

TOWARD A RELIABLE NETWORK MANAGEMENT FRAMEWORK

A DISSERTATION  
IN  
Computer Networking and Communications Systems  
and  
Economics

Presented to the Faculty of the University  
of Missouri–Kansas City in partial fulfillment of  
the requirements for the degree

DOCTOR OF PHILOSOPHY

by  
HAYMANOT GEBRE-AMLAK  
University of Missouri - Kansas City

Kansas City, Missouri  
2018

© 2018

HAYMANOT GEBRE-AMLAK

ALL RIGHTS RESERVED

# TOWARD A RELIABLE NETWORK MANAGEMENT FRAMEWORK

Haymanot Gebre-Amlak, Candidate for the Doctor of Philosophy Degree

University of Missouri–Kansas City, 2018

## ABSTRACT

As our modern life is very much dependent on the Internet, measurement and management of network reliability is critical. Understanding the health of a network via outage and failure analysis is especially essential to assess the reliability of a network, identify problem areas for network reliability improvement, and characterize the network behavior accurately. However, little has been known on characteristics of node outages and link failures in access networks. In this dissertation, we carry out an in-depth outage and failure analysis of a university campus network using a rich set of node outage and link failure data and topology information over multiple years. We investigated the diverse statistical characteristics of both wired and wireless networks using big data analytic tools for network management. Furthermore, we classify the different types of network failures and management issues and their strategic resolution.

While the recent adoption of Software-Defined Networking (SDN) and software-ization of network functions and controls ease network reliability, management, and various network-level service deployments, the task of monitoring network reliability is still very challenging. We find it challenging because it not only requires vast measuring and processing resources but also introduces an additional intermediate network, so-called a 'control-path network', that physically connects the control and data plane networks. We proposed a topology-aware network management framework that utilizes Link Layer Discovery Protocol (LLDP) messages via prudent control of the frequency of LLDP messages and considering tier-based network architecture. It provides fast and effective reliability information for faster recovery from failures. The topology-aware analysis also enables us to explore the economic impact and the cost of various types of network failures with regards to Capital Expenditure (CapEx) and Operational Expenditure (OpEx).

Wireless LAN (Local Area Network) or Wi-Fi has become the primary mode of network access for most users; thus, its performance measurement becomes a critical part of network management in access networks. Through large-scale, extensive analysis of a university campus Wi-Fi network, we found its performance behavior and management issues are very distinctive from a wired network. The study also informs a strategic Wi-Fi Access Point deployment and enhanced Wi-Fi association scheme for better coverage and enhanced user experience.

Most of the current and future networks would involve both wired and wireless

subnets. Our work of understanding the unique issues of each one and their interplay would shed light on managing and improving network reliability in a holistic manner.

## APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Graduate Studies, have examined a dissertation titled “Toward a Reliable Network Management Framework,” presented by Haymanot Gebre-Amlak, candidate for the Doctor of Philosophy degree, and hereby certify that in their opinion it is worthy of acceptance.

### Supervisory Committee

Baek-Young Choi, Ph.D. Committee Chair  
Department of Computer Science & Electrical Engineering

Mathew Forstater, Ph.D., (Co-chair)  
Department of Economics

Cory Beard, Ph.D.,  
Department of Computer Science & Electrical Engineering

Yugyung Lee, Ph.D.,  
Department of Computer Science & Electrical Engineering

Deep Medhi, Ph.D.,  
Department of Computer Science & Electrical Engineering

Sejun Song, Ph.D.,  
Department of Computer Science & Electrical Engineering

## CONTENTS

ABSTRACT . . . . .	iii
LIST OF ILLUSTRATIONS . . . . .	x
LIST OF TABLES . . . . .	xiv
ACKNOWLEDGEMENTS . . . . .	xiv
1 INTRODUCTION . . . . .	1
1.1 Reliability in Wired and Wireless Network . . . . .	1
1.2 Reliability in Wired Network . . . . .	4
1.3 The Cost of Network Reliability . . . . .	6
1.4 Objective of This Study . . . . .	7
1.5 Contribution . . . . .	7
1.6 Organization . . . . .	8
2 UNDERSTANDING THE RELIABILITY OF A UNIVERSITY CAMPUS NET- WORK WITH SPLUNK . . . . .	9
2.1 Motivation . . . . .	9
2.2 Related Work . . . . .	10
2.3 Node and Link Data Source . . . . .	12
2.4 Wired vs. Wireless Networks in Access Layer . . . . .	16
2.5 Characteristics of Layers . . . . .	20
2.6 Analysis of Outage . . . . .	22

2.7	Summary . . . . .	26
3	TOPOLOGY-AWARE RELIABILITY MANAGEMENT FRAMEWORK FOR SOFTWARED NETWORK SYSTEMS . . . . .	28
3.1	Background . . . . .	28
3.2	Motivational Experiments . . . . .	32
3.3	Topology-Aware Reliability Management . . . . .	34
3.4	Evaluations . . . . .	37
3.5	Related Work . . . . .	44
3.6	Summary . . . . .	46
4	PROTOCOL HETEROGENEITY ISSUES OF CAMPUS INCREMENTAL WI- FI UPGRADE . . . . .	48
4.1	Background . . . . .	48
4.2	Related Work . . . . .	50
4.3	Wi-Fi Deployment Design . . . . .	53
4.4	Evaluations . . . . .	60
4.5	Summary . . . . .	66
5	AGILE POLYMORPHIC SOFTWARE-DEFINED FOG COMPUTING PLAT- FORM FOR MOBILE WIRELESS CONTROLLERS AND SENSORS . . . . .	68
5.1	Background . . . . .	68
5.2	Related Work . . . . .	71
5.3	Light-Weight Controller in Software-Defined Network Design . . . . .	75
5.4	Evaluations . . . . .	82



5.5	Summary	88
6	COST OF UNPLANNED NETWORK OUTAGE AND SLA VERIFICATION	93
6.1	Background	93
6.2	Related Work	97
6.3	Cost of Network Failure	100
6.4	Evaluations	105
6.5	Summary	108
7	LESSON LEARNED	110
7.1	Big Data Analytic System Setup	110
7.2	Sensors Communication Challenges and Work-around	112
8	CONCLUSIONS AND FUTURE DIRECTIONS	116
8.1	Conclusions	116
8.2	Future Directions	117
	REFERENCE LIST	1
	VITA	15
8.3	Conference and Journal Publications	17
8.4	Posters Extended Abstracts	18
8.5	Talks	20
8.6	Instructor and Teaching Assistant	21

## LIST OF ILLUSTRATIONS

Figure		Page
1	System Failure to Restoration . . . . .	1
2	Terminologies of System Failure and Repair . . . . .	2
3	Sensor Network . . . . .	3
4	Fog Computing at the Edge Network . . . . .	5
5	Wired and Wireless Down-time: bin size of the x-axis is 5 Minutes . . . .	18
6	Correlation between Wired and Wireless Node Outages . . . . .	18
7	Distribution for Time-to-Repair of Each Layer . . . . .	21
8	Node Outage with Splunk . . . . .	24
9	Link Outage with Splunk . . . . .	25
10	Centralized Reliability Management in SDN . . . . .	29
11	Triptych High Availability (HA) Domains . . . . .	31
12	Topology of LLDP-Discovery and LLDP-Speaker . . . . .	32
13	Control Message Analysis Experimental Setup . . . . .	32
14	Control Messages from the Controller I/O . . . . .	33
15	LLDP Messages Over the Daisy Chain Network . . . . .	33
16	Topology-Aware Reliability Management (TARMan) implementation . .	37
17	LLDP Messages Captured on a Controller . . . . .	40
18	Accumulated LLDP Messages Captured by the Switches . . . . .	40

19	Impact of Control Message Outage . . . . .	42
20	Impact of Data Flow Outage . . . . .	42
21	An Example of Coverage Map . . . . .	49
22	Network Traffic . . . . .	49
23	Reject New Association Due to Maximum Client Limit Reached Message Count . . . . .	57
24	Campus Network Architecture with Three Layers: Core, Distribution, and Access Layers . . . . .	61
25	Impact of Band Redirection . . . . .	62
26	802.11 total vs 802.11(ac) Throughput . . . . .	63
27	2.4 GHz vs 5 GHz Authentication . . . . .	63
28	Count of AP New Association Rejected between 2017 and 2018 . . . . .	64
29	Wifi Coverage in 2014 . . . . .	65
30	Wifi Coverage in 2016 . . . . .	65
31	Wifi Coverage in 2018 . . . . .	65
32	Authentication Failure Due to Roaming . . . . .	65
33	Usage Summary by Protocol . . . . .	65
34	Urban Surveillance Architecture . . . . .	69
35	A Role-based Polymorphic System . . . . .	76
36	A Unified Controller for SDM . . . . .	77
37	Control Mode Transitions and Co-Existence . . . . .	80
38	Types of Control Modes . . . . .	81

39	A libfluid Module Architecture . . . . .	82
40	SDM Implementation on Smart Mobile Devices . . . . .	83
41	The GoPiGo Prototype . . . . .	84
42	A Five Car Network . . . . .	86
43	Average Number of Active Sensors . . . . .	87
44	Power Usage . . . . .	91
45	Processing Time Overhead in Seconds . . . . .	92
46	Total Cost of Ownership (TCO) . . . . .	94
47	Capital Expenditure (CapEx) . . . . .	95
48	Network Outage Operational Expenditure (OpEx) . . . . .	95
49	Effect of Unplanned Outage . . . . .	101
50	Root Cause of Unplanned Data-center Outage in 2016 [114] . . . . .	103
51	GET request to Blackboard without response (AWS disruption) . . . . .	106
52	GET request to PANOPTO without response (AWS disruption) . . . . .	106
53	Syslog Network Outage Messages . . . . .	107
54	Network Outage from 15 Minutes of Power Outage . . . . .	107
55	Unsupported WiFi Router Messages . . . . .	109
56	Python Modules and Config Files . . . . .	111
57	Python Credential for the Splunk Server . . . . .	113
58	Python Search Job . . . . .	114
59	Three GopiGos . . . . .	115
60	Access Points and Devices Modeling . . . . .	119

61	Bipartite Graph Modeling of the APs and Devices . . . . .	119
62	Flow Network Modeling of Wi-Fi Association Problem . . . . .	119

## LIST OF TABLES

Tables		Page
1	Node Outage and Link Failure Data . . . . .	13
2	Data Processed by Node Type . . . . .	16
3	Wired vs Wireless Outage Data . . . . .	17
4	Statistical Information of Node Outages . . . . .	22
5	LLDP Default Settings in ODL . . . . .	34
6	LLDP Message Count in One Minute . . . . .	40
7	LLDP Message Count by Hop in One Minute . . . . .	41
8	Theoretical Impact Factor of Control Messages . . . . .	43
9	Experiment Result of Impact Factor of Control Messages . . . . .	43
10	Wireless Data Sets Used . . . . .	54
11	Data Source and Tools used for Analyzing Coverage, Authentication and Bandwidth Issues . . . . .	54
12	802.11n vs. 802.11ac Wireless Networking Protocols . . . . .	55
13	802.11a vs 802.11b vs 802.11g Wireless Networking Protocols . . . . .	55
14	802.11 (ac) Wireless Upgrade Summary . . . . .	59
15	Estimated Power Consumption . . . . .	85
16	Reliability Variables . . . . .	104

## ACKNOWLEDGEMENTS

First, I would like to thank my advisor, Dr. Baek-Young Choi, and co-discipline chair, Dr. Mathew Forstater for their great advice and guidance in my Ph.D. study. Dr. Choi has enabled me to develop various reliability researches for wired and wireless networks with insightful thoughts and helped me craft my dissertation topic, Toward Reliable Network Management Framework. She has enabled me to learn research skills including writing, discovering ideas, building up projects, and ultimately writing a paper for several conferences. Dr. Forstater has given me guidance in my Economics research and I am grateful for it. I would like to thank Dr. Sejun Song, who has worked tirelessly with me on various reliability in Software Defined Network, Software Defined Sensors, and Fog Computing researches. I would also like to thank Dr. Deep Medhi for all the valuable advising I received earlier in my Ph.D. studies which motivated me to work harder.

I would like to thank my committee members: Dr. Cory Beard and Dr. Yugyung Lee for having their doors open for me any time I have questions or advising. Also, thank you to my committee for their help in my dissertation and insightful comments. Thanks to these comments, this dissertation has improved drastically.

I also sincerely thank the campus network team for providing me with very valuable campus data and access to various network performance and traffic reports that help me conduct my network reliability analysis with the data.

I appreciate both my parents, Belaynesh Adinow and Colonel Debebe Gebre-Amlak. My father inspired me to continue my higher education and my mother who

has always been by my side to encourage me and help my family any time day or night which made her very close to my children. I am grateful to my husband's parents, Adela Regina Mongalo-Ortiz and Mario Mongalo. My wonderful children Alex, Bethany, and Eyassu for being my greatest strength and inspiration throughout my studies. I appreciate my sisters, Sefanit Adkins, who sent me encouraging words every week, Kassatinun Gebre-Amlak, who stayed up late nights proofreading my research work, Selemawit Gebre-Amlak, who kept the interesting brainstorming conversations, my brother Yonas Gebre-Amlak and his family, who called to check on me and my kids, and my big brother Yemane Gebre-Amlak and his family, whose kids have daily conversation with my kids regardless of the distance (grateful for technologies that make it possible!). I thank my brother Solomon who saw the seed of graduate school in my head before leaving town in 2010. He shared one of his energy-saving novel idea and his desire for prototyping. I hope we will get the chance to work together on the prototype. I thank my husband, Mario Mongalo, without whose undivided support, I could not achieve this amazing goal. I thank Jonathan and Daisy who has been an inspirational way before my children were born.

Most of all, I thank God, who always guide me throughout my life and help me appreciate each moment in life.



## CHAPTER 1

### INTRODUCTION

#### 1.1 Reliability in Wired and Wireless Network

The Internet is a massive interacted network of networks. It has the infrastructure which has become the global means for our day to day life. Our jobs depend on the internet. We learn from the Internet and connect and socialize using the Internet. Life without Internet is unimaginable. The old ways of saying, "The system will be down for 8 hours for the system upgrade. It is OK. Right?" or "We cannot have redundancy in everything - that is too expensive!" or "It takes 15 minutes to reboot the network, what is the big deal about that?" are NO longer acceptable. To keep the Internet up and running reliability is critical.

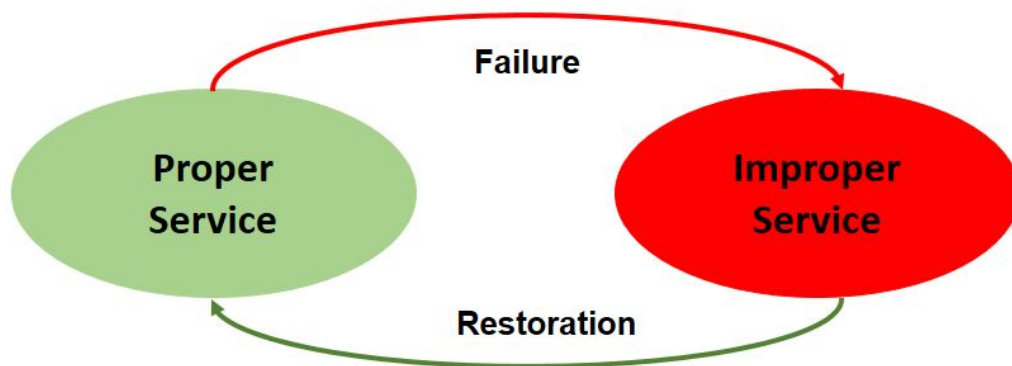


Figure 1: System Failure to Restoration

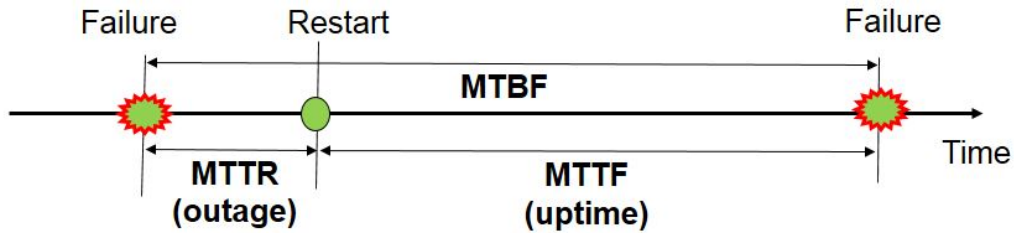


Figure 2: Terminologies of System Failure and Repair

### 1. Reliability

Network reliability is the capacity of the network to offer the same services even during a failure. System failure is a transition from proper service to improper service as illustrated by Figure 1. A single node or link failures account for the vast majority of network failures. Reliability can be calculated by dividing Mean Time To Failure (MTTF) by the sum of Mean Time to Repair (MTTR) and MTTF or dividing Mean Time Between Failure (MTBF) by the sum of MTTF and MTBF as shown in equation 1.1. The terminologies of MEER, MTTF and MTBF is illustrated by Figure 2.

$$Reliability = \frac{MTTF}{MTTF + MTTR} = \frac{MTBF}{MTTF + MTBF} \quad (1.1)$$

Understanding the health of a network via outage and failure analysis is important to assess the availability of a network, identify problem areas for network availability improvement, and model the exact network behavior. However, there has been little work on the statistical characteristics of node outages and link failures in access networks. In this study, we carry out an in-depth outage and failure analysis of a university campus

network using a rich set of node outage and link failure data and topology information. To expedite our analysis of the sheer amount of log data, we used one of the well-known big data analysis tool, Splunk. We investigated the statistical characteristics of various aspects of the wired and wireless network. We categorize the different types of failure and their strategic resolution. Furthermore, we discuss a through a measurement study that brings forth wireless network management issues faced during incremental Wi-Fi deployment on a university campus network.

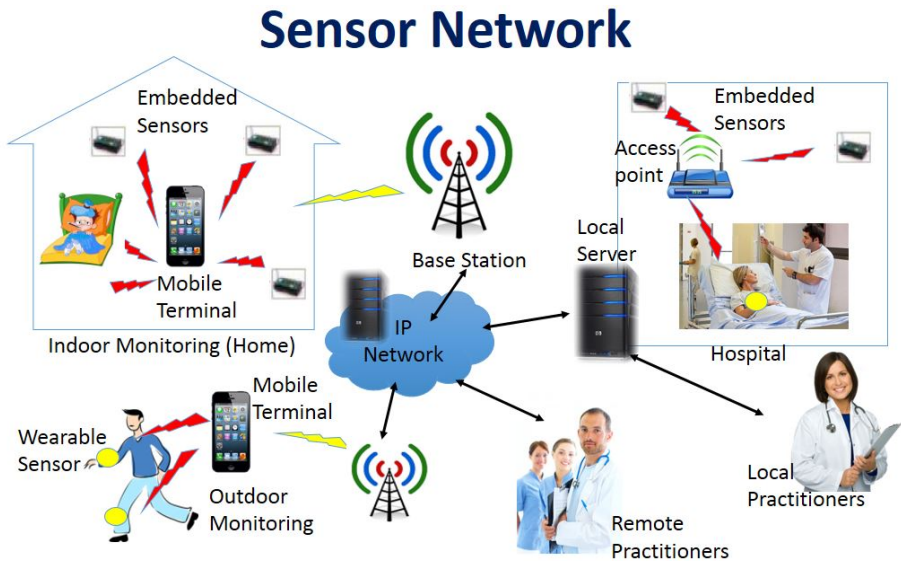


Figure 3: Sensor Network

A recent integration of wireless and mobile cyber-physical systems with the dramatically growing smart sensors enables a new type of pervasive smart sensor network as illustrated by 3. This opens up new opportunities for boosting the accuracy, efficiency, and productivity of uninterrupted wireless service. Wireless sensors provide the tool for

communications and security applications with low-power, multi-functioning, and computational capabilities. To improve the reliability of the wireless sensor we utilize Fog Computing. Fog Computing or edge computing, a recently proposed extension and complement for cloud computing, enables computing at the network edge in a smart device without outsourcing jobs to a remote cloud. Figure 4 illustrates Fog Computing at the edge network. It ensures effective data collection process and promotes efficient information abstraction which enables instant decision making by the end devices. We investigate an effective softwarization approach in a fog computing environment for dynamic big data-driven, real-time urban surveillance tasks of uninterrupted target tracking. we design and prototype an efficient and effective fog system using light-weight agile software-defined control for mobile wireless nodes. We address key technical challenges of node mobility to improve the system awareness. We have built a preliminary proof-of-concept Light-weight controller architecture on both Android-based and Linux-based smart devices and tested various collaborative scenarios among the mobile nodes.

## **1.2 Reliability in Wired Network**

Reliability is very important in the Software Defined Network (SDN) network. Introducing softwarization of network functions, controls, and applications that are promising; they optimize costs and processes while bringing new value to the infrastructures. However, the centralized reliability management in softwarization architecture poses both scalability and latency challenges. TARMan: Topology-Aware Reliability Management

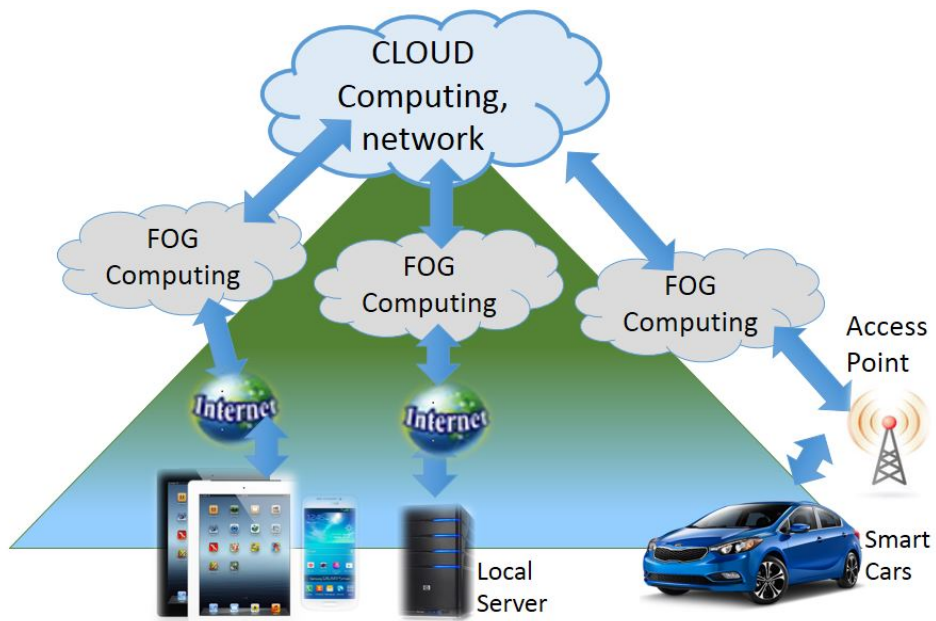


Figure 4: Fog Computing at the Edge Network

for Softwarized Network Systems is a model which design and build a novel topology-aware network reliability management framework that enhances the efficiency of network discovery (Link Layer Discovery Protocol (LLDP)) mechanisms and introduces a three-tier-based algorithm that the controller utilizes to calculate the LLDP-discovery frequency. Based on configurable failure impact, the TARMan platform provides fast and smart decision making information for fast failure detection and recovery, improving MTBF of SDN network.

### **1.3 The Cost of Network Reliability**

The third objective of my research explores the cost of network reliability. Network devices and appliances are intricate to manage, requiring highly skilled personnel to (re)configure and (re)install the system. It also requires additional costs to add, remove or move devices from a network. These changes could have adverse impacts on other parts of the network leading to an unplanned outage. Network service providers use Service Level Agreement (SLA), a time base contract between a service provider and the end user that defines the level of service expected from the service provider. In most cases, these contracts are one-sided with service provider making a promise to end user (customer) the level of service it plans to provide. The customer has no say in the contract or have a way to ensure the service is within the SLA. Human error, natural disaster and equipment failure attributes to 36% of the unplanned network outage in a datacenter. In this study, we present a measurement study to bring forth the various unplanned network outage issues faced on a university campus network and their impact the SLAs.

## **1.4 Objective of This Study**

The objective of my Ph.D. study is to develop a reliable network framework. We start by investigating the network reliability characteristics using SNMP and Syslog data to understand the causes of failures and their impact. Our study shows that the general characteristics of the different layers are very distinct from each other and the wireless network is less reliable compared to the wired network and is directly affected by the performance of the wired network. We perform a measurement study to bring forth wireless network reliability issues faced during incremental wifi deployment on a university campus network. We discuss various design considerations given to incremental deployments of wifi 802.11 (ac) including replacing older wifi versions and addressing compatibility, data rate, coverage, and wifi performance concerns. In addition, we evaluate pre-and-post upgrade results using different network performance analysis tools. This study will shed light on heterogeneous large-scale wifi network management issues, as these will become applicable to the increasing prevalence of large metro area wireless networks.

## **1.5 Contribution**

We analyze the different types of network failures and their recovery strategies based on real-world experience. We propose and prototype of an efficient and effective fog system using light-weight agile software-defined control for mobile wireless nodes. We investigate an effective approach in the Fog environment for dynamic big data-driven, real-time urban surveillance tasks of uninterrupted target tracking. We address the challenges of SDN node mobility to improve the system awareness. We built a preliminary

proof-of-concept light-weight controller architecture on both Android-based and Linux-based smart devices and tested various collaborative scenarios among the mobile nodes. To improve reliability in SDN, we design and build a novel topology-aware network reliability management framework that enhances the efficiency of network discovery (Link Layer Discovery Protocol (LLDP)) mechanisms and introduces a three-tier-based algorithm that the controller utilizes to calculate the LLDP-discovery frequency. A novel impact based per target LLDP-discovery approach enables fast failure detection and recovery for the important targets. To shed light on the cost of reliability, we perform analysis of unplanned network outage using real-world use case. We investigate network failure caused by human error, weather-related and device failure. Our analysis shows that unplanned outages without precautions have adverse impact and takes time to recover (if recoverable) and cleanup.

## **1.6 Organization**

This study follows the order of components that we developed for the Reliable Network Management Framework. Chapter 2 gives our investigation work on reliability with wired and wireless network. Chapter 3 introduces reliability framework in wired, SDN Management research. After that, we elaborate on the component for the Mobile SDN for Fog framework. In chapter 4 and 5 we present reliability in wireless and software defined sensors. Chapter 6 discuss the cost of reliability. Chapter 7 discusses the lessons learned while working on the different area of research and chapter 8 concludes the study.



## CHAPTER 2

### UNDERSTANDING THE RELIABILITY OF A UNIVERSITY CAMPUS NETWORK WITH SPLUNK

#### **2.1 Motivation**

Outage or failure is one of the essential network performance metrics and directly relates to the availability of a network. An outage of network elements can degrade network availability and may eventually cause any service discontinuation. Understanding outage behaviors and conditions are important to determine the failure source for troubleshooting, to monitor customer Service Level Agreement (SLA) conformance, to assess the availability of system components as well as the network, and to identify weak areas for the network availability improvement.

Despite significant efforts made on the network performance issues such as loss, latency, and jitter [1, 2, 3, 4, 5], there has been little attention paid to outage. A few failure measurement and analysis studies [6, 7, 8, 9] have been on a backbone network, and there are little failure measurements and analysis on access networks such as enterprise or campus networks. The study of such networks is critical to provide insights onto potential end-to-end availability expectations.

In this chapter, we investigate the characteristics of a university campus network in various aspects such that the characteristics of different layers in hierarchical network

architecture and the characteristics of wired and wireless networks. We carry out an in-depth analysis with a rich set of university campus network data of both node outages and link failures that span over 9 years. The measurement was conducted at the University of Missouri, Kansas City (UMKC) using the methods of network management system-initiated polling, SNMP-based event notification, and Syslog event logging. Our campus network is designed hierarchically with core, distribution, and access layers and many link redundancies are added for reliability.

Our results indicate that the general characteristics of the different layers are very distinct from each other and the wireless network is less reliable compared to the wired network and is affected by the performance of the wired network.

The remainder of this chapter is organized as follows. Section 2.2 discusses related work on the outage or failure measurement and analysis. Section 2.3 provides information on data sets that we used for analysis. We discuss more details of our analysis through Sections 2.5, 2.4 and 2.6. Section 2.7 summarizes the chapter.

## **2.2 Related Work**

Many methodologies [10, 11, 12, 13, 14, 15, 16] have been proposed to help operators in many ways such as performance analysis, network usage analysis, troubleshooting or diagnosing errors by utilizing the Syslog messages. Their approaches mainly depend on analysis of Syslog data and analyze only the usage patterns of wireless networks. They provide no characteristics of network failures that is important to assess the availability of the network, determine failure source for trouble-shooting, and identify weak

areas for network availability improvement [17]. Another approach utilizing the Syslog data [18, 19, 20, 21] is extracting valuable or meaningful information from the data or clustering the Syslog messages into a smaller number of meaningful categories or events.

