Optimal Cyber Security Placement Schemes for Smart City Infrastructures

by

Md Mahmud Hasan

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the Ph.D. degree in
Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

The conceptual evolution of smart cities is highly motivated by the advancement of information and communication technologies (ICTs). The purpose of a smart city is to facilitate the best quality of life to its inhabitants. Its implementation has to be supported by the compliant utilities and networked infrastructures. In the current world, it can only be achieved by applying ICTs in an extensive manner. The move towards the smart city's seamless connectivity widens the scope of cyber security concerns. Smart city infrastructures to face a high risk of targeted attacks due to extended cyber-physical vulnerabilities. This creates many challenging research issues relevant to the design and implementation of cyber security solutions. Networks associated with city infrastructures vary from a small indoor one to a large geographically distributed one. The context of a network is an essential consideration for security solutions. This thesis investigates a set of optimal security placement problems for enhancing monitoring in smart city infrastructures. It develops solutions to such placement problems from a resource management perspective. Economy and quality-of-security service (QoSS) are two major design goals. Such goals are translated into three basic performance metrics: (i) coverage, (ii) tolerance, and (iii) latency. This thesis studies security placement problems pertaining to three different types of networks: (i) wireless sensor network (WSN), (ii) supervisory control and data acquisition (SCADA) backbone, and (iii) advanced metering infrastructure (AMI) wide area network (WAN). In a smart city, WSNs are deployed to support real time monitoring and safety alert (RTMSA) applications. They are highly resource constrained networks. For WSNs, placement problems for an internally configured security monitor named watchdog have been studied. On the other hand, a smart grid is a key driver for smart cities. SCADA and AMI are two major components of a smart grid. They are associated with two different types of geographically distributed networks. For SCADA backbones, placement problems for a specially designed security device named trust system have been studied. For AMI-WANs, placement problems for a cloud-based managed security service have been studied. This thesis proposes a number of promising solution schemes to such placement problems. It includes evaluation results that demonstrate the enhancements of the proposed schemes.

Acknowledgements

It was a God gifted opportunity for me to have Dr. Mouftah as my supervisor. I gratefully appreciate Dr. Mouftah's patience and tolerance during different phases of this research. He was always available for discussion despite having a busy schedule. His positive attitude always inspires me to move forward.

I wish to thank the members of my thesis examining committee for their valuable time and patience. I am grateful to the Faculty of Engineering Graduate Office for its professional handling of academic matters.

I thank Dr. N. U. Ahmed, Dr. Upal Mahfuz, and Dr. Binod Vaidya for their valuable suggestions on various research matters. I would like to thank all my friends and colleagues, whom I have had many fruitful discussions.

Finally, I would like to thank my family members for their endless support and continuous encouragement.

Dedication

To my parents

Table of Content

List of Figures

List of Tables

List of Acronyms

AES	Advanced Encryption Standard
AIRS2Parallel	Artificial Immune Recognition System2 Parallel
AIS	Artificial Immune System
AMI	Advanced Metering Infrastructure
BCDR	Business Continuity and Disaster Recovery
BM	Bipartite Matching
BS	Base Station
C1WSN	Category 1 WSN
C2WSN	Category 2 WSN
CA	Certification Authority
CLONALG	ClONal selection ALGorithm
CoA	Collaboration-Aware
CPS	Cyber-Physical System
CPU	Central Processing Unit
CSP	Cloud Service Provider
CSA	Cloud Security Alliance

DIDS	Distributed Intrusion Detection System
DLP	Data Loss Prevention
DDoS	Distributed Denial-of-Service
DoS	Denial-of-Service
DSM	Demand Side Management
EaaS	Encryption-as-a-Service
EMS	Energy Management System
ESPRESSO	Encryption as a Service for Cloud Storage Systems
EV	Electric Vehicle
FCH	Full Coverage Heuristic
G2V	Grid-to-Vehicle
GUI	Graphical User Interface
HAN	Home Area Network
HMI	Human Machine Interface
IAM	Identity and Access Management
ICT	Information and Communication Technology
IDS	Intrusion Detection System
IDSaaS	Intrusion Detection System-as-a-Service
IED	Intelligent Electronic Device
ILP	Integer Linear Program
IoT	Internet of Things

IP	Internet Protocol
ISN	Intelligent Sensor Network
ISO	International Organization for Standardization
ISP	Internet Service Provider
IT	Information Technology
ITS	Intelligent Transport System
KDC	Key Distribution Center
LAP	Linear Assignment Problem
LCM	Link Coverage Maximization
LO	Limited Overlapping
LOLP	Loss of Load Probability
LP	Linear Programming
LPP	Linear Programming Problem
LRM	Linear Resource Minimization
M2M	Machine-to-Machine
MANET	Mobile Ad-hoc Network
MILP	Mixed Integer Linear Program
MPT	Minimal Path Tolerance
MST	Minimum Spanning Tree
NAN	Neighborhood Area Network
NIST	National Institute of Standards and Testing

PKI	Public Key Infrastructure
PLC	Programmable Logic Controller
PMU	Phasor Measurement Unit
QAP	Quadratic Assignment Problem
QCLP	Quadratically Constrained Linear Programming
QoS	Quality of Service
QoSS	Quality of Security Service
RAM	Random Access Memory
RTMSA	Real Time Monitoring and Safety Alert
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SECaaS	Security-as-a-Service
SIEM	Security Information and Event Management
SLA	Service Level Agreement
SVM	Support Vector Machine
TCP	Transport Control Protocol
TCA	True Cost Analysis
TLS	Transport Layer Security
TMU	Traffic Monitoring Unit
TTP	Trusted Third Party
UDP	User Datagram Protocol

V2G	Vehicle-to-Grid
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VM	Virtual Machine
VPN	Virtual Private Network
WAMS	Wide Area Measurement System
WAN	Wide Area Network
WSN	Wireless Sensor Network

List of Symbols

α	weighting factor related to topology
β	weighting factor related to latency
γ	path tolerance
γ_M	maximum path tolerance
$\delta_{set}(.)$	set of subpartition node sets
$\delta_{size}(.)$	set of subpartition sizes
$\triangle_{min}$	subpartition node set to be adjusted
ζ_{ij}	auxiliary variable for watchdog placement
η_i	set of nodes in the coverage area of a sensor node i
θ	latency threshold for minimum spanning trees
κ_{ij}	single hop neighborhood indicator between sensor nodes i and j
$\lambda_{mm'}$	collaboration indicator between AMI concentrators m and m'
$\mu_{mm'}$	collaboration level between AMI concentrators m and m'
$\xi_{mnm'n'}$	auxiliary variable for service placement
Ξ	vector of auxiliary variables related to Ω
ρ	tolerance factor of a SCADA network
τ_{max}	maximum MST weight of non-star partition
τ_s^{mst}	MST weight of a segment s
τ_{ref}	reference MST weight
ϕ	loop control variable
χ_{ij}	auxiliary variable for trust node placement
$\psi(e)$	link centrality metric of a link e

ω_{mn}	binary auxiliary variable for service placement
Ω	server incidence vector
A_c	set of AMI concentrator
a_{ij}	shortest hop distance between nodes i and j
c_e^L	centrality weight of a link e
c_u	degree centrality of a node u
c_{sb}	degree centrality of a bordering node b located in a segment s
$B(s)$	set of bordering nodes in a segment s
D_c	set of data center
$d_{min}^{MST}(e)$	minimum degree of link e in the MST
$d_{min}^{MST}(e_{u\leftrightarrow v})$	minimum degree of a link between nodes u and v in the MST
d_u^{MST}	MST degree of a node u
e_{ij}	direct link indicator between nodes i and j
e^s	link in a segment s
$e_{u\leftrightarrow v}$	undirected link between nodes u and v
E	link set of a SCADA Network
E_I	set of MST links to be eliminated to form partitions
E_s^{mst}	set of MST links in a segment s
E^{ps_i}	set of intra-segment MST links in partition s_i
E^{SS}	special set of MST links that form initial partitions
E^{MST}	MST link set of a SCADA Network
f_i	total number of nodes in the coverage area of a sensor node i
$g(.)$	reservation function
$\widetilde{g}(.)$	initial assignment function
$G(V, E)$	SCADA network graph
h_{ij}	total number of common nodes between coverage areas of nodes i and j
k_s	ideal segment size
K	total number of segments to be created
K_{min}	minimum value of K for a given θ
$L_{ss\prime}$	set of inter-segment links between segments s and $s\prime$

m_{sample}	sample mean
M	total number of available trust systems
n_{obs}	sample size
N	size of a SCADA network in terms of node
N_{wd}	Total number of watchdogs in a WSN
N_{wsn}	size of a WSN in terms of node
P_{ij}	node set for the shortest path between nodes i and j
P_{ij}^{*}	node set for the shortest path with the highest hop count
Q	total number of required trust systems
r_n	capacity of data center n
s_i	set of nodes of a segment i
S	set of network segments
t_{mn}	access latency between AMI concentrator m and data center n
t_{NL}	normalized latency
$t_{nn'}$	latency between data centers n and n'
$T(V, E^{MST})$	MST of a given network $G(V, E)$
v_{c1}	set of monitored sensor nodes for non-overlapping watchdogs
v_{sample}	sample variance
v_{wd}	final watchdog set
v_{wd1}	non-overlapping watchdog set
v_{wd2}	secendary watchdog set
V	node set of a SCADA network
V^{Init}	initial set of trust nodes
V^{s}	node set of a network segment s
V^{Trust}	set of trust nodes
V_{no}	coverage node set for non-overlapping watchdogs
V_{wsn}	set of sensor nodes in a WSN
$w(e)$	weight of link e in terms of propagation delay
$\widetilde{w}(e)$	normalized weight of a link e with respect to the maximum link weight
w_{max}^{MST}	maximum MST link weight for a given SCADA network

W	total weight of an MST
x_i	binary variable for trust node selection
x_{sb}	binary variable for bordering node selection
x_I	bordering node incidence variable
X	node incidence vector
$\widetilde{X}$	vector of auxiliary variables related to X
X_B	bordering node incidence vector
$X_I(l)$	variable set for bordering nodes belonging to the inter-segment link l
y_e	binary variable for MST link elimination
Y	link incidence vector
z_i	binary variable for watchdog selection
z-value	value corresponds to the z-distribution
Z	sensor node incidence vector
$\widetilde{Z}$	overlapping incidence vector

Chapter 1

Introduction

1.1 Background

A smart city is an evolving concept that is motivated by technological advancements. It is also synonymous with the digital city, intelligent city, and connected city. In the literature, there is no firm definition for a smart city due to its multidisciplinary nature. It can be defined in many ways, from different perspectives. For us, a smart city is an integrated effort of advanced technologies towards a digital world. At present, 50% of the world population lives in urban areas and this percentage is estimated to reach 70% by 2050 [fao09]. This implies the rapid growth of urban population due to urbanization of rural areas, and the human mobility towards urban areas. It is necessary to have an urban development strategy for the future cities to maintain high standards of living and comfort of their citizens. Smart cities are devised as an all-in-one solution to urban lives. There are numerous initiatives around the world to develop smart cities [Sch12, Bat12, mit12, eu007, ucl12]. Those initiatives include government, industrial, and academic projects focusing on smart cities. Each of them has its own outline for a smart city. The common fact is that smart cities require an extensive deployment of advanced information and communication technologies (ICTs). Eventually, smart cities will comprise a system of systems that is powered by a network of everything. This leads to the development of a number of new concepts such as the urban Internet of Things (IoT) [Zan14].

Smart cities are expected to have six main characteristics: smart economy, smart people, smart mobility, smart governance, smart environment, and smart living [eu007]. The smart economy comprises a competitive market that facilitates innovative business opportunities, entrepreneurship, and international investments. The smart people are basically social and human capital. They are qualified, life-long learners, social, flexible, creative, open-minded, and willing to participate in public life. The smart mobility comprises transportation and ICT. It offers safe transportation, seamless accessibility, and availability of ICT infrastructure. The smart governance ensures citizens participation in policy and decision making processes i.e., e-democracy as well as encourages public together with social services. The smart environment mainly concerns conservation of natural resources. It refers to a protected environment that is pollution free and green. The smart living concerns the quality of life and comprises a number of social, cultural, health, comfort, safety, along with individual factors. A number of building blocks are necessary to attain the aforementioned characteristics. In particular, our current study focuses on the infrastructures and utilities that integrate multiple cyber-physical systems (CPSs). In smart cities, infrastructures and utilities are expected to offer better urban facilities than conventional ones. Four critical infrastructures are frequently mentioned in the available literature [Bat12, Sch12]: (i) rich and seamless ICT infrastructure, (ii) smart grid, (iii) intelligent transport system (ITS), and (iv) real time monitoring and safety alert (RTMSA). The ICT infrastructure provides wireless connectivity, seamless mobility, and broadband Internet access in smart cities. Its components include high capacity optical fiber backbone, cellular networks, wireless access points, Internet service providers (ISPs), integrated database systems, data centers, and cloud-based services. The smart grid is arguably the key driving force behind smart cities. It comprises electric utilities, supervisory control and data acquisition (SCADA) systems, advanced metering infrastructure (AMI), bulk generation, distribution, and demand side management (DSM). Its key offerings include integration of renewable energy resources, consumers' participation in the energy market, and support for green initiatives. An example of such green initiatives is the electrification of vehicles. Electric vehicles (EVs) significantly reduce carbon emissions. The scope of an ITS mainly includes signaling and lighting, vehicular communications, and intelligent sensor networks (ISNs). The roads and

highway signaling and lighting are required to ensure safe transportation. Signaling is an essential part of the traffic control system as it serves the purpose of safety regarding roads and highways. The lighting ensures visibility for vehicle operators and pedestrians. The ITS vehicular communications include vehicle-to-infrastructure (V2I), vehicle-to-vehicle (V2V), and center-to-center communications. The V2I communication helps vehicle route scheduling and congestion management. The V2V communication brings safety and co-operative collision avoidance. The center-to-center communications are used in congestion control and vehicle tracking systems. The ISNs monitor traffic, generate control signals, and initiate alerts to aid roadside safety. They are also deployed to monitor the structural health of transport infrastructures such as bridges, flyovers, underpasses, railways, and highways. The RTMSA system is responsible for monitoring of environment and events. It boosts public safety by generating alerts whenever necessary. It requires a large-scale deployment of wireless sensor networks (WSNs) around the city. Those WSNs collect data that are fed to the central database for further processing.

