoints periodically transmit BFD control messages to one another. Failure is declared at either endpoint when no BFD messages have been received for a certain time. This time T_{down} is independently calculated at each endpoint. It depends on the detection time multiplier M and the transmit interval of the other endpoint T_i and is calculated as: $T_{down} = T_i * M$. In addition, BFD also adds a 0% to 25% jitter to prevent synchronization with other network traffic.

The other mode of operation is the demand mode. In demand mode, the endpoints do not continuously send BFD messages, but rather only perform a short message exchange when the system decides it needs to verify connectivity. The popular software switch Open vSwitch [24] implements BFD with UDP as underlying transport protocol. BFD is used in Open vSwitch to inform fast failover type group tables of the liveness of each configured output port. Recent improvements in Open vSwitch, allow for very low BFD message intervals. This enables Open vSwitch to detect link failures within (3.3 ± 0.8) ms [10]. The relation between fast failover and BFD can be seen in Figure 2. The figure shows how the ports (1, 2, 3) are monitored by BFD, and the status of the port is provided to the group table, which uses the first live port (3) to transmit data.

In Paper II and III, we use an *proactive* approach combining BFD and fast failover in order to provide network resiliency. This is achieved through pre-calculating backup paths for each flow, and pre-installing them in the network devices. BFD monitors the link liveness while fast failover performs the traffic redirection. This helps alleviate packet losses in both ICN and mmWave small cell backhaul networks.

2.3 Traffic Control in SDN

In addition to configuring the forwarding table of forwarding devices, SDN can also include the management of which traffic control algorithm should be used. A traffic control algorithm (also called queueing algorithm) is an algorithm used by the forwarding device to determine how packets should be prioritized. Traffic control is most often performed on the egress port, that is the output port, of the network device. The most commonly used traffic control algorithm is First In, First Out (FIFO). FIFO enqueues outgoing packets on a particular port in a queue data structure, and then equeueing packets in a first-come first-serve manner. Traffic control algorithms can be divided into two types, discriminatory and non-discriminatory algorithms. Discriminatory algorithms classify packets based on preconfigured classification criteria, such as IP address or port number. This classification is then used to determine which packets should be prioritized. An example of a discriminatory algorithm is Hierarchical

Token Bucket (HTB) [26]. Non-discriminatory algorithms do not classify packets, but rather treat them equally. An example of a non-discriminatory algorithm is Stochastic Fair Queueing (SFQ) [27].

The OpenFlow 1.3 specification [28] allow specifying which output queue a packet should be assigned to. This configuration is used to classify packets for discriminatory traffic control algorithms. However, OpenFlow does not specify a method for configuring which algorithm should be used, or what parameters it should be configured with. For this a separate protocol has to be used. One popular protocol for switch configuration, which supports configuring traffic control, is `ovsdb` [29]. This protocol is supported by both controllers, such as OpenDaylight, and switches such as Open vSwitch. In Paper I, our traffic control algorithm extensions to Open vSwitch is configured through `ovsdb`. CloudMAC traffic is sensitive to cross traffic induced delays. Using our extensions, the SDN controller can configure non-discriminatory queueing algorithms in addition to discriminatory. These new algorithms include Active Queue Management (AQM) algorithms that help reduce delays cause by queue buildup. This ultimately improved the reliability of CloudMAC.

2.4 IEEE 802.11 Wireless Local Area Networks

In wireless networks, computers communicate using wireless signals which are transmitted through a wireless medium (such as through air), as opposed to wired networking, where signals are transmitted through a wired medium, such as a network cable. IEEE 802.11 WLAN is one standard for wireless communication, which utilizes the 2.4 GHz frequency band. WLANs consists of wireless devices (called stations) and wireless Access Pointss (APs). The AP acts as an intermediary, receiving messages from stations and forwarding them to other stations. Alternatively, the AP may forward the message through a wired interface, mapping between a WLAN and a Ethernet LAN. These messages, typically called frames, are divided into three different categories: *control frames*, *management frames* and *data frames*. Control frames are messages used in part to compensate for some of the difficulties of wireless transmission. These include Request to Send (RTS)/Clear to Send (CTS) frames and wireless acknowledgements. Management frames are messages which govern the connection and management of stations and access points in the networks, and include authentication, association and information requests. The final type of WLAN frame are the data frames, used for data transfer between stations and access points [13, 30].

Connecting to a WLAN is performed through an exchange of management frames between a station and an access point. This exchange can be seen in Figure 3. The access point periodically transmits beacon frames which contain network information, such as the Basic Service Set Identifier (BSSID) and Service Set Identifier (SSID). The station can use this information to determine which APs are available. The connection attempt starts with the station probing the AP, to gain more information about its capabilities. This