The most relevant works are [6, 7, 8, 9] which characterize link failures of IP backbone provider networks using routing or Syslog messages. They provide good insight onto the failure characteristics of IP backbone network and failure classifications based on probable causes and various statistics from link failure messages. Our work differs from them in that we analyzed an access network taking the network hierarchy into consideration, and studied the various aspects of a campus network. Furthermore, in order to get clear insights of characteristics onto the campus network, we have used both node outage and link failure data and investigate their impact on each other.

Syslog messages have no specific format. They have a minimal structure to compose a message. Due to the minimal structure, their formats differ depending on various vendors request and various router operating systems. We need ways to standardize with a general approach to achieve analyzing Syslog messages from different formats.

Many kinds of researches [10, 11, 12, 13, 14] have proposed methodologies to help operators in many ways such as performance analysis, network usage analysis, troubleshooting or diagnose errors by utilizing the Syslog messages. Their approaches mainly depend on analysis of Syslog data they get from the routers in the networks.

Another approach utilizing the Syslog messages [21, 18, 19, 20] is extracting valuable or meaningful information from the Syslog messages or clustering the Syslog messages into a smaller number of meaningful categories or events. Our implementation

is specifically designed to achieve this purpose. Many papers have been proposed to deal with this approach. [18] applies the Teiresias algorithm to automatically classify Syslog messages. It discovers all patterns in categorical data of at least a user-given specificity and support. [19] mainly analyzes the networking datasets with a mathematical approach. Yamanishi et al. introduced a methodology of dynamic Syslog mining for network failure monitoring. Their methodology typically generates a predictive alarm for system failures. [20] describes mathematical methods used for Syslog messages processing. The goal is to find out some mathematical description of Syslog behavior which would allow them to check if the network behavior is usual or if it needs some special attention. Since the Syslog message structures may change in time, they are looking for some adaptive solution not based on the semantics of the messages. [21] introduces the SyslogDigest system which automatically summarizes the high volume of raw Syslog messages into a small number of meaningful and prioritized network events. They didn't design the SyslogDigest system for troubleshooting, but they assure that it can benefit complex troubleshooting task significantly. One of the issues is they use a combination of words with high frequency to find the subtypes of the Syslog message templates. The problem is they use a static value when they prune the tree ( $k=10$ ). This makes so many false positives which can cause one of the main factors of degradation of the processing time.

### **2.3 Node and Link Data Source**

We have used extensive and complete data sets for network outage and failure analysis. We have collected the node outage data as well as the link failure data from the

Table 1: Node Outage and Link Failure Data

Network events	Source data	Data size	Types	Periods
Node outages	SNMP	20.2 Mb	Wired	Apr., 2005 ~ Dec., 2014
			Wireless	Apr., 2005 ~ Mar., 2010
Link failures	Syslog	39.3 Gb (zipped)	Wired	Dec., 2017 ~ Dec., 2014
			Wireless	Oct., 2008 ~ Dec., 2010

university campus network. The campus network of our study is designed in a hierarchical manner that is a common practice of campus or enterprise networks [22]. Particularly, we used SNMP and Syslog as a source of node outages and link failures, respectively. As for network topology, we have utilized the topology information tool, called ‘Intermapper’. It provides physical locations of all the devices in the campus and relationships between them. Additionally, we have discussed the network operators’ anecdotal comments on special events and actions.

For our analysis, the most valuable data are node outage and link failure data. Node outage data was gathered by SNMP polling and trap, and it is from April 7, 2005, till December 9, 2014, with 248,366 outage events. The polling time varies depending on the monitored devices ranging from 2 minutes to 5 minutes. The outage event time is recorded in the unit of minutes, and the outage duration is measured with second granularity. Link failure data is collected from each device to a central Syslog server. The period of data is from October 1, 2008 to December 9, 2014. While we appreciate various types of Syslog messages, we primarily consider ‘LINK-3-UPDOWN’ messages for the link failure analysis. The 6-year data contains roughly 262 million Syslog messages and

about 44.1 million messages represent 'LINK-3-UPDOWN' messages.

### 2.3.1 Campus Network Architecture

The core serves as a backbone for the network. The individual building blocks are interconnected using a core layer. The core devices are high capacity routers and expected to be very resilient as most of the building blocks depend on it for connectivity. There are only a few routers in this layer for a minimal configuration so as to limit the possibility of operational error. The distribution layer aggregates nodes from the access layer, protecting the core from high-density peering. Additionally, the distribution layer creates a fault boundary providing a logical isolation point in the event of a failure originating in the access layer. Typically there is one distribution node per building and it is deployed as a pair of L3 switches, the distribution layer uses L3 switching for its connectivity to the core of the network and L2 services for its connectivity to the access layer. The access layer is the first point of entry into the network for edge devices and end stations.

The building blocks of modular networks are easy to replicate, redesign, and expand. There is no need to redesign the whole network each time a module is added or removed. Distinct building blocks can be put in-service and taken out-of-service with little impact on the rest of the network. This capability facilitates troubleshooting, problem isolation, and network management. In a hierarchical design, the capacity, features, and functionality of a specific device are optimized for its position in the network and the role that it plays. The number of flows and their associated bandwidth requirements increase as they traverse points of aggregation and move up the hierarchy from access to

distribution and to the core layer.

Redundancy is addressed in several ways. Core routers have redundant power supplies, and all devices connected to them are connected to both, providing redundant paths between core and distribution layers. In the distribution layer, stackable switches provide redundant power supplies and redundant supervisor engines where possible by connecting each device to more than one other switch in the stack. Access-layer uplinks are redundant as well, and some of the access switches have redundant power supplies. Connections to hosts are typically non-redundant, though there is a provision in the data centers for servers to be connected with dual links, with the servers NIC drivers responsible for managing the connections. As for the intra-domain routing protocol, EIGRP is used. It is responsible for traffic management on links between the core and distribution layers; Gigabit EtherChannel provides similar functionality between the distribution and access layers. This eliminates the need for spanning-tree (in our case, Rapid-PVST) recalculation. EtherChannel works by bonding two physical connections into one virtual link. As a result, STP (Spanning Tree Protocol) uses the virtual link in its calculations, thereby avoiding any blocked links.

It is possible that the link failure can occur due to software/hardware malfunction, natural or human-caused incidents, and may not lead to service outage due to redundancy or recovery mechanisms.

There may be some possible artifacts in the data, however, due to in-band (the monitoring data follows the same physical path as the user data) monitoring, the SNMP polling interval, and nature of the protocol. Failure or outage reporting can be affected

Table 2: Data Processed by Node Type

Node Type	Node Outage Processed	Link Failure Processed
Core	1,016	74,167
Distribution	1,354	1.56 Million
Access	184,597	17.8 Million

by the topology of the network. Any failure that is on the path to the monitoring system would likely result in an outage being reported for all devices on the path, though it is possible that the issue only affected one host.

If connectivity is lost between the sending device and the Syslog server, the Syslog event would not be recorded. Additionally, as Syslog uses UDP protocol, data can possibly be lost due to transient network congestion, CPU load, OS patching, EIGRP reconvergence, STP (Spanning Tree Protocol) recalculation, etc.

## 2.4 Wired vs. Wireless Networks in Access Layer

In this section, we explore and compare the characteristics of wired and wireless networks in the access layer. There are 662 nodes, physical network devices attached to the network and capable of creating, receiving, or transmitting information over the internet. There are 960 links connecting these nodes. First, we go over the statistical information of node outages.

Table 3 and 2 summarizes the node outage and link failure events collected through the both of data sets. In the access layer, the node outage data shows that total 49,770 node outage events over nine years from April 2005 to December 2014. However, we



Table 3: Wired vs Wireless Outage Data

Data Source	Network types	% of events
Node outages (SNMP)	Wired	59.64%
	Wireless	40.36%
Link failures (Syslog)	Wired	99.9976%
	Wireless	0.0024%

have wireless outage data until Mar 2010 so we investigate node outages from April 2005 to Mar 2010 (23,411 node outages). About 40% (9,449 node outages) of events coming from the wireless components. Link failure data shows that about 17.8 million link failure events over six years from October 2008 to December 2014. However, since we only have wireless link failure data from October 2008 to March 2010, we compare the link failures of wired and wireless during this period (around 2.37 million link failures). Due to the higher complexity of the network under each switch, it is natural that we have much more link failure events than node outage events. Based on the statistics we had from the node outage events, we expected that there are about 40% of link failures coming from the wireless link failure data. Interestingly, only 178 link failures which is 0.0024% of all link failures are detected from the wireless components in the link failure data. The main reason that we have much smaller percentages of the wireless link failures is due to the topology of our campus network. The APs in our campus network only have one wired connection to a switch in the access layer. That's the only way that the AP carries client traffic to the network, as well as the mechanism for sending link failure messages. If the link goes down, there is no alternative route to reach the Syslog server, as the link

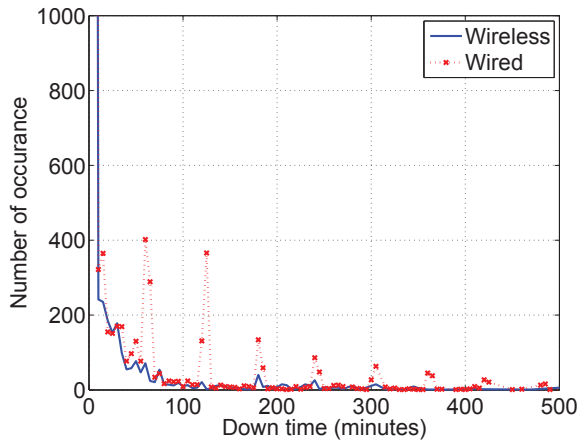


Figure 5: Wired and Wireless Down-time: bin size of the x-axis is 5 Minutes

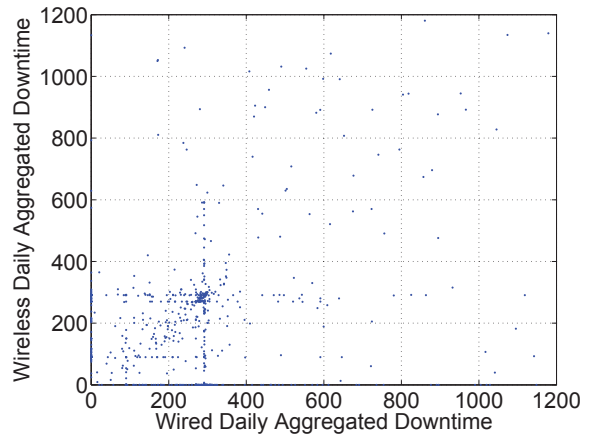


Figure 6: Correlation between Wired and Wireless Node Outages

which is down is the one carrying the message. This limitation restricts us from comparing the wired and wireless networks in the view of link failures. Hence, we prefer to compare both of the networks by using the node outage data in the following analysis. The APs which mainly consists of the wireless network is located at the most bottom of the hierarchical layers in the campus network. Therefore, the performance of the wireless network strongly depends on the performance of the wired distribution network. The impact of outages in the wired distribution network directly affects the outages of the wireless devices connected to the corresponding wired device. In addition, APs themselves are brought down when we reconfigure the device settings, or when the device is rebooting for an upgrade. This extra outage of the APs will be added to the total outage duration of the wireless network. Processing the data of the link failures which include decompressing the file, collecting the key data, and re-compressing the file (to conserve disk space), took 38 hours.

We compare the downtime of wired and wireless node outages. As shown in Figure 5, even though the number of the wireless node outages are two times less than that of the wired node outages, the characteristics are the same such that the majority of the outage is less than 5 minutes duration. This is because wireless devices are directly connected to wired components and are affected by the performance of the wired components. Figure 6 verifies our description on the wired and wireless networks of our campus network. It shows the correlations of the average outage duration of the wired and wireless networks for each day. There are some exceptions where the outage duration of the wired network is longer than those of the wireless network which are located near the x-axis. This is because some switches are not connected to APs so the node outages of these switches did not induce the APs' outage. Additionally, there are two possibilities that caused this issue even though there are APs connected to the switches. As introduced in the previous section, one possibility is that the data plane of the switch is still passing traffic, but the management plane is having an issue of some sort, so it can not receive/respond to monitoring traffic but is capable of passing the monitoring traffic along to the AP. The second possibility is that the switch rebooted (and possibly the AP as well), but the polling cycle was offset just enough that only the switch was noticed by the monitoring system.

In order to validate the correlation between wired and wireless node outages, we aggregate the node outage times daily and compare them. The daily aggregated node outage times are shown in Figure 6. To increase the visibility, we limit the ranges of the x-axis and the y-axis to 1200 second. Each dot represents the daily aggregated node

outage time of the wired and the wireless networks in the access layer simultaneously. 29.36% outage events in the wireless access network occur independently from the wired outages. This indicates that the wireless devices in our network are less reliable and more troublesome than the wired devices. The drastically increasing number of wireless capable devices can explain this phenomenon. 11.43% outage events in the wired access network occur independently from the wireless outages. In terms of the degree of the correlation, the value between wired and wireless access network was 0.6158 (the closer to 1, the stronger correlation) and it shows that the two different networks work dependently on each other. The correlation even gets higher to 0.7377 if we remove the outage events that occurred independently.

## 2.5 Characteristics of Layers

In this section, we explore and compare the characteristics of each layer (i.e., core, distribution, and access layers) of the campus network. First of all, we go over the statistical information of node outages. We found about 40,000 node outages from the SNMP data set and classified them into layers. The percentages and the best fitting distribution of the node outage events of each layer are shown in Table 4. In terms of the percentages of node outage events, it is clear that the core and distribution layers are more stable than the access layer. The hierarchical architecture of the campus network contributes higher node outages in the access layer. This is because one node outage in the upper layer affects the devices in the lower layer that are connected to the corresponding device in the upper layer.

The results of the distributions show that the access layer is well approximated by the Pareto distribution and the core and distribution layers are approximated by the exponential distribution. In order to evaluate the goodness of fit, we used the first dist function in matlab. As you can see from Figure 7, the core and distribution layers are highly skewed from their best fitting distributions. This is because of the switches and routers in the core and distribution layers are monitored actively by network operators and almost 98% of node outages are fixed within 300 seconds. Compared to these two layers, the node outages in the access layer shows less number of events that are lasted less than 100 seconds. On the other hand, it has a much longer tail than the others. This is mainly because some of the switches in the access layer did not get quick attention of network operators due to its low or no impact on other devices or other parts of the network. Therefore, we can see the different characteristics of different layers based on their roles in the hierarchical network architecture.

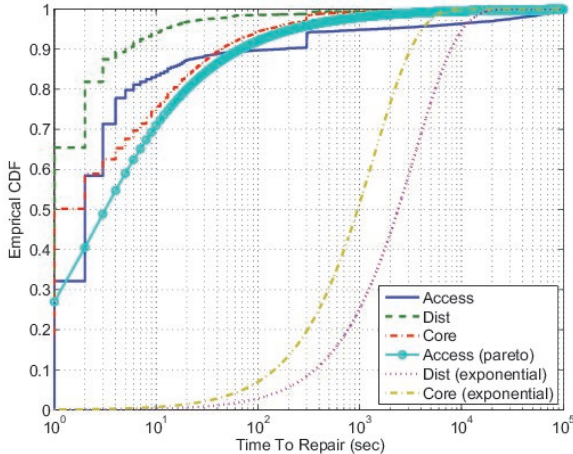


Figure 7: Distribution for Time-to-Repair of Each Layer

Table 4: Statistical Information of Node Outages

Network types	% of events	Distribution
Core layer	0.381%	Exponential ( $\mu$ : 1378.65)
Distribution layer	8.112%	Exponential ( $\mu$ : 3439.01)
Access layer	91.507%	Pareto ( $\sigma$ : 1.69, $\theta$ : 2.41)

Next, the empirical distributions of downtimes of the node outages are shown in Figure 7. Each layer represents the different characteristic of node outages.

## 2.6 Analysis of Outage

### 2.6.1 Analysis with Splunk

Splunk is one of well-known big data analysis tools and it provides powerful classification and statistics in a very easy way by capturing, indexing, and correlating real-time data. It analyzes the similarity between each line of the given data and recognizes the format of the messages or anomalies. It is very useful to quickly check various statistics of big data in real-time. Therefore, it enable us to have an agile visibility on the data and manage systems efficiently. Algorithm 1 and 2 illustrate our Splunk search algorithm used to retrieve selected down messages for all hosts generated from LINK-3-UPDOWN where the status changed from up to down.

As the size of the network increases, network operators usually focus on only important links which are uplinks from a switch to other switches in the upper layer. Considering the limited human resources, it's impossible for them to track all the network

---

**Algorithm 1** Search by Key Field for All Hosts By Link-3-UPDOWN

---

**Input:** allMessages**Output:** selectedData

```
1: for all message  $\in$  queue do ▷ retrieve selected message
2:   if message has LINK-3-UPDOWN and status change to down then
3:     get all hosts
4:     get all months
5:     group by 24 hours
6:     return the search result
7:   end if
8: end for
```

---

---

**Algorithm 2** Splunk Search by Key Field for Down

---

**Input:** allMessages**Output:** selectedData

```
1: for all message  $\in$  queue do ▷ retrieve selected message
2:   if message has down then
3:     get host = core
4:     get all months
5:     group by 12 months
6:     return the search result
7:   end if
8: end for
```

---

messages caused by very end links since the sheer amount of messages are being generated everyday. Currently, the issues with individual interfaces are not monitored well nor fixed unless a user contacts the network operators. However, for better user experience, we need to harness the Syslog messages by providing an automatic tool that detects detrimental network events based on the campus network policies. In order to quickly identify network anomaly, we conducted a quantitative analysis which ranks the number of node outages and link failures. We have used Splunk for this analysis so that we can identify

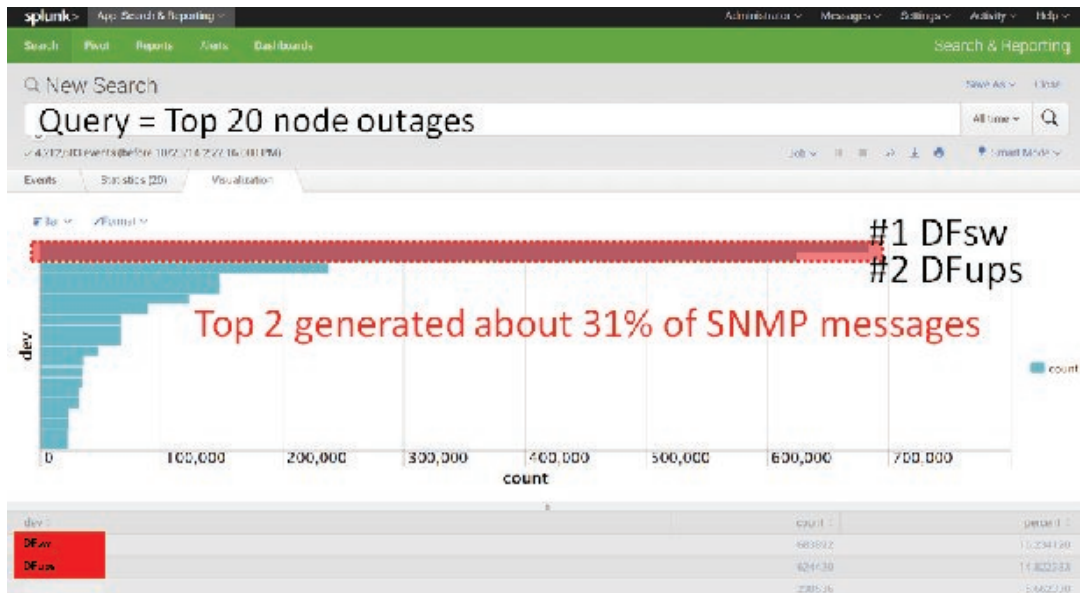


Figure 8: Node Outage with Splunk

the problematic areas in our campus network taking the spatial and temporal aspects into consideration. For example, Splunk identifies that our network has many node outages in the campus soccer field which is a wide open area. Since no students expect that the Wi-Fi is available in this area, no complaint has been issued and it was left unfixed. Figure 8 illustrate the soccer field node outage with Splunk. Splunk also can be used to detect a problematic network component. In this example, Splunk indicates that we have many link failures in one of the switches in the medical building as illustrated by Figure 9. The possible reasons could be related to a bad port on the switch, adapter on a client's NIC, or very old cables such as CAT3. In this case, the old cables caused these errors. After the new wiring installation, these problems were gone. This type of errors has a detrimental impact on only individual network devices. That's why it didn't get urgent attention. We



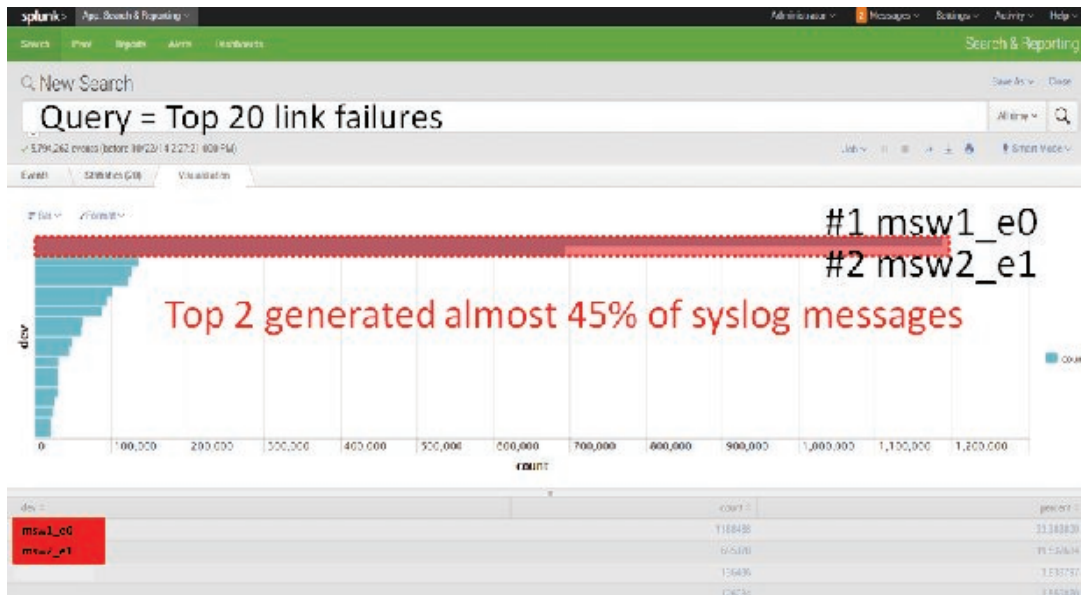


Figure 9: Link Outage with Splunk

also used an application called Intermapper, a cross-platform, network monitoring, and network mapping. Intermapper gives provides physical locations of all the devices in the campus and the relationships between them. Intermapper enables the server and monitor via SNMP probes that can do basic echo pings or read memory and CPU usage, depends on the equipment, at specific intervals and then the server will send us an email if there is an issue with the probe. Which tells us we need to look into the device.

These network events which are captured by Splunk are hard to be monitored by IT administrators since these errors don't have a significant impact on the network operation and there is no way for them to get this information unless they look through all the node outages and link failure events. Network health analysis with Splunk helps IT administrators actively search problematic areas and devices in an efficient and easy way.

---

**Algorithm 3** Calculate Mean Time To Repair

---

**Input:** allMessages**Output:** MTTR

```
1: for all status  $\in$  queue do ▷ retrieve status change
2:   if status change to down then
3:     get timestamp = downtime
4:     get swithid = downswitch
5:   end if
6:   if status change to up then
7:     get timestamp = uptime
8:     get swithid = upswitch
9:     if downswitch == upswitch then
10:      get MTTR = uptime - downtime
11:    end if
12:    return MTTR
13:   end if
14: end for
```

---

The wireless network was the last to come back to operational. Splunk showed there were 159,577 network events for that day. In order to analyze the mean time to repair (MTTR), we developed python script that calculates the interval between the status change to down and the status change to up, shown in the algorithm 3. The result showed that the different interfaces varied in the MTTR.

## 2.7 Summary

We have shown in-depth analysis of link failure and node outage data on a university campus network in order to understand the health of an access network. In addition to long periods of network data such as messages and SNMP data, we incorporated vendors' documents in regards to the causes and recommended actions, and the network operators'

input on special events and actions. We have explored the characteristics of hierarchical architecture of the campus network and the correlations of wired and wireless networks. We found that the general characteristics of the different layers are very distinct from each other and the wireless network is less reliable compared to the wired network and is affected by the performance of the wired network. This study on a campus network provides insights on the behaviors and conditions of access network availability, and potential end-to-end availability expectations.

In this paper, we develop a system SyslogFilter that filters out unnecessary Syslog messages based on their recommended actions and hierarchical locations. SyslogFilter automatically generates the regular expressions for general parameters such as IP addresses, MAC addresses, etc to remove them from the raw Syslog messages and combine with the router names and timestamps to identify signatures of Syslog messages. We evaluated SyslogFilter using real Syslog data collected from our campus network. The weakness in our work is its possible specificity: our results only necessarily apply to our campus network consisting Cisco routers. While we believe many of our observations would hold true in other similar environments, we have not verified this. We would thus like to compare other studies such as SyslogDigest with our system.

## CHAPTER 3

### TOPOLOGY-AWARE RELIABILITY MANAGEMENT FRAMEWORK FOR SOFTWARED NETWORK SYSTEMS

#### **3.1 Background**

The recent softwarization of network functions, controls, and applications is promising, as it improves the cost efficiency, control accuracy, and deployment flexibility of infrastructures. Software-Defined Networking (SDN) [23, 24] is a softwarization technology that logically centralizes the application and control planes (controllers) of a network by separating them from the underlying data plane (forwarders). OpenFlow [25] has been adopted as a southbound communication protocol between the control plane and data plane networks.

In SDN, where the control plane responsibility is moved to the logically centralized controller, the network reliability schemes are operated by the controllers as centralized management protocols. An SDN controller operates periodic heartbeats to discover the initial network topology. The controller maintains up-to-date network visibility of the discovered network topology using the remote node's status notifications and periodic discovery messages. For example, an SDN controller identifies SDN switches when they initiate a TCP connection to the controller according to the controller's configuration. In addition, it discovers the network link topologies by using discovery protocols such as the Link Layer Discovery Protocol (LLDP) [26], Broadcast Domain Discovery Protocol

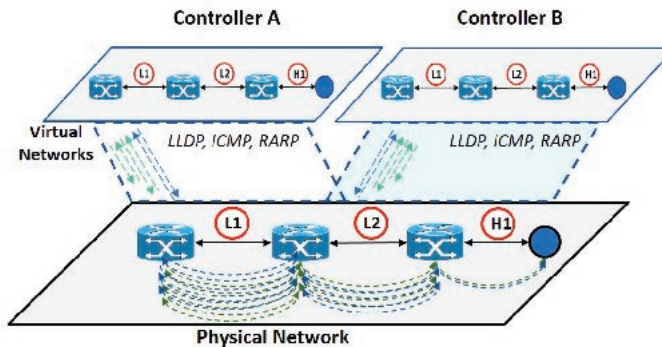


Figure 10: Centralized Reliability Management in SDN

(BDDP) [27], and OpenFlow Discovery Protocol (OFDP) [28]. Optionally, network link failures can be handled by Bidirectional Forwarding Detection (BFD) [29] and OpenFlow Fast Failover [30]. Among them, LLDP is used as a dominant periodic discovery protocol.

However, the centralized discovery protocols in softwarization architecture pose both scalability and latency challenges. In traditional networks, LLDP was configured as an optional distributed protocol for the link layer neighbor discovery (i.e., Cisco Discovery Protocol (CDP)). However, SDN applications rely heavily on LLDP for discovering and maintaining their network visibility. Each SDN controller maintains a *uniform period timer* for a discovery protocol. The total number of LLDP messages for a controller to process for an OFDP discovery period is about twice of the entire number of switch ports including the inter-SDN switch port, host port, and non-SDN switch port. Hence, the control message scalability decreases significantly, if the network size and the discovery

frequency increase. As illustrated in Figure 10, if the forwarding network is used by various virtualized networks as well as is overlapped by the control path (in-band) via linear or tree topology, many redundant control messages will be introduced to the top (i.e., L1 which is near to the controller) of the network switches. We present detailed scalability case studies in Section 3.2. Although there have been a few recent studies that address the issues of failure detection and recoveries in the SDN data plane [31] and control plane [32, 33] networks, respectively, little work has been conducted to the discovery protocols with the network topologies and virtualization domains.

Network reliability management is one of the most crucial operational functions of network service providers (NSPs). A network system fundamentally uses various heartbeat based reliability protocols which are built-in distributed network devices. A node periodically sends and checks heartbeat messages. No heartbeat from a remote node for more than a threshold duration indicates a potential failure of the node or link. Hence, to detect a failure earlier, the heartbeat transmission period can be configured faster.

In this chapter, we propose a Topology-Aware Reliability Management (TARMan) scheme to enhance the scalability and latency issues of the centralized discovery mechanism by dynamically configuring the discovery frequency for a specific target according to the impact instead of using a uniform period for the entire network. For example, in a tree network topology, the impact of a target (node/link) to the network traffic is calculated according to the location, relationship, and functionality (i.e., core, aggregate and edge). In a data center environment, a typical network architecture uses a three-tiered design that has a core tier in the root of the tree, an aggregation tier in the middle, and

an edge tier at the leaves of the tree (i.e., Top of Rack) [34]. TARMan can also facilitate many common reliability monitoring parameters such as protocol type, heartbeat mechanism, period, and target for the registered applications by analyzing both off-line and on-line network topology information. By taking a common corrective action against a failure, it acts as an effective decision making tool leading with improved MTBF. We implemented TARMan into Cisco’s OpenDayLight (ODL) [35] module. The Mininet [36] based experiment results show that TARMan expedites failure detection on the critical network segment without impacting the network scalability.