Due to the pervasive use of communication networks, smart city infrastructures are vulnerable to various forms of cyber-attacks. The major technical challenges include the design of network-specific solutions.

1.2 Motivation

In general, cyber security solutions are designed considering two types of mechanisms: cryptographic and network monitoring. This thesis focuses on the network monitoring. Any networked infrastructure has three basic security requirements: confidentiality, integrity, and availability. The goal of a network monitoring mechanism is to act on cyber-attacks and distrust activities that intend to violate at least one of three basic security requirements. Network monitoring is a common preliminary task in anomaly detection, traffic filtering, intrusion detection, firewalling, and trust management. Security monitors are deployed to perform such monitoring tasks. Depending on the network, a security monitor can be an internal entity (watchdog in WSNs), an installed device (trust system in

SCADA), or an externally managed system (cloud-based security service). This thesis is motivated by three aspects of security deployment strategies: (i) efficient utilization of resources, (ii) quality-aware economical solutions, and (iii) adaptation with the new technological advancement. In particular, its key drivers are the following research challenges.

Security placement in high density resource-constrained networks. This category includes WSNs in smart cities that have limited computational capability and energy concerns. Security monitors are internally configured on the selected sensor nodes.

Security placement in geographically distributed utility control networks. This category includes smart grid SCADA networks. Security monitors are installed as specialized devices that provide intrusion detection and firewall services. In a large network, only a selected set of nodes is equipped with such devices due to budgetary restrictions.

Security service placement for geographically distributed utility metering networks. This category includes AMI wide area mesh networks. Its primary source of traffic is customer owned meters. As those meters are low-trust entities, the collected traffic are required to pass through security monitors. The potential of cloud computing can be exploited in such a way that security monitoring becomes part of a managed service.

1.3 Objectives

The objective of this thesis is to study in-depth the cyber security placement problems in the area of smart city infrastructures and to develop quality-aware and resource-efficient solutions. In particular, this thesis aims at solving the following research problems.

• Enhancement to the desired level of protection deploying minimal amount of security resources.

• Efficient utilization of a given amount of security resources for obtaining maximal benefits.

• Design of optimal security deployment strategies considering network topology, resource availability, latency, and distributed monitoring.

• Resource management for a cloud-based security as a service (SECaaS) model that is devised for geographically distributed cyber-physical systems.

1.4 Contributions

In this thesis, we study the cyber security placement problems for three different network infrastructures that are pivotal in smart cities. The major considerations in designing security solutions for those infrastructures include their applications, network topologies, and the availability of resources. The main contributions of this thesis are as follows.

• An investigation of watchdog placement problems in WSNs from a resource management perspective. Four novel optimization models are presented and evaluated for realistic topologies.

• A novel segmentation-based trust system placement scheme for smart grid SCADA networks. The proposed scheme exploits the graph theoretic properties of minimum spanning trees (MSTs). It provides network topology-aware solutions that are computationally low overhead.

• A centrality-based trust system placement scheme for resource constrained scenarios of smart grid SCADA networks. It extends the aforementioned segmentation-based scheme to resource constrained problems.

• A latency-aware segmentation-based trust system placement scheme for smart grid SCADA networks. It creates network segments using a latency threshold. It addresses the time criticality in the alert propagation.

• Two novel optimization models for trust node assignment to enhance distributed monitoring in smart grid SCADA networks. Those models are proposed to utilize trust systems' active/router mode of operation, whereas the aforementioned schemes are most likely appropriate for the tunnel/gateway mode. They are evaluated and compared to obtain the best trust node assignment strategy.

• An architectural framework for cloud-based security services for smart grid AMIs. The framework's main goal is to deliver a collaborative monitoring service.

• An optimization model for cloud-centric collaborative security service placement for AMIs. It considers a geographically distributed private cloud provisioning problem. The model minimizes the overall latency.

1.5 Thesis Outline

The rest of this thesis is organized as follows. Chapter 2 presents a survey of the relevant literature in the area. Three major parts are included: cyber security issues in WSNs, cyber security concerns in the smart grid, and cloud-based security services. Chapter 3 presents a study of watchdog placement problems in WSNs. It includes three optimization models for resource unconstrained scenarios and also extends a model to resource constrained scenarios. Chapter 4 presents a trust system placement scheme for smart grid SCADA networks. It focuses on a topology-aware segmentation approach to select trust nodes. Chapter 5 presents investigations of constrained cases of trust system placement problems. It includes two major parts: the study of resource-constrained and the latency-constrained scenarios. Chapter 6 presents trust system placement schemes for distributed monitoring in SCADA networks. It proposes two approaches to distributed monitoring: link-by-link coverage and hop-by-hop coverage where the limitations of security resources are considered in both cases. Chapter 7 introduces a cloud-centric collaborative security service placement scheme for AMIs. It proposes an SECaaS architecture for smart grid monitoring. It also investigates latency-aware placement for both types of collaboration, distributed and centralized. Finally, Chapter 8 provides a conclusion to this thesis and discusses the future work.

1.6 List of Publications

Book Chapter

[Has17b] M. M. Hasan and H. T. Mouftah. "Cyber-physical vulnerabilities of wireless sensor networks in smart cities". In *Security and Privacy in Cyber-Physical Systems: Foundations and Applications*. John Wiley, London, UK, pp. 263-280, 2017.

Journal

[Has16c] M. M. Hasan and H. T. Mouftah. "Optimal trust system placement in smart grid SCADA networks". *IEEE Access*, Vol. 4, pp. 2907-2919, May 2016.

[Has17c] M. M. Hasan and H. T. Mouftah. "Optimization of watchdog selection in wireless sensor networks". *IEEE Wireless Communications Letters*, Vol. 6, No. 1, pp. 94-97, Feb. 2017.

[Has17a] M. M. Hasan and H. T. Mouftah. "Cloud-centric collaborative security service placement for advanced metering infrastructures". *IEEE Transactions on Smart Grid*, 2017. submitted.

Conference

[Has16a] M. M. Hasan and H. T. Mouftah. "Latency-aware segmentation and trust system placement in smart grid SCADA networks". In *Proceedings of the CAMAD2016*, Toronto, Canada, pp. 37-42, 2016.

[Has16b] M. M. Hasan and H. T. Mouftah. "A study of resource-constrained cyber security planning for smart grid networks". In *Proceedings of the EPEC2016*, Ottawa, Canada, pp. 1-6, 2016.

[Has15] M. M. Hasan and H. T. Mouftah. "Encryption as a service for smart grid advanced metering infrastructure". In *Proceedings of the ISCC2015*, Larnaca, Cyprus, pp. 216-221, 2015.

[Has13a] M. M. Hasan and H. T. Mouftah. "Cloud-based security services for the smart

grid". In *Proceedings of the CASCON2013*, Markham, Canada, pp. 388-391, 2013.

[Has13b] M. M. Hasan and H. T. Mouftah. "Security and privacy challenges in a smart city". In *Proceedings of the Fifth WiSENSE2013*, Ottawa, Canada, pp. 8-12, 2013.

Chapter 2

Cyber Security Monitoring in Smart City Infrastructures: State of the Art

2.1 Introduction

A smart city meets a set of technical and non-technical criteria that offer the best urban facilities. Those criteria are proposed by various initiatives from academia, governments, and industries around the world. Despite variations among the proposed sets, one obvious thing is the extensive implementation of ICTs. Eventually, the operation of a smart city is going to be supported by the *network of everything*. The evolution towards smart cities incurs rapid increment of the flow of sensitive information over supporting communication networks. At the same time, cyber security concerns are greatly intensified. It requires the integration of a number of cyber-physical infrastructures that facilitate transportation, energy, safety, and clean environment. Those infrastructures are critical due to their essentiality in urban life. Sensing is one of the most useful tasks in smart city operations. This leads various forms of WSNs to be an integral part of smart city infrastructures. Those WSNs can be an easy target for cyber-attacks as well as physical-attacks. They need to be protected first to ensure the security of the rest. The smart grid is considered as one of the key ingredients of a smart city. It comprises two major components: SCADA systems and AMIs. These components are included into key smart city infrastructures that are

exposed to potential cyber risks [oci15]. Smart grid SCADA systems are assumed to be Internet-based. Thus one of the major sources of security threats is the Internet. An AMI is connected to numerous untrusted devices that are located at customer premises. Those devices are a source of potential security threats. Smart grid components require specialized security solutions to ensure smooth operations. The design of a security solution is motivated by some crucial factors such as network topology, availability of resources, and latency. In addition to the existing solutions, new security paradigms are being devised to meet the challenge of future contexts. The SECaaS is a model that exploits the potential of cloud computing for delivering cyber security [csa11]. It can be used to develop new security paradigms for smart city infrastructures. Our current study includes three major areas: security in WSNs, security in smart grid networks, and SECaaS. In this chapter, we survey the most relevant literature that covers the current area of interests.

2.2 Cyber Security and Optimal Placement Problems

The operation of a smart city infrastructure is heavily dependent on supporting communication networks. The flow of information in a communication network has three main security requirements: confidentiality, integrity, and availability. The confidentiality refers to the protection of information from the access of unauthorized parties. The integrity refers to the prevention against the unauthorized modification of information. The availability refers to the assurance for authorized parties to have uninterrupted access to information. In a successful attack, at least one of these basic requirements is violated. These requirements can be achieved by implementing carefully designed security solutions. In addition to the basic security requirements, a smart city operation may require non-repudiation and user level privacy. Non-repudiation is a way of identifying the attacking entities since it stores strong signatures of activities. It is mainly useful in preventing internal attacks. For example, utilities are required to deploy non-repudiation techniques to secure consumption data and revenue information [Xia13]. It is also a way of detecting energy theft, which is a common problem around the world. The privacy concerns arise at the consumer level where grid data can be exploited for tracking personal information. Unprotected smart

metering data can reveal information about corresponding dwellers. In [Cav09], a set of guidelines called SmartPrivacy has been proposed for the Canadian smart grid. The reputation of a grid operator is affected when it fails to preserve consumers' privacy. It is a discouraging message since the participation of customers is one of the basic conditions for a smart grid environment.

Cryptography and security monitoring are two major means of enhancing cyber security in a networked infrastructure. Their combined actions are necessary to fulfill the aforementioned requirements.

The cryptographic encryption is the most prominent way of enhancing cyber security. It provides confidentiality and integrity to messages. There are two basic encryption techniques: symmetric key cryptography and public key cryptography. In the symmetric cryptography, the same key is used to encrypt and decrypt messages. In the public key cryptography, two different keys are used to encrypt and decrypt messages. Those keys are known as the public key and private key respectively. It requires a trusted third party (TTP) in both types of cryptography. The TTP can be a key distribution center (KDC) or a certification authority (CA). The role of a KDC is to distribute keys among communicating parties as it pertains to the symmetric cryptography. It can also be adopted in the public key cryptography. For the symmetric cryptography, implementation of a central KDC is not mandatory. It is possible to establish keys in a distributive manner, where communicating parties use a cryptographic key distribution protocol such as the Diffie-Hellman. In the public key cryptography, each node owns a pair of public and private keys. It requires a public key infrastructure (PKI) to support the cryptographic operation. The private key of a node is only known to its owner. The public key of a node can be known to other nodes in the same network. The role of a CA is to tell which public key is owned by which node. Only authorized nodes have true public keys. Thus, it prevents communications between authorized and unauthorized nodes. In general, PKI is a resource intensive solution that expends high overheads regarding communication and computation. It also exposes high authentication delay and power hungriness.

An intrusion detection system (IDS) is an ideal example of a network security monitor.

It makes four possible types of decisions: (i) true positive, (ii) true negative, (iii) false positive, and (iv) false negative. These decisions are obtained from behavioral analysis of network traffic. A decision is positive when an IDS classifies a behavior as an intrusion. A decision is negative when an IDS classifies a behavior as normal. The true decisions are made when an IDS acts correctly. On the other hand, the false decisions are made when an IDS acts incorrectly. An effective IDS generates a low rate of false decisions. Depending on the application, IDSs are categorized and implemented. There are two basic detection methodologies are used in behavioral analysis: (i) anomaly-based and (ii) signature-based. In the anomaly-based detection, the behavior of each node is compared with a pre-defined normal behavior. If the deviation exceeds a threshold value, an intrusion is detected. The behavior of a communicating entity is extracted from its traffic pattern. The settings regarding normal behavior and threshold affect the accuracy of detection. To improve the accuracy, settings are required to be updated periodically. In the signature-based detection, a set of rules is pre-defined based on the previously known attacks. If the behavior of any node violates one of those rules, an intrusion is detected. New or unknown attacks cannot be detected from the signature-based IDS. This is why the set of rules is required to be updated from time to time. In particular, whenever a new attack signature is developed, a rule should be created to add to the set.

Optimal placement problems are useful resource management tools for large-scale distributed networks. Their applications in the area of security monitoring have drawn significant attention in recent years [Alm16, Pas15, Hua12, Tha16, Sha16]. Economy and quality of security are two basic aspects of such problems. Their exact solutions are computationally expensive in general. They are pertaining to (i) cost-effective deployment strategies for security hardening devices and (ii) efficient utilization of security resources. Coverage, tolerance, and latency are commonly addressed quality metrics that are associated with optimal security placements. An optimal placement solution can also be useful in the deployment of cryptographic facilities such as KDC and CA, in a networked environment.

2.3 Review of Cyber Security in Wireless Sensor Networks

In a smart city, WSNs have big roles to play to support city-based operations. It requires large-scale deployments of WSNs around the city for sensing numerous events. A massive amount of information will then be collected from those WSNs for analyses and decision making. The most challenging part is to secure the information from various forms of attacks. Detection and prevention procedures in response to cyber-physical attacks are resource intensive. In this section, we provide an overview of cyber-physical vulnerabilities of WSNs in the context of smart cities. We also discuss possible mitigation approaches.

2.3.1 WSN Applications

Smart city infrastructures are expected to deploy a wide range of WSN-based applications. The major application areas include smart home, smart grid, ITS, and RTMSA. Commercial sensor networks can be categorized into two basic groups: category 1 (C1WSNs) and category 2 (C2WSNs) [Soh07]. The C1WSNs consist mesh-based systems with multihop radio connectivity. They are more suitable for large geographic area coverage and mobile applications such as ITS and RTMSA. The C2WSNs consist point-to-point or multipoint-to-point systems generally with single-hop radio connectivity. They are more suitable for small area or indoor applications such as smart home and smart grid distribution substation monitoring. Communication standards IEEE 802.15.4, WiFi, and ZigBee are popularly chosen for commercial deployments of WSNs. In the context of smart cities, the outdoor deployment of WSNs such as ITS and RTMSA are more exposed to security threats.