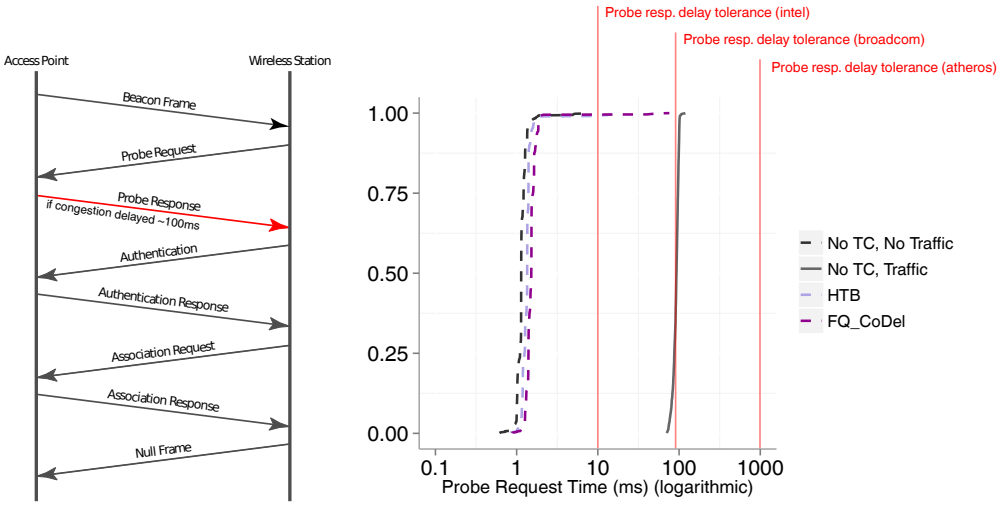


Figure 3: WLAN connection process and probe response deadlines for different manufacturers.

is done through sending a probe request to the AP. The probe request is responded to by a probe response, which contains information about the AP. Next, the station sends an authentication management frame and gets an authentication response back. The authentication frame contains credentials for the network, if any are required. Then, the station sends an association request and the AP responds with an association response. Finally, a null frame is sent back from the wireless station to the AP.

There are strict timing requirements imposed on some wireless management and control frames. For example wireless acknowledgements should be received within a very short time interval (9 µs by default using 802.11a or 802.11g), and are therefore typically generated in firmware or hardware. Another type of management frame, which has specific timing requirements, is the probe response frame. From our experiments performed in Paper I, we can see that different manufacturers impose different delay tolerances on probe response frames, and use this to determine if a network should be connected to or not. In traditional WLANs, the control and management frames are processed and generated at the AP, so the delays are typically very low. However, in split MAC architectures, which may increase the management frame delay, there can be difficulties in providing a stable connection process. One such network is the CloudMAC [31], which moves the processing of management frames to the cloud. By processing such frames on a separate machine, the network between the AP and the cloud machine may introduce delays to the probe response frames. Paper I investigates what delay tolerances are permissible with various Wireless Network Interface Card (WNIC) vendors and attempted to improve the processing speed by using discriminatory traffic control strategies which prioritize CloudMAC frames over normal network traffic.

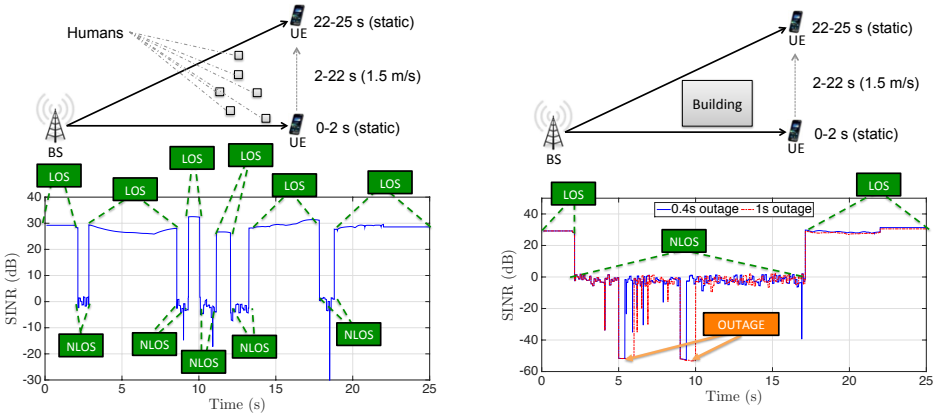
2.5 Industrial Control Networks

Industrial Control Networks (ICNs) are networks used in industrial applications. The application areas are very diverse, including manufacturing systems, energy and power systems, automotive communications, avionics and aerospace, railway control systems and many more. What these systems share in common is the need for fast reliable machine to machine communication. For example, avionics might require communication between heading sensors and navigation systems while a manufacturing plant could need synchronization between motor actuators and photosensitive sensors. Historically, the majority of systems have been fieldbus systems. Fieldbus system can have varied characteristics, but commonly refer to a data bus which provides communication between industrial controllers and instrumentation devices [32]. While fieldbus systems have historically dominated ICN, nowadays many industrial networks use Industrial Ethernet, and the dominance of Ethernet in other sectors (such as home users and data centers) further increases its prevalence.

An ICN typically consists of sensor nodes, which observe data and transmit it through the network (e.g. temperature sensors); actuator nodes, which can receive messages and perform one or more operations (e.g. motors); and some form of industrial controller which receives messages from sensors and sends responses to the actuators depending on the sensor values received. Depending on the application, there may be high demands on the network communication speed and reliability. Observed sensor values must reach the industrial controller, and messages from the industrial controller must reach the actuators within certain predefined deadlines. This means that long delays and consecutive packet losses should not occur, something that must be taken into account when designing and deploying an ICN [33, 5].

Many fieldbus systems have already taken these constraints into account during the design process, which makes them very suitable for industrial applications. However, limitations (mainly in interoperability and diagnostics) of fieldbus systems have made Ethernet more attractive. One of the biggest challenges with classical 10 Mbps Ethernet is its usage of Carrier Sense, Multiple Access, Collision Detection (CSMA/CD). CSMA/CD Ethernet monitors the link for collisions, and if a collision is detected, both the sender and receiver will backoff and retransmit [13]. Using this method, however, means that a collision could slow down message arrival time significantly. Later technologies such as Time Division Multiplexing (TDM) and Collision Avoidance (CSMA/CA) made Ethernet a more desirable technology in industrial applications and led to the development and adoption of Industrial Ethernet [5].