The remainder of this chapter is organized as follows. Section 3.2 discusses the observations that motivated this work. We introduce our proposed solution and the implementation details in Section 3.3. Section 3.4 provides the experimental results. Section 3.5 describes the related work, and Section 3.6 summarizes the chapter.

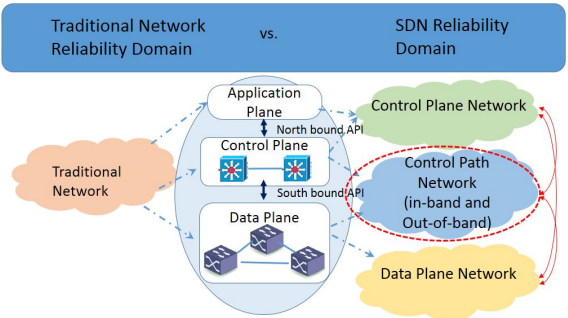


Figure 11: Triptych High Availability (HA) Domains

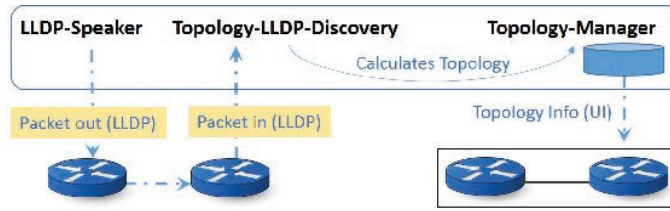


Figure 12: Topology of LLDP-Discovery and LLDP-Speaker

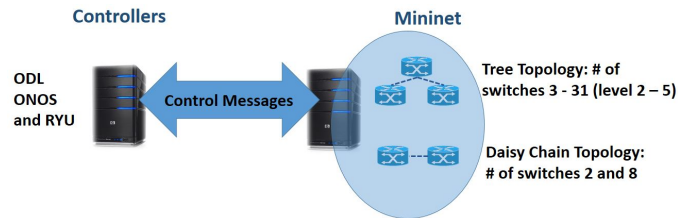


Figure 13: Control Message Analysis Experimental Setup

### 3.2 Motivational Experiments

**Background:** The LLDP has been used by distributed network devices for advertising their local information such as identity, capabilities, and neighbors. In traditional networks, this vendor-neutral link layer protocol is configured as an optional component in network management and monitoring applications. However, in SDN, OFDP, or an SDN LLDP is a centralized discovery protocol that transmits information about the current status of a device and the capabilities of its interfaces. As illustrated in Figure 12, the SDN controller has LLDP facilities including an LLDP Speaker and LLDP Discovery. When a switch is connected to a controller, an LLDP Speaker periodically sends dedicated LLDP packets in the Packet-Out messages for all the interfaces of the network switches.



The switch floods the LLDP packets through all of its ports. Upon receiving an LLDP packet, an SDN switch sends a Packet-In message to its controller for acknowledging a direct link between the switches.

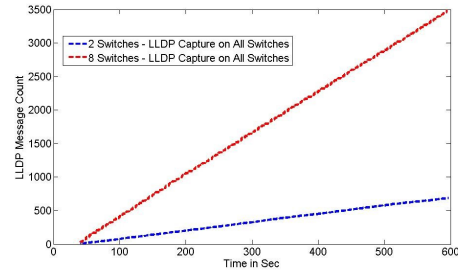
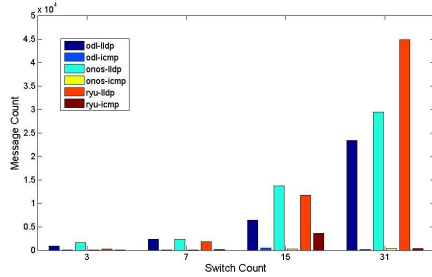


Figure 14: Control Messages from the Con- Figure 15: LLDP Messages Over the Daisy  
troller I/O Chain Network

**Experiments:** As illustrated in Figure 13, we have conducted Mininet based control message experiments with tree topology based forwarding networks by varying the tree depth from 2 to 5 levels (assuming 2 fan outs, there are 3, 7, 15, and 31 switches, respectively) as well as daisy chain based networks by varying the switches between 2 and 8. We have captured control messages for 10 mins with various SDN controllers including ONOS [37], RYU, and ODL.

According to Figure 14, there are various initial control messages to setting up the network. However, among the control messages, periodic LLDP messages are dominant. Specifically, ODL generates the smallest number of control messages. ONOS has a comparatively small number of control messages, which increases linearly as the number of switches increase. On the other hand, RYU has a relatively large number of control messages and the number of control messages increase remarkably as the number of switches increase. This suggests that an RYU controller can be easily congested with

Table 5: LLDP Default Settings in ODL

Parameter	Default
Global LLDP	Disabled
LLDP on interfaces	Enabled(*)
LLDP hold time (before discarding)	120 seconds
LLDP reinitialization delay	2 seconds
LLDP timer (packet update frequency)	30 seconds
LLDP TLVs	Enabled
LLDP receive	Enabled(*)
LLDP transmit	Enabled(*)
DCBXP	Enabled, with enabled LLDP

its own control messages.

Figure 15 presents accumulative LLDP message counts on the daisy chain topology networks with 2 and 8 switches, respectively. As the network size increases, the amount of control messages dramatically increases due to the path dependency among the network switches. For example, as shown in Figure 10, if an SDN network segment is an in-band network or over the tree topology, a link failure on L1 can have a more significant impact on the entire network than a link failure on L2 because more components are depending upon the L1 link. Hence, increasing the LLDP discovery frequency will worsen the scalability of the network, but decreasing the LLDP discovery frequency will delay the detection of the network status changes.

### 3.3 Topology-Aware Reliability Management

The Topology-Aware Reliability Management (TARMan) framework enhances the scalability and latency issues of the centralized discovery mechanism by assigning

target specific discovery frequencies instead of using a uniform period for the entire network. It promotes configurable hierarchical network topologies in an order of importance based on the core, aggregation, or edge. The impact of a target (node/link) to the network traffic is calculated according to the location, relationship, and functionality (i.e., core, aggregation, and edge). We used a zone concept to differentiate the discovery message period according to the network topology. In this paper, we statically use a simple 3 depth binary tree topology (dynamic zone discovery and other network topologies are not discussed in this paper). The root node, such as the core switch, is categorized as Zone1. The middle/aggregation tier of the topology is categorized as Zone2. The edge nodes are categorized as Zone3. For example, a failure in a core switch may impact the most data traffic and control messages. By sending more frequent discovery messages to the core (the more important nodes), TARMan can expedite a failure detection and recovery on the critical network segment while sending less frequent discovery messages to the edge nodes. Considering the network distance to travel, TARMan can achieve faster detection without worsening the network scalability issues. Similarly, for non-tree topology, TARMan can be configured into two tiers. The core switch that is closest to the controller will have the most weight in the two tier approach. The rest of the switches will have evenly distributed weight. This will enable the controller to keep closer look at the main switch.

As illustrated in Figure 16, the TARMan module is implemented in the LLDP-Speaker module of an ODL controller. LLDP-Speaker is an application in the Openflow module for sending LLDP frames. A NodeConnectorInventoryEventTranslator() thread listens on DataChange events such as nodes added or removed and node-links added or

removed from the network nodes. It maintains all the information of connected node IDs and node-connectors in the LLDP-speaker module. The node-connector consists of the node ID and port number. The LLDP-speaker module runs a thread that sends the LLDP frames packaged into Openflow PACKET\_OUT messages to all learned nodes for every 5 secs. We intercept this routine to embed our TARMAN module. Algorithm 4 presents an LLDP discovery frequency function of the TARMAN module that returns an LLDPfrequency value for each target switch (switchID). The LLDP frequency determines how often the LLDP-Speaker sends a probe to a specific switch for requesting the LLDP message. First, it calls the getZones function to check the right zone of the target. It reads a switch functionality from the configuration (ZoneInfo) using the switch ID. Using the switch functionality (i.e., core, aggregation, and edge), it assigns the zone number. Second, using the zone number, it calls the readFrequency function. It reads an LLDP control message period from the configuration using the zone value. The frequency value can be dynamically assigned according to the network condition. With the returned LLDPfrequency (period) and switchID (target), the TARMAN module calls a packetProcessingService.transmitPacket(PacketInput) API. The PacketInput consists of the NodeConnector, LLDP payload, and other pointers. In an Openflow packet processing module, the packetProcessingService.transmitPacket(PacketInput) function calls the messageservice.packetOut(); that is, an ODL API to send the packets out to switches.

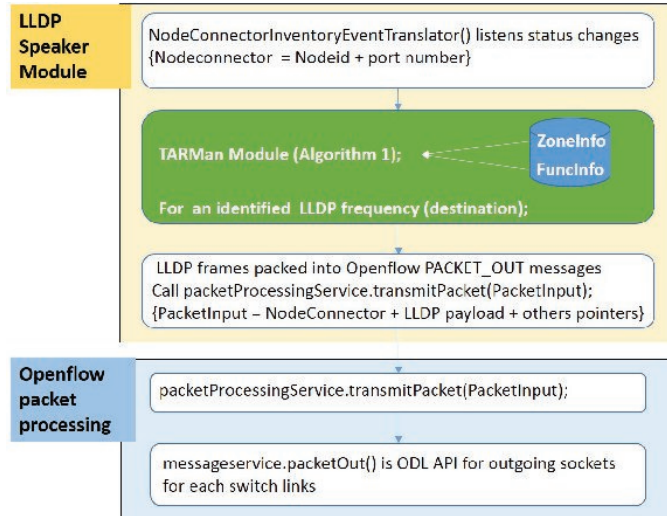


Figure 16: Topology-Aware Reliability Management (TARMan) implementation

### 3.4 Evaluations

#### 3.4.1 Experimental Setup

We investigate our proposed architecture through emulation of real implementation. For the emulation environment we used DELL PowerEdge T320 server with Intel (R) Xeon (R) CPU ES-2403 V2 @ 1.80 GHz x4 and Ubuntu 14.04 OS. We have conducted four experiments: 1) testing LLDP message overhead from the controller I/O in Figure 17, 2) testing accumulated control message hop counts by capturing LLDP messages from the switches in Figure 18, 3) testing the impact of control message outage over the in-band networks in Figure 19, and 4) testing the impact of data flow outage in Figure 20. For aforementioned experiments, we built a balanced binary tree topology network with a depth of three (7 switches and 8 hosts) by using Mininet. This enables

---

**Algorithm 4** LLDP Discovery Frequency

---

**Input:** switchID**Output:** lldpfrequency

```
1: for all SwitchID  $\in$  queue do
2:   zone = getZones(switchID)
3:   lldpfrequency = readFrequency(zone)
4:   // read a LLDP control message period from the configuration using the zone
   value
5:   return lldpfrequency
6: end for
7: function GETZONES(switchID)
8:   // read a switch functionality from the configuration using the switch ID
9:   if readSWfunc(switchID)  $\in$  core then
10:    return 1
11:  else
12:    if readSWfunc(switchID)  $\in$  agg then
13:      return 2
14:    else
15:      return 3
16:    end if
17:  end if
18: end function
```

---

us to create data center like topology architecture. Ping is used to generate packets and Wireshark is used to capture messages from the loop-back interface. The ODL controller uses the default 5 second uniform intervals in sending LLDP PACKET\_OUT messages to the switches, named an ODL (5,5,5) set. In the TARMan module, we configured a couple of simplified frequency sets. The TARMan (1,3,5) set is with intervals of 1, 3, and 5 seconds for the core (top), aggregation (middle part of the topology), and edge switches, respectively. The TARMan (1,3,5) set creates the more frequent LLDP messages for the core and aggregation than the ODL (5,5,5) set in order to detect failures faster from those

important links or nodes. The TARMan (1,5,10) set is with intervals of 1, 5, and 10 seconds for the core, aggregation, and edge switches, respectively. The TARMan (1,5,10) set creates the more frequent LLDP messages to the core switches than the ODL (5,5,5) set to have the quicker updates from the core links or switches. However, it creates the less frequent LLDP messages to the edge switches than the ODL (5,5,5) set that could have the slower updates from the edge links and switches. We use the default 3 consecutive LLDP message failures to change a status.

### 3.4.2 Control Message Overheads

The goal in proposing three different configurable frequencies is to allow for different failure discovery time based on the role of the node on the overall network. To decrease the delay due to failures, it is key to protect the traffic involved from the link and node failures [33]. For example, if a failure happens at the root of the network, then the controller needs to know about it sooner than if it happens in the middle of the network or the edge of the network because that is the closest node. Without the core, the controller cannot reach the rest of the network. For that purpose, we configured the LLDP PACKET.OUT frequency to every one second. At the aggregate, the controller would want to know the state of the network sooner than the edge network because the aggregate network is closer to the controller than the edge. Therefore, we configure the aggregate network between three to five seconds. The edge network is the farthest to the controller with the least impact on the overall network and we configure the interval to be between five to ten seconds.

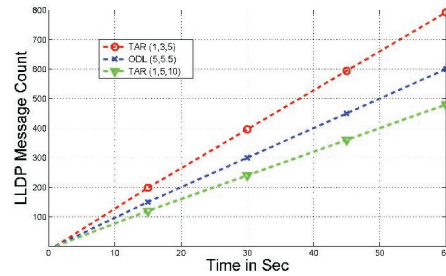
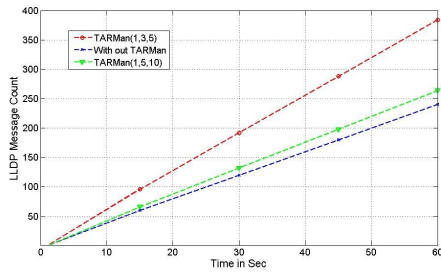


Figure 17: LLDP Messages Captured on a Controller

Figure 18: Accumulated LLDP Messages Captured by the Switches

Table 6: LLDP Message Count in One Minute

Time	TARMan(1/3/5)	TARMan(1/5/10)	Without TARMan
15 sec	96	66	60
30 sec	192	132	120
45 sec	288	198	180
60 sec	384	264	240

According to Figure 17, it is obvious that both the TARMan (1,3,5) and TARMan (1,5,10) sets create relatively more LLDP messages from the controller than the ODL (5,5,5) set. However, the LLDP messages are not significantly increased. Especially, the TARMan (1,5,10) set generates the similar amount of LLDP messages to the ODL (5,5,5) set. On the other hand, Figure 18 presents the practical network usage (accumulated LLDP messages per network hop) by the LLDP messages in case of the in-band networks. The presented results are the accumulated LLDP messages captured by the network switches. Although the TARMan (1,3,5) set still has slightly higher accumulated LLDP messages than the ODL (5,5,5) set, the TARMan (1,5,10) set creates less accumulated LLDP messages than the ODL (5,5,5) set. This is because the important core and



Table 7: LLDP Message Count by Hop in One Minute

Time	TARMan(1/3/5)	TARMan(1/5/10)	Without TARMan
15 sec	198	120	150
30 sec	396	240	300
45 sec	594	360	450
60 sec	792	480	2600

aggregation nodes are closer to the controller with the in-band network, which creates less accumulated LLDP messages. This results are promising because creating more frequent messages to the core does not create a significant control message overhead.

### 3.4.3 Impact of Control Message Outages

Our goal of this experiment is to appreciate the impact of a link or node failure to the control messages, especially, over the in-band network. In the in-band network where a controller is connected to the network via a core switch, a failure of a core switch could cause failures to the entire network, which is connected to the core switch. For example, when a core switch is down, the controller may lose access to the rest of the network. Hence, the outage time of the control message is bounded by the detection time of the failed switch. We define an Impact ( $I_c$ ) as an accumulated outage time in seconds, which is a product of the number of impaired/impacted nodes ( $N_i$ ) and the outage time of the top impacted node ( $O_t$ ) (i.e.,  $(I_c) = (N_i) * (O_t)$ ).

Figure 19 presents Impact ( $I_c$ ) of the core, aggregation, and edge switch failures to the TARMan (1,3,5), TARMan (1,5,10), and ODL (5,5,5) sets. It shows that a core switch failure cause far less impact to both the TARMan (1,3,5) and TARMan (1,5,10)

sets than the ODL (5,5,5) set, which means the TARMan sets could react much faster to the important core switch failure. Although the TARMan (1,5,10) set impacts greater than others on the edge switch failure, overall, the TARMan sets cause far less outage than the ODL (5,5,5) set.

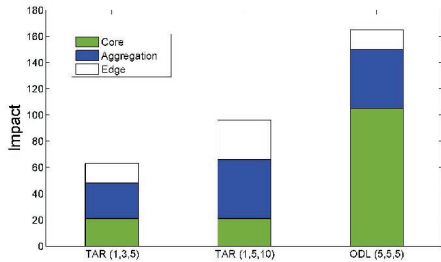


Figure 19: Impact of Control Message Outage

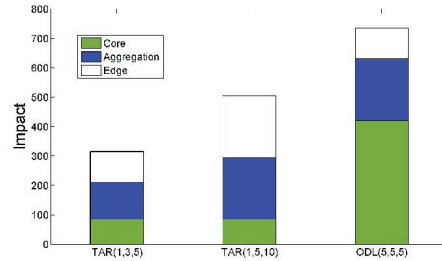


Figure 20: Impact of Data Flow Outage

### 3.4.4 Impact of Data Flow Outage

Our goal of this experiment is to evaluate the impact of a link or node failure to the data flows over the tree topology network. In the tree topology network, a switch failure creates a couple of separate networks where the data flows could not communicate with each other. Hence, the outage time of the data flows is bounded by the detection time of the failed core switch and a half of the network traffic. In our experiment, we assumed that there could be a total of 56 bi-directional flows from 8 hosts. A core node failure would result in 28 bi-directional flows impacted by the outage in theory. We define an Impact ( $I_f$ ) as an accumulated outage time in seconds, which is a product of the number of potential impaired/impacted data flows ( $F_i$ ) and the outage time of the top impacted node ( $O_t$ ) (i.e.,  $I_f = (F_i) * (O_t)$ ). For ( $O_t$ ), we used 3 times of the LLDP message

frequency. Total impacted bi-directional flows( $F_i$ ) were tested by enabling ping flows from all the hosts. Figure 20 presents Impact ( $I_f$ ) of the core, aggregation, and edge switch failures to the TARMan (1,3,5), TARMan (1,5,10), and ODL (5,5,5) sets. It shows that a core switch failure cause far more impact to the ODL (5,5,5) set than both the TARMan (1,3,5) and TARMan (1,5,10) sets. This indicates that the TARMan sets could cause much shorter data flow outage than the ODL (5,5,5) set during the important core switch failure. Although the TARMan (1,5,10) set causes the longer failure than others on the edge switch failure, the real impact on the edge could be minimal (as the edge is less important switch than others). Most significantly, the results show that the TARMan sets cause far less data flow outage than the ODL (5,5,5) set.

Table 8: Theoretical Impact Factor of Control Messages

Topology	TARMan(1/3/5)	TARMan(1/5/10)	Without TARMan
Core	84	84	420
Aggregate	126	210	210
Edge	105	210	105

Table 9: Experiment Result of Impact Factor of Control Messages

Topology	TARMan(1/3/5)	TARMan(1/5/10)	Without TARMan
Core	96	96	480
Aggregate	360	600	600
Edge	360	720	360

we set up our experiment using Wireshark. Wireshark is a well-known protocol

analyzer tool. It is highly used by both industries and educational institutions, and it allows us to see what's happening on our network at a microscopic level. It gives us real time capture of all the packets as they are being exchanged between the controller and all the switches in the network. The desire to capture the Link Layer Discovery Protocol (LLDP) messages between the controller and the switches. LLDP is vendor-neutral link layer protocol in the Internet Protocol (IP) Suite used by network devices for advertising their identity, capabilities, and neighbors on an IEEE 802 local area network, principally wired Ethernet [38]. In a traditional network, LLDP protocol message is exchanged between the adjacent neighbors to update each other's table with their identity and capabilities. However, in softwarized networks where the control plane responsibility is moved to the centralized controller, the LLDP protocol message is initiated by the controller to understand the overall network and the physical links.

### **3.5 Related Work**

There have been a few recent studies that focus on the failure recoveries in each network domain, such as OpenFlow switches or links connecting them in the data plane [39] and the controller cluster networks for improving reliability [33]. However, little work was found that orchestrates the overarching failure detection and recovery modes of all of the network domains [40]. Furthermore, many existing reliability researches start the recovery process only after detecting a failure. Information fusion from the various sources (i.e., exploiting the network topology and the low level link signal along with the heartbeat messages) not only detects failures quickly, but it also proposes another potential

enhancement by overlapping the failure detection and recovery windows. Some research focused on efficient SDN topology discovery [28] where the authors proposed an enhancement to OpenFlow Discovery Protocol (OFDP), named OFDPv2. OFDPv2 limits the number of LLDP Packet-Out messages sent to each switch by using an OpenFlow OFPT FEATURES REQUEST at the connection establishment message between OF and the switches. Utilizing the establishment message allows the controller to have one-to-one mapping of all the MAC addresses and Port IDs. In OFDPv2, the number of LLDP Packet-Out messages is reduced to achieve a reduction of CPU time. Their objective is met by reducing the CPU load imposed by the SDN controller's topology discovery service [41]. Failure recovery in an SDN is determined by the specific software logic running at the controller [42]. A proposed solution is a run time system that automates failure recovery and enables network developers to write simpler failure-agnostic codes. Upon detecting a failure, their approach first spawns a new controller instance in an emulated environment excluding the failed elements followed by quick replay inputs observed by the controller before the failure occurred, and leading the emulated network into the forwarding state that accounts for the failed elements. It then recovers the network by installing the difference rule set between emulated and current forwarding states [43]. An SDN fast failure recovery work [30] shows that if a failure occurs along the data traffic path, both control and data traffic can be affected. The paper explains how failure recovery can be deployed in such a network and focus on failure recovery mechanisms for the in-band OpenFlow network. They proposed restoration and protection techniques for control traffic, while utilizing their previously proposed restoration and protection mechanisms of

the out-of-band network for data traffic. In order to achieve the quality of carrier-grade OpenFlow networks, both control and data traffic should recover from a failure within 50 ms. However, their failure recovery experiment using NOX [44] and Mininet presents that the traffic recovery cannot be done within 50 ms in a large-scale network.

Our work differs from the existing because we propose to reduce the overall control message yet focus on retaining key protocol messages and as a result increase the reliability of the network. We introduced an hierarchy-based control scheme that maintains different control frequencies according to the topology. It gives more attention to the important components (i.e., core nodes), while the overall control messages can be reduced. Comparing the fast failure recovery research, our work differs from theirs because we enhance the network reliability through efficient link/node failure detection while they left the focus on the switch or controller side of the network for future work. \*) denotes after LLDP is enabled globally

he remainder of this paper is organized as follows.

### **3.6 Summary**

Little attention has been paid to the network discovery protocols in softwarized network systems that suffer from scalability and latency issues. We proposed a novel Topology-Aware Reliability Management (TARMan) framework that exploits network tier architecture and deliberately controls the frequency of LLDP messages depending on each tier's failure impact. Based on failure impact, we enabled the TARMan platform to

provide fast and smart decision making information for fast failure detection and recovery. A prototype is implemented on Cisco's OpenDayLight (ODL). Extensive experiment results exhibit that our algorithm achieves effective and efficient network failure detection while generating limited LLDP message overhead.

## CHAPTER 4

### PROTOCOL HETEROGENEITY ISSUES OF CAMPUS INCREMENTAL WI-FI UPGRADE

#### **4.1 Background**

The availability of versatile and resilient wireless networks with high-speed performance and Wi-Fi on-the-go service with always-connected features for bandwidth-intensive applications is considered a basic necessity these days. Faster connection and speedy performance are the bare requirements for emerging bandwidth intensive services like high resolution video uploading (i.e., YouTube), video streaming (i.e., Netflix), virtual reality, augmented reality, real-time updating data (i.e., Facebook), online gaming, and live stream video (i.e. surveillance camera data). For example, Cisco announced the Zettabyte Era [45] is expecting the global traffic to extend to 3.3 ZB per year by 2021 or 278 EB per month [46]. Devices and connections are growing at a 10 percent compound annual growth rate (CAGR) [47], and North American growth is expected to be the highest rate in the world [46]. Adaptation of IPv6 throughout the Internet allows for an effortless interaction among the Internet of Things (IoT) [48]. Internet traffic is increased by internet video, video-streaming, gaming, video-conferencing, and video specific applications. Most internet users use the internet on their mobile devices such as tablets, phone, laptop, etc., and they get their connection through wireless data services. They require high throughput and seamless connectivity without outage from their network



services. Mobile customers are dependent on Wi-Fi networks for higher data rate and

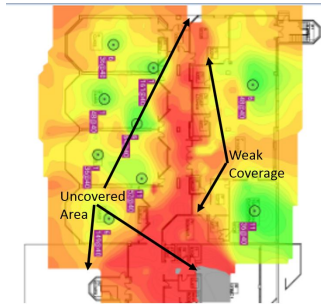


Figure 21: An Example of Coverage Map

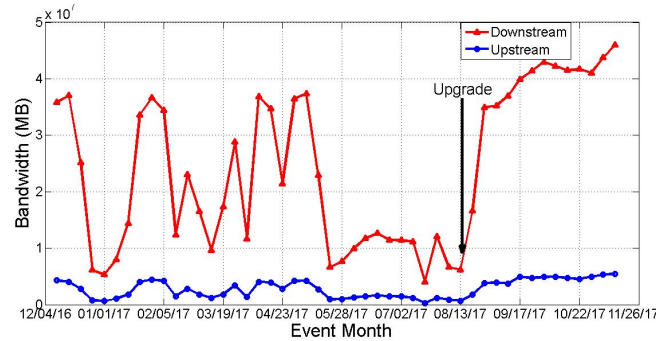


Figure 22: Network Traffic

good coverage area. There is high demand for Wi-Fi hot-spot. Wireless technology with the different advance feature are the bare expectation for standards these days. These requirements bring forth the evolution and enhancement of IEEE 802.11 [49] Wi-Fi standards according to the technological and application specific need. From the total list of IEEE 802.11 [50] wireless network standards, which is the most publicly used, have evolved to meet these growing requirements in terms of features like extended channel binding, multi-user Multiple In Multiple Out (MIMO) [51] ability, wide range of modulation, beam-forming, MAC modification, coexistence mechanisms, throughput and much more [52].

University Wi-Fi network accommodate thousands of mobile and wireless devices being used on a campus network. Multiple features of network utilization such as device variation, bandwidth requirements, application variability, operating system variability, user authentication and data off-loading, can be observed in a campus network. Each university strives to provide the best network service to its users despite the challenge of

a fixed budget. To support the growing demand, it was necessary to upgrade the Wi-Fi network to improve coverage area, network speed, and throughput. The network team has taken two major steps for this enhancement. First, increase the number of access points (APs) to address the coverage gap. Second, upgrade the Wi-Fi network to 802.11 (ac) protocol to increase performance. In this chapter, we conduct an extensive measurement analysis of the Wi-Fi network condition throughout the access layer of a university campus network, the University of Missouri - Kansas City (UMKC). We found some interesting phenomena like the variety of user density in the different locations with diversified bandwidth demands, lacking coverage in some area resulting in coverage gap, data rate degradation in certain points, and protocol compatibility issues. Figure 21 represents the signal strength and coverage map of a floor in a campus building prior to the upgrade. Furthermore, our observation indicates that between December 2016 and October 2017, the network's traffic (upstream and downstream) has increased by up to 20%, even though the number of clients has decreased, as illustrated in Figure 22. The remainder of the chapter is organized as follows. Section 4.2 discusses the related work. Section 4.3 outlines the deployment consideration and design. Section 4.4 presents our analysis and evaluation. We summarize the chapter in Section 4.5.

## **4.2 Related Work**

In this section, we first discuss the related studies that have been carried out using the various approaches for High-density [53] Wi-Fi, Wi-Fi management, and security. We then discuss our work in the context of those studies. In their work, Zhu et al., propose a

user-centric network management framework to optimize the throughput of users operating in the high-density WLANs taking into consideration the network conditions sensed by users and their access priorities. Their proposed framework is built around an information pipeline that facilitates the sharing of the information needed for optimal management of communication resources. Their theoretical analysis and simulations were presented on two management activities: AP association and channel selection. From their finding, they demonstrated that their proposed user-centric network management framework significantly outperformed the traditional network management framework in the high-density deployment environment [54]. Furthermore, inspired by cloud-RAN, [55] proposed Amorphous Wi-Fi (AmorFi), a new way of deploying WLANs to handle peak traffic demands with average-case provisioning. Their fundamental idea is to decouple base-band processing from RF transmission using the algorithm and introduce software programmability to flexibly allocate Wi-Fi capacity in real time based on varying traffic demands.