2.3.2 Possible Forms of Attacks

Cyber adversaries always look to exploit the vulnerabilities of their targets. The vulnerabilities refer to the inherent weaknesses that can be exploited in a cyber-attack. A CPS

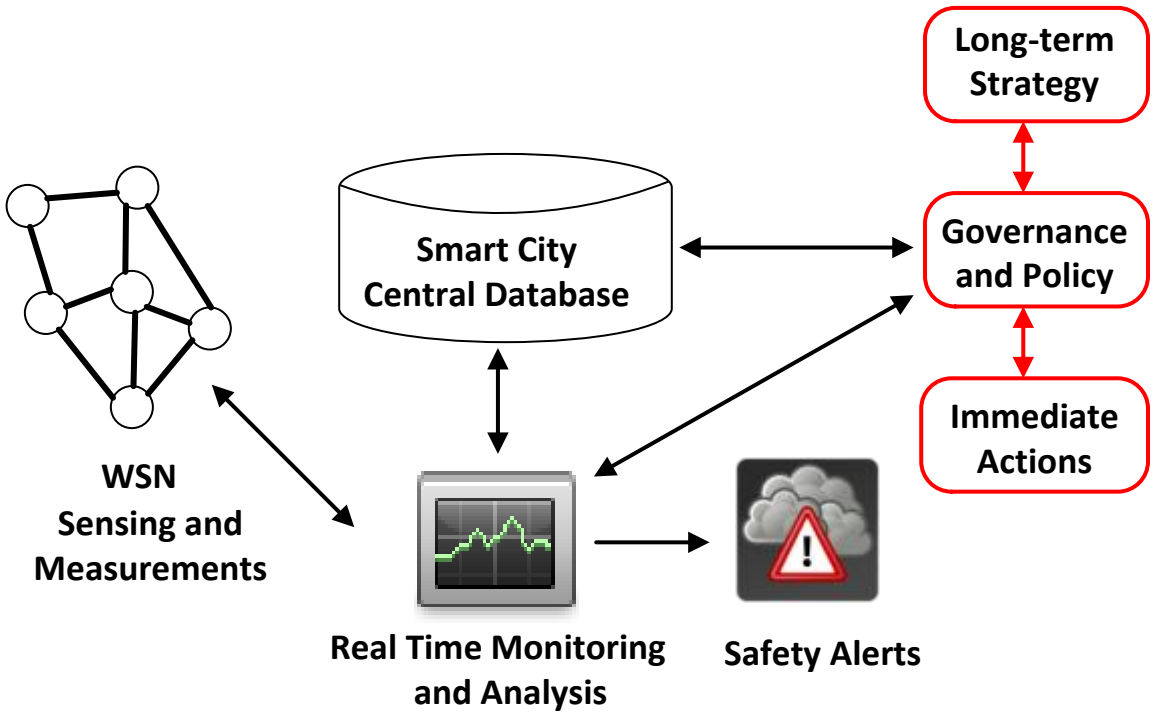


Figure 2.1: The real-time monitoring and safety alert system in a smart city.

consists of interactions between computation and physical processes. The cyber system is responsible for computation processes. The physical system comprises infrastructures and control systems where physical processes take place. These two systems are connected through a communication network. Wireless sensors are deployed at the physical system. In [Wu11], the roles of WSNs in a CPS have been thoroughly studied. A precise description of such a CPS is as follows. WSNs capture data from physical processes and deliver them to the communication network. The cyber system uses the delivered data for computations and makes control decisions. Those decisions are used to govern actuation processes. This description reveals that the system functionalities rely on the data that are initially captured from sensors. This is why the assurance of data integrity is a fundamental requirement of the system. In a CPS, data corruption is one of the major security threats to WSNs. It can result from both kinds of attacks, cyber and physical. An attacker can use compromised sensor nodes to inject bad data to a CPS. It can also place fake sensor nodes to inject false data. Tampering and physical destruction can also be used to corrupt data.

Figure 2.2 shows cyber-physical interactions and threats. As WSNs work at the lowest layer of information processing, they can be a preferred gateway for attackers. For WSNs, it is difficult to separate cyber and physical vulnerabilities.

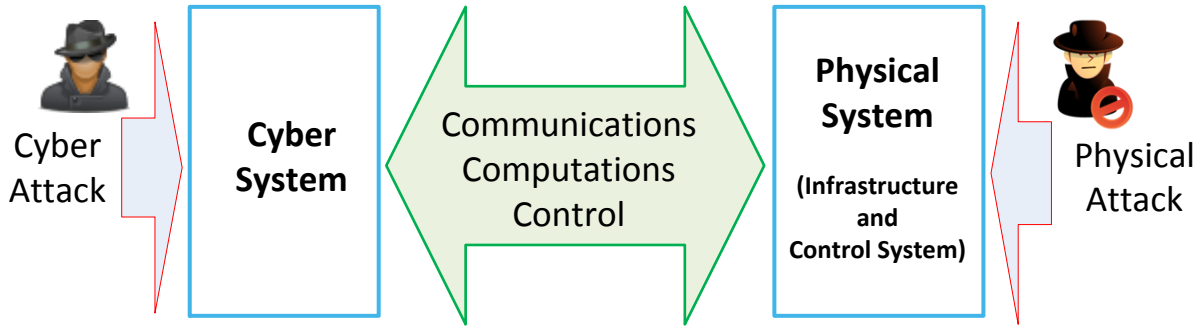


Figure 2.2: Cyber-physical threats and interactions.

There are two main ways of attacking against WSNs: physical and cyber. The major physical-attacks include physical destruction, physical tampering, environment tampering, and physical intrusion. Table 2.1 summarizes those attacks. In the physical destruction, sensor nodes are destroyed completely so that sensing operation becomes unavailable. A physical tampering can be invasive or non-invasive. For the invasive one, attackers access and modify internal structure of the devices. In a non-invasive attack, attackers get physical information without physically accessing the sensors. Significant information such as power consumption and processing time are leaked [Ben08]. These information are useful in side-channel attacks that extract secrets. In an environment tampering, the deployment area is tampered, which result in erroneous sensing. For example, an attacker can artificially change the temperature of an area. As a result, sensors will measure wrong temperatures. Physical intrusions in the deployment area can create a situation for three other types of physical-attacks. It can also create an environment that is suitable for cyber-attacks. Lots of possibilities exist such as fake sensor deployment to corrupt data, unauthorized WSN deployment to collect sensitive information, and malicious sensor node placement to corrupt routing process.

The major categories of cyber-attacks include routing attacks, denial-of-service (DoS),

Table 2.1: Summary of Common Physical-Attacks

Physical-Attack Type	**Impact on WSN**	**Remarks**
Physical Destructions	Sensing unavailable	The prime targets are the sensors that are deployed in an open environment: ITS, RTMSA applications.
Physical Tampering	Sensor malfunctioning (invasive), physical information disclosed (non-invasive, side-channel attacks)	All types of applications can be targeted. Physical intrusion in the deployment area is a prerequisite
Environment Tampering	Unreliable or misleading data	The prime targets are the sensors that are deployed for monitoring and control of environmental parameters: smart home and substation applications.
Physical Intrusion	Creates environment for physical and cyber attacks, deploys unauthorized sensing devices.	All types of applications can be targeted.

insider attack, and cyber intrusion. Routing attacks can have a number of forms: Sybil, wormhole, sinkhole, and selective forwarding. In the Sybil attack, fake identities of the same nodes are created. The same compromised node works under multiple identities. In the wormhole attack, multiple malicious nodes work together. They eavesdrop and move another area and retransmit in another area. In the sinkhole attack, a fake base station is created so that all traffic is directed to a compromised entity. Thus data can be lost and tampered. In the selective forward attack, a malicious sensor node only forwards a selected portion of received messages. This harms the sensing operation. There are different types of DoS attacks. The common idea is unnecessary consumptions of network resources and blocking of useful operations. There can be a distributed DoS (DDoS) where a number of nodes work together and cause a service outage. Hello flood attack is one of the examples of

DoS attack. Where Hello messages are broadcasted using high power to consume network resources. The insider attack is powered by node cloning. A fake replica of a valid node is created to orchestrate attacks [Wal06]. Cyber intrusion enables unauthorized access to sensor information. Table 2.2 summarizes common cyber-attacks.

Table 2.2: Summary of Common Cyber-Attacks

Cyber-Attack Type	**Impact on WSN**	**Remarks**
Routing Attacks: Sybil, Wormhole, Sinkhole, Selective Forwarding	Adversaries capture actual data (break of confidentiality), corrupts data (break of integrity), misleads sensing operations (break of availability)	The prime targets are the large-scale sensor networks: ITS, RTMSA applications.
Service Denial: DoS, DDoS, Flooding	Unnecessary consumption of resources: battery (affected lifetime), bandwidth, processor, add delays in routing (affected time criticality)	The prime targets are the large-scale sensor networks: ITS, RTMSA applications.
Insider Attack	Forgery, initiates other types of attacks	All types of applications can be targeted.
Cyber Intrusion	Compromise, privacy leak, initiates other types of attacks	All types of applications can be targeted.

2.3.3 Security Monitoring Approaches

WSNs in smart cities can be an easy target for highly motivated attackers. The most common motivations include financial benefits, privacy infiltration, impeding operations, vandalism, and sabotage. These motivations can lead to both types of attacks, cyber and physical. The role of solution approaches is to ensure the basic security requirements of the sensed data and relevant information. The term relevant information here refers to any other sensitive information than the sensed data such as sensors' energy status. Security solutions for WSNs are constrained by a number of design factors including energy

consumption, communication and computational overheads, storage capacity, and costs. Simultaneous consideration of low energy, low overheads, and low cost is always being a research challenge.

Monitoring of WSNs is a way of enhancing cyber security. There are two main types of monitoring approaches: intrusion detection systems (IDSs) and watchdog systems [Abd13, Ren16]. As intrusion events can trigger harmful activities in many ways, the deployment of intrusion IDSs is an effective security measure. In WSNs, IDSs are mainly responsible for identifying malicious sensor nodes. IDSs consume computational resources and battery energy. This incurs additional computational overheads and reduction in battery lives. Thus sensor nodes require additional resources to maintain the quality standard of sensing services. As WSNs are operated in resource-limited conditions, design and implementation of IDSs are challenging tasks.

The relevant literature of IDS in WSNs has been comprehensively studied in [Abd13] and [But14]. A discussion on the adoption of mobile ad-hoc network (MANET) IDS solutions in WSNs was also included in [But14]. Among the adoptable solutions, agent-based distributed-collaborative IDSs and clustering-based IDSs are very promising for large-scale WSNs. It consumes a significant amount of resources when each node in a WSN hosts an IDS. This consumption can be reduced by deploying zone-based IDSs [Sun07]. For zone-based IDSs, a WSN is divided into a number of non-overlapping zones. Alerts are generated and distributed by IDS agents locally inside each zone. Only gateway nodes are responsible for the final detection and network wide alarms. The concept of watchdog system was derived from the trust management in WSNs. It was originally proposed for MANETs and then adopted in WSNs. Though the objective is the same, its working mechanism is different from IDSs. In a WSN, a number of intermediate nodes are selected as watchdogs. Each watchdog overhears its neighbors' message transmission in a promiscuous mode. If it detects any irregularity in message transmission or a corrupted message from a neighbor, it reports the neighbor as misbehaving. In [Lia10], watchdog systems have been studied for misbehavior detection in both ad-hoc and sensor networks. The study reported some advantages of the watchdog mechanism over end-to-end misbehavior detection sys-

tems: (i) no reduction in the network throughput since only watchdogs are responsible for detection; other nodes do not spend resources for detection, (ii) watchdogs have the knowledge about the location of misbehaving nodes since each watchdog only overhears its neighbors. The main challenges for watchdog systems are the uncertainty of wireless channels and energy management. In [Zho15], an optimized task scheduling method has been proposed to improve the energy-efficiency of watchdog systems. It focused on two major factors: (i) selection of an optimal location for each watchdog and (ii) selection of a watchdog task frequency for each target node. Each watchdog is placed close to its target node in such a way that communications consume minimal energy. On the other hand, task frequencies are selected depending on target nodes trustworthiness. Lower task frequency is acceptable where target nodes are trustworthy. This saves energy by reducing the number of transmissions.

Besides the conventional approaches, new solutions are developed using emerging technologies. In [Wu16], a novel hierarchical attack mitigation scheme for WSNs in smart cities has been proposed. It uses software defined networking and network function virtualization technologies to mitigate ongoing attacks. It deploys two attack detection tools called data crystallization and KeyGraph.

2.4 Review of Cyber Security in Smart Grid Networks

The concept of a smart grid was developed to improve power system operations using appropriate ICTs. It adds new cyber threats and vulnerabilities to energy infrastructures [Sri12, Fou12]. For any communication network, the concern of cyber security arises when there is a possibility of targeted attacks. In the case of smart grid networks, there are a good number of reasons for being attractive to cyber-attackers [Kna13]. Cyber-attackers or unauthorized parties intend to acquire confidential information for financial benefits or reconnaissance. They can be motivated to break the integrity of messages to make the system malfunctioning. For example, erroneous control messages are capable of shutting

down the grid operation. For the authorized parties, availability of information is necessary for computation and processing tasks. It can be hampered due to cyber-attacks. Security deployment strategies need to be up-to-date to enhance an acceptable level of protection. In [Wan13], a comprehensive study of cyber security in the smart grid has been reported. It identified major cyber vulnerabilities of smart grid components in the light of relevant literature. It studied three major forms of target-oriented cyber-attacks: (i) DoS attacks, (ii) integrity attacks, and (iii) confidentiality attacks. The DoS attacks target on the availability information or messages. They cause unnecessary consumption of network resources so that authorized parties are unable to exchange information. The integrity attacks corrupt information and messages. They result in erroneous state estimation. The confidentiality attacks cause unauthorized acquisition of confidential information. They open the scope of unsolicited activities. The study also discussed two broad categories of countermeasures: network and cryptographic. Network countermeasures include traffic monitoring and filtering. They were described as an effective protection against DoS attacks. On the other hand, cryptographic countermeasures were recommended against integrity and confidentiality attacks. In a smart grid, a set of cyber-physical systems including AMI and SCADA are integrated. It also requires the deployment of some cyber-physical mechanisms [Sri12]. Physical attacks on an infrastructure can also affect cyber activities. For example, a grid service can be unavailable because of a destruction of communication devices. This can affect the decision process of the grid. Similarly, cyber-attacks are also capable of physically harming grid entities by sending erroneous control messages. The scope of smart grid network security is very large. We have chosen two of its operational components to study their security concerns: AMI and SCADA.