These advancements in wired networking technology, however, do not completely prevent packet losses or delays. Packets may still be lost due to problems such as *faulty hardware* and/or *human errors*. To further increase the resiliency of an industrial network, technologies such as SDN could be used to provide both centralized network repair through traffic re-routing, and fast local link repair. SDN can also provide parallel path operation through packet duplication. Paper II investigates how some of these resiliency improvements,



(a) Examples of NLOS state changes.

(b) Examples of outage state changes.

Figure 4: Examples of state changes in mmWave transmissions. [34]

enabled by SDN, can improve the overall robustness of an industrial network.

2.6 mmWave Wireless Backhaul Networks

With the emergence of the 5th Generation (5G) mobile networks, it is expected that there will be a significant increase in the bandwidth requirements of customers. Cisco has predicted [35], that in 2021, there will be over 11 billion mobile devices connected, generating in total 49 EB of traffic. The 4th Generation (4G) Long Term Evolution (LTE), which is the currently deployed solution for high data rate mobile networking, have seen significant capital and academic investment in developing new technologies to provide even higher throughput. These improvements include the deployment of Massive Multiple-Input and Multiple-Output (MIMO), the introduction of Co-ordinated Multipoint (CoMP), improved coding and modulation schemes [36]. However, despite these improvements, traditional frequency bands encompass a small, heavily used spectrum and cannot provide the bandwidth required for such high-speed transmissions.

In order to cope with the ever increasing bandwidth demands, one considered solution for 5G mobile networks is to utilize parts of the Extremely High Frequency (EHF) spectrum, such as the mmWave bands (typically between 30 GHz and 300 GHz) [37, 38, 39, 40]. These were traditionally not considered due to their short-range and non-line-of-sight coverage issues, where even rain and thick foliage may incur propagation losses. In order to deal with these difficulties, technologies such as directional beamforming antenna, based on antenna arrays, are often deployed. These antenna arrays can circumvent some blockage [41]. However, the prevalence of blockage in urban environments may still cause intermittent outages, and thus fluctuation in available throughput. According to the model presented in [42], the link quality moves between three states: line-of-sight (LOS), non-line-of-sight (NLOS) and outage (see Figure 4). One of the most important challenges in the widespread deployment

of backhaul networks using mmWave bands is coping with the fast channel attenuation.

The usage of SDN to configure backhaul networks has been suggested [43, 44], as current backhaul technologies typically are of a proprietary nature. SDN could open up the backhaul networks for fast innovation and deployment of new networking technologies, while also simplifying management through increased flexibility and centralized control. However, in mmWave backhaul networks, the intermittent connection failures of the mmWave links may happen rapidly and often. Thus, a reactive link recovery approach may be too slow. Paper III, investigates utilizing BFD and fast failover to provide local link repair in mmWave backhaul networks, and evaluates their effectiveness in coping with the rapid changes in link quality.

3 Research Objectives

One of the fundamental issues with networking today, is how to provide reliable transmission of data. Loss of connectivity, even for a brief moment, can severely impact the QoE of the user. In VoIP, connection loss may result in inferior call quality, or a dropped call. For an avid or professional gamer, dropped packets can cause delays which alter the outcome of tournament games, and for machine-to-machine communication such as in industrial factories, packet loss in a sensor-actuator loop can have severe consequences.

SDN increases the flexibility of computer networks by opening up and centralizing the control plane, which enables faster innovation and deployment of new network architectures. This centralization allows the network devices to notify the controller of detected link failures, and the SDN controller can subsequently re-calculate new routing information for each device such that the link failure is mitigated.

However, the reaction time of a controller-based link repair may be too long for networks with high availability requirements, or in networks where the link liveness changes frequently. Fortunately, recent extensions of the OpenFlow protocol allow the network devices to perform local failure recovery through the use of alternate paths. The OpenFlow protocol also allows duplicating packets and spreading them over multiple disjoint paths. Furthermore, the `ovsdb` protocol can, in conjunction with OpenFlow, be used to configure the traffic control algorithms and their packet classification rules.

This work investigates using SDN-based resiliency and traffic control in networks where high availability, even during unfavourable channel conditions, is required. We evaluate different configuration parameters and their effects on packet loss rate, delay, throughput, overhead etc.

3.1 Research Questions

1. *Can SDN, combined with fast failover and BFD, be used to meet stringent packet delivery ratio requirements while also coping with rapidly varying*

link quality?

Computer networks which require high availability, such as ICNs, and networks which suffer from intermittent connection outages, such as mmWave backhaul networks, could benefit from the flexibility and reliability which SDN provides. The centralized controller can monitor link liveness and re-route in case a link fails. However, the reaction time can be too long for applications which require high availability. With the recent introduction of local link repair through fast failover in OpenFlow 1.1, the reaction time can be significantly reduced. Paper II, investigates the effect of fast failover and BFD on ICN, and shows that using BFD and packet duplication techniques reduces the packet loss rate, increasing network resiliency. Paper III, shows the usage of similar techniques in 5G mmWave backhaul networks, which are shown to reduce packet loss rate and increase throughput.

2. *How can the latency of IEEE 802.11 WLAN management frames be reduced using traffic control in networks which separate the processing of WLAN MAC frames into two separate entities?*

   In CloudMAC-like networks, due to the timing requirements of WLAN transmission, congestion may prevent wireless stations from connecting. Using state of the art AQM, and SDN-based packet prioritization and traffic control configuration, WLAN management frames may be prioritized over data frames, providing rapid delivery of frames required for connection. Paper I shows that utilizing OpenFlow 1.3 QoS functionality, in conjunction with the `ovsdb` switch configuration protocol, the network can be dynamically configured such that WLAN management frames are prioritized over data frames, enabling successful connections even in heavily congested networks.