#### 4.2.1 Wi-Fi Management

The challenge to provide seamless mobility emerges as a key topic in various standardization bodies as explored by [56] discusses the support of seamless handover between homogeneous networks. The continued effort in pursuit of gigabit wireless communications has been most noticeable in the IEEE 802.11 WLAN [57] in recent years. In 2010, the Wireless Gigabit (WiGig) Alliance, formed by a consortium of industry leaders,

has completed the defined first draft of a unified architecture to enable tri-band communications over the frequency bands of 2.4, 5, and 60 GHz in their WiGig specification [58]. The WiGig specification, which aims to achieve multi-gigabit wireless communication in the 60 GHz band, has since been contributed to the new 802.11ad amendment building on the existing 802.11 standards where interoperability with the 2.4 and 5 GHz bands are based on the existing 802.11b/a/g/n and the upcoming 802.11ac standards. A Dartmouth College study done by [59] performed analysis of 802.11 WLAN data for 550 access points and 7000 users over seventeen weeks. They employed several measurements techniques like Syslog messages, telephone records, SNMP polling and tcpdump packet captures and found out that user heterogeneity characteristics regarding wireless network usages and user mobility with increased number of user, P2P application, streaming data and AP utilization.

#### 4.2.2 Wi-Fi Performance

Wi-Fi network supports varying levels of performance, depending on the technology standard. In their research [60] introduce the key mandatory and optional PHY features, as well as the MAC enhancements of 802.11ac over the existing 802.11n standard in the evolution towards higher data rates. They compare the MAC performance between 802.11ac and 802.11n over three different frame aggregation mechanisms, viz., aggregate MAC service data unit (A-MSDU), aggregate MAC protocol data unit (A-MPDU), and hybrid A-MSDU/A-MPDU aggregation through numerical analysis and simulations. Their results showed that 802.11ac with a configuration of 80MHz and single (two) spatial

stream(s) outperforms 802.11n with a configuration of 40 MHz and two spatial streams in terms of maximum throughput by 28% (84%). Furthermore, they showed that hybrid A-MSDU/A-MPDU aggregation yields the best performance for both 802.11n and 802.11ac devices, and its improvement is a function of the maximum A-MSDU size.

### 4.2.3 Wi-Fi Security

Security has become an essential measure in wireless data communication. Research revealing vulnerabilities and weaknesses of WEP protocol which is used in IEEE 802.11b has been done by [61]. They reveal the major issue of WEP protocol as lack of a proper fundamental management technique and propose a method to overcome the security by introducing a dynamic key for authentication and data transmission on per data frame basis.

Our research differs from these authors; we focused on the design and implementation of long-term incremental Wi-Fi upgrade and analyze the different issues discovered during the upgrade. For example, coverage area gap, bandwidth overload, and association failures were uncovered and addressed during the upgrade. Additionally, we explore roaming issue addressed by the upgrade.

## 4.3 Wi-Fi Deployment Design

Incremental deployments were done over four years period. How to design heterogeneous compatibility, data rate, coverage, and performance were factored into the deployment plan and roll-out schedules [62].

Table 10: Wireless Data Sets Used

Data Source	Data Size	Duration
Syslog	15.3 billion records	Jan 2014 - Jan 2018
Cisco Prime	(14 categories)/(126 items) report	Oct 2016 - Jan 2018
Ekahau Survey Map	41 Buildings Blue Print & Signals of 1137 Access Points	Jan 2014 - Jan 2018

Table 11: Data Source and Tools used for Analyzing Coverage, Authentication and Bandwidth Issues

Issues	Coverage Gaps	Authentication Failure	BW Overload
Analysis	Ekahau	Splunk & Syslog	Cisco Prime & WLC
Data Sources	Survey Map & Syslog	Netflow	Radius Logs
Tools	Ekahau & Splunk	Cisco Prime	Ekahau

#### 4.3.1 Analysis Tool

Wireless devices need to be compatible with most of the wireless protocols and decide to select the best channel and wireless protocol based on wide range of parameters. The upgrade had to be backward compatible to accommodate heterogeneous devices [63]. The University Information Services department use network log events with Splunk and Ekahau Spectrum Analyzers to locate the coverage gaps through out the campus Wi-Fi network. There were many errors from re-authentication failure, and co-channel interference due to the wide range of the 2.4 GHz APs. Additionally, Wi-Fi network faced bandwidth overload in most of the 2.4 GHz APs due to the extensive coverage area which reaches too many clients far more than the APs' capacity.

Deploying an upgrade to replace the older APs which supported 802.11 (b/g/n) [64]

Table 12: 802.11n vs. 802.11ac Wireless Networking Protocols

Features	IEEE 802.11 n	IEEE 802.11 ac
Frequency Band	2.4 GHz & 5 GHz	5 GHz only
Channel BW	20, 40 MHz	20, 40, 80 MHz 160 MHz optional
Spatial Streams	1 - 4	1 - 8 total up to 4 per client
Multi-user MIMO	No	Yes
Single Stream [1 x 1] Maximum Client Data Rate	150 Mbps	450 Mbps
Three Streams [3 x 3] Maximum Client Data Rate	450 Mbps	1.3 Gbps

Table 13: 802.11a vs 802.11b vs 802.11g Wireless Networking Protocols

Features	IEEE 802.11 a	IEEE 802.11 b	IEEE 802.11 g
Frequency Band	5 GHz and 3.7 GHz	2.4 GHz	2.4 GHz
Channel BW	20 MHz	22 MHz	20 MHz
Spatial Streams	N/A	1 - 8 total N/A	N/A
Multi-user MIMO	No	No	No
Single Stream [1 x 1] Maximum Client Data Rate	54 Mbps	11 Mbps	54 Mbps
Three Stream [3 x 3] Maximum Client Data Rate	54 Mbps	11 Mbps	54 Mbps

with newer APs that supports both 802.11 (b/g/n/ac) and the adaptive 802.11 (r) [65] with FT feature would address these rising concerns. Table 13 and Table 12 illustrates the comparisons of a/b/g and a/ac. There are several tools used for the analysis of the upgrade and the different issues resolved by the upgrade as listed in Table 11. Table 10 discusses the data set used for our analysis. We had 15.3 billion records generated by Syslog server over four years period. The second data source is Cisco Prime consisting of 1 hr, 1 week, 1 month and/or 1 year reports data from 14 different Categories, Autonomous AP, Clean Air, Clients, Compliance, Composite, Device, Guests, Identity Service Engin, Mesh, Network Summary, Performance, Raw NetFlow, Security and System Monitoring. There are 126 items reports under these categories.

#### 4.3.2 Channel Planning and Band Select Feature

Several questions came to mind while planning the Wi-Fi protocol. What channels do the heterogeneous devices in the campus network support? Many older devices may not support the Extended channels and when that happens, the upgrade should only be providing redundant coverage. What is the student capacity in each class rooms? High density deployments were needed in some of the large auditoriums, which required increasing the number of APs in addition to the Wi-Fi upgrade. Consideration was given for APs to be allocated where coverage is optimized with minimum number of APs. Clients inside a pair of APs need to be within coverage distance from at least one of APs. The 2.4 GHz band covers a larger range than 5 GHz and is commonly used on the campus network. Cisco APs by default send quicker probe response from 2.4 GHz band, resulting



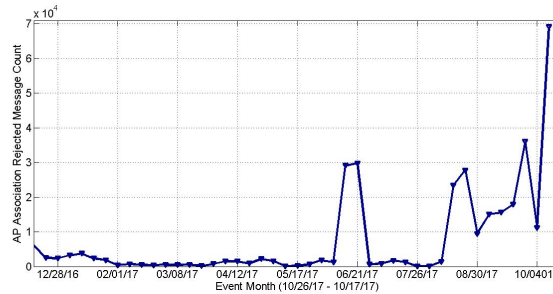


Figure 23: Reject New Association Due to Maximum Client Limit Reached Message Count

in more connection to 2.4 GHz band until the AP is overloaded and can no longer accept new association requests. Our campus Wi-Fi network experienced a surge in the "Reject New Association" due to maximum client limit reached in AP Radio during the month of June 2017 and between August - October 2017 as illustrated by Figure 23. To overcome this crowding issue, Cisco introduced a feature called 'Band Select' [66], which allows dual-band clients to prefer the 5 GHz band over the 2.4 GHz. Band selection works by regulating probe responses to clients by making 5 GHz channels more attractive to dual-band clients from delaying 2.4 GHz probe responses. For example, the number of device association that is higher than N (our network set N to be 12) will have 2.4 GHz response suppressed M (our network set M to be 4) times before allowing the clients to receive Probe Response from 2.4 GHz wireless band. Algorithm 5 illustrates the band redirection. This may cause a small delay for non-dual band 2.4 GHz only devices, however, will alleviate the crowding issue of 2.4 GHz wireless band and send dual-band clients to 5 GHz wireless band creating a more balanced network.

---

**Algorithm 5** Redirect 2.4 GHz Requests  $> N$  to 5 GHz

---

**Input:** 2.4 GHz Prob Request**Output:** 5 GHz Association for dual-band clients

```
1: Initialize
2: attempt_count = 0
3: for all MACID  $\in$  queue do
4:   ap_count = 2.4 GHz Prob Request Count
5:   if ap_count  $> N$  and attempt_count  $< M$  then
6:     suppress 2.4 GHz prob request
7:     send to 5 GHz prob response
8:     attempt_count += 1
9:   else
10:    device is not dual band enabled
11:    send 2.4 GHz prob response
12:   end if
13:   // devices with dual band capability will be switched to 5 GHz
14: end for
```

---

### 4.3.3 Addressing Roaming Issue

Wi-Fi 802.11 (ac) has the fastest data rate, 1300 Megabits per second (Mbps) and compared to 802.11 (n) [50] typically 450 Mbps, it is 3x faster. Fast Transition(FT) [67], a feature of 802.11 (ac), makes roaming between two adjacent 5 GHz band seamless. However, if a client roams outside of 802.11 (ac) or even between two adjacent 5 GHz band and 2.4 GHz band, there is bound to be a delay. FT feature, included in the Wi-Fi deployment, permit continuous connectivity aboard wireless devices in motion, with fast and secure hand-offs from one base station to another while the client is roaming. FT will not be explored in this work. However, it will be scoped as part of future works.

#### 4.3.4 Managing Coverage Gap

It is part of the design process to locate the holes in the existing Wi-Fi coverage area in order to identify the precise locations to place the new APs to provide the ultimate coverage. The initial installation had the APs stacked in one location of the building providing inefficiency and weak coverage to no coverage outside of the stacked APs coverage area. The current design fans out these APs to maximize the coverage area with minimum number of APs. Additional APs were placed throughout the campus locations to manage the load capacity for high-density areas.

Table 14: 802.11 (ac) Wireless Upgrade Summary

Model	Desc	AP Count
1810	802.11 (ac) Wave 2 MU-MIMO	735
3802 I	decisions based on Wave 1 end-device activities	65
3702	Wave 1 150 Mbps	199
3602 Radio	dual-band 2.4/5-GHz - Wave 1 integrated radios	130

#### 4.3.5 Wi-Fi Upgrade Summary

The incremental Wi-Fi upgrade targeted the wireless infrastructure of several buildings including the university dormitories, and an off-campus network. There were 1187 APs upgraded to 802.11 (ac). Several of the old APs were replaced by the new APs with updated features. Majority of the APs (735), had Model 1810 Wave 2 with feature MU-MIMO[51]. MU-MIMO provides concurrent downstream communications to multiple

wireless devices allowing client devices to get on and off the network faster, enabling more clients to use the network. 199 of the APs were with Model 3702 Wave 1 dual-band radios with a data rate of 150 Mbps. Table 14 lists the model and the number of AP which were deployed over the summer/winter weeks targeting minimal impact to clients.

## **4.4 Evaluations**

In this section we perform pre-and-post upgrade evaluation using different network performance analysis tools. Cisco Prime Infrastructure reporting [68] is a tools used by network team to help monitor the system and network health and is used for troubleshooting network problems. To evaluate the sheer volume of Syslog messages generated from the Wi-Fi network during the upgrade period between 2015 and 2018, we use Big Data analytic tool named Splunk. We used Ekahau Wi-Fi Site Survey tool to capture the coverage map before and after the deployment. We used Wireshark to capture the packets of the roaming clients and FT authentications. We also used Wireless LAN Controller (WLC) and Radius logs to gather the Wi-Fi traffic activities.

### **4.4.1 Campus Network Setup**

The campus network is hierarchical with three layers which is a common practice for campus or enterprise networks, illustrated by Figure 24. UMKC network's core routing depends mainly on two core routers, one as primary and the second as fail-over and load-balance. There is a third core router that is dedicated to the University housing and eduroam network. The second or middle layer is the distribution layer, connects the core layer (using routers) to the access layer (using switches). The lower layer includes all

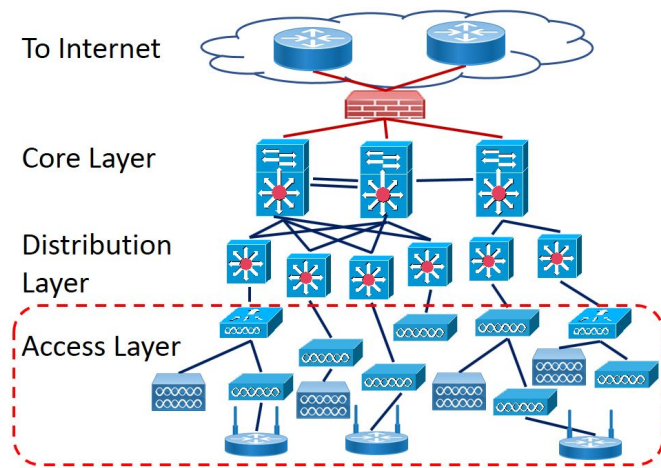


Figure 24: Campus Network Architecture with Three Layers: Core, Distribution, and Access Layers

the edge switches, and APs which connect to the end users. The Wi-Fi network extends from our main university campus to nearby campus in downtown Kansas City, to Union Station of Kansas City, which is about 15 miles away. Within our university network, there are many networks, each serving a specific service. UMKCWPA is the main SSID for the UMKC wireless network. The eduroam is the network SSID for the educational network access for all the eduroam [69] parties. eduroam (education roaming) is an international roaming service for users in research, higher education and further education. It provides a single authentication access for all the mobile connectivity requirements of an institution across 78 countries in thousands of locations. The Wi-Fi network is primarily accessed by students, faculty, staff. Guest accounts use the guest network, which is dedicated for guests with very limited temporary access. The media network is the network that connects the media devices such as ROKUs [70], Firesticks [71], Chromecast [72],

and Digital Video Recorder (DVR) boxes.

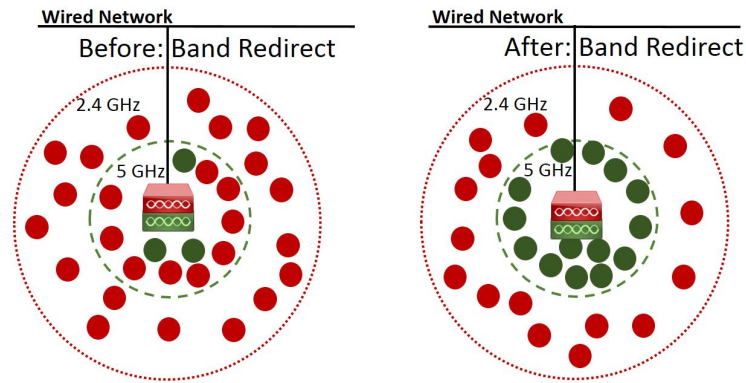


Figure 25: Impact of Band Redirection

#### 4.4.2 Client Redirection of 2.4 GHz to 5 GHz Band

5 GHz band offers a lot more space compared to 2.4 GHz, though it is not always fully utilized. Figure 25 illustrates authentication before and after the Band Select Feature suppresses client's association request to 2.4 GHz multiple times guiding the requests to 5 GHz band. We can see that "Before: Band Redirect" most clients were connecting to 2.4 GHz band by default. However, "After: Band Redirect", the dual band clients are guided toward the 5 GHz band. Our campus analysis indicated that there were more 5 GHz band authentications than 2.4 GHz as most laptops are only 5 GHz enabled. Between October 2016 and 2017 we observe the gradual decline in use of 2.4 GHz band and increase in 5 GHz band as illustrated by Figure 27. Each dot on the graph represents the aggregated authentication count for a week. The lower peaks are due to holiday seasons, such as season break where student attendance is low. Comparing the first five weeks of the graph

with the last five weeks, we can see that the client using 5 GHz band increased by up to 1150 count a week while clients using 2.4 GHz band decreased by up to 750 counts a week.

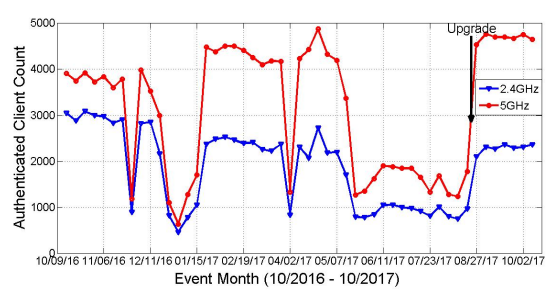
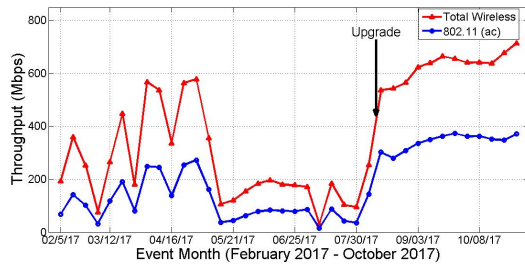


Figure 26: 802.11 total vs 802.11(ac) Throughput

Figure 27: 2.4 GHz vs 5 GHz Authentication

#### 4.4.3 Maximum Client Limit Reached

When the AP reaches the maximum number of clients per radio, the AP will reject new association and send out messages to the Syslog, "Maximum Client Limit Reached in AP Radio. Reject new Association". Prior to the incremental Wi-Fi upgrade in 2017 there were large number of rejections. After additional APs were installed in the campus network in addition to the band redirect from 2.4 GHz to 5 GHz to balance the client load on the APs, those numbers reduced significantly improving the Wi-Fi network, as illustrated by Figure 28.

#### 4.4.4 Throughput: Total 802.11 vs 802.11 (ac)

There were several activities during the last year that impacted the throughput to fluctuate during the Spring semester. The week of February 27th the WiFi network faced

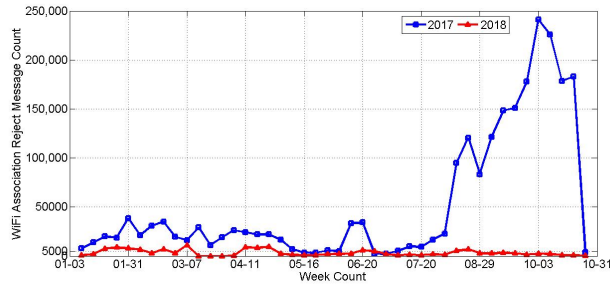


Figure 28: Count of AP New Association Rejected between 2017 and 2018

brief outage causing a dip in the traffic report. However, the controller/fail-over instantly corrected itself and traffic resumed. The week of April 2nd following spring break, our campus network upgraded to a new router resulting in short interruption. The third dip during April 23rd was due to performance issue which caused the router to reboot. Total Wireless network vs. 802.11 (ac) illustrated by Figure 26 shows the weekly aggregated throughput in Mbps(Mega bit per second) between February 2017 and October 2017, with an increase in throughput for both Wi-Fi 802.11 (ac) and Total wireless (Wi-Fi 802.11) between spring and fall semester. Each dot on the graph represents the value of aggregated throughput for one week. Shortly after the Wi-Fi upgrade, throughput increased for both Wi-Fi 802.11 as a whole and 802.11(ac).

#### 4.4.5 Coverage Gap

After the recent implementation in January 2018 which included installation of several new APs to provide coverage to the red zone, the coverage improved significantly. Figure 29 represents the initial coverage with 9 APs in 2014, where the majority of the



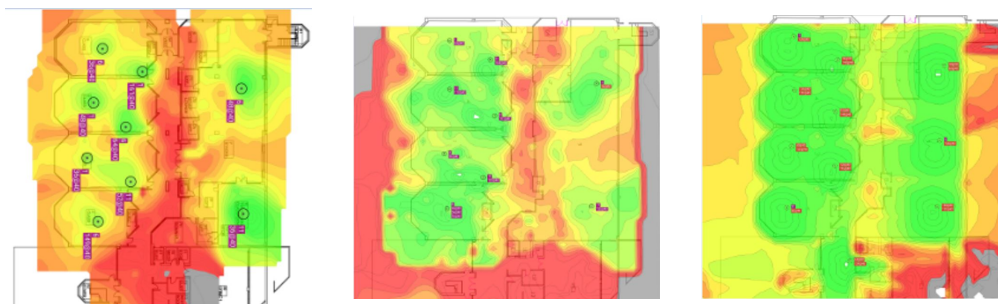


Figure 29: Wifi Coverage in 2014      Figure 30: Wifi Coverage in 2016      Figure 31: Wifi Coverage in 2018

area was red and orange indicating coverage gaps. Figure 30 illustrates the coverage map in 2016 after 2 additional APs were installed and coverage showed improvement. Figure 31, recent coverage representation January 2018, with total of 19 APs and majority of them were dual band high-density.

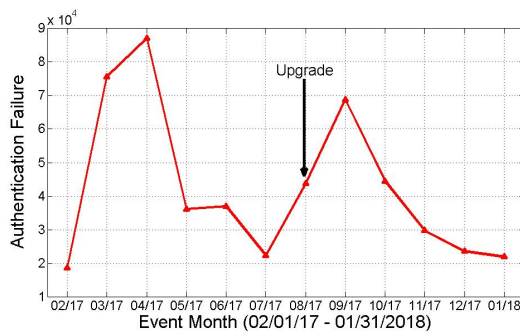


Figure 32: Authentication Failure Due to Roaming

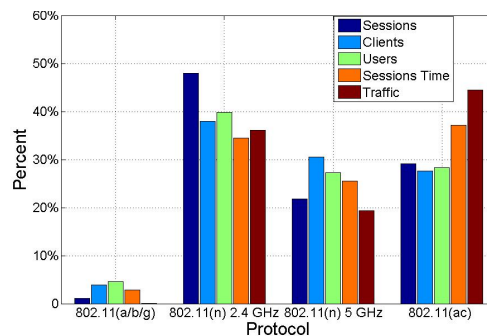


Figure 33: Usage Summary by Protocol

#### 4.4.6 Authentication Failure Due To Roaming

Figure 32 illustrates authentication failure due to roaming between February 2017 and January 2018. The highest failure event in the spring semester happened in April with the total of 87K messages. In the fall semester, the peak month was September total

failure message of 68.7K, showing a decrease of 18K compared to the spring semester. The upgraded feature enables fast and smooth roaming transition between 802.11 (ac), decreasing the authentication failure caused by roaming clients.

#### 4.4.7 Heterogeneous Protocols

Figure 33 illustrates the percentage of all the protocols used in the campus network over a 31 days period grouped by sessions, clients, users, session time and traffic. Although, the top three protocols were 802.11 (n) 2.4 GHz, 802.11 (ac) and 802.11 (n) 5 GHz in respective order. Protocol 802.11 (ac) shows the highest Session to Traffic ratio and the most Session Time with the least clients. Protocols 802.11 (a/b/g) are the least used protocol in the campus network. However, the campus network has to support these heterogeneous protocols to provide the environment for all devices needing Wi-Fi access.

### 4.5 Summary

In this research, we have conducted extensive analysis of heterogeneity issues encountered during an incremental campus Wi-Fi protocol deployment. We have explored various design considerations for heterogeneous compatibility, data rate, coverage, and performance. Furthermore, we discussed the issues around coverage gap, load balance, and roaming. This study of a campus incremental Wi-Fi deployment provides insights on the behaviors of Wi-Fi access network availability, and potential end-to-end expectations for high-speed and bandwidth-intensive clients. After the deployment, the most traffic is generated by 802.11 (ac) protocol. However, heterogeneous protocols such as 802.11 (a/b/g/n) still need to coexist until all devices are 802.11 (ac) compatible. In the future,

we plan to explore fast transition, performance enhancement, power saving, and time improvement benefits gained from upgrading to Wi-Fi 802.11 (ac) protocol.

## CHAPTER 5

### AGILE POLYMORPHIC SOFTWARE-DEFINED FOG COMPUTING PLATFORM FOR MOBILE WIRELESS CONTROLLERS AND SENSORS

#### **5.1 Background**

An array of traditional urban surveillance infrastructures, such as stationary video cameras and RFID sensors, become smarter when they are connected to central control systems by networks where Cloud services and applications can store, analyze, and control the collected data to help make various decisions. Furthermore, a recent integration of wireless and mobile computing technologies with the dramatic growth of smart devices enables a new type of crowd-based pervasive smart and mobile urban surveillance infrastructures. This opens up new opportunities for boosting the accuracy, efficiency, and productivity of uninterrupted target tracking and situational awareness. In order to ensure effective data collection, efficient information abstraction, and instant decision making, a fog computing [73] is used at the network edge that keeps data and their processing among heterogeneous smart mobile devices. However, a few challenging issues of building a smart device based fog computing system includes the device mobility and service heterogeneity controls.

Figure 34 illustrates an example of system architecture of fog computing paradigm for data driven, real-time Information, Surveillance, Target Acquisition, and Reconnaissance (ISTAR) [74] in an urban area sensing. Multiple sensor units, including satellites,

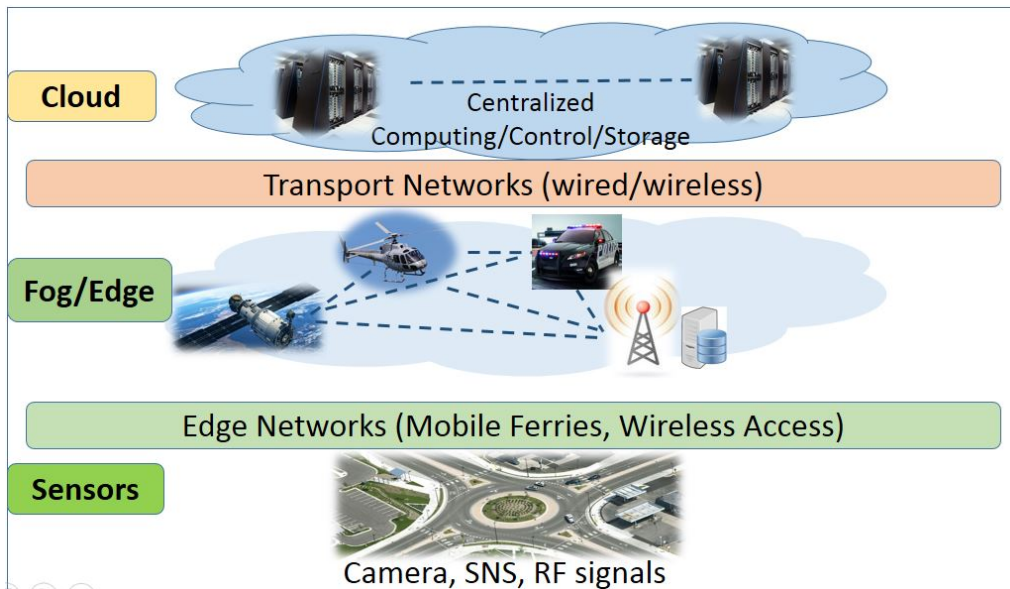


Figure 34: Urban Surveillance Architecture

UAVs [75], mobile robots, vehicles, social networks, and first responders are monitoring the area concurrently from different positions. When they are collecting real-time data streams, each of them will also need processed information for instant decision making. Although it is ideal that all the collected data are sent back to a central cloud facility for thorough global analysis, there is no guarantee that a reliable communication network to a remote cloud center is always available. In addition, not all data are globally significant and thus do not create necessary traffic in the networks. Instant on-site decision making also reduces the risk of exposing data to eavesdroppers in transmission channels. Just like a powerful centralized processing center, a well-equipped cloud may be separated by uncertainties. It is not agile enough to handle latency and connectivity sensitive tasks and it cannot guarantee reliable and secure communication services for many mission

critical situations. Therefore, a fog computing architecture, which consists of the computing devices carried by the units in or near the sensing area, can fulfill the requirements very well. Live event processing necessitates a service-oriented, dynamic data driven work model. Modern virtualization technology enables the fog to provide a uniform task-oriented, homogeneous computing platform on top of the heterogeneous computing and storage devices. In this paper, we present an efficient and effective fog system using software-defined control over a light weight mobile and wireless environment. We have built a reference fog system prototype using both Android- and Linux-based smart devices. We also designed a unified software-defined controller with role-based flexibility. Furthermore, we have identified and researched a few challenging issues of building a smart device-based fog computing system including the mobility control issues. Elastic node churn, dynamic controller churn, location-variability, aware applications, perishable and unpredictable service demands, and updating, predicting and maintaining the network latency and dynamic routes in different granularity as well as the softwarization issues (including cooperation among different east-west controllers, cross-layer collaboration, controller placement, mobility and latency control, controller scalability, and control reliability) were also identified in this research. We further investigate the algorithms and protocols and prototype them into mobile robots using a Linux based Raspberry Pi system (GoPiGo mobile platform). Applying the emerging softwarization technologies to the fog/edge networks and sensors, we improve the control agility and collaboration among groups of sensors and networks. We created a programmable and dependable software

defined sensor network and tested its usability including power consumption and system efficiency during the collaboration among the community of mobile sensor networks. The rest of the chapter is organized as follows. Section 5.2 discusses the related work. Section 5.3 describes our agile light-weight controller design and prototype. Section 5.4 presents a mobile sensor system and our experimental results. We summarize the chapter in Section 5.5.