2.4.1 Advanced Metering Infrastructure

An AMI bridges between customers, market, and utilities. The operation of an AMI unifies all types of utilities in a household: electricity, gas, and water. It is mainly responsible for metering and billing and also delivers market related information to consumers and grid. In the literature, a smart grid AMI network architecture is described by hierarchical

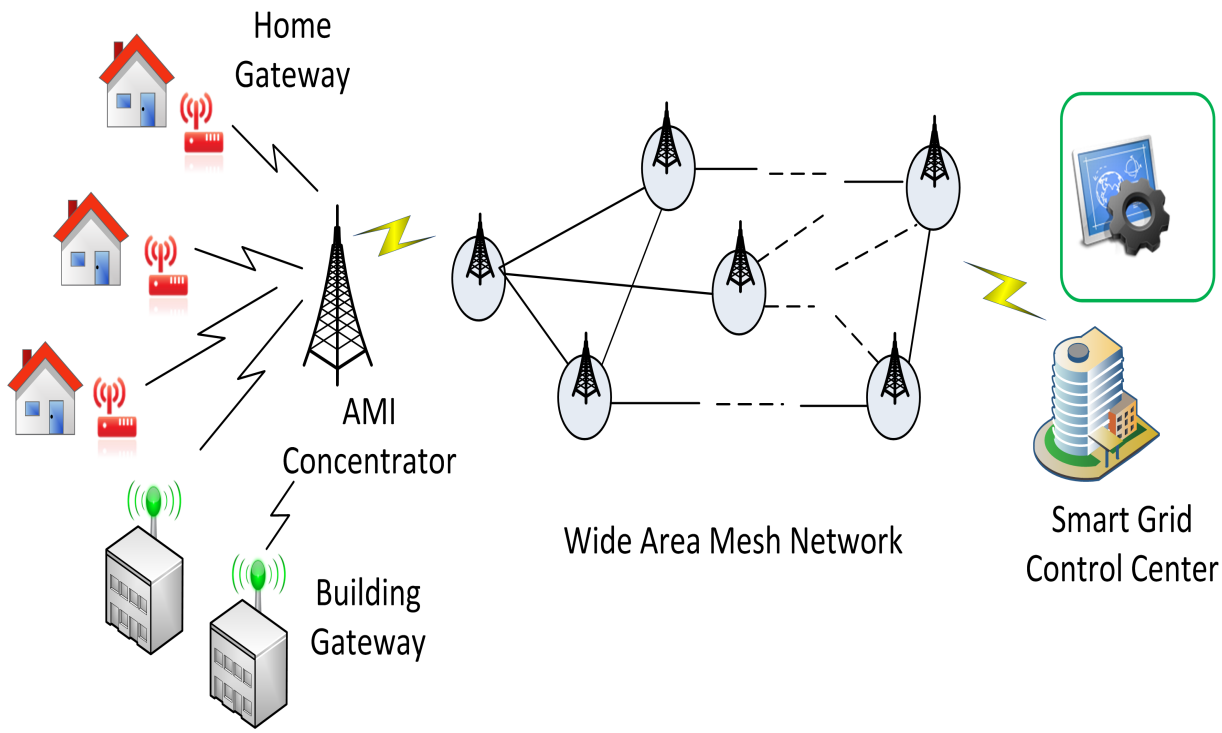


Figure 2.3: A smart grid AMI Network.

layers [Fou12, Zha11, Lu12]. Its hierarchy includes smart meters and a smart grid control center at the lowest and highest levels respectively. There are three basic layers: home area network (HAN), neighborhood area network (NAN), wide area network (WAN). In the first layer, smart sensors and smart meters collect in-home consumption information. Each home equipped with a HAN and it has a home gateway to communicate with the next layer of the network. In a building with multiple apartments, a building gateway does the same job. The second layer includes the network of home and building gateways. It is governed by an access controller or NAN gateway. The NAN gateway is hosted by a nearby AMI concentrator. The AMI concentrator collects all the acquired information from home and building gateways. The AMI concentrators periodically gather information from corresponding gateways. In the third layer, the AMI concentrators deliver collected information to the smart grid control center. A wide area mesh network takes place between the concentrators and the control center. Each concentrator regularly sends information to the control center. The third layer is a WAN of NAN gateways and the control center. An

AMI requires multihop and heterogeneous communication networks. It may include IEEE 802.11n, ZigBee, and WiMAX in different layers. The control center performs various analyses on consumers information for billing, forecasting, and policy making. Figure 2.3 shows a smart grid AMI network.

In [Zha11], a distributed intrusion detection system (DIDS) has been proposed for an AMI. The proposed DIDS applies the artificial immune system (AIS) and support vector machine (SVM). It deploys an IDS in each gateway of every layer of the AMI namely HAN IDS, NAN IDS, and WAN IDS. These IDSs work on both egress and ingress traffic passing through them to detect malicious content. The ZigBee, IEEE 802.11n, and WiMAX were considered as the communication standards for HAN, NAN, and WAN respectively. The HAN IDS has six modules. It collects consumption data using an information acquisition module. A data segmentation module partitions the collected data into suitable segments. A preprocessing module prepares the segments for detection algorithms. An analyzing module detects the suspicious events. A trained SVM/AIS algorithm is used to classify intrusions. An output module records and prints out intrusion types, attack time, and associated addresses. The whole procedure is coordinated by the control module. The NAN IDS and WAN IDS consist of similar modules to HAN IDS as well as follow the same procedure. The SVM is a machine learning technique that classifies data based on certain baseline characteristics. It solves a convex quadratic program. The AIS uses two bio-inspired modeling algorithms to classify attacks: clonal selection algorithm (CLONALG) and arti

cial immune recognition system2 parallel (AIRS2Parallel). In [Fai15], a data-stream-based IDS has been proposed to secure AMI. The proposed IDS considers an architecture comprised of three basic components: smart meter, data concentrator, and AMI head-end. Each component is equipped with an IDS. A stream mining algorithm is used to classify data for each component. In [Pat17], a collaborative intrusion detection and prevention architecture for smart grid has been proposed. The proposed architecture combines the functionalities of both network-based and host-based monitoring. It is dedicated to the smart grid's large-scale distributed heterogeneous networks. To improve the accuracy of

intrusion detection, it uses the SVM technique, network traffic ontology, and fuzzy logic. In [Car14], a framework for evaluating IDS architectures in AMIs has been proposed. The proposed framework explores costs and benefits of various IDS architectures in AMI. It compared three architectures: centralized, distributed, and embedded. It identified a number of factors that affect costs and benefits. Those factors include network density, area coverage, and the target of cyber-attacks.

2.4.2 Supervisory Control and Data Acquisition (SCADA)

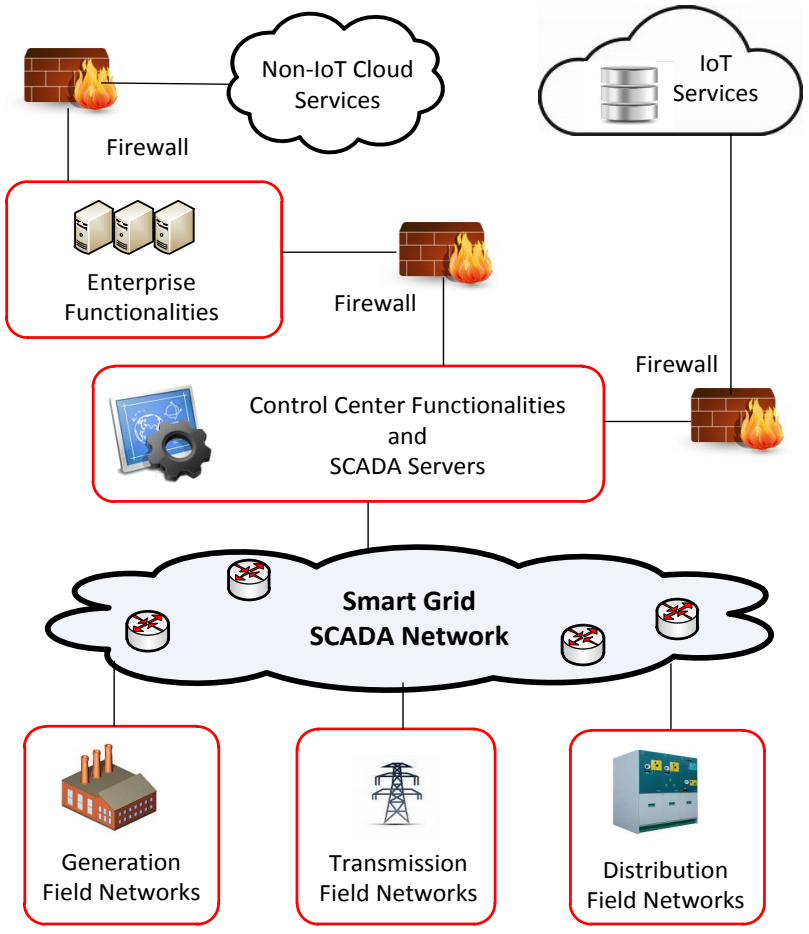


Figure 2.4: A conceptual depiction of smart grid SCADA systems in the IoT era.

The fourth generation of SCADA architectures is currently being implemented. It is increasingly adopting advanced technologies such as cloud computing and IoT. Thus,

cloud-based services are now available for supporting SCADA systems [Saj16]. In the previous generations, SCADA activities were basically confined to proprietary networks. In contrast, the current generation is mostly Internet-based. This incurs additional cyber security risks in the system. Electric power grids are one of the most prominent application areas of SCADA systems. Their operations are heavily dependent on ICTs under the smart grid environment. Smart grid architectures include the Internet-based SCADA for supporting measurements and control functionalities [Moh11]. Thus the fourth generation is compliant to the smart grid perspective. SCADA networks are deployed to serve three operational domains: generation, transmission, and distribution. As these domains are interconnected, cyber-attacks can easily propagate from one to another. Each domain is equipped with field level devices such as remote terminal units (RTUs), programmable logic controllers (PLCs), and intelligent electronic devices (IEDs). These field level devices communicate with locally deployed sensors and actuators using a field network. They acquire measurements from sensors and deliver control commands to actuators. An operational domain includes a number of geographically distributed field networks. Typically, each power system bus is supported by a SCADA node and a field network. The SCADA node works as a gateway router for the field network. The SCADA network takes place between the smart grid control center and field networks. It conveys the acquired measurements to the control center and then conveys control messages to field networks. The control center provides a wide range of functionalities such as system management, energy management, human machine interface (HMI) stations, forecasting and demand response, fault management, and asset management. The IoT technologies offer new Internet-based interfaces to ease control center functionalities. This creates opportunities for new services that use advanced data analysis and predictive analytics [Al15, Per14, Lu15]. Figure 2.4 shows a conceptual depiction of smart grid SCADA systems in the IoT era. The enterprise functionalities include business tasks such as billing and dynamic pricing. They are separated from the control center and SCADA. The non-IoT cloud services are offered to enterprise functionalities. On the other hand, the IoT services are offered to SCADA and control center functionalities. The control center and SCADA components form a control network. There are three firewalls to monitor network traffic. Two of them are placed

between smart grid and external service providers. The remaining one is placed between the enterprise and control networks. In the conventional power systems, the interface between external service providers and control network was absent. The smart grid adds new vulnerabilities to the SCADA network. Cyber-attacks can originate from both the internal and external domains.

In a smart grid environment, Internet-based SCADA systems are devised to perform measurements and control functionalities. In [Moh11], a novel Internet-based attack on smart grid SCADA system has been reported. In that attack, intruding agents alter load status information in a system. The possible impacts include inappropriate control messages, unnecessary change in demand side price information, and erroneous input to load distribution algorithms. Depending on targets, there are three types of such attack: (type I) targets power plants; it disrupts operation or generation, (type II) targets power distribution and control systems; it corrupts state information that may lead to instability, (type III) targets consumer premises; it causes increments in load that can damage the grid. Defense mechanisms to these attacks include private key encryption for unicast communications, message authentication codes, group key encryption for multicast communications, user authentication, password protected access, and firewalling of SCADA traffic. In [Gia13], a comprehensive study of data integrity attack on SCADA systems has been reported. As data integrity influences the state estimation process, it is a major security concern for an energy management system (EMS). The study focused on an unobservable low sparsity cyber-attack that requires coordination of less than five energy meters. It is infeasible to place a meter in each power flow line in a large-scale electric grid. Therefore, a phase measurement unit (PMU) placement algorithm was proposed. In [Som10], a comparison among SCADA cyber security standards has been presented. The most important comparison criteria include countermeasures, possible attacks, and threats. In [Zha16], a model has been proposed to include the impact of cyber vulnerabilities for SCADA in power system reliability. The model uses a game theoretic approach to solving the optimal security resource allocation problem. Those resources include IDSs and firewalls that are deployed to protect substations from targeted cyber-attacks. It assumes a successful cyber-attack when the deployed security resources in a substation are insufficient to combat the

adversary. Thus the loss of load probability (LOLP) is calculated as an impact of cyber vulnerabilities.

2.4.3 Trust System Placement Problems

Trust systems are compact security devices that are mainly designed to defend energy SCADA systems against cyber adversaries [Coa08, Coa10]. A trust system comprises specialized hardware and software agents. It is devised to provide a number of functionalities including firewall, IDS, encryption and authentication, and routing and switching. It offers flexible implementations depending on the utility operator's need. It intercepts and assesses SCADA traffic from both directions, ingress and egress. It initiates and distributes appropriate alert messages depending on the assessment of status messages and control commands. It has been tested for UDP/IP and TCP/IP communications [Coa10]. The conceptual development of the trust system was motivated by the future Internet-like utility Intranet. Thus it complies with smart grid communication networks. Its optimal placement brings technical and economic benefits to grid operators. Technical benefits come from the improvement in quality of protection, while economic benefits come from the reduction of expenditures. In the literature, the problem of trust system placement is commonly reported as an NP-hard problem [Gon11, Zha13b].