4 Research Methods

Computer Science can essentially be divided into three different categories: hardware, which examines the physical structure of computational devices, software which examines the software which is executed on the hardware, and theoretical computer science, which examines the capabilities of computational methods, through analyzing and designing algorithms and data structures [45]. In order to answer the research objectives laid out in Section 3, the main focus of this thesis will be on investigating software which improves the reliability of computer networks. To answer these questions, we utilize the scientific method. The scientific method is an iterative process of scientific inquiry which typically goes through these stages:

Hypothesize Formulate a hypothesis which can explain an observation. The resulting hypothesis must be falsifiable, otherwise it cannot be tested. For example: In the early stages of Paper I, we crafted a hypothesis which

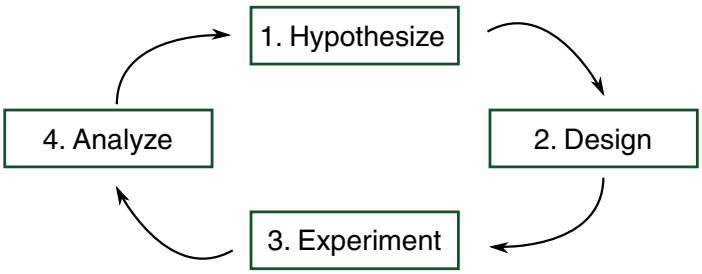


Figure 5: The iterative process of the scientific method

stated that the cross-traffic in CloudMAC negatively impacts connection establishment.

Design Create an experiment which can test your hypothesis. In Paper I, the hypothesis was tested by simulating cross-traffic and measuring connection success rate.

Experiment Perform the experiment. Repeating the experiment multiple times help improve our confidence in the result. The number of repetitions depends on the variability of the experiment, and the requirements of the analysis.

Analysis Analyze the results of the experiment(s). If they do not match your hypothesis, revise or reformulate your hypothesis or formulate a new one. In Paper I, the results showed that the connection success rate was impacted. We observed that the management frame delay determined if the connection was a success, in conjunction with which wireless hardware that was used. This resulted in the formulation of a new hypothesis, that using traffic control in the CloudMAC network would improve the success rate, and the process was re-started.

How these stages relate to each other can be seen in Figure 5. Following this process will strengthen the hypothesis over time [45, 46].

Hypothesis formulation is typically done through literature review. By reading and understanding the current state of the art, specific problem areas can be identified. For example: where a model might be improved, an experiment repeated for additional validation or a new protocol might solve a previously unsolved problem.

Experimentation in computer networking typically has three main methods: *simulation*, *experimentation* and *emulation*. Simulations often use a software which contains algorithmic and mathematical models for how a network behaves. These models can be extended with any additional models that may be required for the hypothesis to be verified. Simulated computer networks generally come with high accuracy, as long as the model used to simulate the network is realistic and comprehensive. However, some parts of computer networking can be especially difficult to simulate, such as wireless networking due to complicated propagation environment. Experimentation, on the other

hand, uses real-world hardware, software and configurations to perform the experiment. This can provide very accurate results. However, experimentation typically comes at the cost of obtaining equipment, which may require significant capital investment. For certain experiments, there may not even be equipment available, as it has not yet been created. Another weakness with experimentation is the increased amount of variability, which make repetition more difficult. For example, performing a wireless experiment may have interference caused by nearby people or the equipment used for a certain experiment may no longer be available. Finally we have emulation, which is a hybrid approach between the simulation and experimentation. In emulation, physical hardware is used for some part of the network, such as the host machines, while another part of the network is simulated.

While all of these methods have their respective advantages and disadvantages, it is important to note that for a comprehensive confirmation of a hypothesis, a combination of the approaches should be used. This can be done over the course of several studies. In this thesis, both emulation and experimentation are used to validate our hypothesises.

In Paper I, we performed the evaluation using an experiment involving physical hardware for the whole network. This included desktop PCs, wireless access points and off-the-shelf switches. To evaluate the performance of the network, we looked at the latency and throughput, the probe response time and the WLAN connection success rate. This was tested with and without cross-traffic, using various traffic control algorithms. We chose this method for validation because CloudMAC is designed to work with off-the-shelf wireless stations, and should continue to do so even after our improvements. In both Paper II and Paper III, an emulation approach was used, utilizing the CORE emulator. This emulator uses Linux containers and network namespaces in order to create a large number of virtual machines running on the same physical computer. These virtual machines are connected using virtual bridges, which can be configured using the traffic control system in Linux. Through this configuration, the emulator can emulate link characteristics such as bandwidth, delay, jitter and packet loss, while using real traffic and applications. In both experiments (Paper II and III), we extended the CORE emulator to support Open vSwitch [24], a popular software switch which support OpenFlow. We chose the method of emulation because performing physical experiments with large networks or new technologies can be quite challenging, and emulation gives us a near-realistic environment while allowing us to construct a cost- and time-effective testbed. In Paper II, as we are interested in the performance of local link repair using fast failover in ICN, we mainly focus on the packet loss rate and latency. However in Paper III, as we investigate the feasibility of using fast failover in future mmWave 5G networks, we are interested in more parameters, including: throughput, delay, BFD failover count and overhead.