## 5.2 Related Work

In this section, we first discuss the related works that have been done on the various approaches for Software Defined Sensors. We will then present our contributions with respect to these works. There are a number of research that have identified approaches for Software Defined Sensors. Some have focused on architecture and security while others have focused on sensor hardware. Fog Computing provides an efficient platform for a community of sensors by extending the Cloud Computing paradigm to the edge of the network, thus enabling quicker process time [76]. Fog Computing is essential for many lightweight devices to communicate directly with each other in real time. When considering mobility, most IP mobility solutions require a mobility anchor in the network, which is responsible for tracking the location of the mobile hosts and redirecting its traffic towards its current topological location. It is not possible to fully fit IP Mobility solutions into flat and decentralized environment. It may be desirable not to have a traffic re-routed from the previous point if the mobile host roams from one access point to another. The main problem in this case would be that the mobility anchor becomes the bottleneck of data

transmission for mobile users, which then greatly degrades and hinders network scalability and impacts user experience. Additionally, the mobility anchor would be susceptible as a single point of failure and attack possibly leading to the whole network failure. Furthermore, no route optimization is performed for mobile users. The proposed solution by [77] introduces new distributed IP Mobility management approach by using the SDN technique. Their approach is designed to avoid problems like the single point of failure and the lacking of route optimization in flat mobile network architecture. On one hand, While multiple controllers harmonize the distributed mobility management mechanisms, on the other hand, it alleviates the burden and increase the robustness of the centralized controller.

### 5.2.1 Sensor Hardware

Development of a flexible software-defined sensor architecture that enables distributed data collection in real-time over the Internet was presented in a recent finding by [78]. In this research the authors designed hardware sensor components and built low-cost commercial off-the-shelf sensors that may be useful in application areas such as dynamic spectrum access in cognitive radios. They implemented and evaluated different sensing strategies and noise reduction techniques. [79] addresses some of the practical and physical limitations associated with miniaturization of wireless sensor networks as the push for low cost, high performance, low power communication and computation wireless sensor networks increase. Sensor technology in automotive applications is used



to measure position, pressure, torque, exhaust temperature, angular rate, engine oil leakage, quality, flexible fuel composition, linear acceleration, night vision, speed, timing, long-range distance, short-range distance and ambient gas concentrations inside automobiles, and enhances their safety, security and reliability. the research by [80] used Inter-integrated circuit mode (I2C) software to communicate between sensors and embedded control system in their research. They assembled sensors which were built as part of a system-on-board that included sensors, a Q2 microcontroller and interface circuitry hardware, which in turn included a PIC18 (Peripheral Interface Controller) processor, a field programmable gate array (FPGA) chip and peripherals. The design featured compact, high-level integration, reliability, high precision and high-speed communication, low-power consumption, security and flexibility for expansion.

### 5.2.2 Sensor Architecture and Security

Mobility in wireless sensor networks was studied by [81]. Their work proposed a two-tier approach based on an algorithm of local interactions among sensors, on global tasks of mobile agents and on location predictions. A localization algorithm computes periodically the node's location. A slope is then computed based on the direction of the node. Once the slope of the movement is estimated, the location that a node will move to next is estimated in the same way. This algorithm is used for node location prediction. In the mobile agent algorithm, the nodes execute an algorithm that would allow the knowledge of their location at any given moment, with some error. The nodes then estimate their future positions and the slope of their trajectories and maintain a list of

the neighbours they have already reached. Their proposed scheme did not utilize routing protocols, and all location prediction and mobile agent decisions are of linear complexity. In their evaluation, performance was better as mobility degree and node density grew. The work in [82] exploits OpenFlow technology to address reliability in sensor networks where they unified an OpenFlow based sensor, which is called 'flow-sensor' and communication controller. Flow-sensor is more reliable compared to typical sensor because data packets, control packets, packet route and even the sensor nodes can be easily observed, regulated and routed whenever required. In their sensor network communication among controller, access point (gateway) and flow-sensor using OpenFlow protocol. Their goal was achieving increase reliability in OpenFlow messages (control packet) flow without interrupting data packet flows between flow-sensors using TCP/IP layered stack. Furthermore, achieving a robust routing between sensors and access point where route alternation is possible when required in both upstream and downstream directions would lead to increased reliability. In their simulated evaluation flow-sensor generated less number of packets than the typical sensor. Additionally simulation time seems to be increasing but at a slightly higher rate for typical sensor. Hence, flow-sensor displays far better performance where its OpenFlow architecture turns it to be a very much reliable. There has also been recent research on sensor nodes in SDS networks that provided insight on the discovery of dynamically reprogrammable sensors for different sensing tasks via the over-the-air-programming technique [83]. Furthermore, [84] in their proposed load balance architecture for data center and cloud stressed the need to move toward distributed controllers. Their research showed that depending on the traffic, the controller pool is

dynamically grown or shrunk and automatically balances the load across controllers ensuring good performance. The controllers are in communication with each-other about the switches connected to them. A switch connected to more than one controllers can be migrated to another controller by an established master/slave role. The migration of the role from slave to master or master to slave will happen in 4 different phases to minimize the possibility of packets loss from controller's role transition.

Our research is unique from other existing projects just discussed, as our does not utilize mobile IP. Light-Weight Controller builds a polymorphic Software-Defined Mobility (SDM) [85] architecture by importing the modularization algorithms and direct communication protocols. Focusing on the idea of creating a community of sensors, our proposed solution enables efficient and effective sensor collaboration and achieves energy efficiency, while the the previous research discussed mainly focused on the architecture and security of sensors and hardware. Applying a novel softwarization approach to the sensors, it enables a logically centralized sensor network management tool [86].

### **5.3 Light-Weight Controller in Software-Defined Network Design**

In this section, we present our Light-Weight Controller which utilizes a softwarization approach in the Fog environment to accomplish real-time urban surveillance tasks in support of uninterrupted, adaptive target tracking [87]. The key contribution of this work lies in Software-Defined Mobility (SDM), which efficiently and effectively integrates a Software-Defined Network (SDN) [88] at the edge of the networks to create

a unified and agile computing platform that integrates light-weight high device mobility [77] and heterogeneous hardwares and their functionalities.

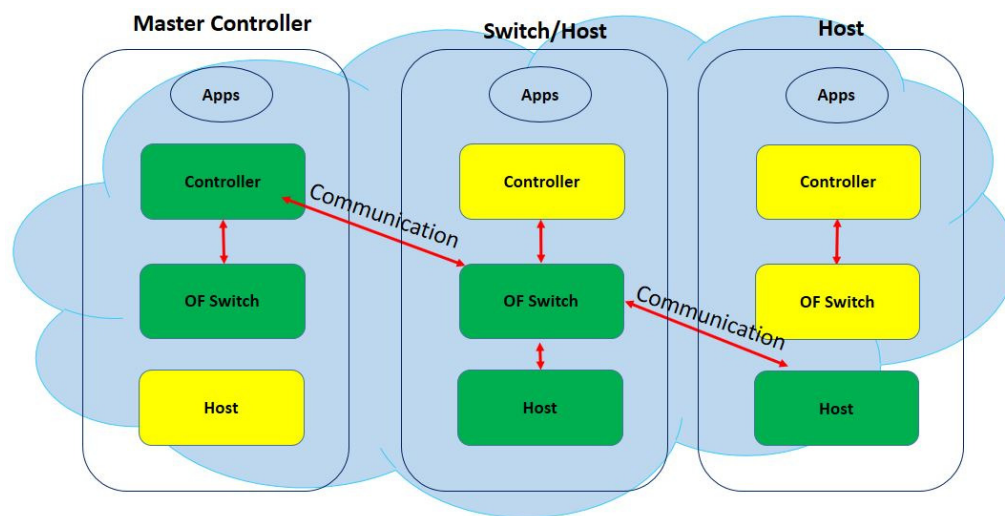


Figure 35: A Role-based Polymorphic System

### 5.3.1 Software-Defined Mobility (SDM) Architecture

We design SDM as a role-based polymorphic system. Algorithm 6 illustrates the process for determining a master controller. If a light-weight mobile device is the first in the network, then it will be assigned the highest available sequence number from the predefined pool, and that sequence number will be removed from the pool. The controller id of that device will be a master controller, and "I am a master" heartbeat message will be broadcast to the network to inform any other device that enters the network. Devices that join the network after the master controller has been established will request the next available sequence number and join the master controller as a slave. This approach allows

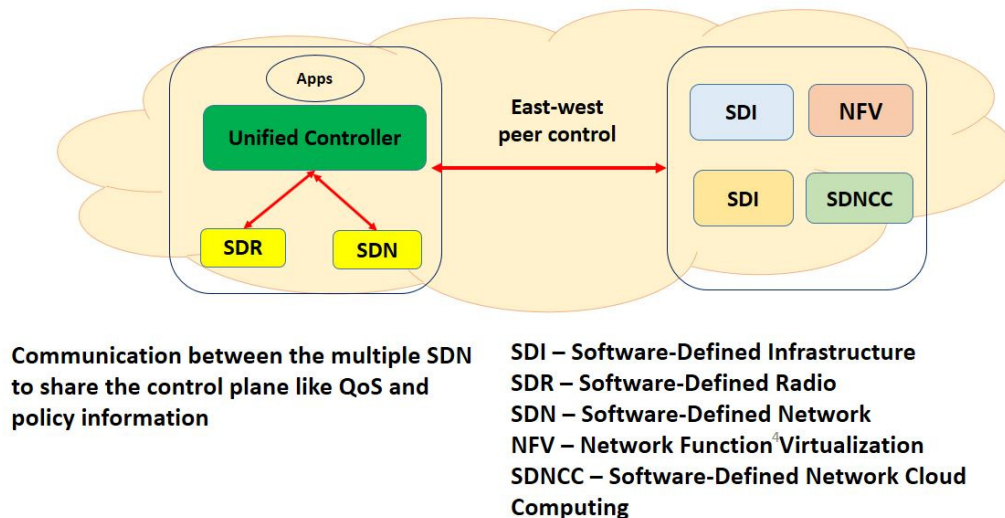


Figure 36: A Unified Controller for SDM

flexibility. Instead of designating a dedicated role such as controller, switch, or host for each smart mobile device, we make SDM adaptively change its role according to circumstances. As illustrated in Figure 35, each node has the capability of multiple functionalities as controller, switch, and host. A node becomes a controller for a group and a host becomes one for other groups at the same time. In this way, the system can provide the agility and reliability to the mobile system in response to frequent membership changes. However, the design decision of implementing all of the SDN functionalities within a node may impose a heavyweight system overhead onto a relatively resource limited mobile device [89]. To ensure a light-weight system design, we took an approach to use a library bundle instead of importing the existing SDN controllers and switches into SDM. We chose to use and modify a libfluid library. According to our initial evaluation on an

---

**Algorithm 6** Determine a Master Controller

---

**Input:** switchID**Output:** masterControllerID, seqNum

```
1: for all SwitchID  $\in$  queue do
2:   masterControllerID = getLeaderID(switchID)
3:   if masterControllerID = 0 then
4:     masterControllerID = switchID
5:     getSeqNumber(switchID)
6:     start LibFluid controller
7:     send the heartbeat messages
8:   else
9:     getSeqNumber(switchID)
10:    join the masterControllerID
11:  end if
12: end for
13: function GETLEADERID(switchID)
14:   if leaderID(switchID)Exists then
15:     return leaderID
16:   else
17:     return 0
18:   end if
19: end function
20: function GETSEQNUMBER(switchID)
21:   return the highestSeq
22:   remove the sequence from the pool
23: end function
```

---

Android smartphone, a controller and switch design of the combined size of the libfluid-based approach is an order of magnitude smaller than the size of any existing controller and switch combinations. Algorithm 7 illustrates the process of changing the master role. When a master controller is out of range or is disconnected from the network, the role is switched over to the next node with the highest sequence number. When there is no "I am master" heartbeat message received from the master controller for three consecutive seconds, the switches in the network will send inquiry message "who is the master" along

with their sequence number. The algorithm performs the prediction of the next master controller by looking at the sequence numbers from the nodes in the network and the node with the highest sequence number will establish the role as the new master. The new master will then start light-weight LibFluid controller and send out "I am a master" heartbeat message and other devices that are in the network will join the new master. If the old master comes back to network, it will be treated as new device and be required to get new sequence number and join the existing network as a slave with the newly established master. The process of establishing new master will start over again when the newly assigned master controller moves out of range or become unavailable. As illustrated in Figure 36, an SDM system orchestrates the softwarized controls within the system as well as other peer (east-west) systems under wireless environments. The controller provides several functions: select and manage software-defined control modes; coordinate frequency hopping and channel selection (as in Software-Defined Radio (SDR) with multi-hop, ad hoc, and peer-to-peer route selection i.e., SDN); resolve intermittent connectivity, congestion, and packet loss issues; facilitate a priority-based traffic reservation/emergency route; provide wireless network virtualization service (spatiotemporal); and adjust the power level of each wireless node. We design a dynamic task and data management architecture to cope with elastic controllers and node churn.

As shown in Figure 38, the software-defined control modes can be a Logically Centralized Control (LCC) [90], Legacy Distributed Control (LDC), or Partially Centralized Control (PCC). We design an algorithm to select an efficient control mode according to the elastic node churn, dynamic controller churn, location-variable and aware

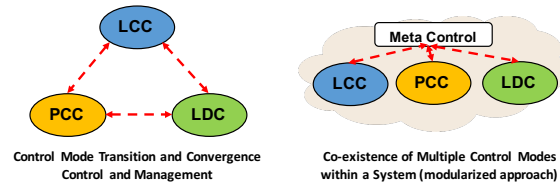


Figure 37: Control Mode Transitions and Co-Existence

applications, perishable and unpredictable service demand, and updating, predicting and maintaining the network latency and dynamic routes in a different granularity. We also investigate the mode transition, modularization, and convergence control approaches. As illustrated in Figure 37, during the transition the control mode can be further modularized into several different modes instead of a simple mode transition. According to the changes, there should be a facility designed to handle a dynamic resource provisioning and intra or inter-nodes task migration. We also investigate and design an intelligent direct node-to-node control protocol unlike the traditional centralized discovery protocols (LLDP, BDDP, OFDP) [91] that cannot provide the agile, reliable, and secure topology learning in mobile and wireless environments. Specifically, we investigate the Fog computing platform in terms of network protocols. There are a couple of main differences from the traditional communication platform. First, a large portion of the device-to-client (D2C) [92] and device-to-device (D2D) [93] communications is not the final target, because the final results need to be written back to the cloud server in a secure and consistent manner. Second, there are heterogeneous devices coexisting in the edge networks of the Fog, which may not use the same protocol to communicate.



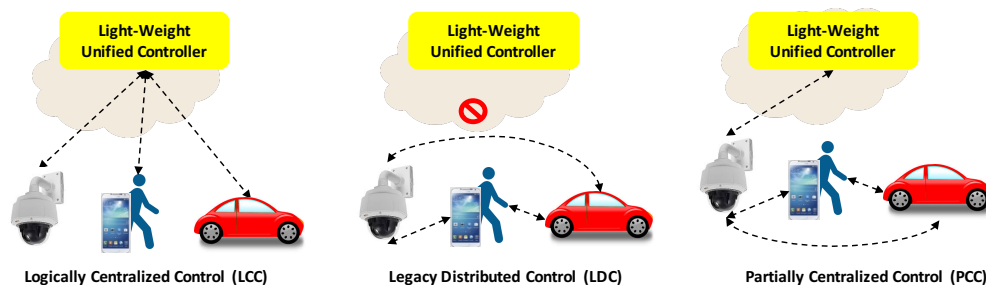


Figure 38: Types of Control Modes

### 5.3.2 Prototype

We have designed and built a proof-of-concept SDM architecture on both Android smartphones and Linux-based Raspberry Pi mobile robots using a libfluid library. We achieve the proposed role-based polymorphic SDM system by applying a light-weight libfluid library based controller and switch onto smart mobile devices. The initial work shows that the library approach is promising in the flexibility and scalability aspect. As an initial prototype, SDM is designed based upon a Logically Centralized Control (LCC) mode. We also develop a simple unified controller by using a beacon-based D2D communication protocol and adding a libfluid controller application. Algorithms 6 and 7 detail the mobility transition control and group management, and Figure 40 illustrates how the unified controller application can control both internal and external OpenFlow switches as well as communicate with external SDN controllers (both south-north and east-west controls) and internal software-defined controllers (such as SDR).

Our SDM is the first "libfluid" library [94] ported to an Android system. As illustrated in Figure 39, libfluid is a library bundle that provides the basic facilities to implement an OpenFlow controller. It consists of two main libraries including libfluid\_base and

libfluid\_msg, libfluid\_base are classes for creating an OpenFlow server (controller) that establishes connections to clients (OpenFlow switches), relays the client calls, and handles event callbacks for the applications. libfluid\_msg are classes for enabling applications to parse OpenFlow format messages. By integrating SDM into different mobile platforms, including Pi robots and Android smartphones, our solution facilitates the sound technical foundations for a practical deployment of a large-scale effective software-defined (mobile) Fog computing system as a part of the urban surveillance system.

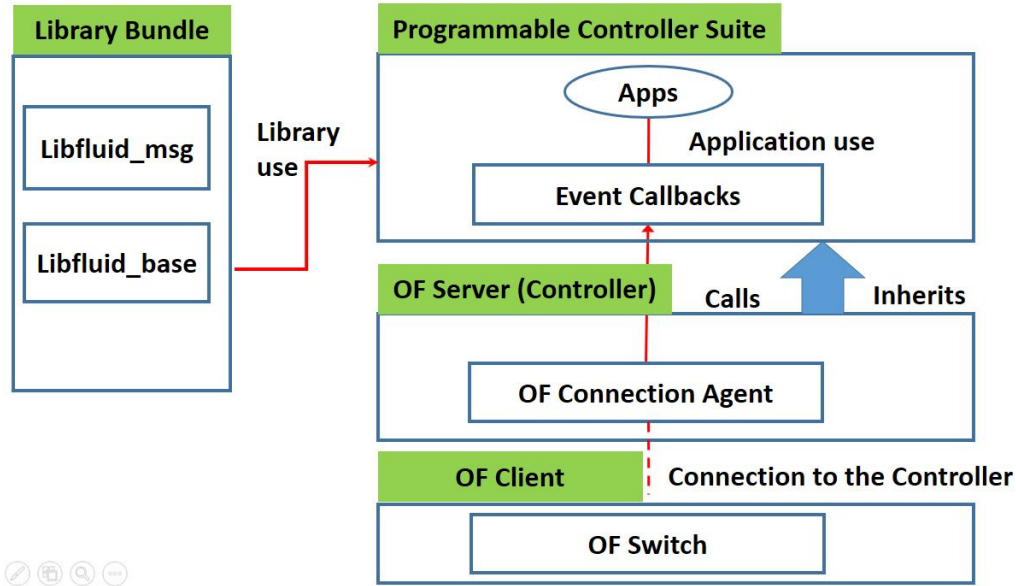


Figure 39: A libfluid Module Architecture

### 5.4 Evaluations

We have built a prototype of a role-based polymorphic Software-Defined Mobility (SDM) system architecture on both Android smartphones and Linux-based Raspberry

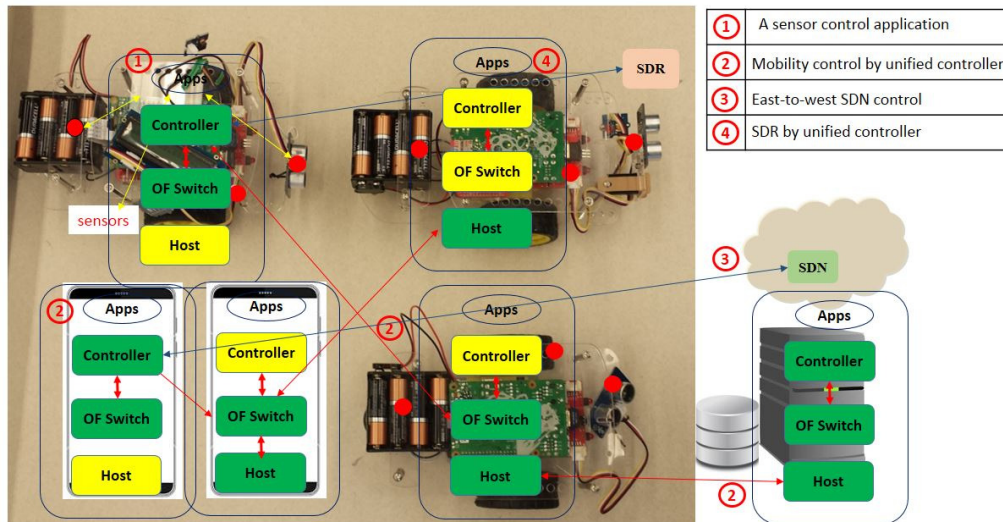


Figure 40: SDM Implementation on Smart Mobile Devices

Pi mobile robots using a libfluid library. Applying the softwarization technologies to networks, communications, smart devices, and sensors, we improve the control agility and collaboration among sensors. To evaluate the feasibility of the SDM system, we choose to tackle multiple mobile sensor scenarios similar to [95]. For example, small Unmanned Aerial Systems (UAS) [96] such as CubeSat, Small-Sat, drones, and mobile robots carry a variety of mission critical sensors [97]. However, current sensor systems are not optimized in their SWaP (Size, Weight, and Power) and are not designed to collaborate efficiently within a system [98] or among the swarm of UAS [99]. Sensors are often redundant, occupy an inconvenient amount of space, consume power inefficiently, and add undesirable weight [100]. Fog computing with SDM can optimize the cost, and facilitates low SWaP sensors (less redundant, more reliable, and less power) in the system. Specifically, we evaluate the SDM system performance using metrics such as the number

of active sensors, device power usage, and processing time overhead.

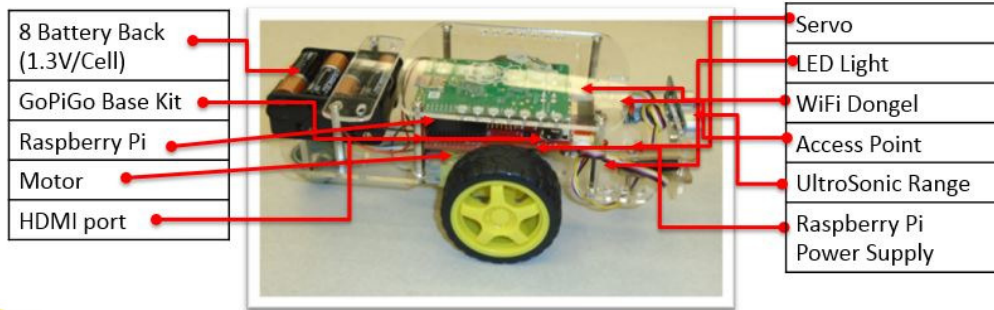


Figure 41: The GoPiGo Prototype

#### 5.4.1 Experimental Setup

We deploy a libfluid based SDM system over three Raspberry Pi mobile robots (GoPiGo) [101] and a couple of Android smartphones. Among the sensors illustrated in Figure 41, we use UltraSonic range sensors. The UltraSonic range sensor features include input voltage of 5V with a current draw of 20mA maximum. The digital output is 5V with 0V low and the frequency is 40kHz. The distance range of the UltraSonic is between 2cm to 400 cm. We develop an application using Python script to customize the network topology and control the mobility of the GoPiGo devices as well as manage the sensor community. The detail of sensor types and their power consumption is estimated in Table 15. Table 15 shows that a range sensor consumes 20mA power on average. The battery capacity is estimated as 5V for a AA cell. Total capacity is calculated by multiplying the individual cell ( $C_n$ ), Cell capacity ( $C_c$ ) and Cell voltage ( $C_v$ ) and dividing by 5V. USB power capacity is measured at 3.7V [102].

Table 15: Estimated Power Consumption

<b>Sensor Type</b>	<b>Power Consumption</b>
Camera	310mA
UltraSonic Range	20mA
Led	1.1uA
Temperature	45mA
Humidity	25mA
Sound	20mA

#### 5.4.2 Active Sensors

The experiment shows how many sensors can be turned off through the collaboration of sensors. In this experiment, we assume each car is equipped with 8 sensors (2 sensors in each direction). Sensors continuously detect events and generate data. As the sensors are not designed to collaborate with each other, for each individual car, the entire 8 sensors are actively used to detect surrounding events. However, as illustrated in Figure 42, when cars can collaborate with each other, one side of the sensors can be turned off or in a sleep mode and neighbor's car on the side can be detected by the neighbor's car sensors by receiving event messages. Theoretically, the more cars that are in a collaboration network, the greater number of sensors that can be turned off. Figure 43 shows the correlation between the number of cars in a collaboration network and the number of active sensors within a car. Given an environment where sensors can work together efficiently, we can see the benefit of collaboration. When there are 5 cars in a collaboration network, 8 sensors in total can be turned off (1.6 sensors for each car on average) in the best case scenarios.

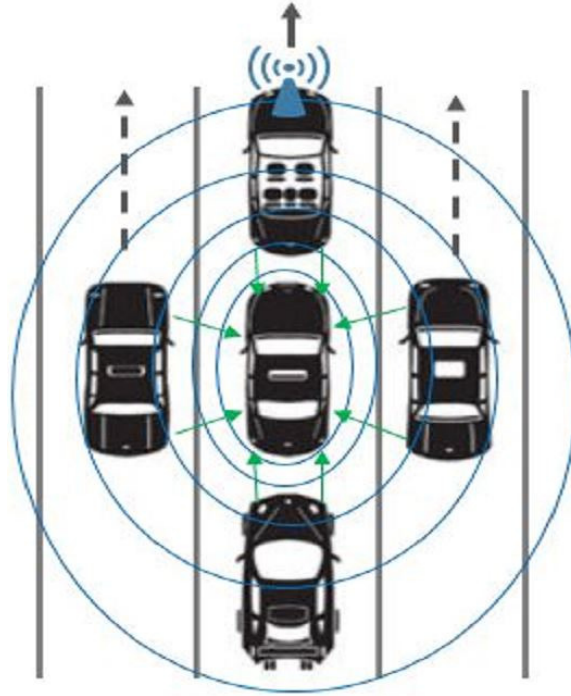


Figure 42: A Five Car Network

### 5.4.3 Power Consumption

The experiment shows how much power consumption can be saved through the collaboration of sensors. In this experiment, we make similar assumption that each car is equipped with 8 sensors (2 sensors in each direction). For each individual car, the entire 8 sensors are actively used to detect surrounding events. The more cars that are used, the more sensors are used in total and more power is consumed. However, as illustrated in Figure 42, when cars can collaborate with each other, one side of the sensors can be turned off to save power consumption and an event on the side can be detected by the neighbor's car sensors by receiving event messages. Theoretically, the more cars that are

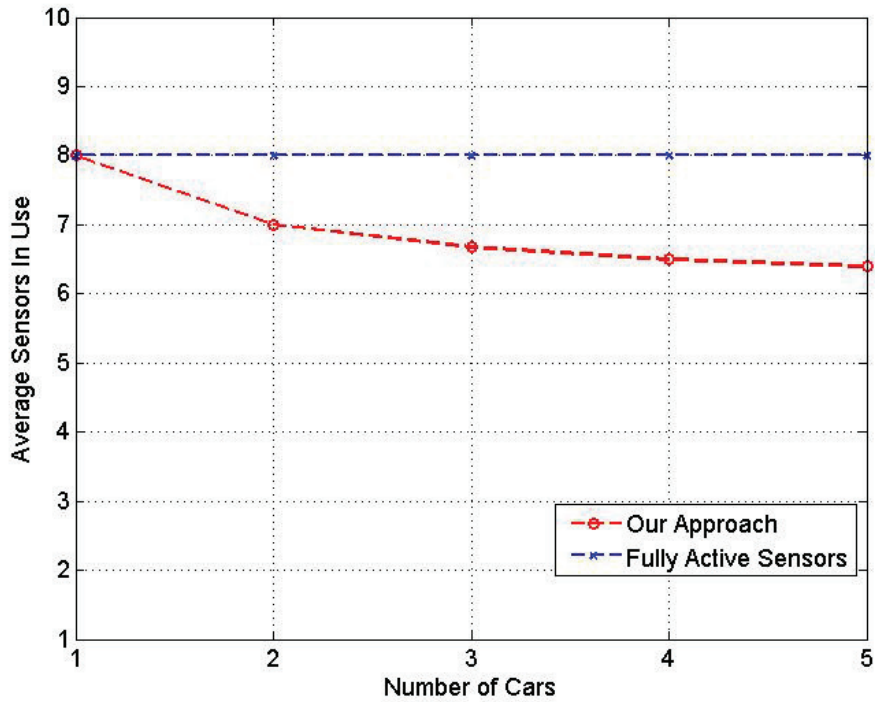


Figure 43: Average Number of Active Sensors

in a collaboration network, then the more sensors can be turned off. Figure 44 compares total power consumption by the cars between fully active sensors without collaboration and the best collaboration case of our approach. For example, when 5 cars are in collaboration, the best possible power saving is 100 watts. The more sensors are used in a car, the more power saving can be achieved.