In [Gon11], a heuristic solution has been proposed for a known number of trust systems. The solution uses a mixed integer linear programming (MILP) formulation. Communication links are weighted by propagation delays to address the time criticality of trust systems' response. Those delays are calculated considering an optical fiber network for the SCADA backbone. In the solution, the SCADA network is segmented into small pieces called domains or compartments. Those domains are formed based on preset timing thresholds. The number of segments is determined by the solution. It is not always possible to meet the time criticality due to resource constraints.

In [Zha13b], the trust system placement problem has been studied for a layered network architecture. The architecture was devised for smart grid AMIs. The study formulated a set packing problem to compute the hierarchical set of trust nodes. It introduced a constraint

named tolerance level. The tolerance level sets the maximum number of intermediary nodes between two trust nodes in each layer. A heuristic solution was proposed for secure routing between different layers. It considered the shortest path routing strategy.

The existing heuristics are appropriate for certain scenarios of smart grid networks: (i) a given number of trust systems with timing thresholds [Gon11], and (ii) a layered network architecture with tolerance levels [Zha13b]. There are still many more possible scenarios that require further contributions where a few such cases are considered for our current research.

2.5 Review of Cloud-Based Security Services

Cloud computing has introduced a number of emerging service paradigms [Koc11]. The SECaaS is a promising cloud-based service model [csa11, Hus11]. It is devised to offer cyber security solutions to customers from both domains, cloud and non-cloud. The main idea of SECaaS is to exploit the potential of cloud computing to offer security solutions. It is an attractive option to enterprise clients for its managed service packages. The main advantages of SECaaS are simplicity and cost-effectiveness. The cloud security alliance (CSA) has defined ten main categories of services for SECaaS: (i) identity and access management (IAM), (ii) data loss prevention (DLP), (iii) web security, (iv) e-mail security, (v) security assessment, (vi) intrusion management, (vii) security information and event management (SIEM), (viii) encryption, (ix) business continuity and disaster recovery (BCDR), and (x) network security. The IAM provides verifiable identities to people, processes, and systems that are eligible for accessing enterprise resources. It also defines the level of access for each entity. The DLP uses rule-based actions and monitoring to protect data that are at rest, in motion, and in use. The web security prevents malware from entering the enterprise through online activities. The e-mail security controls the content of inbound and outbound e-mails. The security assessment provides third-party audits of services and systems. It includes tests that follow well-defined standards such as national institute of standards and testing (NIST) and international organization for standardization (ISO).

The intrusion management provides packet inspection that detects and prevents intruders. It has two main architectural options: traffic monitoring (where the SECaaS provider collects packets for analyzing and alert initiation) and traffic filtering (where packets are routed through the SECaaS provider). The SIEM collects and analyzes log and event information. It provides real-time reports and alerts on incidents. The encryption provides cryptographic services such as key management and distribution. The BCDR ensures operational resiliency when natural or man-made disasters or disruptions occur. The network security deals with threats and security controls for resources. Its key requirement is the visibility of traffic.

There are four main deployment models for cloud-based services: public cloud, private cloud, community cloud, and hybrid cloud. A public cloud is open for public use. A private cloud is solely operated for a particular organization. It can be hosted and managed by a third party. In a community cloud, the infrastructure is shared by several organizations. A hybrid cloud comprises two or more types of clouds. For an enterprise customer, we found that the private cloud model is the most suitable one. This is because it offers a dedicated service provider. A smart city infrastructure can be considered as an enterprise customer. It is much simpler when a service provider is dedicated to a particular customer since it enhances the quality of service (QoS). Figure 2.5 shows a summarized view of the SECaaS managed services.

2.5.1 Managed Security Services

The major research issues in SECaaS includes novel services and architectures, novel application areas, and software tools. The SECaaS model aims to deliver faster, simpler, and cost-effective services. Figure 2.6 provides an idea of the cost-effectiveness of cloud computing in security management. It shows comparative estimated costs for a facility with one hundred SPARC T4 cryptographic processors [ora15]. These costs are obtained from the IBM smart cloud simulator that uses three years true cost analysis (TCA) [ibm15]. The IT costs include expenditures associated with facilities, labor, hardware, and software. The operating costs and strategic costs are associated with a number of factors such as business

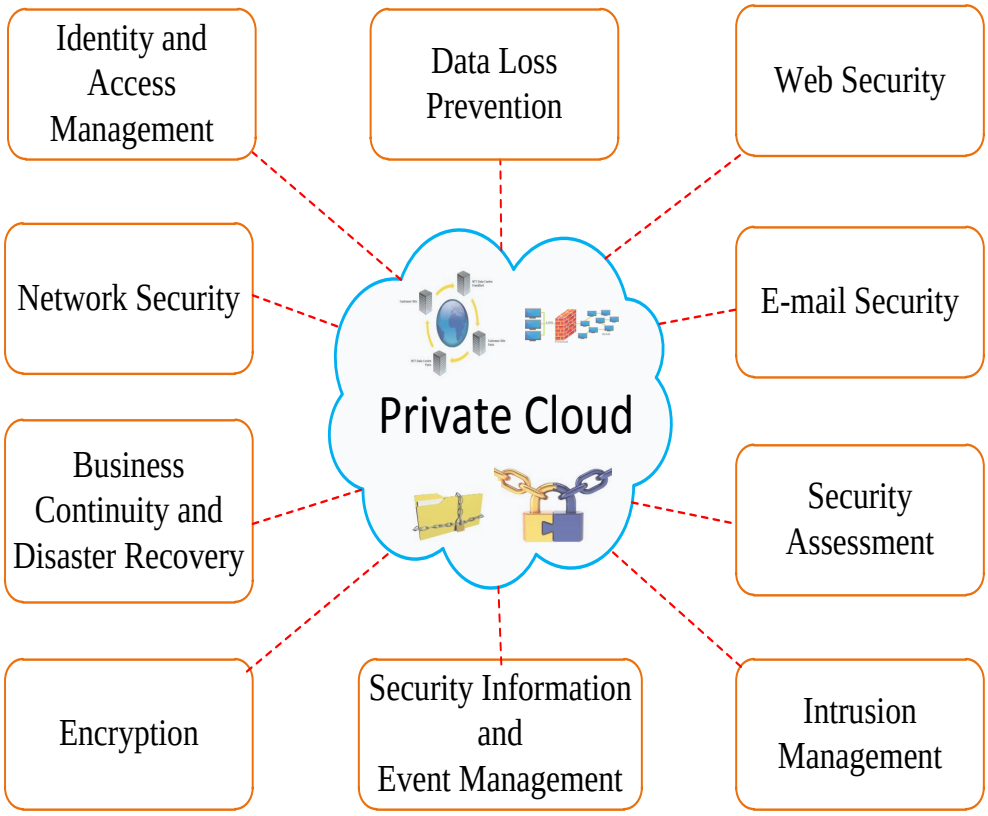


Figure 2.5: The SECaaS at a glance.

agility, planned downtime, and unplanned downtime. The conventional owned information technology (IT) infrastructure is much more expensive than cloud-based solutions.

The emergence of SECaaS in the future was predicted in [Hwa10a]. It has proposed a reputation system between service providers and data owners. The reputation system relies on a trust-delivery network over multiple data centers. It also made some recommendations for the future implementations: (i) data coloring and software watermarking techniques, (ii) access control for sensitive data in both public and private clouds, (iii) live migration of the virtual machines (VMs) that are created for building DIDS, and (iv) defense schemes to protect user data from servers. The first recommendation is made for protecting shared data objects and massively distributed software modules. An insulation between owner and service provider is provided using data coloring. A cloud service provider (CSP) can

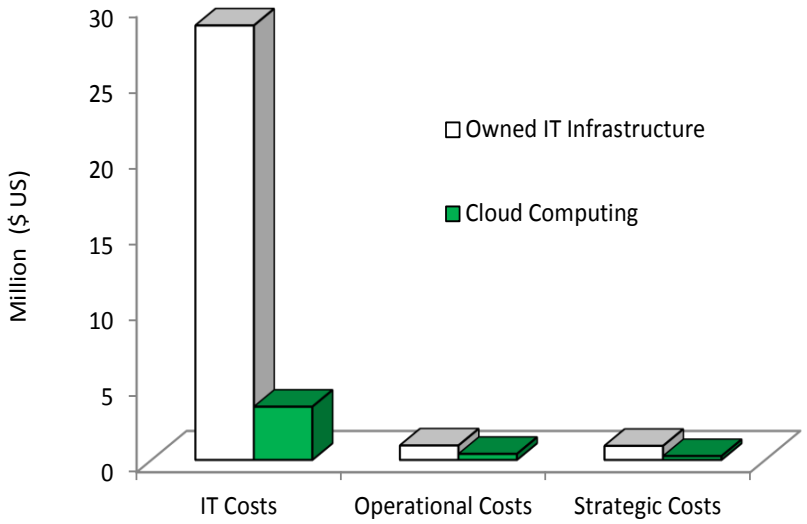


Figure 2.6: Comparative costs in security management.

deploy multiple IDS VMs at various resource sites such as the data centers. The DIDS design demands trust negotiation among PKI domains. Its security policy needs to be updated periodically.

In [Get12], opportunities and concerns of the SECaaS paradigm have been studied where its emphasis was managed security services. It discussed four possible security tools that can be offered as a service: e-mail filtering, web content filtering, vulnerability management, and identity management. The e-mail filtering includes malware scanning, anti-phishing, and anti-spamming. The web content filter runs on a proxy server of SECaaS provider to perform screening, monitoring, and analyzing of contents. The vulnerability management includes cloud-based anti-virus and virtual IDS. The identity management provides interoperable identity solutions.

In [Rak13], the SECaaS model has been studied for specifying service level agreement (SLA)-aware parameters. It reported three main phases in SLA lifecycle: (i) negotiation, (ii) monitoring, and (iii) enforcement. In the negotiation phase, SLA is not fully defined. Customers evaluate service specification, performance, and costs. On the other hand, a service provider evaluates requested services and matches them with the required capabili-

ties. It also assesses possible risks. In the monitoring phase, the SLA is signed. This phase also defines penalties for SLA violations. In the enforcement phase, actions are needed to respect the SLA i.e., maintain a quality of security service (QoSS).

The SECaaS model opens the possibilities of cloud-based PKI or cloud-based TTP [Bro11, Mes12]. The challenges of such PKI and corresponding solutions have been discussed in [Bro11]. The discussion also revealed the necessity of PKI in a cloud environment for secure machine-to-machine (M2M) communications. The cloud-based PKI includes an enterprise CA that generates certificates for a restricted community such as an organization or an application. The main advantages of an enterprise CA include: it provides a fully consumer controllable PKI, on-demand certification, and on-demand revocation. However, there are some implementation challenges. Firstly, a VM snap shot contains all confidential data including private keys. As a result, private keys lifetime becomes shorter. Secondly, transport layer security (TLS) connection is compromised. Thirdly, session keys can also be leaked. As session keys are generated using random number generator, its internal state can be revealed. This, in turn, may help attackers in predicting session keys. The recommended solutions to these challenging problems are: (i) the removal of replication of VMs; snapshots can only be taken in the trusted hosts and (ii) every power ON should be issued a new certificate and every power OFF cause revocation of certificate. The major requirements for cloud-based TTP have been identified in [Mes12]. Firstly, it requires the ability to transform top-policies to low-level component requirements such as quality, measure, configuration, and compatibilities. Secondly, it requires continuous monitoring for quality assurance of security services. Thirdly, it is important to identify vulnerabilities and solutions. Finally, it requires historical recording of security information and storage. The encryption as a service (EaaS) is a SECaaS category that deals with cloud-based encryption solutions. In the available literature, it is mainly proposed for data storage applications [Hua11, Kan14]. Its major research issues include business model, encryption and communication overheads, along with users' privacy. In [Hua11], the business model considered encryption and decryption as separate services. The reason behind this separation is the privacy concern of users' data. The model is aware of the risk of disclosure where encryption and decryption keys are stored by the same service provider.

In this model, a user needs to have SLA with two different service providers. The design and implementation of an encryption service named ESPRESSO have been reported in [Kan14]. ESPRESSO stands for encryption as a service for cloud storage systems. It uses the advanced encryption standard (AES) algorithms. Its design is dedicated to cloud storage systems where users data are protected. It offers server-side encryption to users and exposes additional communication overhead.

In [Sub11, Khu12, Tit13], SECaaS architectures have been proposed for securing smart phones. The important functional components of those architectures are replicas and proxy servers. They are required to avail smart phone traffic in the cloud domain. A replica in the cloud is created for each smart phone. The replica is synchronized with the actual smart phone's operating system files and other authorized files. It maintains a similar state with the physical smart phone. A server farm becomes the interface between the cloud and the smart phone. A proxy server controls traffic in and out of the smart phone with the replica being a VM. The architecture in [Sub11] described three main components of a CSP: synchronization module, interpreter, and controller. It was designed to offer a number of cloud-based security functionalities such as anti-virus, secure browsing, operating system integrity checks, remote wiping and versioning, policy control, and secure storage. The architecture in [Khu12] introduced a cloud-based IDS for Android smart phones. It requires an emulator to emulate a smart phone traffic. Its functionalities include memory scanner, anomaly detector, and antivirus software. A proxy server is required to forward the smart phone traffic to the emulator. The architectures in [Tit13] focused on malware detection in smart phones that included a virtual replica hosted by the physical device. A security manager coordinates all the functionalities: emulation, storages, interactions, reactions, and malware detections. It requires three modules to be deployed by a smart phone: reaction enforcement, user interaction recorder, and image creation.

A cloud-based intrusion detection service has been proposed in [Alh12]. It focused the public cloud consumers who create their own virtual private clouds. The intrusion detection system as a service (IDSaaS) framework has been developed to protect the virtual private cloud. It offers full control to the public cloud consumers and independent of the

implementation model. The IDSaaS includes application-oriented monitoring and reaction on consumer's virtual private cloud. It also identifies the attack type and supports on-demand elasticity since IDS core changes with the amount of traffic. The IDSaaS architecture consists of five components: (i) intrusion engine to pre-process network packets based on the attack signature, (ii) output processor to feed log files into the event database, (iii) event database to store the formatted events, (iv) alert management to provide the graphical user interface (GUI) facility, and (v) Rule-base manager to update the rule-base to cope with new threats. A generic framework for cloud-based IDS has been proposed in [Yas12]. The framework is designed to monitor cloud networks to detect malicious activities. It has three basic components: user data collector, cloud service component, and a cloud intrusion detection component. The collector is responsible for collecting users' information before the cloud IDS. The service component either delete or forward information to the cloud IDS. It analyzes and translates information for the detection component. The detection component performs pattern matching, reports activities, accesses attack signature database, and user data database.