5 Main Contributions

Throughout this work, we have investigated reliability issues of different types of computer networks, from industrial wired networks to wireless mmWave backhaul networks. In particular, we have attempted to find solutions to these reliability issues using SDN. First, we investigated using SDN-configured traffic control in the CloudMAC architecture as a way to improve the reliability of WLAN connection establishment. In doing so, we contributed to the inclusion of multiple AQMs, such as FQ-CoDel, in mainline Open vSwitch. Furthermore, we showed that both non-discriminatory (such as FQ-CoDel) and discriminatory (such as HTB) traffic control strategies can be used to improve the connection reliability in CloudMAC. These contributions were presented in Paper I. Further extensions, including QoS improvements using IEEE 802.11ae access categories and Grafana and InfluxDB integration in the controller were presented in [47].

We also evaluated fast failover and BFD in two types of networks which both require high resiliency. First, we configured a testbed to emulate an ICN, using the CORE network emulator. Next, we used the testbed to show that using packet duplication and fast failover may be a potential solution to resiliency issues in future SDN-enabled ICN. This was presented in Paper II. Using the same emulator, we configured a 5G mmWave backhaul scenario where we evaluated utilizing fast failover to cope with the rapid channel quality changes, which are typical of mmWave networks. The idea was presented as a poster in [48], expanded using improved accuracy and LTE assist mode in [49]. Further, it was expanded including TCP and UDP throughput, overhead costs and failover count metrics in Paper III.

6 Summary of Appended Papers

This section consists of a summary of the appended papers and their main contributions.

Paper I - QoS Enabled WiFi MAC Layer Processing as an Example of a NFV Service

CloudMAC [31] splits the WLAN MAC layer processing, moving the management and data frame processing to a Virtual Access Point (VAP) in the cloud. The WLAN APs are simplified into Wireless Termination Points (WTPs), which serve as simple radio heads, transmitting the traffic they receive from the VAP to the wireless medium, and forwarding all received wireless traffic to the VAP. Between the WTPs and the VAPs, a SDN enabled network connects the WTPs to the VAPs. The SDN controllers in this network continuously monitor the signal strength of each station connected to the CloudMAC network, updating forwarding rules between the WTPs and VAPs when the signal strength weakens. As the connection state is kept in the VAP, this handover is seamless.

In Paper I, we investigated the impact of cross-traffic on wireless management frames in the CloudMAC network, in particular on the probe response/request frames which are essential to a successful association attempt. This is caused by Wireless Network Interface Cards (WNICs) not connecting to wireless networks which do not respond to probe requests within a certain deadline. Different manufacturers have different delay tolerances for probe responses, and in a network which could potentially increase the delay of management frames, investigation showed that in a heavily congested network, the increased probe response delay did indeed increase the prevalence of connection failures. As a solution to the issue, we proposed reducing the delay in a congested network using various of the traffic control strategies available in Linux-based switches and routers. We extended Open vSwitch to include support for multiple additional traffic control strategies, such as SFQ [27], Controlled Delay (CoDel) [50], FlowQueue-Codel (FQ-CoDel) [51] [1], and made this available through the SDN controller, using the `ovsdb` [29] interface.

In order to verify that the added traffic control algorithms would improve the connection success rate, we performed an evaluation using the above-mentioned algorithms, and also the HTB [26] traffic control algorithm (already implemented in Open vSwitch). The evaluation was performed by loading the SDN-based network with cross-traffic and attempting to connect with a wireless station using CloudMAC. With the results from this paper, we increased the connection success rate in CloudMAC using traffic control, causing an overall increase in the network reliability under high load (Connection success went from 50% to almost 100% with for example a Broadcom WNIC). Furthermore, we contributed a substantial patch to the available traffic control algorithms in Open vSwitch.

Paper II - Resilient Software Defined Networking for Industrial Control Networks

Transmission reliability and fast data delivery are important aspects of ICNs, where consecutive loss of data could have severe consequences for a particular industrial process. Providing latency and packet delivery rate guarantees is difficult in Ethernet, especially when repair times have to be below 10 ms.

In Paper II, we proposed using SDN to improve the resiliency of Industrial Sensor Actuator Networks (ISANs). Due to the centralized view of the network enabled through a SDN-based architecture, link failures can be repaired as soon as the controller receives information on the link status. However, while the link detection in SDN-enabled switches such as Open vSwitch is typically slow, utilizing the new features in OpenFlow 1.1, such as fast failover and group tables, the reaction time can be greatly reduced, to as low as (3.3 ± 0.8) ms [10]. Another method which can reduce the Packet Loss Rate (PLR), is the duplication of packets at the sensor network gateways. This packet duplication, while effectively halving the available link throughput, prevents the loss of packets

[1] The changes are available in Open vSwitch mainline since 2016, at commit hash: 677d9158fc0aa16f875198d83c7bd8f87238aed5

in the interval between the failures and the failure detection.

An emulation-based testbed using the CORE emulator was set up to evaluate the impact of using fast failover in ICNs. The testbed was configured with both the wireless and the wired part, both experiencing packet loss. Furthermore, the wired portion of the network experienced link failures throughout the test. Our experiments show the packet loss is in the emulated network using different BFD message intervals. The PLRs are high due to the configured PLR in the emulated wireless portion of the sensor network. As we can see in the table, a lower BFD interval typically gives a lower PLR. Furthermore, increasing the amount of gateways and the amount of copies in the network, significantly decreases the PLR. This indicates that using SDN with fast failover and packet duplication could increase the reliability of a ICN.