#### 5.4.4 Processing Time Overhead

The experiment in Figure 45 shows a processing time difference between an individual local decision and a regional collaborative decision based on the sensor communications. When cars can collaborate with each other, one side of a car's sensors can be turned off and an event on the side can be detected by the neighbor's car sensors by receiving event messages. The X axis represents the elapsed time where a car with an active sensor sends a message for a 1 second interval and the Y axis shows the processing time. In this experiment, multiple cars are communicating with each other and we observed if the regional communication cause noticeable overhead or not. All the sensors in one sensor (GoPi 1) are enabled while only some of the sensors in other sensors (GoPi 2 and 3) are turned on. Event messages are sent from GoPi 1 to the other cars (GoPi 2 and GoPi 3) simultaneously. Upon receiving an event message, a car calculates the processing time. The current local time is an end time and the start time is found from the event message. The difference between them is a processing time. Overall, the processing time overhead was negligible and the overhead of processing due to the regional collaboration of sensors did not impact the sensing performance.

### 5.5 Summary

As more and more sensors are being increasingly and incrementally deployed, diverse networked sensor applications should embrace heterogeneity and mobility of various sensors through an efficient fog system. We proposed an agile, dynamic and lean software-defined network framework for mobile sensor applications. We also developed



a proof-of-concept prototype of fog system for nodes with various and dynamic roles. We discussed our algorithm for selection of a master controller and switchover of the master controller role when a master controller is no longer available within the network. In our evaluation, we have shown its feasibility and efficiency of mobile sensor collaboration under various scenarios. To the best of our knowledge, this work is the first to propose and develop a fog computing framework that supports agile architecture of Software-Defined Network for mobile smart devices. We believe with the proliferation of cyber-physical systems and their dynamic nature, our proposed agile and unifying platform will be beneficial to deploying sensors and controllers of diverse and dynamic roles.

---

**Algorithm 7** Master Heartbeat and Role Switchover

---

**Input:** switchID, seq\_num, heartbeat**Output:** new\_master

```
1: for all switchID  $\in$  queue do
2:   first_time = 1
3:   if first_time = 1 then
4:     set timer = 0
5:     no_heartbeat = 0
6:     seqTable = 0
7:     first_time = 0
8:   end if
9:   if Exists heartbeat then
10:    Master is alive and in range
11:    timer =timer + 1
12:  else
13:    Master is out of range
14:    no_heartbeat = no_heartbeat + 1
15:    for all seq_num  $\in$  switchID do
16:      if seqTable <seq_num then
17:        set seqTable = seq_num
18:      else
19:        Next
20:      end if
21:    end for
22:  end if
23:  if no_heartbeat = 3 and switchID = seqTable then
24:    switchID = new_master
25:    start LibFluid controller
26:    send the heartbeat messages
27:  else
28:    join the the new_master
29:  end if
30:  sleep for 1 second
31: end for
```

---

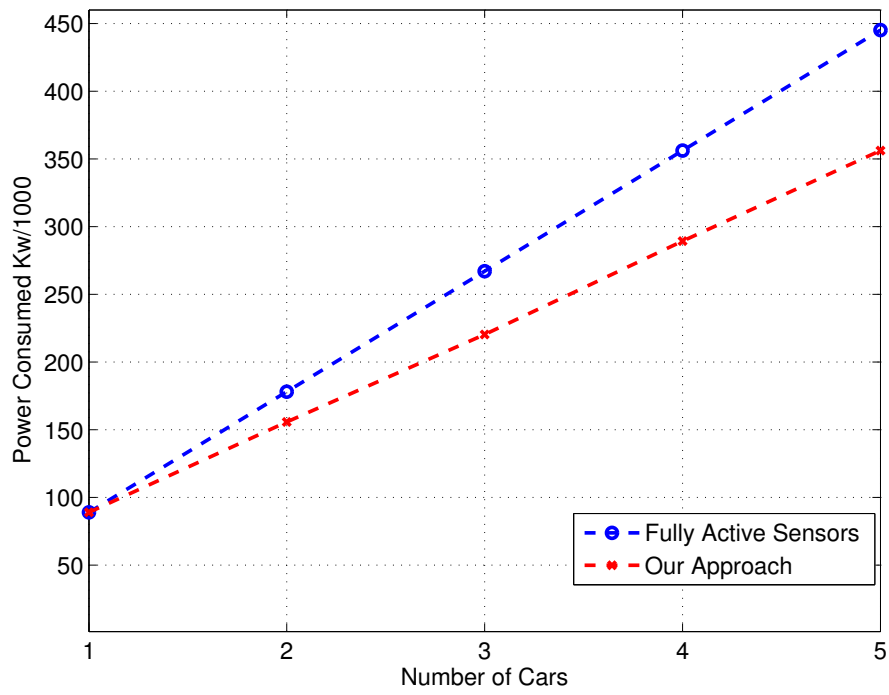


Figure 44: Power Usage

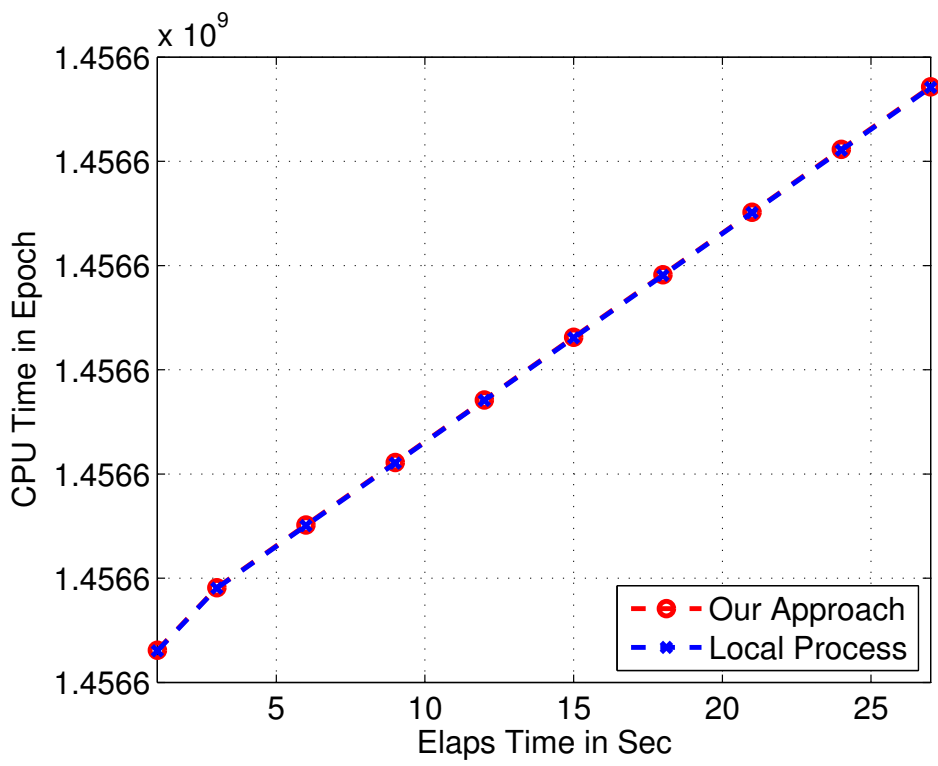


Figure 45: Processing Time Overhead in Seconds

## CHAPTER 6

### COST OF UNPLANNED NETWORK OUTAGE AND SLA VERIFICATION

#### **6.1 Background**

Internet is massive intricately connected network and require high-skilled personnel to (re)configure and (re)install devices due to the network's complexity. It also requires additional cost to add, remove or move devices from a network with multiple switches, routers, etc. These changes have cascading impacts to Access Control Lists (ACLs), Virtual Local Area Networks (VLANs) and some other network connections. Additionally, the crosslinking of multiple networks allows a simple node or link failure to propagate and be felt worldwide. Global networking takes full credit for the new trend of doing business. For example, in recent years remote work and off-shoring are part of doing businesses. While this improves how businesses operate, it opens another possibility of risk from the rippling effect that would be felt globally from a simple network failure. Cascading failure is the usual mechanism by which failures propagate to cause larger impact and occur commonly in congested complex networks, where it may be expressed as the process of generation, diffusion and dissipation of congestion.

Network service provider use Service Level Agreement (SLA), a time base contract between a service provider and the end user that defines the level of service expected from the service provider. In most cases these SLAs are one sided with service provider

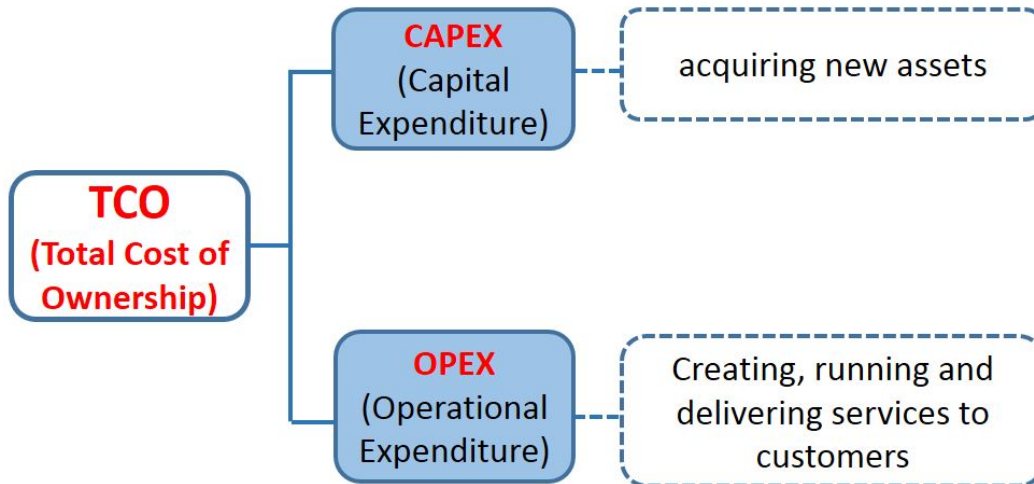


Figure 46: Total Cost of Ownership (TCO)

making promise to end user (customer) the level of service it plans to provide. The customer has no say in the contract or have a way to ensure the service is within the SLA's promise. For example, Amazon had service disruption that occurred in the Northern Virginia (US-EAST-1) Region on the morning of February 28th, 2017. This disruption impacted a subsystem which was necessary to serve all GET, LIST, PUT, and DELETE requests. This outage impacted 148,213 websites and 121,761 unique domains including our university. However, SLA was not impacted on this disruption because it was based on application's availability rather than network availability. Most SLA are time based and has no cost associated to it. On the contrary, The Wall Street Journal reported the outage, "cost companies in the S&P 500 index \$150 million, according to Cyence Inc., a startup that specializes in estimating cyber-risks. Apica Inc., a website-monitoring company, said 54 of the Internet's top 100 retailers saw website performance slow by 20% or

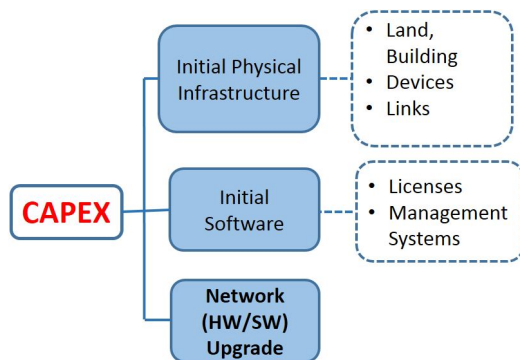


Figure 47: Capital Expenditure (CapEx)

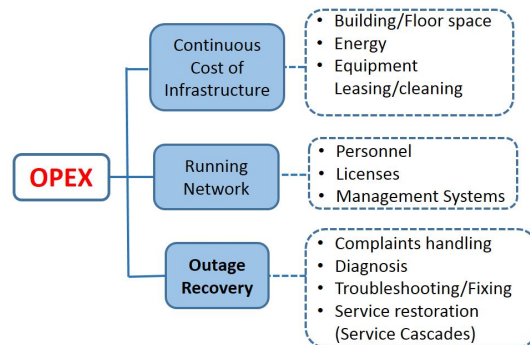


Figure 48: Network Outage Operational Expenditure (OpEx)

more” [103].

It is important to understand what the cost of network planning, design and implementing is to estimate cost as accurately as possible when it comes to decisions. Such estimations will enable a trade-off between the required availability of the network and the associated cost. Equipment cost model is used to estimate capital expenditure (CapEx) costs. Operational expenditure (OpEx) costs is associated with continuous running of the network. CapEx mainly contributes to the fixed infrastructure of a company, and they depreciate over time [104]. They include the purchase of land and buildings (e.g., to house the personnel), network infrastructure (e.g., optical fiber and IP routers), and software (e.g., network management system). Buying equipment has always been considered part of CapEx, regardless of whether the payment is made all at once or spread over time. Additionally, interest paid for a loan is included in CapEx. Land and building which is composed of network land/building and personnel building contribute toward CapEx as illustrated by 47. OpEx represents the cost of keeping the company operational and include costs of technical and commercial operations, administration, etc. The

majority contributors to OpEx for network service providers can be classified into three major categories: the portion directly related to continues cost of infrastructure, running the network and outage recovery. operating an existing network (which has already been set up), equipment installation, and some general expenditure (aspects not specific to a network operator). Although, Miscellaneous OpeEx such as payroll, talent management, cost of infrastructure heating, building cleaning and administration costs are not included in the figure, they contribute toward OpEx. In addition to OpEx and CapEx, Total Cost of Ownership (TOC) illustrated by Figure 46 includes service providers factor in service penetrations, population density per geographic area and service demand per user to determine the benefit of setting up network.

In this chapter, we present a measurement study to bring forth the various unplanned network outage issues faced on a university campus network. We discuss the type of the network outage and the impact to the SLA. In addition, we perform evaluations using different network performance analysis tools. This study will shed light on unplanned network outage issues, as these will become applicable with the increasing prevalence of large metro area wireless networks. The rest of the chapter is organized as follows. Section 6.2 discusses the related work. Section 6.3 describes our contribution, Cascading Failure Framework (CAFF). Section 6.4 presents a our evaluation and analysis. We summarize the chapter in Section 6.5.



## 6.2 Related Work

In this section, we first discuss the related works that have been done on the various cost analysis of network expenditure, network failure and cascading network failure. We then present our contributions with respect to these works.

### 6.2.1 Network Expenditure

The study [105] present unit cost for a service with Quality of Service (QoS) parameters. They characterize the unit cost for a service with respect to CapEx, OpEx, and workload of a network during certain period. They also investigate the relationship between the unit cost of a service and scalability of a network. They apply the proposed scheme in Software Defined Network (SDN) based architectures: centralized single controller architecture, distributed controllers' architecture, and hierarchic controller's architecture. In their experiments they reveal that there is an inverse relation between unit cost of a service and control plane scalability of the architectures: more scalable architectures result in lower unit costs for services. Furthermore, the research Hernandez-Valencia et. al. [106] provides a view into the operational costs of a typical service provider and discusses how the NFV/SDN attributes can be expected to influence the business equation. The direction and drivers of OpEx change, and the different categories of OpEx most affected, based on our analysis from interactions with many service providers worldwide, are also presented in their structured analysis. While their work describes normal network operating cost, we chose to mainly focus on the cost of network outage and the cost of cascading network failure.

### 6.2.2 Network Failure and Dependability

The study Schrank and Whitford [107] support a theory of "network failure" analogous to more familiar theories of organizational and market failure already prevalent in the literature on economic governance. In their research, they entertain a range of new questions of substantial theoretical and practical importance. When are relative network failures remediable? When do they give way to devolution? And why? Do institutions like legal systems, corporate governance arrangements, and training regimes influence the quality as well as the quantity of network governance? Does national culture play a role? And do the answers vary by industry? How, if at all, do organizational structures, cultures, and compositions influence the likelihood and type of network failure? And does network structure influence the likelihood and/or character of network performance? In their research work [108] give precise definitions characterizing the various concepts that come into play when addressing the dependability and security of computing and networking. It was surprisingly difficult to clarify these concepts when they discuss systems in which there are uncertainties about system boundaries. Furthermore, in their findings the very complexity of systems (and their specification) was often a major problem, and the determination of possible causes or consequences of failure can be a very subtle process. There are (fallible) provisions for preventing faults from causing failures. Our work approaches the question of network failure mainly from an economic perspective rather than a global overview of network failure.

### 6.2.3 Cascading Network Failure

Cascade failures from overloaded networks are usually initiated when a heavily loaded node is lost for some reason, and the load on that node (i.e. the flow passing through it) must be redistributed to other nodes in the network. However, this redistribution may cause other nodes to exceed their capacity, causing them to also fail. Sometimes even if an overloaded node does not actually fail, some protection mechanisms design may force it to shutdown to prevent the node failure. Hence the number of failed or stressed nodes increases and this propagates throughout the network. In particularly serious cases, the entire network is affected. In an early research, Motter et al. [109] show that for networks where loads can redistribute among the nodes, intentional attacks can lead to a cascade of overload failures, which can in turn cause a substantial part of the network, or even the entire network, to collapse. The authors demonstrate that the heterogeneity of these networks makes them particularly vulnerable to attacks, and a large-scale cascade may be triggered by disabling a single key node. This brings obvious concerns on the security of such systems. A different study done by [110] proposes an evolutionary algorithm to evolve complex networks that are resilient to cascading failure. In their research they analyze networks for topological regularities that explain the source of such resilience. The analysis reveals that clustering, modularity, and long path lengths all play an important role in the design of robust large-scale infrastructure. Furthermore, [111] defines cascading failure for power blackouts and gives a review of the current understanding, industrial tools, and the challenges and emerging methods of analysis and simulation.

Research work done by [112] investigates two network scenarios based on this

OF solution in a techno-economic analysis: (scenario 1) focuses on software-defined, and non-shared networks and (scenario 2) looks at the virtualized, shared networks and compare it against the current situation. By doing so, their work provides insights on the relative cost savings that a mobile network operator can reach through Software Defined Networking (SDN) and network sharing. The techno-economic analysis indicates that SDN and virtualization of the first aggregation stage and second aggregation stage network infrastructure leads to substantial capex cost reductions for the mobile network operator. As a consequence, mobile network infrastructure virtualization through the use of OpenFlow could be one of the problem solvers to tackle the issue of rising costs and decreasing profitability. Still, we did not take into account the direct effect on operational expenditures and the indirect effect that network sharing can adversely affect the ability of the operators to differentiate themselves.

Our research is unique from prior published works discussed, as we focus on the cost based SLA of network failure, and the indirect impact of failure that is not provisioned. We address the various types of unplanned network outage and cascading impact.

### **6.3 Cost of Network Failure**

Network outage could impact both CapEx and OpEx. It can be classified into two major categories: planned network outage and unplanned network outage. Planned network outage mainly impacts continuous cost of structure and maintenance (device upgrade and cleaning) in OpEx. For example, Internet Service Providers (ISP) send out an announcement on their web page when they schedule a planned outage for upgrade or

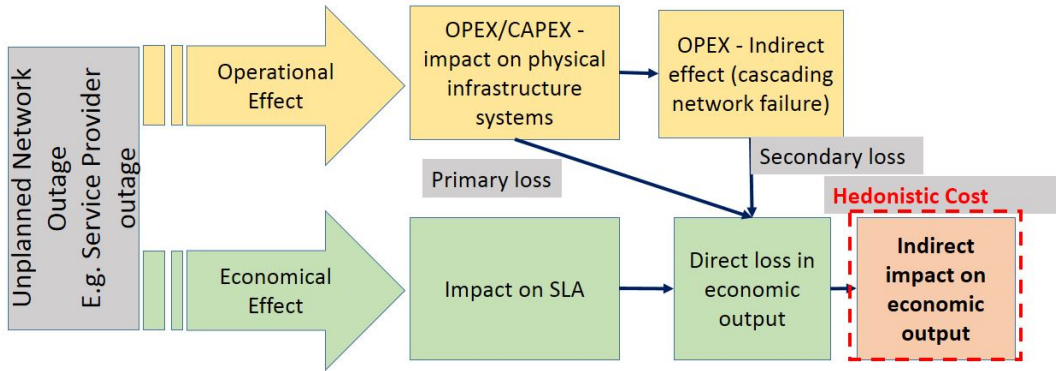


Figure 49: Effect of Unplanned Outage

maintenance, informing their clients that service will not be available during the upgrade or maintenance window. However, with SDN, the cost of planned outage is minimal and customers should not experience any outage.

Most planned outages are provisioned in the Service Level Agreement (SLA) [113], a contract between a service provider and its internal or external customers that documents what services the provider will furnish, and defines the performance standards the provider is obligated to meet. Unplanned network outage can be grouped into three sub-categories: human error (where a planned outage goes wrong), natural disaster (such as thunder storm, excessive heat leading to electrical fire, etc...) or device failure (where unsupported devices get introduced to the network causing network failure). For example, an unplanned outage could be triggered by human error from mis-configuration while doing planned outage or from unplugging the wrong device, etc. Figure 49 illustrates effects of network outage. SDN makes several set of attributes that minimize the

impact on OpEx. First, mechanized and automated roll-out (creation/provisioning/ removal/termination) of capacity of SDN functions including transport connections, based on near-realtime demand, application performance, and so on, are enabled by features such as network programmability and open APIs. This enables automated elasticity for fast deployment of network service roll-out [106]. Furthermore, SDN removes the dependency between software and hardware. Software service can be deploy on any Hardware. SDN also enables multi-tenancy and resource pooling for multiple software functions on the same hardware. Hypervisors and associated management and orchestration software facilitate virtualization of the network functions and the automation of network processes are attributes found in SDN, lacking from physical network. SDN consolidates and optimizes Service agility through enabled service abstraction and automation. Operating model change by blurring staff responsibility of network and IT [106].

### 6.3.1 Root Cause of Unplanned Outage

There is a major difference between planned and unplanned network outage. With planned outage, applications and servers are brought down gracefully in preparation to the outage with little need for cleanup and restoration. Unplanned outage depending on the impact could be as low as a glitch that unnoticed by end users to high unrecoverable network. Human error is the second highest root cause of unplanned failure in a data-center reported in 2016 [114]. Figure 50 illustrated the different root cause of unplanned data-center outage.

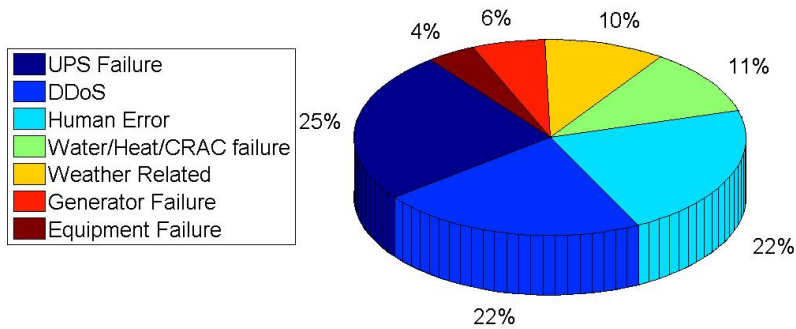


Figure 50: Root Cause of Unplanned Data-center Outage in 2016 [114]

### 6.3.2 Cost of Reliability

Reliability is achieved by minimizing these four categories: recovery cost, revenue loss, productivity loss, and hedonistic cost. The impact of network failure varies based on the network dependency of the network [115]. Eg. Is revenue generated primarily online? Is revenue highly dependent on the use of email, databases, or other online resources?

#### 1. Impact Factor ( $\alpha$ )

Impact factor ( $\alpha$ ) is % the reliant on up-time as illustrated by Table 16.

$$\alpha(n) = \% \text{ of dependent on up-time} \quad (6.1)$$

#### 2. Recovery Cost (RC)

Recovery cost depend on the impact factor( $\alpha$ ), number of employee(E) and wage (W), hours to recover (HW) and the cost of service (such as device, tool, date etc.)

Table 16: Reliability Variables

Variable Name	Description
$\alpha$	% reliant on network up-time
E	Number of employee
HW	Number of Hours worked
W	Median Wage / Hour
GR	Gross Revenue Generated / Year
TH	Total Business Hour / Year
HD	Hours of Downtime
S	Cost of service to recover
L	% Potential loss to competitor

to recover (S).

$$RC(n) = \alpha \times (E \times W) \times (HW) + S \quad (6.2)$$

### 3. Revenue Loss (RL)

Revenue Loss is affected by the impact factor and average revenue generated per day.

$$RL(n) = \alpha \times \left(\frac{GR}{TH}\right) \times HD \quad (6.3)$$

### 4. Production Loss (PL)

Production Loss is affected by the number of employees, the average wage, hour of downtime and impact factor.

$$PL(n) = (NE \times W) \times HD \times \alpha \quad (6.4)$$

### 5. Hedonistic Loss (HL)



Hedonistic Loss includes all intangible loss impacted by the total sell, impact factor and % lost potential business to competitors. For example, customers leaving because they are frustrated by the network outage.

$$HL(n) = \left(\frac{R}{\alpha}\right) \times L \quad (6.5)$$

#### 6. Total Cost of Outage (TC)

Total cost of outage include all the above.

$$TC(n) = RC + RL + PL + HL \quad (6.6)$$

### 6.4 Evaluations

In this section we evaluate real world use case, three unrelated unplanned network outages caused by human error, weather related and device failure.

Our UMKC main campus, home to roughly 17K students, and about 1.2K faculty members [2], had experienced multiple cascading failures this year that we applied our cascading framework to UMKC, with three core networks and one data-center, is a client of Blackboard Inc. for its educational technology. Blackboard, an educational technology company, offering services nationwide has 28 global data-centers managed by IBM [116]. Blackboard uses Amazon Web Services (AWS), the world's largest public cloud provider, to store its Blackboard contents and Panopto videos on its Simple Storage Service (S3). Amazon has 14 data-centers regions to provide nationwide coverage.

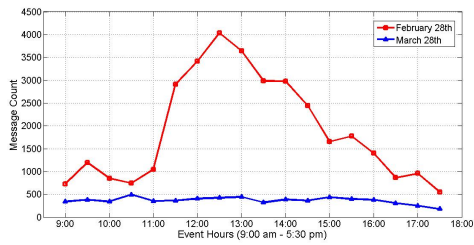


Figure 51: GET request to Blackboard without response (AWS disruption)

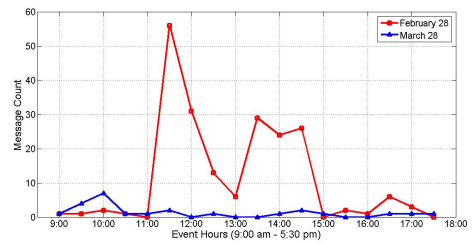


Figure 52: GET request to PANOPTO without response (AWS disruption)

#### 6.4.1 Unplanned Outage Due to Human Error

The February 28, 2017 disruption caused by human error impacted the Blackboard service. Figure 51 and 52 illustrates the increased number of GET request messages without response due to the AWS disruption that lasted over 4 hours [117] on February 28 compared to regular schedule on March 28, 2017.