2.5.2 Monitoring-as-a-Service

The SECaaS model can be combined with the cloud-assisted IoT to meet the demand of the future. This can benefit security monitoring of smart city infrastructures where IoT technologies are deployed. In [Men13], the monitoring-as-a-service architecture has been proposed for a cloud environment. The same concept can be adopted for cloud-based monitoring of non-cloud applications. This type of adoption requires highly available cloud radio access networks. This requirement complies with the context of smart cities. The emerging concept of collaborative security adds a new dimension to security monitors [Pan16, Men15]. It can be applied in the cyber protection of smart cities. The collaborative security monitoring is a proven mechanism for enhancing accuracy in the filtering of malicious traffic. It is achieved through the collaboration among monitoring entities placed in different locations in a network. The monitors are required to share a certain set of information on their traffic.

2.6 Summary

In this chapter, the state of art of cyber security monitoring in smart city infrastructures is surveyed. The main focus was to explore research issues and proposed solutions in the light of the currently available literature. It included three major areas: cyber security in WSNs, cyber security in smart grid networks, and cloud-based security services. In the context of smart cities, the most frequently addressed topics include cyber-physical vulnerabilities, cryptographic techniques, network traffic monitoring and filtering, anomaly detection, and security resource management. For WSNs, the limitation of computational capability and battery life are two major issues in security implementations. For smart grid networks, adoption and adaptation are the two most important parts in the designing cyber security solutions. It requires the adoption of new security paradigms and adaptations of the existing solutions. Cloud computing has opened the door to new possibilities with potential that can be exploited to offer managed security services to critical infrastructures. Its relevant architectural frameworks that are proposed in the recent years have been studied.

Chapter 3

Watchdog System Placement in Wireless Sensor Networks

3.1 Introduction

In a smart city, WSNs are deployed to support a wide range of applications. Such sensor networks provide economical and faster solutions to data collection. Smart cities integrate a number of cyber-physical systems through a communication network. WSNs are used to capture data from physical processes and to deliver them to the communication network [Wu11]. The acquired data from sensing operations are utilized to govern actuations. This basic concept is adopted to many emerging applications that are devised to create a smart city environment [Bat12]. As those applications deploy large-scale WSNs in an open environment, sensing nodes are more exposed to cyber-physical vulnerabilities. This intensifies the security concerns regarding the integrity of sensed data. For example, outdoor sensors can be an easy target to disrupt the real time monitoring systems in a city. Any disruption in an RTMSA system affects the public safety. The behavioral monitoring of sensor nodes is a way of detecting corrupted entities in a WSN [But14, Lia10, Zho15].

A watchdog system is a method of behavioral monitoring of sensor nodes. In such a system, a number of sensor nodes are selected as watchdogs. In the literature, it is considered as an effective countermeasure to various cyber-attacks such as DoS, Sybil,

sinkhole, and selective forwarding. Watchdogs are deployed to detect misbehaving nodes in a WSN. Each watchdog is responsible for its single hop neighbors. It periodically sends behavioral reports to the base station (BS). It is also responsible for event driven reporting when anomalies are detected. There are two basic mechanisms of collecting behavioral information: overhearing and communicating [Lia10, Zho15, Abd13, Ren14]. The former exploits the broadcasting nature of wireless channels as it overhears neighbors. The latter exchanges messages with neighbors for behavioral monitoring. As watchdogs are mainly dedicated to the monitorial tasks, sensing operations lose resources. Optimal selection of watchdogs can reduce resource consumptions in monitorial tasks.

Most of the previous studies in this area focus on energy-awareness in selecting watchdogs [Zho15, Abd13, Dua14, Mon15]. In [Zho15], a stochastic solution to the watchdog optimization was proposed to reduce energy consumption. Watchdogs were located based on the probability of attacks. In a realistic scenario, that probability is an unknown parameter or expressed in other words, a security designer does not have the direct knowledge. In [Dua14], every sensor node was assumed to be a watchdog for its single hop neighbors. It is a resource intensive solution since it incurs computational and energy overheads that affect all sensor nodes. In [Mon15], monitoring nodes were selected in such a way that each sensor node is monitored by at least two of them. It considered the residual energy as a selection criterion and used a regular grid topology for evaluation. This is also a resource intensive solution for other types of realistic WSN topologies due to the required number of watchdogs. As realistic topologies are application specific, the regular grid has a limited scope.

Note that our coverage problem is not to be confused with the k-coverage problem studied in [Hua05]. In the k-coverage problem, sensor nodes are the monitoring entity. They are deployed for monitoring certain events. In contrast, in our case, sensor nodes are the monitored entity. Their responsive behaviors are monitored by watchdogs. In the k-coverage problem, coverage is defined as the region or area covered by the deployed sensors to observe some events. For example, the art gallery problem described in [Hua05], where sensors are deployed to provide surveillance of visitors. The parameter k measures

the number of sensor nodes that are capable of monitoring the same event or a certain spot of a covered area. In our case, the coverage of a watchdog means the number of sensor nodes located in its coverage area. In a simple manner, it can be said, sensor nodes monitor the events; while watchdogs monitor the sensor nodes. Therefore, our problem basically addresses how the integrity of the deployed monitors (sensors) can be monitored in an efficient manner. Another clarification includes that our current work considers only static topologies, meaning that mobile sensors are not included. Our focus is the static WSN deployments for smart city applications such as the RTMSA.

In this chapter, we present four optimization models for watchdog selection in a WSN. Three of the models consider resource unconstrained cases. The models are developed considering two major facts: overlapping and coverage. An overlapping occurs when a sensor node is monitored by multiple watchdogs. It is an inevitable event due to the propagation characteristics of wireless signals. Though monitoring performance may improve, it results in additional consumption of resources and unfairness issues. The full coverage occurs when each sensor in a WSN is either monitored by at least one watchdog or working as a watchdog. Our models exploit the single hop neighborhood information in selecting watchdogs. It is a realistic approach since sensor nodes exchange *Hello* messages at the provisioning stage to discover their single hop neighbors. Each sensor node is a network resource that becomes a security resource when it works as a watchdog. Our models introduce novel formulations that help with the understanding of watchdog selection problems from a resource management perspective.

3.2 System Model and Problem Definition

Watchdogs are selected by the BS at the network provisioning stage. The BS is assumed to be informed about the connectivity of every node in the WSN. Initially, each sensor node scans their single hop neighbors and sends the information to the BS. Figure 3.1 shows a simple example that consists of two watchdogs. Their coverage areas are shown by the large circles where such areas are dependent on the sensing range of watchdogs. The

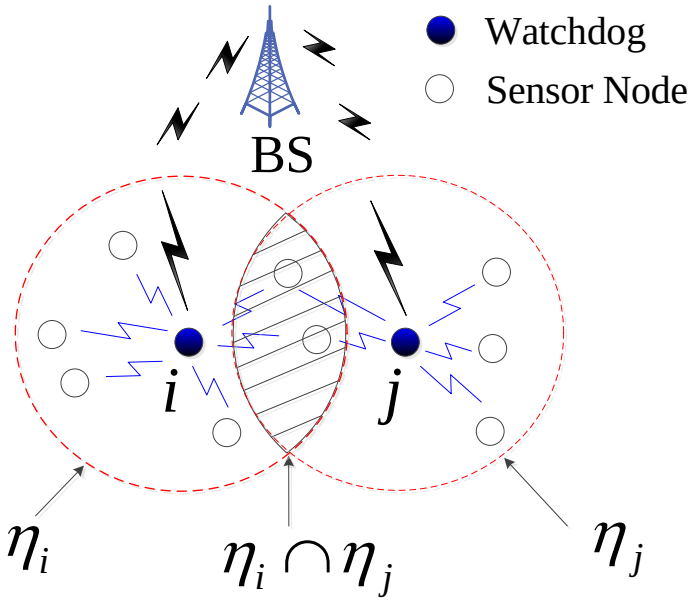


Figure 3.1: A simple example of a WSN covered by two watchdogs.

overlapping between coverages is shown by the shaded area.

The coverage set of a watchdog includes itself and its single hop neighbors. The coverage sets of watchdogs i and j are denoted by η_i and η_j, respectively. The individual coverages of watchdogs i and j are as follows,

$$f_i = |\eta_i| \quad \text{and} \quad f_j = |\eta_j|. \tag{3.1}$$

The overlapping between coverages is given by,

$$h_{ij} = |\eta_i \cap \eta_j|. \tag{3.2}$$

Therefore, the overall coverage from two watchdogs i and j is given by,

$$|\eta_i \cup \eta_j| = |\eta_i| + |\eta_j| - |\eta_i \cap \eta_j| = f_i + f_j - h_{ij}. \tag{3.3}$$

The overall coverage for two watchdogs is expressed in terms of individual coverages and overlapping. This expression is the basis of our developed optimization models. Note

that the overlapping here is not just the overlapping between two areas. It is the number of sensor nodes located inside an overlapped area.

3.3 Optimization Models

In this section, we present four optimization models for watchdog selection. The set of nodes in a WSN is denoted by V_{wsn}. The first model limits the overlapping for each sensor up to a certain number. The second model heuristically targets the full coverage with minimal overlapping. The third model tries to use minimal resources and avoids non-linear terms. The fourth model is a resource-constrained model, which is an extension of the first one. The decision variable for these models is $Z = (z_i)_{N_{wsn}\times 1}$, where N_{wsn} is the number of nodes in a WSN. It is a node incidence vector such that,

$$z_i = \begin{cases} 1, & \text{if node } i \text{ is selected as a watchdog;} \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

3.3.1 Limited Overlapping Model

The limited overlapping (LO) model is a quadratic assignment problem (QAP). It maximizes the overall coverage. The first approach can be formulated as follows.

$$\text{(QAP1)} \quad \max_{Z} \sum_{i=1}^{N_{wsn}} f_i z_i - \frac{1}{2} \sum_{i=1}^{N_{wsn}} \sum_{j=1, j\neq i}^{N_{wsn}} h_{ij} z_i z_j, \tag{3.5}$$

subject to

$$\sum_{j=1, j\neq i}^{N_{wsn}} \kappa_{ij} z_j \leq 2, \qquad \forall i \in V_{wsn}, \tag{3.6}$$

$$\kappa_{ij}(z_i + z_j) \leq 1, \qquad \forall i, j \in V_{wsn}, i \neq j, \tag{3.7}$$

where

$$\kappa_{ij} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ are single hop neighbors;} \\ 0, & \text{otherwise,} \end{cases} \tag{3.8}$$

$$z_i \in \{0, 1\}, \qquad \forall i \in V_{wsn}. \tag{3.9}$$

The objective given in (3.5) expresses the overall coverage. The linear term associated with individual coverages. The quadratic term associated with overlapping between coverages. Constraint (3.6) ensures that a sensor node will be monitored by maximum two watchdogs. Note that (3.5) is only valid for (3.6). For overlapping more than two watchdogs, the expression of coverage will incur cubic and other higher order terms. Constraint (3.7) ensures that two watchdogs cannot be single hop neighbors.

It is necessary to linearize QAP1 to reduce the time complexity. We use the linearization technique described in [She07]. The linearized objective function is given by,

$$\max_{Z,\widetilde{Z}} \sum_{i=1}^{N_{wsn}} f_i z_i - \frac{1}{2} \sum_{i=1}^{N_{wsn}} \sum_{j=1, j\neq i}^{N_{wsn}} h_{ij} \zeta_{ij}. \tag{3.10}$$

The linearization technique incurs a set of auxiliary variables $\widetilde{Z} = (\zeta_{ij})_{(\sum_i^{N_{wsn}} \sum_{j,j\neq i}^{N_{wsn}} \mathbf{1})\times 1}$. It can be termed as the overlapping incidence vector such that,

$$\zeta_{ij} = \begin{cases} 1, & \text{if both } i \text{ and } j \text{ are selected as watchdogs;} \\ 0, & \text{otherwise.} \end{cases} \tag{3.11}$$

The relationship between the auxiliary and decision variables is defined by the following additional constraints:

$$\zeta_{ij} \geq z_i + z_j - 1, \qquad \forall i, j \in V_{wsn}, i \neq j, \tag{3.12}$$

$$\zeta_{ij} \geq 0, \qquad \forall i, j \in V_{wsn}, i \neq j. \tag{3.13}$$

3.3.2 Full Coverage Heuristic

The full coverage heuristic (FCH) tries to minimize the impact of overlapping. It solves two of optimization problems. The first one is a quadratically constrained linear programming (QCLP). The second one is a pure linear programming (LP). The first problem is solved to find the set of non-overlapping watchdogs that maximizes the coverage. Two watchdogs are said to be non-overlapping if their coverage sets don't have any common elements. The first problem is described as follows.

$$\text{(QCLP1)} \qquad \max_{Z} \sum_{i=1}^{N_{wsn}} f_i z_i, \tag{3.14}$$

subject to

$$\sum_{i=1}^{N_{wsn}} \sum_{j=1, j\neq i}^{N_{wsn}} h_{ij} z_i z_j = 0, \tag{3.15}$$

$$z_i \in \{0, 1\}, \qquad \forall i \in V_{wsn}. \tag{3.16}$$

Objective (3.14) expresses the overall coverage for the non-overlapping set while constraint (3.15) ensures zero overlapping between coverages. To reduce the time complexity, the quadratic constraint (3.15) is replaced with the following equivalent linear constraint,

$$h_{ij}(z_i + z_j) - h_{ij} \leq 0, \qquad \forall i \neq j, h_{ij} \geq 0. \tag{3.17}$$

From QCLP1, the set of non-overlapping watchdogs v_{wd1} is obtained. The set of sensor nodes monitored by v_{wd1} is v_{c1}. Therefore, the overall coverage from the non-overlapping watchdogs is given by,

$$V_{no} = \{v_{wd1} \cup v_{c1}\} = \bigcup_{i \in v_{wd1}} \eta_i. \tag{3.18}$$

As $V_{no} \subseteq V_{wsn}$, the second problem is required to be solved to achieve the full coverage. It is given as follows.

$$\text{(LP1)} \qquad \max_{Z} \sum_{i=1}^{N_{wsn}} \widetilde{f}_i z_i, \tag{3.19}$$

where

$$\widetilde{f}_i = \begin{cases} f_i - |\eta_i \cap V_{no}|, & \text{if } i \in \{V_{wsn} \setminus V_{no}\}; \\ 0, & \text{otherwise,} \end{cases} \tag{3.20}$$

subject to

$$z_i = 0, \qquad \forall i \in V_{no}, \tag{3.21}$$

$$\kappa_{ij}(z_i + z_j) \leq 1, \qquad \forall i, j \in \{V_{wsn} \setminus V_{no}\}, i \neq j, \tag{3.22}$$

$$z_i \in \{0, 1\}, \qquad \forall i \in V_{wsn}. \tag{3.23}$$

Note that LP1 only deals with the nodes that are not included in V_{no} and constraint (3.21) ensures the fact. In addition, the term $\widetilde{f}_i$ is introduced to modify coverages based on the uncovered nodes. Objective (3.19) tries to select the rest of the nodes as watchdogs. Constraint (3.22) ensures that two watchdogs cannot be single hop neighbors. The set of watchdogs computed from LP1 is v_{wd2}. Algorithm 3.1 shows the details of the FCH model.