Paper III - Low Frequency Assist for mmWave Backhaul - The case for SDN resiliency mechanisms

In order to cope with the increased demand for mobile network bandwidth, one approach considered for the 5G standard is utilizing the 60 GHz frequency bands, also called the mmWave bands. However, these high-frequency bands suffer from high levels of signal attenuation, shadowing and blockage. These frequency bands typically require near-LOS conditions in order to transmit data at throughput with acceptable packet loss rates. To cope with the path loss, mmWave transmitters employ antenna arrays and beamforming techniques to help improve signal strength. Still, despite these improvements, intermittent objects may cause signals to be blocked, especially at high sender-receiver distances.

Traditionally mmWave backhaul networks are closed and proprietary, hindering fast innovation and deployment of new networking technologies. In Paper III, in order to open the backhaul network, we propose using a SDN-based approach for configuring mmWave backhaul links. One of the most popular SDN protocols, OpenFlow, enables support for fast failover, which can be used to cope with intermittent objects in the transmission path. Fast failover allows configuration of backup paths in the networks, which can be used should the primary path be blocked. When BFD is used, monitoring packets are continuously sent over each link, reporting the link status. As the monitoring packets keep being sent, the link quality is continuously reassessed, so that the node can choose whether to transmit through the primary path or through one of its configured backup paths.

An experiment testbed was setup using the CORE emulator, including the extensions used in Paper II. The emulation was further extended by implementing a three state model suggested from measurements in [42]. This model changes the quality of the link by randomly switching between three states: LOS, NLOS, and outage. Aspects such as PLR, delay, throughput, overhead and failover count were also measured. From the results in the paper we can conclude that a low BFD interval can improve the packet loss of a mmWave backhaul network.

7 Conclusions and Outlook

Within this work, the possibility of using SDN-based resiliency and traffic control mechanisms to reduce the impact of link failures and cross-traffic has been investigated. Support for AQM QoS algorithms was introduced into Open vSwitch, and utilized to improve the connection establishment in CloudMAC. The usage of fast failover, combined with BFD, to improve the packet delivery in both ICN and in 5G mmWave small cell backhaul networks was investigated. In combination, the results obtained may strengthen the image of SDN as a provider of resilient, high-availability networks.

While the SDN protocol OpenFlow opens up the control plane of networking devices, increasing flexibility and manageability, the data plane remains closed and relatively fixed. Fortunately, recent developments in programmable data planes, such as P4 [52], seeks to solve that problem. Thus, for the future, we seek to expand our investigation of SDN resiliency into the data plane.

Furthermore, some unresolved issues with the CloudMAC architecture remain. These include evaluating how CloudMAC copes in a multi-tenant cloud, where a large amount of virtual machines may introduce delays due to context switching. Also, programmable data planes, like P4, could allow us to significantly increase the handover speed, as the decision could be taken locally, in the switch.

References

[1] A. P. Guimarães et al. “Availability analysis of redundant computer networks: A strategy based on reliability importance”. In: *2011 IEEE 3rd International Conference on Communication Software and Networks* (May 2011), pp. 328–332. DOI: 10.1109/ICCSN.2011.6014733.

[2] D. Rodrigues, E. Cerqueira, and E. Monteiro. “QoE Assessment of VoIP in Next Generation Networks”. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2009, pp. 94–105. DOI: 10.1007/978-3-642-04994-1_8. URL: https://doi.org/10.1007%2F978-3-642-04994-1_8.

[3] T.-k. Chua and D. Pheanis. “QoS evaluation of sender-based loss-recovery techniques for VoIP”. In: *IEEE Network* 20.6 (Nov. 2006), pp. 14–22. DOI: 10.1109/mnet.2006.273116. URL: https://doi.org/10.1109/mnet.2006.273116.

[4] K.-T. Chen, P. Huang, and C.-L. Lei. “How sensitive are online gamers to network quality?” In: *Communications of the ACM* 49.11 (Nov. 2006), p. 34. DOI: 10.1145/1167838.1167859. URL: https://doi.org/10.1145/1167838.1167859.

[5] R. Zurawski. *Industrial Communication Technology Handbook, Second Edition*. Industrial Information Technology. CRC Press, 2014. ISBN: 9781482207330. (*Mainly chapter 16: Switched Ethernet in Automation*).

[6] Open Networking Foundation. *Software-Defined Networking: The New Norm for Networks*. Tech. rep. Apr. 2012.

[7] S. Sharma et al. "OpenFlow: Meeting carrier-grade recovery requirements". In: *Computer Communications* 36.6 (2013). Reliable Network-based Services, pp. 656–665. ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2012.09.011.

[8] A. Sgambelluri et al. "OpenFlow-Based Segment Protection in Ethernet Networks". In: *Journal of Optical Communications and Networking* 5.9 (Aug. 2013), p. 1066. DOI: 10.1364/jocn.5.001066. URL: https://doi.org/10.1364/jocn.5.001066.

[9] N. McKeown et al. *OpenFlow: Enabling Innovation in Campus Networks*. Tech. rep. 2008.

[10] N. L. M. van Adrichem, B. J. Van Asten, and F. A. Kuipers. "Fast Recovery in Software-Defined Networks". In: *Software Defined Networks (EWSDN), 2014 Third European Workshop on* (Sept. 2014), pp. 61–66. DOI: 10.1109/EWSDN.2014.13.

[11] M. S. Seddiki et al. "FlowQoS : QoS for the Rest of Us". In: *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*. HotSDN '14 (2014), pp. 207–208. DOI: 10.1145/2620728.2620766. URL: http://doi.acm.org/10.1145/2620728.2620766.