#### 6.4.2 Unplanned Outage Due to Weather Related

UMKC is powered by Kansas City Power & Light (KCP&L), an electric utility company serving the Kansas City metropolitan area with 596,021 [118] customers in the state of Missouri. Early in May our campus experience cascading network outage that lasted 15 minutes due to power outage caused by trees falling around the Kansas City Metro Area [119] The outage started at about 2:10 and ended at 2:25, disrupting the normal class schedule. Figure 10 showed there were network status change messages sent right after the power outage started. This indicates that as the devices were losing power, the syslog server was receiving status change message for these devices. However, displayed by Figure 53, the syslog messages were not all off during the power outage

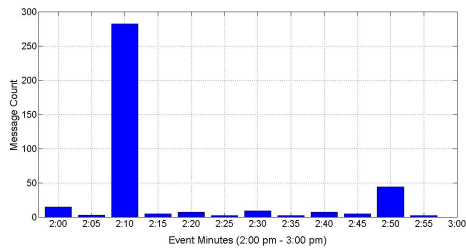


Figure 53: Syslog Network Outage Messages

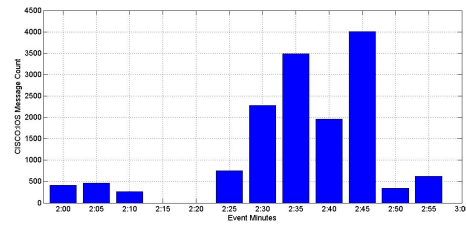


Figure 54: Network Outage from 15 Minutes of Power Outage

showing the presence of battery backup or Uninterruptible Power Supply (UPS) [120]. The majority of the UPS in the campus network failed to keep the network up during the power outage. Figure 54 illustrates that Cisco:ios were completely down during the power outage and fast recovering after the outage. There was not SLA agreement between the power supplier company and the university that covers natural disaster. Cyber insurance is available for natural disaster related outage [121]

### 6.4.3 Unplanned Outage Due to Equipment Failure

Though only 4% of unplanned outage it attributed to equipment failure, we examine real case device failure our network experienced in March. UMKC’s main channel for education resource sharing between students and instructors is provided by Blackboard, an educational technology company. Blackboard is the most used educational resource on our campus. Students receive their daily class announcements and assignments, access to class materials and grades, and connect with other students. It is also the students’s main online resource for lecture recording and online exams. Instructors use Blackboard to upload the class materials, students’s assignment, send out announcements and upload

students grades. Furthermore, universities use Blackboard for faculty training. Blackboard experienced connectivity and recording issues between March 15th and 16th [122] that impacted their Australia (AU), Canada (CA), Europe (EU) and United States (US) data-centers, affecting clients using Web Conferencing with the Ultra experience hosted by these servers. This maintenance outage affected clients attempting to join sessions. The reported issue stated that certain WiFi routers were not handling media packet prioritization correctly, resulting in dropped end-user connections. The patch put in place by Blackboard was to fix the way Collaborate declares the packet prioritization, which accommodates for WiFi routers not supporting the full array of packet prioritization declarations. Blackboard experienced 2 hrs. of outage. The SLA was minimally impacted. However, our UMKC campus experienced 2 days of outage where Network Operators had to work overtime to recover and restore services back to operational. Figure 55 illustrates the "Unsupported WiFi Router" messages from Blackboard during the two days outage our campus experienced.

## **6.5 Summary**

The Internet is intricately connected network of networks, and it would require highly skilled personnel manage it due to its complexity. These changes could result with adverse impact for unplanned network outage. SLAs do not cover most unplanned outages with indirect impact. We perform analysis of unplanned network outage using real world use case. We investigate network failure cause by human error, weather related and device failure. Our analysis shows that unplanned outage has adverse impact and

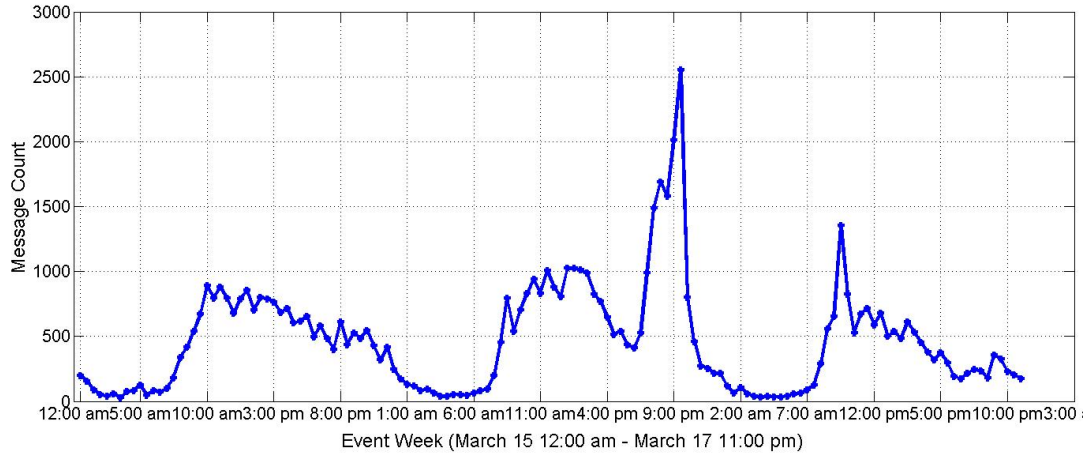


Figure 55: Unsupported WiFi Router Messages

takes time to recover and cleanup.

## CHAPTER 7

### LESSON LEARNED

#### 7.1 Big Data Analytic System Setup

Working with Splunk, big data analytic tool, we faced some issue where our queries were limited by the browser's capability and most queries would not complete. We created a python code using splunk API which helped with executing the query much faster and completing and retrieving the desired data. Figure 56 illustrates the modules and the config files setup in the python code. Importing modules: `splunklib.client` as `client`, `splunklib.results` as `results` and `splunklib.binding` as `binding` help set up the python api for the splunk server. Importing `ConfigParser` help parse the credential config file. We set the credential as config file for security reason not to hard-code the `userid/password` in the python code. This code has two input variables. The first variable is for the search query, and the second variable is optional for the desired report name to be created. If the second variable is not supplied, the default value is the user ID as the report name. Figure 57 illustrates the credential information to set up for the desired splunk server. The config file has the `HOST` (Splunk server), `userid`, and `password` information that will be used to connect to the Splunk server. If information of the port, owner and application is available in the config file, those information will be populated. However, they are not required to connect to the Splunk server. Figure 58 illustrates the Splunk Search Keyword to set up the query. This section shows that the input variable (key search value) is run

```
Crimson Editor - [C:\Program Files (x86)\Crimson Editor\template\SplunkSearch.py]
File Edit Search View Document Project Tools Macros Window Help
SplunkSearch.py
import os
import os.path
import sys
import ConfigParser
import splunklib.client as client
import splunklib.results as results
import splunklib.binding as binding
from time import sleep

#if the excute command does not have the correct input variable
#display the usage and exit the application
if len(sys.argv) == 1:
    print "Please provide the search command and report name"
    print "usage: python splunkSearch.py (required search command) (optional report name)"
    sys.exit(0)

""" Input variable search command"""
search_command = sys.argv[1]

""" Input variable report name"""
report_name = sys.argv[2]

""" Get config files for the credentials """
configFilePath = 'splunk.txt'
config = ConfigParser.ConfigParser()
config.sections()
config.read (configFilePath)

Ready Ln 30, Ch 1 77 ASCII, DOS READ REC COL OVR
```

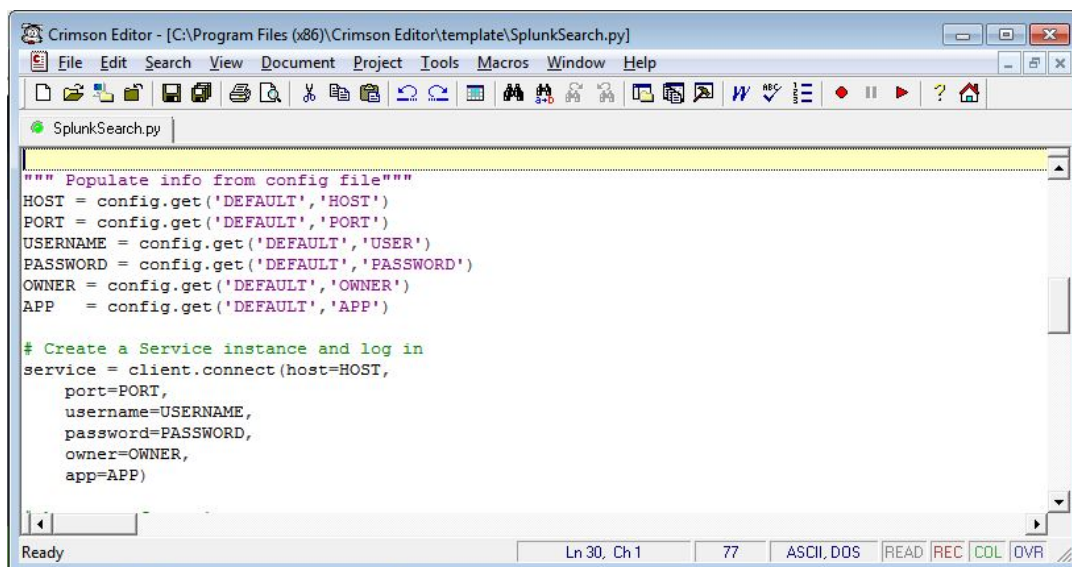
Figure 56: Python Modules and Config Files

until the job is complete and ready to with the desired result.

## **7.2 Sensors Communication Challenges and Work-around**

Working on Agile Polymorphic Software-Defined Fog Computing Platform for Mobile Wireless Controllers and Sensors research, we faced several issues. Figure 59 illustrate the three GoPiGos we used for our research. First we attempted was using the campus Wifi to communicate between the three cars. During our initial testing we discovered that the campus network blocked the broadcasting capability which was needed for our internal communication. To overcome that, we used the Cisco Access Point which helped us connect all the three GoPiGo and remove the broadcasting obstacle the campus wifi imposed. However, that approach added extra weight to our raspberrypi which worked against the optimization goal we were trying to achieve. Converting one of the raspberrypi as Access Point removed that obstacle and gave us the mechanize we needed to connect to each other and saved the added weight came with the Cisco Access Point. Another issue faced was the mininet limitation to access outside of its internal topology. For that issue, we had to create custom topology and use Network Address Translator (NAT). NAT allowed us to ping outside of mininet and with that we were able to communicate with the other GoPiGos.





The image shows a screenshot of a text editor window titled "Crimson Editor - [C:\Program Files (x86)\Crimson Editor\template\SplunkSearch.py]". The editor contains a Python script with the following code:

```
""" Populate info from config file """
HOST = config.get('DEFAULT','HOST')
PORT = config.get('DEFAULT','PORT')
USERNAME = config.get('DEFAULT','USER')
PASSWORD = config.get('DEFAULT','PASSWORD')
OWNER = config.get('DEFAULT','OWNER')
APP = config.get('DEFAULT','APP')

# Create a Service instance and log in
service = client.connect(host=HOST,
    port=PORT,
    username=USERNAME,
    password=PASSWORD,
    owner=OWNER,
    app=APP)
```

The status bar at the bottom of the editor shows "Ready", "Ln 30, Ch 1", "77", "ASCII, DOS", "READ", "REC", "COL", and "OVR".

Figure 57: Python Credential for the Splunk Server

```
Crimson Editor - [C:\Program Files (x86)\Crimson Editor\template\SplunkSearch.py]
File Edit Search View Document Project Tools Macros Window Help
SplunkSearch.py
#the type of service
search_response = service.get("apps/local")

#check the saved searches
#saved_report = {}
#savedsearches = service.saved_searches

#for savedsearch in savedsearches:
#    saved_report[savedsearch.name] = 1 #print " " + savedsearch.name
#    #print saved_report
#    #print "    Query: " + savedsearch["search"]
#if report name is provided as input variable, then search and save with report name
#otherwise, the report name will be the userid
if(report_name):
    search_response = service.post('saved/searches', name=report_name, search=search_command)
else:
    search_response = service.post('saved/searches', name=owner, search=search_command)

#this is for excuting the search job to save the result in local directory
#pipe the result or desplay on screen
#if top is provided (TODO)
#searchquery_normal = "search " + search_command | head 220

searchquery_normal = "search " + search_command
kwargs_normalsearch = {"exec_mode": "normal"}
job = service.jobs.create(searchquery_normal, **kwargs_normalsearch)

# A normal search returns the job's SID right away, so we need to poll for completion
while True:
    while not job.is_ready():
        #
```

Ready Ln 46, Ch 1 77 ASCII, DOS READ REC COL OVR

Figure 58: Python Search Job

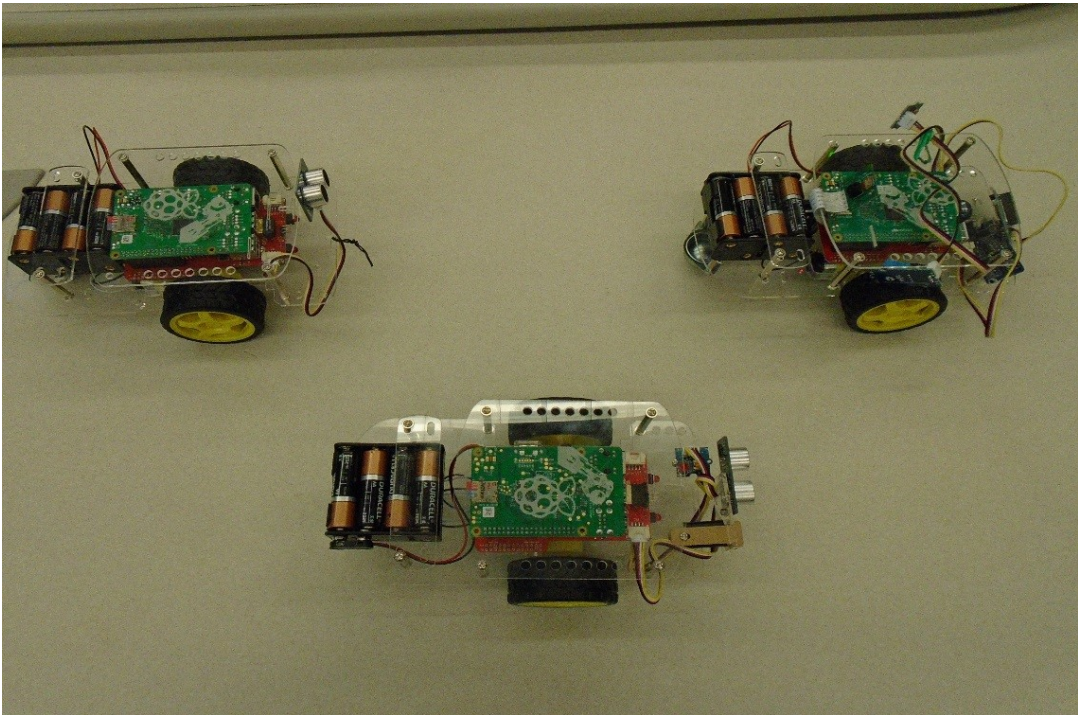


Figure 59: Three GopiGos

## CHAPTER 8

### CONCLUSIONS AND FUTURE DIRECTIONS

#### **8.1 Conclusions**

Network reliability is crucial for uninterrupted internet on the go. Our research, *Understanding the Reliability of a University Campus Network with Splunk*, carry out in-depth statistical characteristics of node outages and link failures in access networks of a university campus network. We investigated the statistical characteristics of various aspects of the network. Using Big Data analytics tool, our study shows that the general characteristics of the different layers are very distinct from each other and the wireless network is less reliable compared to the wired network and is affected by the performance of the wired network.

Networking paradigm has shifted toward virtualization and softwarization of network functions and controls, easing the daunting task of network management. Applications are promising, as they optimize costs and processes and brings new values in the infrastructures. However, the issue of monitoring Service Level Agreements (SLAs) to maintain system performance and connectivity in softwarized architecture is still a very difficult task that requires vast measuring and processing resources. We design and build a novel topology-aware network reliability management framework that utilizes Link Layer Discovery Protocol (LLDP) messages efficiently. It introduces tier-base algorithms that calculate the frequency of the LLDP messages which enables the topology-aware network

management framework to provide fast and smart decision-making information for fast failure detection and recovery.

Network devices and appliances are intricate to manage, requiring highly skilled personnel to (re)configure and (re)install the system. It also requires additional costs to add, remove or move devices from a network. These changes have cascading impacts on other parts of the network. Cascading failure is the usual mechanism by which failures propagate to cause larger impact, and occur commonly in congested complex networks, where it may express itself as the process of generation, diffusion, and dissipation of congestion. Most Service Level Agreement (SLA) only provision the direct economic impact of the initial cause of the cascading failure. Most unplanned network outages are not covered by the SLAs. We perform analysis of unplanned network outage using real-world use case. We investigate network failure caused by human error, weather-related and device failure. Our analysis shows that unplanned network outages without proper precaution have adverse impact and takes time to recover (if recoverable) and cleanup.

## **8.2 Future Directions**

Network reliability continues to be a concern that needs to be addressed. Outdated infrastructure and congestion contribute to these reliability concerns. Our research work, on measurement analysis of the Wi-Fi network condition throughout the access layer of a university campus network, sheds the light on some of these reliability concerns. Exploring the benefit of fast transition, performance enhancement, and power saving gained from the WIFI 802.11 (ac) upgrade can explain the benefit of 802.11 (ac) besides what

is revealed in our work. Furthermore, 802.11ah also called Wi-Fi HaLow provides extended range Wi-Fi networks, compared to conventional Wi-Fi networks operating in the 2.4 GHz and 5 GHz bands. improve coverage range, enable wide area-based sensor networks, sensor backhaul systems, and potential Wi-Fi off-loading functions. Ideal for the Internet of Things (IoT) with lower energy consumption, allowing the creation of large groups sensors that cooperate. Wi-Fi HaLow has two features, DAC (Distributed Authentication Control) and Central Authentication Service (CAS). DAC is self-adaptive and access point knows nothing of the connecting stations and its means of control of these stations are very limited. On the other hand, CAS does require an algorithm to dynamically control its parameters with global knowledge. It allows a single sign-on protocol for the web and allows web applications to authenticate users without gaining access to a user's security credentials, such as a password. For example, Figure 60 illustrate APs with their coverage area. Figure 61 represents the bipartite graph modeling of the AP with devices. Even though device 4 ( $d_4$ ) is within the range of AP2, it can access AP3.

CAS of the HaLow feature gives global view of the network. By having global view of the source (s) and target (t) nodes as shown by Figure 62, we can apply max flow algorithm to improve reliability.

Our contribution, TARMan:Topology-Aware Reliability Management Framework for Softwarized Network Systems improves reliability by reducing MTBF. TARMan framework focuses on adjusting the controller side of the SDN. Adding additional reliability on the switch side by applying bidirectional flow table for fast fail-over and switch-over of link failures [123] can improve TARman reliability further.

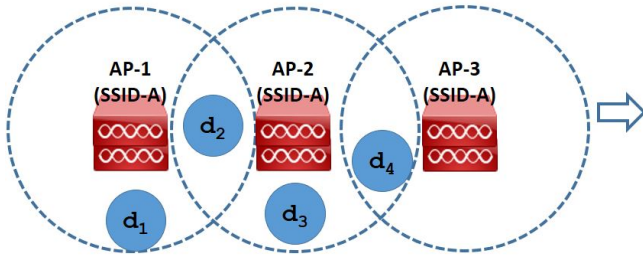


Figure 60: Access Points and Devices Modeling

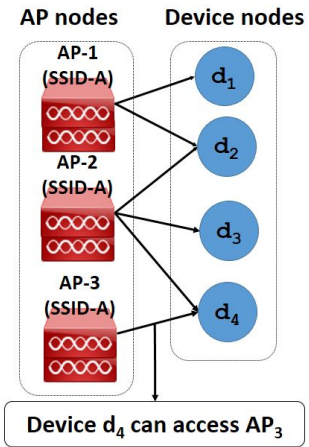


Figure 61: Bipartite Graph Modeling of the APs and Devices

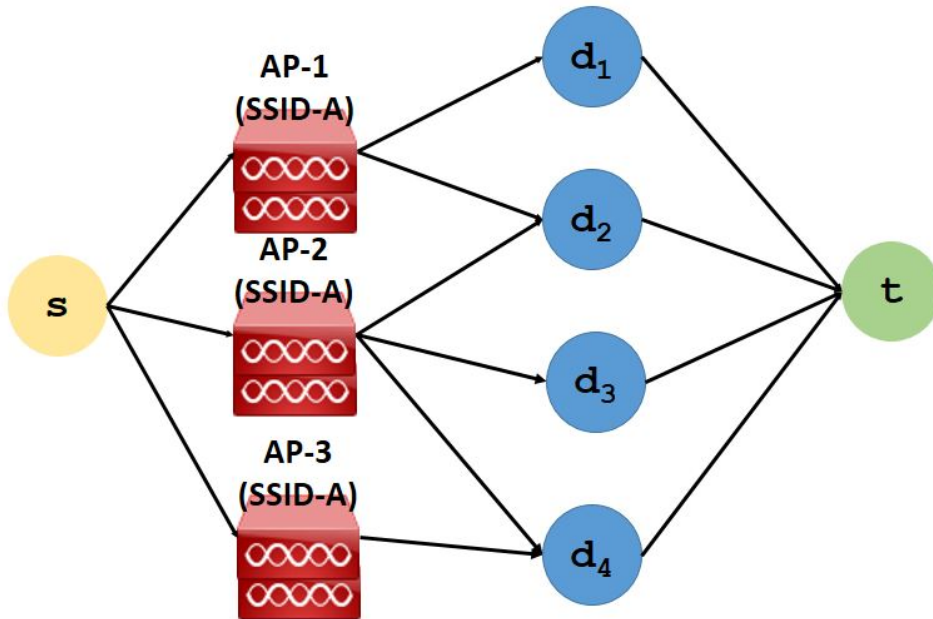


Figure 62: Flow Network Modeling of Wi-Fi Association Problem

## Bibliography

- [1] C. Boutremans, G. Iannaccone, and C. Diot. “Impact of link failures on VoIP performance”. In: *NOSSDAV*. 2002.
- [2] C. Fraleigh, F. Tobagi, and C. Diot. “Provisioning IP Backbone Networks to Support Latency Sensitive Traffic”. In: *Proceedings IEEE INFOCOM 2003, The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, San Francisco, CA, USA, March 30 - April 3, 2003*. 2003, pp. 375–385. DOI: 10.1109/INFCOM.2003.1208689.
- [3] B.-Y. Choi et al. “Analysis of Point-to-point Packet Delay in an Operational Network”. In: *Computer Networks: The International Journal of Computer and Telecommunications Networking*. Vol. 51. 13. New York, NY, USA: Elsevier North-Holland, Inc., Sept. 2007, pp. 3812–3827. DOI: 10.1016/j.comnet.2007.04.004.
- [4] B.-Y. Choi et al. “Practical Delay Measurement Methodology in ISPs”. In: *Proc. ACM CoNext*. 2005.
- [5] J. Wang et al. “Every Packet Counts: Fine-Grained Delay and Loss Measurement with Reordering”. In: *Proc. of ICNP*. 2014, pp. 95–106.
- [6] G. Iannaccone et al. “Feasibility of IP restoration in a tier-1 backbone”. In: *IEEE Network* (2004).
- [7] A. Markopoulou et al. “Characterization of Failures in an IP Backbone”. In: *Proc. of INFOCOM*. 2004, pp. 2307–2317.
- [8] A. Markopoulou et al. “Characterization of Failures in an Operational IP Backbone Network”. In: *IEEE/ACM Transactions on Networking* 16.4 (2008), pp. 749–762.
- [9] D. Turner et al. “California Fault Lines: Understanding the Causes and Impact of Network Failures”. In: *Proc. of SIGCOMM*. 2010, pp. 315–326.



- [10] R. Rao Kompella et al. “Detection and Localization of Network Black Holes”. In: *Proc. of INFOCOM*. 2007, pp. 2180–2188.
- [11] T. Henderson, D. Kotz, and I. Abyzov. “The changing usage of a mature campus-wide wireless network”. In: *Proc. of MOBICOM*. 2004, pp. 187–201.
- [12] D. Schwab and R. Bunt. “Characterising the Use of a Campus Wireless Network”. In: *Proc. of INFOCOM*. Vol. 2. 2004, pp. 862–870.
- [13] S. Gautam and G. Landge. *Characterizing the User Mobility and Network Usage in U.C. Davis Campus Wireless Network*. Tech. rep. U.C. Davis, 2004.
- [14] X. (George) Meng et al. “Characterizing Flows in Large Wireless Data Networks”. In: *Proc. of MOBICOM*. 2004, pp. 174–186.
- [15] D. Kotz and K. Essien. *Characterizing Usage of a Campus-wide Wireless Network*. TR2002-423. Dartmouth College, 2002.
- [16] X. Chen et al. “Session Lengths and IP Address Usage of Smartphones in a University Campus WiFi Network: Characterization and Analytical Models”. In: *Proc. of IPCCC*. 2013.
- [17] B.Y. Choi et al. “Outage Analysis of a University Campus Network”. In: *Proc. of ICCCN*. 2007, pp. 675–680.
- [18] J. Stearley. “Towards Informatic Analysis of Syslogs”. In: *Proc. of CLUSTR*. 2004, pp. 309–318.
- [19] K. Yamanishi and Y. Maruyama. “Dynamic syslog mining for network failure monitoring”. In: *Proc. of ACM SIGKDD*. 2005, pp. 499–508.
- [20] K. Slavicek et al. “Mathematical Processing of Syslog Messages from Routers and Switches”. In: *Proc. of ICIAFS*. 2008, pp. 463–468.
- [21] T. Qiu et al. “What happened in my Network? Mining Network Events from Router Syslogs”. In: *Proc. of IMC*. 2010, pp. 472–484.

- [22] D. Teare and C. Paquet. *Campus Network Design Fundamentals*. Cisco Press, 2005. ISBN: 1587052229.
- [23] H. Yan et al. “Tesseract: A 4D Network Control Plane”. In: *Proceedings of the 4th USENIX Conference on Networked Systems Design and Implementation*. NSDI’07. Cambridge, MA: USENIX Association, 2007, pp. 27–27. URL: <http://dl.acm.org/citation.cfm?id=1973430.1973457>.
- [24] M. Casado et al. “Ethane: Taking Control of the Enterprise”. In: *SIGCOMM Comput. Commun. Rev.* 37.4 (Aug. 2007), pp. 1–12. ISSN: 0146-4833. DOI: 10.1145/1282427.1282382. URL: <http://doi.acm.org/10.1145/1282427.1282382>.
- [25] *OpenFlow*. <http://archive.openflow.org/documents/openflow-spec-v0.8.9.pdf>. Accessed: 2016-09-21.
- [26] *LLDP*. <http://www.cisco.com/>. Accessed: 2016-09-21. 2016.
- [27] L. Ochoa Aday, C. Cervelló Pastor, and A. Fernández Fernández. “Current Trends of Topology Discovery in OpenFlow-based Software Defined Networks”. In: (2015).
- [28] F. Pakzad et al. “Efficient topology discovery in OpenFlow-based Software Defined Networks”. In: *Computer Communications* 77 (2016), pp. 52 –61. ISSN: 0140-3664. DOI: <http://dx.doi.org/10.1016/j.comcom.2015.09.013>. URL: <http://www.sciencedirect.com/science/article/pii/S0140366415003527>.
- [29] D. Katz and D. Ward. “Bidirectional Forwarding Detection (BFD)”. In: *RFC 5880 (Proposed Standard)*, Internet Engineering Task Force. 2010.
- [30] S. Sharma et al. “Fast failure recovery for in-band OpenFlow networks”. In: *Design of Reliable Communication Networks (DRCN), 2013 9th International Conference on the*. 2013, pp. 52–59.
- [31] N. Feamster, J. Rexford, and E. Zegura. “The Road to SDN”. In: *Queue* 11.12 (Dec. 2013), 20:20–20:40. ISSN: 1542-7730. DOI: 10.1145/2559899.2560327. URL: <http://doi.acm.org/10.1145/2559899.2560327>.

- [32] R. Bifulco et al. “Improving SDN with InSPiRed Switches”. In: *Proceedings of the Symposium on SDN Research*. SOSR ’16. Santa Clara, CA, USA: ACM, 2016, 11:1–11:12. ISBN: 978-1-4503-4211-7. DOI: 10.1145/2890955.2890962. URL: <http://doi.acm.org/10.1145/2890955.2890962>.
- [33] A. Tootoonchian and Y. Ganjali. “HyperFlow: A Distributed Control Plane for OpenFlow”. In: *Proceedings of the 2010 Internet Network Management Conference on Research on Enterprise Networking*. INM/WREN’10. San Jose, CA: USENIX Association, 2010, pp. 3–3. URL: <http://dl.acm.org/citation.cfm?id=1863133.1863136>.
- [34] M. Al-Fares, A. Loukissas, and A. Vahdat. “A Scalable, Commodity Data Center Network Architecture”. In: *SIGCOMM Comput. Commun. Rev.* 38.4 (Aug. 2008), pp. 63–74. ISSN: 0146-4833. DOI: 10.1145/1402946.1402967. URL: <http://doi.acm.org/10.1145/1402946.1402967>.
- [35] *ODL OpenDayLight*. <https://www.opendaylight.org/>. Accessed: 2016-09-22.
- [36] D. Kreutz, F. M. V. Ramos, and P. Verissimo. “Towards Secure and Dependable Software-defined Networks”. In: *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*. HotSDN ’13. 2013.
- [37] P. Berde et al. “ONOS: Towards an Open, Distributed SDN OS”. In: *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*. HotSDN ’14. Chicago, Illinois, USA: ACM, 2014, pp. 1–6. ISBN: 978-1-4503-2989-7. DOI: 10.1145/2620728.2620744. URL: <http://doi.acm.org/10.1145/2620728.2620744>.
- [38] K. Matheus and T. Königseder. *Automotive Ethernet*. 1st. New York, NY, USA: Cambridge University Press, 2015. ISBN: 1107057280, 9781107057289.
- [39] M. Desai and T. Nandagopal. “Coping with Link Failures in Centralized Control Plane Architectures”. In: *Proceedings of the 2Nd International Conference on COMMunication Systems and NETWORKS*. COMSNETS’10. Bangalore, India:

- IEEE Press, 2010, pp. 79–88. ISBN: 978-1-4244-5487-7. URL: <http://dl.acm.org/citation.cfm?id=1831443.1831452>.
- [40] D. Kim and J.-M. Gil. “Reliable and Fault-Tolerant Software-Defined Network Operations Scheme for Remote 3D Printing”. In: *Journal of Electronic Materials* 44.3 (2015), pp. 804–814.
- [41] M. Reitblatt et al. “FatTire: Declarative Fault Tolerance for Software-Defined Networks”. In: *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*. HotSDN ’13. Hong Kong, China: ACM, 2013, pp. 109–114. ISBN: 978-1-4503-2178-5. DOI: 10.1145/2491185.2491187. URL: <http://doi.acm.org/10.1145/2491185.2491187>.
- [42] M. Kuźniar et al. “Automatic Failure Recovery for Software-defined Networks”. In: *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*. HotSDN ’13. Hong Kong, China: ACM, 2013, pp. 159–160. ISBN: 978-1-4503-2178-5. DOI: 10.1145/2491185.2491218. URL: <http://doi.acm.org/10.1145/2491185.2491218>.
- [43] N. L. M. Van Adrichem, B. J. Van Asten, and F. A. Kuipers. “Fast Recovery in Software-Defined Networks”. In: *Proceedings of the 2014 Third European Workshop on Software Defined Networks*. EWSDN ’14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 61–66. ISBN: 978-1-4799-6919-7. DOI: 10.1109/EWSDN.2014.13. URL: <http://dx.doi.org/10.1109/EWSDN.2014.13>.
- [44] N. Gude et al. “NOX: towards an operating system for networks”. In: *ACM SIGCOMM Computer Communication Review* 38.3 (2008), pp. 105–110.
- [45] L. Saunier and K. A. Delic. “Corporate Security is a Big Data Problem: Big Data (Ubiquity Symposium)”. In: *Ubiquity* 2018.July (July 2018), 1:1–1:11. ISSN: 1530-2180. DOI: 10.1145/3158348. URL: <http://doi.acm.org/10.1145/3158348>.