3.3.3 Linear Resource Minimization Model

Initial formulations of LO and FCH models contain non-linear terms with the major concern being overlapping. They are required to be linearized. We introduce a linear formulation that provides the full coverage and aims to use a minimal amount of resources. The linear resource minimization (LRM) model is given as follows.

Algorithm 3.1 Full Coverage Heuristic

1: **Initialization** $v_{wd} = \varnothing, V_{no} = \varnothing$;

2: $v_{wd1} \leftarrow$ **Solve QCLP1**; // Non-overlapping watchdogs

3: Compute v_{c1}; // Sensor nodes watched by v_{wd1};

4: Compute V_{no}; // Non-overlapping coverage set

5: $v_{wd2} \leftarrow$ **Solve LP1**; // The rest of the watchdogs

6: $v_{wd} = \{v_{wd1} \cup v_{wd2}\}$; // The final watchdog set

7: **return** v_{wd}

$$\text{(LP2)} \qquad \min_{Z} \sum_{i=1}^{N_{wsn}} z_i, \tag{3.24}$$

subject to

$$z_i + \sum_{j=1, j\neq i}^{N_{wsn}} \kappa_{ij} z_j \geq 1, \qquad \forall i \in V_{wsn}, \tag{3.25}$$

$$\kappa_{ij}(z_i + z_j) \leq 1, \qquad \forall i, j \in V_{wsn}, i \neq j, \tag{3.26}$$

$$z_i \in \{0, 1\}, \qquad \forall i \in V_{wsn}. \tag{3.27}$$

Objective (3.24) tries to minimize the resource requirement. Constraint (3.25) ensures there will be at least one watchdog in each single hop neighborhood. The constraint addressing prevention of single hop between watchdogs appears in (3.26). Note that LP2 does not consider any bound for overlapping as such analysis incurs non-linearity to a model.

3.3.4 Resource-Constrained Model

The resource-constrained model is an extension of the LO model that also requires linearization. It maximizes the overall coverage for a given number of watchdogs (N_{wd}) with

the only difference being the resource-constrained. The linear form is given as follows.

$$\max_{Z,\tilde{Z}} \sum_{i=1}^{N_{wsn}} f_i z_i - \frac{1}{2} \sum_{i=1}^{N_{wsn}} \sum_{j=1, j\neq i}^{N_{wsn}} h_{ij} \zeta_{ij}. \tag{3.28}$$

subject to

$$\sum_{i=1}^{N_{wsn}} z_i \leq N_{wd}, \tag{3.29}$$

$$\sum_{j=1, j\neq i}^{N_{wsn}} \kappa_{ij} z_j \leq 2, \qquad \forall i \in V_{wsn}, \tag{3.30}$$

$$\kappa_{ij}(z_i + z_j) \leq 1, \qquad \forall i, j \in V_{wsn}, i \neq j, \tag{3.31}$$

$$\zeta_{ij} \geq z_i + z_j - 1, \qquad \forall i, j \in V_{wsn}, i \neq j, \tag{3.32}$$

$$\zeta_{ij} \geq 0, \qquad \forall i, j \in V_{wsn}, i \neq j, \tag{3.33}$$

$$z_i \in \{0, 1\}, \qquad \forall i \in V_{wsn}, \tag{3.34}$$

$$\zeta_{ij} \in \{0, 1\}, \qquad \forall i, j \in V_{wsn}, i \neq j. \tag{3.35}$$

3.4 Numerical Results and Analysis

In this section, we present numerical results that are obtained from case studies using realistic random topologies (see Appendix A for details). The WSN topologies are generated

for a 40m × 40m area using GENSEN [Cam07]. Two of topologies comprise exactly the same set of sensor nodes but have different sensing ranges 6 and 8 meters. Two of topologies are with the same sensing range but have different number of sensor nodes 70 and 100. An evaluation of the performance of the four optimization models is outlined in this section. All experiments are implemented using a MATLAB-CPLEX integrated solver on a desktop machine with Intel Core i3 3.30 GHz CPU and 4 GB RAM. The solver combines two commercially available optimization software MATLAB and CPLEX [mat, cpl].

3.4.1 Study of Resource-Unconstrained Problems

Figures 3.2, 3.3, and 3.4 depict the results obtained from different experimental settings. The Watchdogs are represented by the small solid circles. Their coverage areas are indicated by the large circles and the rest of the small circles outline sensor nodes. These results are summarized in Figures 3.5(a), (b), and (c). Figure 3.5 also includes results for the non-overlapping watchdog selection strategy to have a more comprehensive explanation of FCH. Figure 3.5(a) shows a comparison of watchdog requirements. Though the formulation of the LRM model aims to minimize resource requirements, it demands more resources than others. This is because of the ignorance of the overlapping phenomenon. The inclusion of the overlapping phenomenon adds non-linearity to a model. For the lower sensing ranges, FCH requires less amount of resources than LO and in the higher sensing range, LO requires less amount of resources than FCH. In all cases, the non-overlapping selection requires the least amount of resources but it does not provide the full coverage. In general, resource requirements are reduced in the higher range. Figure 3.5(b) shows the comparative coverages. For each topology, all three models provide the full coverage while the non-overlapping selection provides coverages above 93%. For the remaining coverages, FCH acquires additional 5-6% sensor nodes as watchdogs. Figure 3.5(c) shows a comparison of overlapping of coverages however the non-overlapping selection does not appear. LO and FCH exhibit the same relative trend as Figure 3.5(a). More watchdogs are required for more overlapping. FCH suffers from excessive overlapping for the higher sensing range. For LRM, overlapping increases along with network density increased. The network density is

increased either by adding more nodes or by increasing sensing range. Though LRM seems to have a lower overlapping than LO in the first case, it results a significant number of nodes that are monitored by more than two watchdogs.

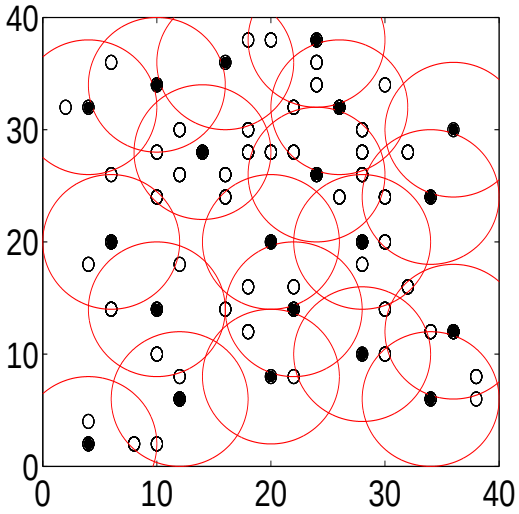
Table 3.1 shows a comparison among the optimization models in terms of average computation times. For each combination of experimental setup, the computation time is observed for 100 runs. This reveals the higher computational overhead for LO model while two other models exhibit much lower overheads.

The experiments are conducted considering a circular sensing range for each sensor node. In a practical WSN, the sensing range may not be circular due to obstacles and imperfections of the hardware [Hwa10b]. It may require more overlapping between watchdogs to achieve the full coverage that can affect the full coverage of the LO model. The two other models can still provide the full coverage. FCH initially selects the non-overlapping watchdogs. It relaxes the overlapping constraint in the final stage to ensure the full coverage. On the other hand, LRM achieves the full coverage because of constraint (3.25).

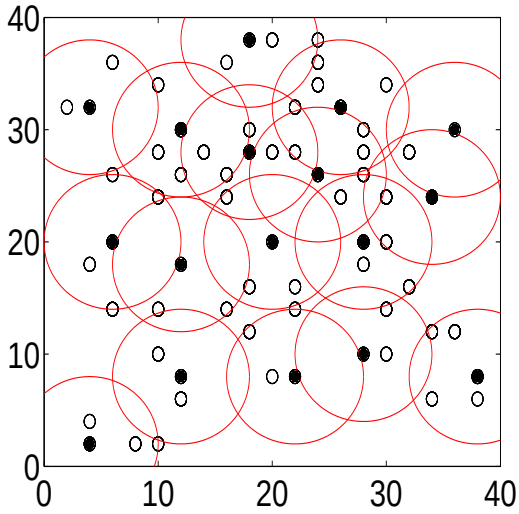
As our system model assumes the watchdog selection process is a part of network provisioning stage, an equal energy level is considered for all sensor nodes. It is not a necessary condition for the presented models. Our models can be applied at any stage of a WSN whose nodes are still capable of forming a connected graph.

Table 3.1: Comparative Computation Time

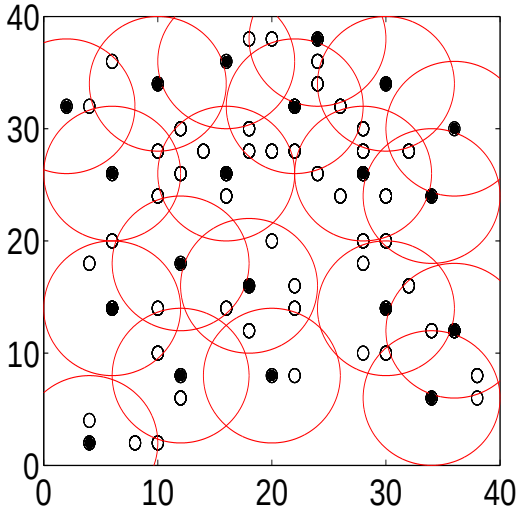
WSN Topology Settings (Area: 40m×40m)	**Limited Overlapping** (sec.)	**Full Coverage Heuristic** (sec.)	**Linear Reource Minimization** (sec.)
70 Nodes, 6 meters range	1.54	0.042	0.201
70 Nodes, 8 meters range	9.01	0.040	0.214
100 Nodes, 6 meters range	11.64	0.063	0.473



(a) LO: 6 m, $N_{wsn} = 70$.

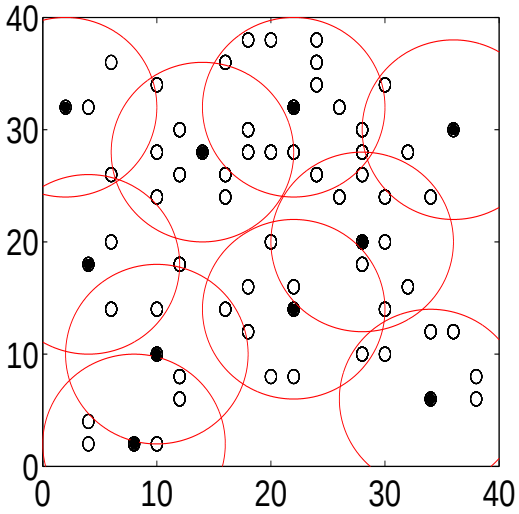


(b) FCH: 6 m, range, $N_{wsn} = 70$.

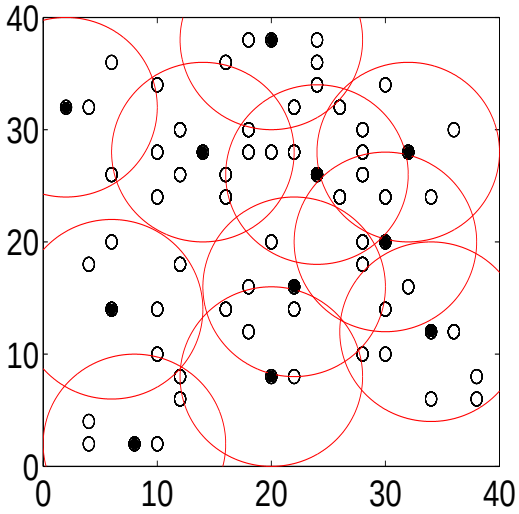


(c) LRM: 6 m, $N_{wsn} = 70$.

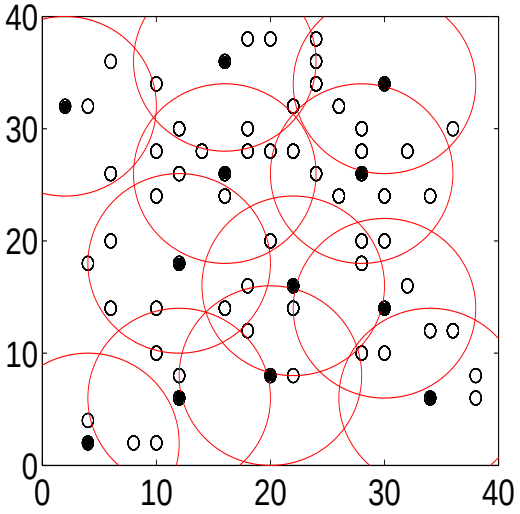
Figure 3.2: Watchdog system placement results for a WSN deployed in a 40m×40m area with sensing range = 6 meters, and $N_{wsn} = 70$.



(a) LO: 8 m, $N_{wsn} = 70$.

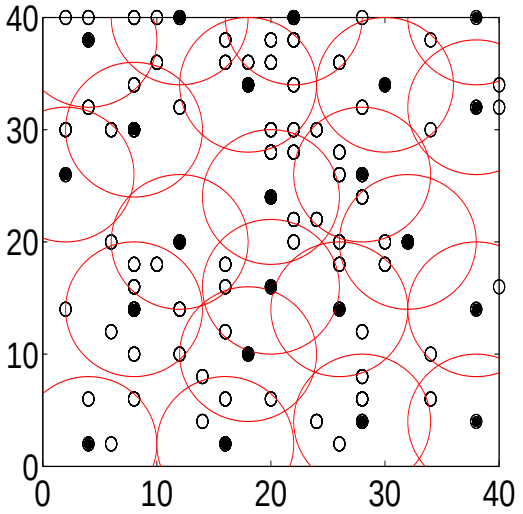


(b) FCH: 8 m, $N_{wsn} = 70$.