[12] D. Katz and D. Ward. *Bidirectional Forwarding Detection (BFD)*. RFC 5880 (Proposed Standard). June 2010. URL: http://www.ietf.org/rfc/rfc5880.txt.

[13] P. L. Dordal. *An Introduction to Computer Networks*. Department of Computer Science Loyola University Chicago, 2017.

[14] D. Levi and H. D. *Definitions of Managed Objects for Bridges with Rapid Spanning Tree Protocol*. RFC 4318 (Informational). Dec. 2005. URL: http://www.ietf.org/rfc/rfc4318.txt.

[15] A. Giorgetti et al. "Performance Analysis of Media Redundancy Protocol (MRP)". In: *IEEE Transactions on Industrial Informatics* 9.1 (Feb. 2013), pp. 218–227. ISSN: 1551-3203. DOI: 10.1109/TII.2012.2186584.

[16] H. Kirrmann, H. M, and P. Muri. "prp: Bumpless recovery for highly available, hard real-time industrial networks". In: *Emerging Technologies and Factor Automation* (Sept. 2007).

[17] M. Rentschler and H. Heine. "The Parallel Redundancy Protocol for industrial IP networks". In: *Industrial Technology (ICIT), 2013 IEEE International Conference on* (Feb. 2013), pp. 1404–1409. DOI: 10.1109/ICIT.2013.6505877.

[18] International Electrotechnical Commission (Ginebra). *IEC 62439-3: Industrial Communication Networks : High Availability Automation Networks. Parallel Redundancy Protocol (PRP) and High availability Seamless Redundancy (HSR)*. 3. IEC, 2010.

[19] A. Akella and R. Mahajan. “A Call to Arms for Management Plane Analytics”. In: *Proceedings of the 13th ACM Workshop on Hot Topics in Networks - HotNets-XIII*. ACM Press, 2014. DOI: 10.1145/2670518.2673883. URL: https://doi.org/10.1145/2670518.2673883.

[20] C. Sieber et al. “Towards a programmable management plane for SDN and legacy networks”. In: *2016 IEEE NetSoft Conference and Workshops (NetSoft)*. IEEE, June 2016. DOI: 10.1109/netsoft.2016.7502428. URL: https://doi.org/10.1109/netsoft.2016.7502428.

[21] D. Kreutz et al. “Software-Defined Networking: A Comprehensive Survey”. In: *arXiv preprint arXiv:1406.0440* (2014), pp. 1–62. ISSN: 0018-9219. DOI: 10.1109/JPROC.2014.2371999. eprint: 1406.0440. URL: http://arxiv.org/abs/1406.0440.

[22] G. Bianchi et al. “OpenState: Programming Platform-independent Stateful OpenFlow Applications Inside the Switch”. In: *ACM SIGCOMM Computer Communication Review* 44.2 (2014), pp. 44–51. ISSN: 01464833. DOI: 10.1145/2602204.2602211. URL: http://dl.acm.org/citation.cfm?doid=2602204.2602211.

[23] *PICOS®The First Two-in-One Open Network Operating System (NOS) Coupling Full Enterprise Support with “Classic” SDN*. Pica8. Mar. 2018.

[24] B. Pfaff et al. “The Design and Implementation of Open vSwitch”. In: *12th USENIX Symposium of Networked Systems Design and Implementation* (2015).

[25] Open Networking Foundation. *OpenFlow Switch Specication: Version 1.1.0 Implemented (Wire Protocol 0x02)*. Feb. 2011. URL: http://archive.openflow.org/documents/openflow-spec-v1.1.0.pdf.

[26] J. L. Valenzuela et al. “A hierarchical token bucket algorithm to enhance QoS in IEEE 802.11: proposal, implementation and evaluation”. In: *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th* 4 (2004), 2659–2662 Vol. 4. ISSN: 1090-3038.

[27] P. E. McKenney. “Stochastic fairness queueing”. In: *INFOCOM ’90, Ninth Annual Joint Conference of the IEEE Computer and Communication Societies. The Multiple Facets of Integration. Proceedings, IEEE* (June 1990), 733–740 vol.2. DOI: 10.1109/INFCOM.1990.91316.

[28] O. N. Foundation. *OpenFlow Switch Specification Version 1.3.3 (Protocol version 0x04)*. URL: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.3.pdf.

[29] B. Pfaff and B. Davie. *The Open vSwitch Database Management Protocol*. Internet Requests for Comments. RFC. Dec. 2013. URL: http://www.ietf.org/rfc/rfc7047.txt.

[30] J. F. Kurose and K. W. Ross. *Computer Networking: A Top-Down Approach*. 6th. Pearson, 2012. ISBN: 978-0-13-285620-1.

[31] P. Dely et al. “CloudMAC - An OpenFlow based architecture for 802.11 MAC layer processing in the cloud”. In: *Globecom Workshops (GC Wkshps), 2012 IEEE* (2012), pp. 186–191.

[32] International Electrotechnical Commission. *IEC 61158-1: Digital data communications for measurement and control - Fieldbus for use in industrial control systems, Part 1: Introduction*. 1. IEC, 2003.

[33] J. Akerberg, M. Gidlund, and M. Bjorkman. “Future research challenges in wireless sensor and actuator networks targeting industrial automation”. In: *Industrial Informatics (INDIN), 2011 9th IEEE International Conference on* (July 2011), pp. 410–415. DOI: 10.1109/INDIN.2011.6034912.

[34] M. Zhang et al. In: *CoRR* abs/1603.0 (2016). URL: http://arxiv.org/abs/1603.02701.