- [46] *The Zettabyte Era: Trends and Analysis*. <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html>. Accessed: 2017-10.
- [47] *Compound Annual Growth Rate - CAGR*. <http://www.investopedia.com/terms/c/cagr.asp>. Accessed: 2017-10.
- [48] R. Buyya and A. V. Dastjerdi. *Internet of Things: Principles and Paradigms*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2016. ISBN: 012805395X, 9780128053959.
- [49] *The Working Group for WLAN Standards*. <http://www.ieee802.org/11/>. Accessed: 2017-10.
- [50] C.-Y. Wang and H.-Y. Wei. “IEEE 802.11N MAC Enhancement and Performance Evaluation”. In: *Mob. Netw. Appl.* 14.6 (Dec. 2009), pp. 760–771. ISSN: 1383-469X. DOI: 10.1007/s11036-008-0129-2. URL: <http://dx.doi.org/10.1007/s11036-008-0129-2>.
- [51] J. Xiong et al. “MIDAS: Empowering 802.11Ac Networks with Multiple-Input Distributed Antenna Systems”. In: *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*. CoNEXT ’14. Sydney, Australia: ACM, 2014, pp. 29–40. ISBN: 978-1-4503-3279-8. DOI: 10.1145/2674005.2675014. URL: <http://doi.acm.org/10.1145/2674005.2675014>.
- [52] C. T. Chou, A. Misra, and J. Qadir. “Low-Latency Broadcast in Multirate Wireless Mesh Networks”. In: *IEEE Journal on Selected Areas in Communications* 24.11 (2006), pp. 2081–2091. ISSN: 0733-8716. DOI: 10.1109/JSAC.2006.881621.
- [53] L. Ho and H. Gacanin. “Design principles for ultra-dense Wi-Fi deployments”. In: *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. 2018, pp. 1–6. DOI: 10.1109/WCNC.2018.8377375.

- [54] Y. Zhu et al. “A user-centric network management framework for high-density Wireless LANs”. In: *2009 IFIP/IEEE International Symposium on Integrated Network Management*. 2009, pp. 218–225. DOI: 10.1109/INM.2009.5188813.
- [55] R. K. Sheshadri et al. “AmorFi: Amorphous WiFi Networks for High-density Deployments”. In: *CoNEXT*. 2016.
- [56] M. Emmelmann et al. “Moving Toward Seamless Mobility: State of The Art and Emerging Aspects in Standardization Bodies”. In: *Wireless Personal Communications* 43.3 (2007), pp. 803–816. ISSN: 1572-834X. DOI: 10.1007/s11277-007-9255-6. URL: <https://doi.org/10.1007/s11277-007-9255-6>.
- [57] G. R. Hiertz et al. “The IEEE 802.11 universe”. In: *IEEE Communications Magazine* 48.1 (2010), pp. 62–70. ISSN: 0163-6804. DOI: 10.1109/MCOM.2010.5394032.
- [58] A. S. Sadri. “Defining the Future of Multi-gigabit mmWave Wireless Communications”. In: *Proceedings of the 2010 ACM International Workshop on mmWave Communications: From Circuits to Networks*. mmCom '10. Chicago, Illinois, USA: ACM, 2010, pp. 1–2. ISBN: 978-1-4503-0142-8. DOI: 10.1145/1859964.1859966. URL: <http://doi.acm.org/10.1145/1859964.1859966>.
- [59] T. Henderson, D. Kotz, and I. Abyzov. “The changing usage of a mature campus-wide wireless network”. In: *Computer Networks* 52.14 (2008), pp. 2690–2712.
- [60] E. Hwee Ong et al. “IEEE 802.11ac: Enhancements for very high throughput WLANs”. In: *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*. 2011, pp. 849–853. DOI: 10.1109/PIMRC.2011.6140087.
- [61] M. Sandirigama and R. Idamekorala. “Security weaknesses of WEP protocol IEEE 802.11b and enhancing the security with dynamic keys”. In: *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*. 2009, pp. 433–438. DOI: 10.1109/TIC-STH.2009.5444462.

- [62] A. Dubey and J. Hudepohl. “Towards Global Deployment of Software Engineering Tools”. In: *2013 IEEE 8th International Conference on Global Software Engineering*. 2013, pp. 129–133. DOI: 10.1109/ICGSE.2013.24.
- [63] A. Zubow and R. Sombrutzki. “Adjacent channel interference in IEEE 802.11n”. In: *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. 2012, pp. 1163–1168. DOI: 10.1109/WCNC.2012.6213952.
- [64] “Next Generation IEEE 802.11 Wireless Local Area Networks”. In: *Comput. Commun.* 75.C (Feb. 2016), pp. 1–25. ISSN: 0140-3664. DOI: 10.1016/j.comcom.2015.10.007. URL: <http://dx.doi.org/10.1016/j.comcom.2015.10.007>.
- [65] K. Sui et al. “Characterizing and Improving WiFi Latency in Large-Scale Operational Networks”. In: *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. MobiSys ’16. Singapore, Singapore: ACM, 2016, pp. 347–360. ISBN: 978-1-4503-4269-8. DOI: 10.1145/2906388.2906393. URL: <http://doi.acm.org/10.1145/2906388.2906393>.
- [66] S. Sur et al. “WiFi-Assisted 60 GHz Wireless Networks”. In: *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. MobiCom ’17. Snowbird, Utah, USA: ACM, 2017, pp. 28–41. ISBN: 978-1-4503-4916-1. DOI: 10.1145/3117811.3117817. URL: <http://doi.acm.org/10.1145/3117811.3117817>.
- [67] M. Vanhoef and F. Piessens. “Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2”. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’17. Dallas, Texas, USA: ACM, 2017, pp. 1313–1328. ISBN: 978-1-4503-4946-8. DOI: 10.1145/3133956.3134027. URL: <http://doi.acm.org/10.1145/3133956.3134027>.
- [68] *Prime Report*. [https://www.cisco.com/c/en/us/td/docs/net/\\_mgmt/prime/infrastructure/3-0/user/guide/pi/\\_ug/rep.html](https://www.cisco.com/c/en/us/td/docs/net/_mgmt/prime/infrastructure/3-0/user/guide/pi/_ug/rep.html). accessed: 2017-10.

- [69] A. E. Redondi et al. “Understanding the WiFi usage of university students”. In: *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*. 2016, pp. 44–49. DOI: 10.1109/IWCMC.2016.7577031.
- [70] M. K. Edwards. *Magic in a Black Box: Why Roku is Popular Tips, Tricks and Hacks*. USA: CreateSpace Independent Publishing Platform, 2015. ISBN: 150782050X, 9781507820506.
- [71] R. Garner. *Fire Stick: The Complete User Manual To Starting With And Using Amazon Fire TV Stick, Plus Little-Known Tips And Tricks!* USA: CreateSpace Independent Publishing Platform, 2016. ISBN: 1537013645, 9781537013640.
- [72] O. Barrick. *Chromecast: 25 Incredible Things Your Chromecast Can Do to Change the Way You View Entertainment [Booklet]*. 2016.
- [73] S. Yi et al. “Fog computing: Platform and applications”. In: *Hot Topics in Web Systems and Technologies (HotWeb), 2015 Third IEEE Workshop on*. IEEE. 2015, pp. 73–78.
- [74] R. G. Driggers et al. “Sensor Performance Conversions for Infrared Target Acquisition and Intelligence–Surveillance–Reconnaissance Imaging Sensors”. In: *Applied Optics* 38.28 (1999), pp. 5936–5943.
- [75] A. M. Khaleghi et al. “A DDDAMS-based UAV and UGV team formation approach for surveillance and crowd control”. In: *Proceedings of the 2014 Winter Simulation Conference*. IEEE Press. 2014, pp. 2907–2918.
- [76] F. Bonomi et al. “Fog Computing and Its Role in the Internet of Things”. In: *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*. MCC ’12. Helsinki, Finland: ACM, 2012, pp. 13–16. ISBN: 978-1-4503-1519-7. DOI: 10.1145/2342509.2342513. URL: <http://doi.acm.org/10.1145/2342509.2342513>.
- [77] Y. Li et al. “Software defined networking for distributed mobility management”. In: *Globecom Workshops (GC Wkshps), 2013 IEEE*. IEEE. 2013, pp. 885–889.



- [78] D. Pfammatter, D. Giustiniano, and V. Lenders. “A Software-Defined Sensor Architecture for Sarge-Scale Wideband Spectrum Monitoring”. In: *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*. ACM. 2015, pp. 71–82.
- [79] K. S. J. Pister. “Smart dust-hardware limits to wireless sensor networks”. In: *23rd International Conference on Distributed Computing Systems, 2003. Proceedings*. 2003, pp. 2–. DOI: 10.1109/ICDCS.2003.1203445.
- [80] E. Feng, J. Zheng, and C. Liu. “An integrated reliability model of hardware-software system”. In: *2014 10th International Conference on Reliability, Maintainability and Safety (ICRMS)*. 2014, pp. 577–580. DOI: 10.1109/ICRMS.2014.7107261.
- [81] F. C. Jabour, E. Giancoli, and A. CP Pedroza. “Mobility support for wireless sensor networks”. In: *Computer and Electrical Engineering, 2008. ICCEE 2008. International Conference on*. IEEE. 2008, pp. 630–634.
- [82] A. Mahmud and R. Rahmani. “Exploitation of OpenFlow in wireless sensor networks”. In: *Computer Science and Network Technology (ICCSNT), 2011 International Conference on*. Vol. 1. IEEE. 2011, pp. 594–600.
- [83] D. Zeng et al. “Evolution of Software-Defined Sensor Networks”. In: *Mobile Ad-hoc and Sensor Networks (MSN), 2013 IEEE Ninth International Conference on*. IEEE. 2013, pp. 410–413.
- [84] A. A. Dixit et al. “Elasticon: an Elastic Distributed SDN Controller”. In: *Proceedings of the tenth ACM/IEEE Symposium on Architectures for Networking and Communications Systems*. ACM. 2014, pp. 17–28.
- [85] L. M. Contreras et al. “Software-Defined Mobility Management: Architecture Proposal and Future Directions”. In: *Mob. Netw. Appl.* 21.2 (Apr. 2016), pp. 226–236. ISSN: 1383-469X. DOI: 10.1007/s11036-015-0663-7. URL: <http://dx.doi.org/10.1007/s11036-015-0663-7>.

- [86] S. Basagni et al. *Mobile ad hoc Networking*. John Wiley & Sons, 2004.
- [87] A. Sivaraman et al. “DC.P4: Programming the Forwarding Plane of a Data-center Switch”. In: *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research*. SOSR '15. Santa Clara, California: ACM, 2015, 2:1–2:8. ISBN: 978-1-4503-3451-8. DOI: 10.1145/2774993.2775007. URL: <http://doi.acm.org/10.1145/2774993.2775007>.
- [88] S. F. Hasan. “A discussion on Software-Defined Handovers in Hierarchical MIPv6 networks”. In: *Industrial Electronics and Applications (ICIEA), 2015 IEEE 10th Conference on*. IEEE. 2015, pp. 140–144.
- [89] R. JM. Vullers et al. “Energy harvesting for autonomous wireless sensor networks”. In: *IEEE Solid-State Circuits Magazine 2.2* (2010), pp. 29–38.
- [90] D. Levin et al. “Logically centralized?: state distribution trade-offs in software defined networks”. In: *Proceedings of the first workshop on Hot topics in software defined networks*. ACM. 2012, pp. 1–6.
- [91] D. Romascanu et al. *Auto-attach using LLDP with IEEE 802.1aq SPBM networks*. Internet-Draft draft-unbehagen-lldp-spb-02. Work in Progress. Internet Engineering Task Force, July 2016. 18 pp. URL: <https://tools.ietf.org/html/draft-unbehagen-lldp-spb-02>.
- [92] W. Q. Murdock. *Internet Sports Computer Cellular Device aka Mega Machine*. US Patent App. 11/901,552. 2007.
- [93] V. A. Bohara. “Measurement Results for Cooperative Device-to-Device Communication in Cellular Networks”. PhD thesis. IIT-Delhi, 2016.
- [94] *Libfluid, The ONF OpenFlow Driver*. <http://opennetworkingfoundation.github.io/libfluid/index.html>. Accessed: 2017-03.
- [95] S. R. Gandham et al. “Energy efficient schemes for wireless sensor networks with multiple mobile base stations”. In: *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE*. Vol. 1. IEEE. 2003, pp. 377–381.

- [96] Farid Kendoul. “Survey of Advances in Guidance, Navigation, and Control of Unmanned Rotorcraft Systems”. In: *J. Field Robot.* 29.2 (Mar. 2012), pp. 315–378. ISSN: 1556-4959. DOI: 10.1002/rob.20414. URL: <http://dx.doi.org/10.1002/rob.20414>.
- [97] R. Jurdak, A. G. Ruzzelli, and G. M. P. O’Hare. “Radio Sleep Mode Optimization in Wireless Sensor Networks”. In: *IEEE Transactions on Mobile Computing* 9.7 (July 2010), pp. 955–968. ISSN: 1536-1233. DOI: 10.1109/TMC.2010.35. URL: <http://dx.doi.org/10.1109/TMC.2010.35>.
- [98] K. R. Ganti, F. Ye, and H. Lei. “Mobile Crowdsensing: Current State and Future Challenges.” In: *IEEE Communications Magazine* 49.11 (2011), pp. 32–39.
- [99] L. J. G. Villalba. *Advances on Software Defined Sensor, Mobile, and Fixed Networks*. 2015. URL: <http://www.hindawi.com/journals/ijdsn/si/392960/cfp/>.
- [100] I. S. A. Dhanapala et al. “White Space Prediction for Low-Power Wireless Networks: A Data-Driven Approach”. In: *14th International Conference on Distributed Computing in Sensor Systems, DCOSS 2018, New York, NY, USA, June 18-20, 2018*. 2018, pp. 9–16. DOI: 10.1109/DCOSS.2018.00010. URL: <https://doi.org/10.1109/DCOSS.2018.00010>.
- [101] *GoPiGo*. <http://www.dexterindustries.com/GoPiGo/programming/python-programming-for-the-raspberry-pi-gopigo/installing-gopigo-python-library/>.
- [102] Raspberrypi. *Pi Power Estimator Android App*. 2015. URL: <http://www.raspberrypi-spy.co.uk/tools/pi-power-estimator-app/>.
- [103] *Amazon And The \$150 Million Typo*. <https://www.npr.org/sections/thetwo-way/2017/03/03/518322734/amazon-and-the-150-million-typo>. Accessed: 2018-12. 2018.
- [104] S. Verbrugge et al. “Methodology and Input Availability Parameters for Calculating OPEX and CAPEX Costs for Realistic Network Scenarios”. In: 17 (Jan. 2006).

- [105] M. Karakus and A. Durresi. “Service Cost in Software Defined Networking (SDN)”. In: *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*. 2017, pp. 468–475. DOI: 10.1109/AINA.2017.111.
- [106] E. Hernandez-Valencia, S. Izzo, and B. Polonsky. “How will NFV/SDN transform service provider opex?” In: *IEEE Network* 29.3 (2015), pp. 60–67. ISSN: 0890-8044. DOI: 10.1109/MNET.2015.7113227.
- [107] A. Schrank and J. Whitford. “The Anatomy of Network Failure\*”. In: *Sociological Theory* 29.3 (2011), pp. 151–177. ISSN: 1467-9558. DOI: 10.1111/j.1467-9558.2011.01392.x. URL: <http://dx.doi.org/10.1111/j.1467-9558.2011.01392.x>.
- [108] A. Avizienis et al. “Basic concepts and taxonomy of dependable and secure computing”. In: *IEEE Transactions on Dependable and Secure Computing* 1.1 (2004), pp. 11–33. ISSN: 1545-5971. DOI: 10.1109/TDSC.2004.2.
- [109] A. E. Motter and Y.-C. Lai. “Cascade-based attacks on complex networks”. In: *Phys. Rev. E* 66 (6 2002), p. 065102. DOI: 10.1103/PhysRevE.66.065102. URL: <https://link.aps.org/doi/10.1103/PhysRevE.66.065102>.
- [110] J. Ash and D. Newth. “Optimizing complex networks for resilience against cascading failure”. In: *Physica A: Statistical Mechanics and its Applications* 380.C (2007), pp. 673–683. URL: <https://EconPapers.repec.org/RePEc:eee:phsmap:v:380:y:2007:i:c:p:673-683>.
- [111] R. Baldick et al. “Initial review of methods for cascading failure analysis in electric power transmission systems IEEE PES CAMS task force on understanding, prediction, mitigation and restoration of cascading failures”. In: *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*. 2008, pp. 1–8. DOI: 10.1109/PES.2008.4596430.
- [112] B. Naudts et al. “Techno-economic Analysis of Software Defined Networking as Architecture for the Virtualization of a Mobile Network”. In: *2012 European*

- Workshop on Software Defined Networking*. 2012, pp. 67–72. DOI: 10.1109/EWSDN.2012.27.
- [113] *Monitoring Service Level Agreements for Internet Service Providers*. <https://netbeez.net/blog/monitoring-service-level-agreements-for-internet-service-providers/>. Accessed: 2017-10. 2017.
- [114] Ponemon Institute.
- [115] D. Bjerke and N. Gabela. *THE COST OF NETWORK DOWNTIME*. <http://www.datacenterjournal.com/30529-2/>. Accessed: 2018-12. 2018.
- [116] *Amazon Web Services (AWS) Outage - Blackboard Products Affected*. [https://blackboard.secure.force.com/btbb\\_exportarticlepdf?id=kAA390000004CeGGAU&pdf=true](https://blackboard.secure.force.com/btbb_exportarticlepdf?id=kAA390000004CeGGAU&pdf=true). Accessed: 2017-10. 2017.
- [117] *Summary of the Amazon S3 Service Disruption in the Northern Virginia (US-EAST-1) Region*. <https://aws.amazon.com/message/41926/>. Accessed: 2017-10. 2017.
- [118] *Storm knocks out power for thousands in KC area*. <http://www.kshb.com/news/local-news/storm-knocks-out-power-for-thousands-in-kc-area>. Accessed: 2017-10. 2017.
- [119] *KCPL Outages Map*. <http://outagemap.kcpl.com/external/default.html>. Accessed: 2017-10. 2017.
- [120] *uninterruptible power supply (UPS)*. <http://searchdatacenter.techtarget.com/definition/uninterruptible-power-supply>. Accessed: 2017-10. 2017.
- [121] Lockton. *How North American natural disasters could affect your cyber insurance*. <https://www.locktoninternational.com/articles/how-north-american-natural-disasters-could-affect-your-cyber-insurance>. Accessed: 2018-12. 2018.
- [122] *March 15, 2017 and March 16, 2017 Emergency Hotfix*. <https://blackboard.secure.force.com/publickbarticleview?id=kAA390000004CeVGAU&homepage=true>. Accessed: 2017-10. 2017.

- [123] Y. Lin et al. “Fast failover and switchover for link failures and congestion in software defined networks”. In: *2016 IEEE International Conference on Communications (ICC)*. 2016, pp. 1–6. DOI: 10.1109/ICC.2016.7510886.

## VITA

Haymanot Gebre-Amlak graduated from University of Kansas in 1992 with Bachelor of Science in Economics and minor in Computer science. She received her MBA from Washburn University in Information Systems in 1994. After graduation she started working for Sprint as software engineer. She was responsible for automating the human resource department and later joined the IT Network team and became responsible for billing mediation automation. While working at Sprint, she earned her Masters in Telecommunications Management in 2002 from Webster University. In 2010 she joined interdisciplinary Ph.D. program at University of Missouri - Kansas City. After attending a semester, she accepted a position with Twtelecom in Littleton, Colorado. After working for two years in Colorado, she moved back to Kansas in 2012 to finish her Ph.D. Her main discipline was Computer Networking and communications Systems and co-discipline is Economics.

Her area of research include reliable network, campus network traffic and performance analysis and modeling, software-defined network, wireless sensors, Internet of Things(IoT), Big data, and Artificial Intelligence (AI).

She received Outstanding Ph.D. student award from Telecommunications and Computer Networking discipline in 2017. She has also received several IEEE travel grant, and is a member of Institute of Electrical and Electronic Engineers (IEEE).

### 8.3 Conference and Journal Publications

- Haymanot Gebre-Amlak, Hoang (Mark) Nguyen, Jesse Lowe, Ala-Addin Nabulsi, and Narisa Chu, "Spatial Correlation Preserving EEG Dimensionality Reduction Using Machine Learning" IEEE International Conference on Bioinformatic and Biomedicine (BIBM 2018), Madrid, Spain, December, 2018
- Haymanot Gebre-Amlak, Md Tajul Islam, Daniel Cummins, Mohammed Al Mansoori and Baek-Young Choi, "Protocol Heterogeneity Issues of Incremental High-Density Wi-Fi Deployment" The 16th International Conference on Wired/Wireless Internet Communications (IFIP WWIC), Boston, USA, June, 2018
- Haymanot Gebre-Amlak, Goutham Banala, Baek-Young Choi, Taesang Choiz , Henry Zhu and Sejun Song "TARMan: Topology-Aware Reliability Management for Softwarized Network Systems" The 23rd IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN17), Osaka, Japan, June, 2017
- Sejun Song, Hyungbae Park, Haymanot Gebre-Amlak, Goutham Banala, Baek-Young Choi, Taesang Choiz and Henry Zhu "User-centric Adaptive Traffic Management Framework and Algorithms for Softwareization Networks" The 3rd International Conference on Network Softwarization (NetSoft'17), Bologna, Italy, July, 2017



- Haymanot Gebre-Amlak, Seoungjin Lee, Abdoh M. A. Jabbari, Yu Cheny, Baek-Young Choi, Chin-Tser Huangz, and Sejun Song, "MIST: Mobility-Inspired Software-Defined Fog System" The 12th International Conference on Consumer Electronics (ICCE'17), Las Vegas, NV, USA, January, 2017
- Hyungbae Park, Haymanot Gebre-Amlak, Baek-Young Choi, and Sejun Song "Understanding university campus network reliability characteristics using a big data analytics tool," The 11th IEEE International Conference on Design of Reliable Communication Networks (DRCN), 2015, Kansas City, MO, March, 2015
- Haymanot Gebre-Amlak, Abdoh M. A. Jabbari, Yu Cheny, Baek-Young Choi, Chin-Tser Huangz, and Sejun Song "Agile Polymorphic Software-Defined Fog Computing Platform for Mobile Wireless Controllers and Sensors" International Journal of Internet Technology and Secured Transactions:Cloud Computing, Big Data and Data Science, 2017

#### **8.4 Posters Extended Abstracts**

- Haymanot Gebre-Amlak, Sejun Song, and Baek-Young Choi, "Understanding the Cost of Cascading Failure," The 11TH Central Area Networking AND Security Workshop (CANSec'17) Student Poster Session, Missouri University of Science

and Technology, Rolla, MO, October 2017

- Haymanot Gebre-Amlak, Md. Tajul Islam, Mohammed Al Mansoori, and Baek-Young Choi, "Gradual Deployment Issues of Campus Wi-Fi: Co-Existence of Multiple Protocols," The 11TH Central Area Networking AND Security Workshop (CANSec'17) Student Poster Session, Missouri University of Science and Technology, Rolla, MO, October 2017
- Haymanot Gebre-Amlak, Seoungjin Lee, Abdoh M. A. Jabbari, Baek-Young Choi, and Sejun Song, "Toward an Energy Efficient Cooperative Software - Defined Sensor System," The 2nd IEEE International Conference on Network Softwarization (NetSoft'16) Student Poster Session, Seoul, Korea, June 2016
- Haymanot Gebre-Amlak, Seoungjin Lee, Abdoh M. A. Jabbari, Baek-Young Choi, and Sejun Song, "Toward an Energy Efficient Cooperative Software - Defined Sensor System," The 2nd IEEE SMARTCOMP (SMARTCOMP'16) Student Poster Session, St. Louis, MO, USA, May 2016.
- Haymanot Gebre-Amlak, Seoungjin Lee, Abdoh M. A. Jabbari, Yu Cheny, Baek-Young Choi, Chin-Tser Huangz, and Sejun Song, "Toward an Energy Efficient Cooperative Software - Defined Sensor System," The 9th Central Area Networking and Security Workshop (CANSec 2016), University of Central Missouri, Warrensburg, MO, April 2016
- Hyungbae Park, Haymanot Gebre-Amlak, Baek-Young Choi, and Sejun Song "Understanding University Campus Network Reliability Characteristics using a Big

Data Analytic Tool,” MINKWIC, University of Missouri-Kansas City, Kansas City, MO, October 2015

- Haymanot Gebre-Amlak ”Wireless Priority Service,” MINKWIC, University of Missouri-Kansas City, Kansas City, MO, October 2013

### **8.5 Talks**

- ”Spatial Correlation Preserving EEG Dimensionality Reduction Using Machine Learning” The 1st International Workshop on Machine Learning for EEG Signal Processing in conjunction with The IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2018, Madrid, Spain, December 2018
- Panelist: JCCC #CoLabTalk 10/4/2018 - Privacy and The Personal Device Revolution October 2018
- ”Protocol Heterogeneity Issues of Incremental High-Density Wi-Fi Deployment” The 16th International Conference on Wired/Wireless Internet Communications (IFIP WWIC), Boston, USA, June 2018
- ”Machine Learning Approach for Cognitive Control Enhancement Training Analysis”, IEEE Brain Data Bank 2018, Xi’an, China, July, 2018. (Second Place Winner)
- ”Can We Predict Fluid Intelligence Response to Training”, IEEE Brain Data Bank Challenge 2017, Boston MA, December 2017

- "Sensors in Software Defined Network" Bahir Dar University, Bahir Dar, Ethiopia, July 2017
- "MIST: Mobility-Inspired Software-Defined Fog System," The 12th International Conference on Consumer Electronics (ICCE'17), Las Vegas, NV, USA, Jan. 2017
- "NetAware: Network Architecture-Aware Reliability Management Schemes for Softwarized Network Systems," The 10th Central Area Networking and Security Workshop (CANSec 2016), Fontbonne University, St. Louis, MO, October 2016
- "Network Architecture" Bahir Dar University, Bahir Dar, Ethiopia, July 2016
- "Understanding University Campus Network Reliability Characteristics using a Big Data Analytic Tool," UND Early Career Big Data Summit, University of North Dakota, Grand Forks, ND, April. 2016.
- "Understanding University Campus Network Reliability characteristics using a Big data Analytic Tool," The 11th IEEE International Conference on Design of Reliable Communication Networks (DRCN), 2015, Kansas City, MO, March 2015

### **8.6 Instructor and Teaching Assistant**

- Instructor of Network Architecture 1, Graduate, UMKC, Kansas City, Missouri, Fall, 2016
- Instructor of Discrete Structures, Graduate, UMKC, Kansas City, Missouri, Spring 2016

- TA of Network Architecture 1, Graduate, UMKC, Kansas City, Missouri, Spring 2016
- TA of Network Architecture 1, Graduate, UMKC, Kansas City, Missouri, Spring and Fall, 2015
- TA of Cloud Computing , Graduate, UMKC, Kansas City, Missouri, Fall, 2015.
- Instructor of Principles of Microeconomics, Undergraduate, Webster University, Kansas City, Missouri, Fall, 2010