[35] Cisco. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update*. Tech. rep. 2017. URL: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html.

[36] J. Wannstrom. *LTE-Advanced*. URL: http://www.3gpp.org/technologies/keywords-acronyms/97-lte-advanced.

[37] W. Roh et al. “Millimeter-wave beamforming as an enabling technology for 5G cellular communications: theoretical feasibility and prototype results”. In: *IEEE Communications Magazine* 52.2 (Feb. 2014), pp. 106–113. ISSN: 0163-6804. DOI: 10.1109/MCOM.2014.6736750.

[38] Huawei. *5G Spectrum Public Policy Position*. URL: http://www-file.huawei.com/-/media/CORPORATE/PDF/public-policy/public_policy_position_5g_spectrum.pdf.

[39] Ericsson. *SoftBank and Ericsson to demonstrate 5G 28GHz*. URL: https://www.ericsson.com/en/press-releases/2017/3/softbank-and-ericsson-to-demonstrate-5g-28ghz.

[40] NTT docomo. *Sēkai shuyō benda to no 5G jikken no gaiyō*. URL: https://www.nttdocomo.co.jp/corporate/technology/rd/tech/5g/5g_trial/.

[41] H. Xu, V. Kukshya, and T. S. Rappaport. “Spatial and temporal characteristics of 60-GHz indoor channels”. In: *IEEE Journal on Selected Areas in Communications* 20.3 (Apr. 2002), pp. 620–630. ISSN: 0733-8716. DOI: 10.1109/49.995521.

[42] M. R. Akdeniz et al. “Millimeter Wave Channel Modeling and Cellular Capacity Evaluation”. In: *IEEE Journal on Selected Areas in Communications* 32.6 (June 2014), pp. 1164–1179. ISSN: 0733-8716. DOI: 10.1109/JSAC.2014.2328154.

[43] K. Seppänen, J. Kilpi, and T. Suihko. "Integrating WMN Based Mobile Backhaul with SDN Control". In: *Mob. Netw. Appl.* 20.1 (Feb. 2015), pp. 32–39. ISSN: 1383-469X. DOI: 10.1007/s11036-015-0574-7. URL: http://dx.doi.org/10.1007/s11036-015-0574-7.

[44] J. Núñez-Martínez, J. Baranda, and J. Mangues-Bafalluy. "A service-based model for the hybrid software defined wireless mesh backhaul of small cells". In: *Network and Service Management (CNSM), 2015 11th International Conference on* (Nov. 2015), pp. 390–393. DOI: 10.1109/CNSM.2015.7367388.

[45] D. Reed. *A Balanced Introduction to Computer Science*. 3rd. Pearson, 2012.

[46] G. Dodig-Crnkovic. "Scientific methods in computer science". In: *Proceedings of the Conference for the Promotion of Research in IT at New Universities and at University Colleges* (2002), pp. 126–130.

[47] J. Vestin and A. Kassler. "QoS Management for WiFi MAC Layer Processing in the Cloud: Demo Description". In: *Proceedings of the 11th ACM Symposium on QoS and Security for Wireless and Mobile Networks*. Q2SWinet '15 (2015), pp. 173–174. DOI: 10.1145/2815317.2815343. URL: http://doi.acm.org/10.1145/2815317.2815343.

[48] J. Vestin and A. Kassler. "Resilient SDN based small cell backhaul networks using mmWave bands". In: *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)* (June 2016), pp. 1–3. DOI: 10.1109/WoWMoM.2016.7523543.

[49] J. Vestin and A. Kassler. "SDN enabled resiliency in LTE assisted small cell mmWave backhaul networks". In: *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)* (Mar. 2017), pp. 199–202. DOI: 10.1109/ICIN.2017.7899411.

[50] K. Nichols and V. Jacobson. *Controlling Queue Delay*. Tech. rep. 2012.

[51] T. Hoeiland-Joergensen et al. *FlowQueue-Codel*. Mar. 2014. URL: https://tools.ietf.org/html/draft-hoeiland-joergensen-aqm-fq-codel-00.

[52] P. Bosshart et al. "P4: Programming Protocol-independent Packet Processors". In: *SIGCOMM Comput. Commun. Rev.* 44.3 (July 2014), pp. 87–95. ISSN: 0146-4833. DOI: 10.1145/2656877.2656890. URL: http://doi.acm.org/10.1145/2656877.2656890.

SDN-Enabled Resiliency in Computer Networks

In computer networking, failures, such as breaking equipment, cable cuts, power failures and human errors continuously cause communication interruptions. Such failures may result in dissatisfied customers, loss of product reputation, violation of SLAs and even critical failures in industrial systems. SDN, which logically centralizes the control plane, is an emerging technology in computer networking. The global view provided by the SDN controller can be used to reconfigure the network in case of a link failure. However, this reconfiguration may take too long for high availability networks. With the introduction of proactive link repair, backup paths are preinstalled in the forwarding devices, reducing path recovery time.

This thesis addresses the usage of SDN to provide resiliency in high availability networks. First, we consider how SDN can be used for increasing the reliability of ICNs. Second, we investigate how similar technologies could be applied to deal with fast channel attenuation and resulting outage in mmWave backhaul networks. Finally, we look at CloudMAC-based Wireless LAN, and how SDN-enabled QoS improvements could improve connection reliability.

ISBN 978-91-7063-848-0 (tryck)

ISBN 978-91-7063-943-2 (pdf)

ISSN 1403-8099

LICENTIATE THESIS | Karlstad University Studies | 2018:17