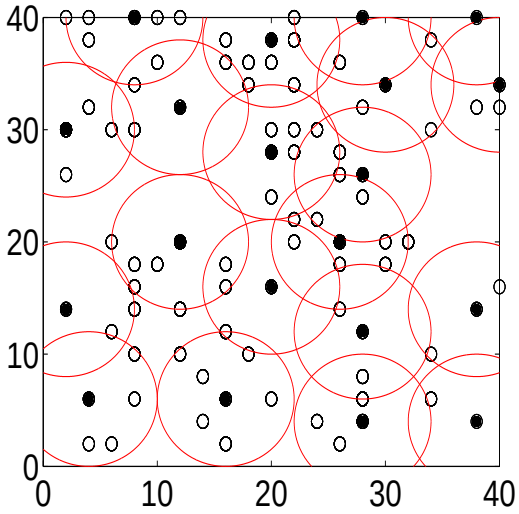


(c) LRM: 8 m, $N_{wsn} = 70$.

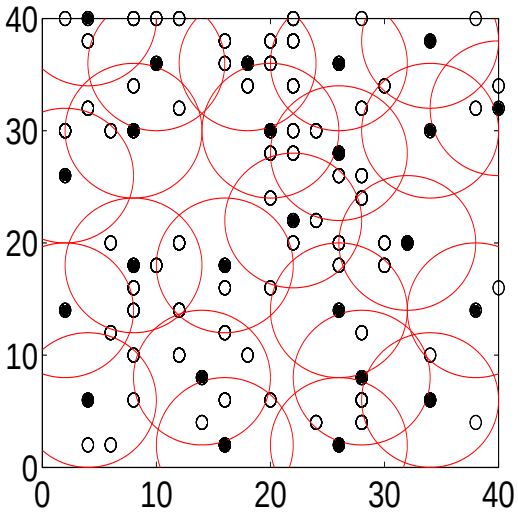
Figure 3.3: Watchdog system placement results for a WSN deployed in a 40m×40m area with sensing range = 8 meters, and $N_{wsn} = 70$.



(a) LO: 6 m, $N_{wsn} = 100$.

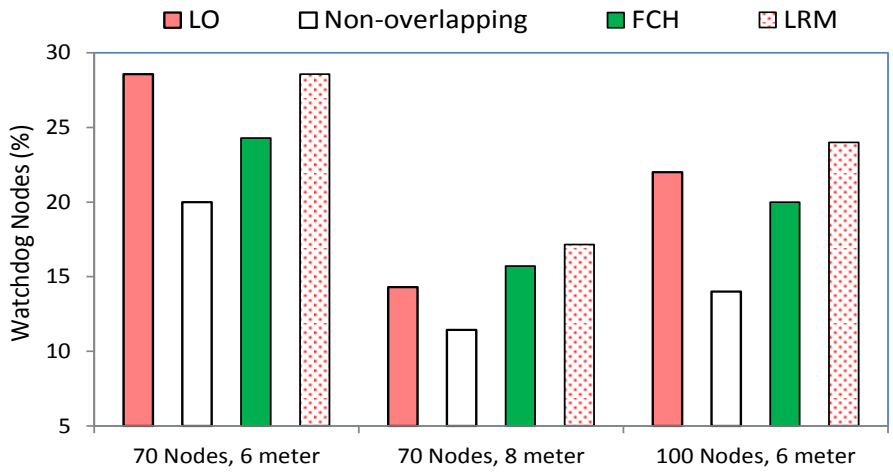


(b) FCH: 6 m, $N_{wsn} = 100$.

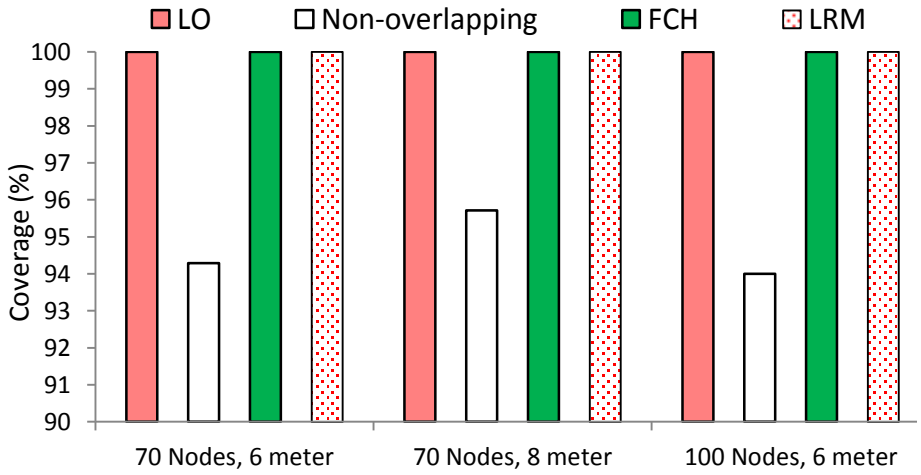


(c) LRM: 6 m, $N_{wsn} = 100$.

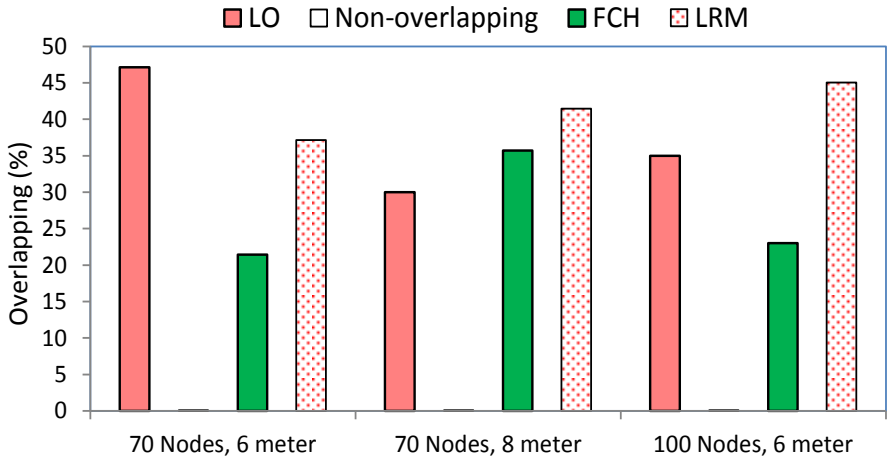
Figure 3.4: Watchdog system placement results for a WSN deployed in a 40m×40m area with sensing range = 6 meters, and $N_{wsn} = 100$.



(a) Watchdog requirements.



(b) Monitoring coverages.

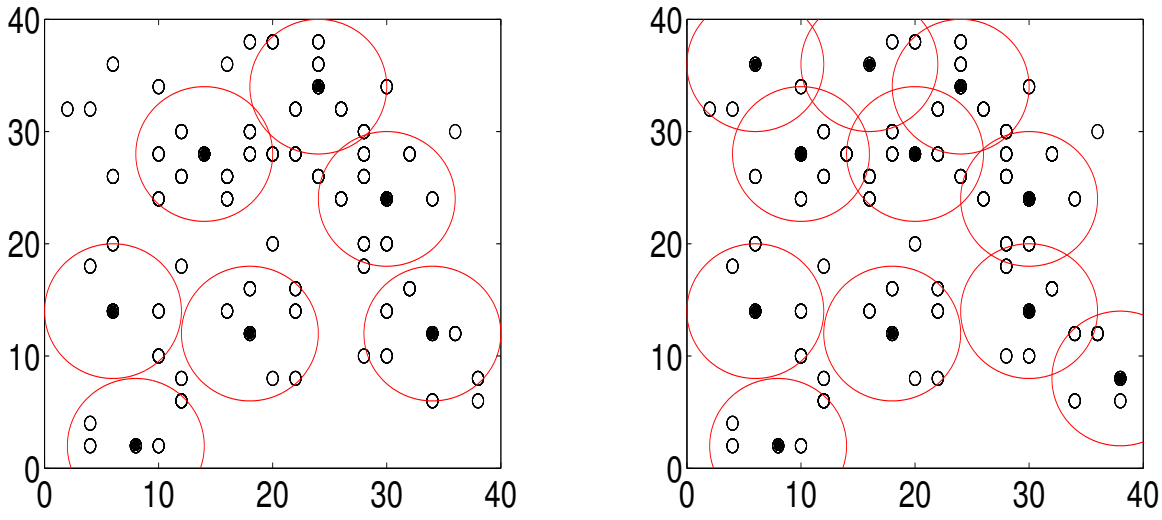


(c) Overlapping of coverages.

Figure 3.5: Comparative evaluation of the optimization models.

3.4.2 Study of Resource-Constrained Problems

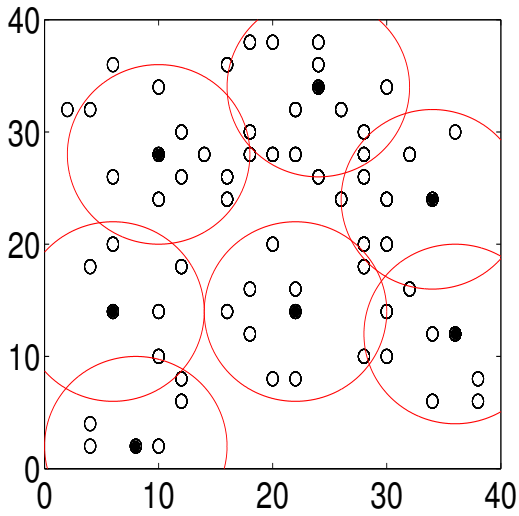
To evaluate the resource-constrained model, we compare two different percentage for watchdogs, 10% and 15%. Such percentages indicate the maximum amount of watchdogs to be deployed in a given WSN. Figures 3.6, 3.7, and 3.8 compare results for different WSN settings. Watchdogs are represented by the small solid circles and their coverage areas are indicated by the large circles. The rest of the small circles represent sensor nodes. One important observation is that the watchdog selection set for 10% is not necessarily a subset of the one for 15%. The model has to find a better set to avoid or limit overlapping. Figure 3.9 summarizes the comparative coverage for resource-constrained scenarios. For the higher density WSN, the gap between coverages is smaller. This is because of the more number of single hop neighbors. Only one case of the full coverage is observed for 15% at the higher sensing range. Figure 3.10 summarizes the comparative for resource-constrained scenarios. For the higher sensing range, overlapping is observed for both resource settings.



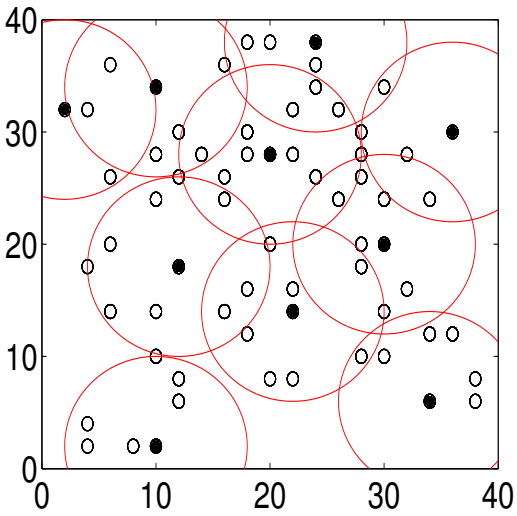
(a) 10% Watchdogs, 6 m, $N_{wsn} = 70$.

(b) 15% Watchdogs, 6 m, $N_{wsn} = 70$.

Figure 3.6: Watchdog system placement results for resource-constrained scenarios considering a 40m×40m deployment area with sensing range = 6 meters, and $N_{wsn} = 70$.

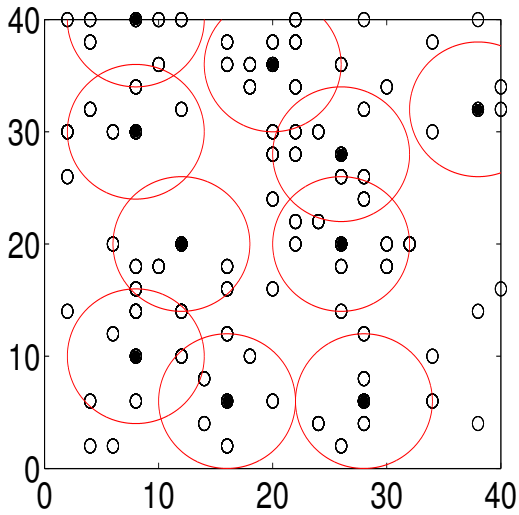


(a) 10% Watchdogs, 8 m, $N_{wsn} = 70$.

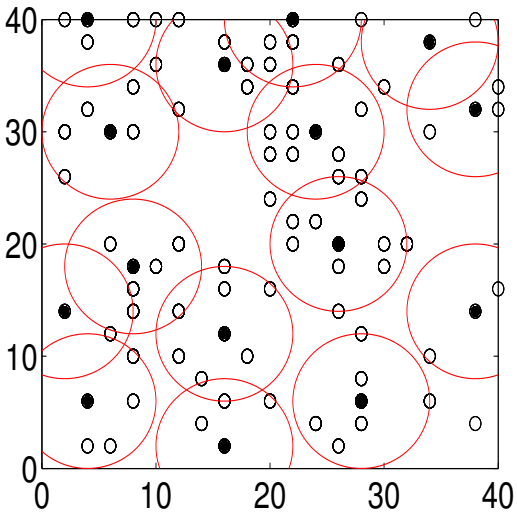


(b) 15% Watchdogs, 8 m, $N_{wsn} = 70$.

Figure 3.7: Watchdog system placement results for resource-constrained scenarios considering a 40m×40m deployment area with sensing range = 8 meters, and $N_{wsn} = 70$.



(a) 10% Watchdogs, 6 m, $N_{wsn} = 100$.



(b) 15% Watchdogs, 6 m, $N_{wsn} = 100$.

Figure 3.8: Watchdog system placement results for resource-constrained scenarios considering a 40m×40m deployment area with sensing range = 6 meters, and $N_{wsn} = 100$.

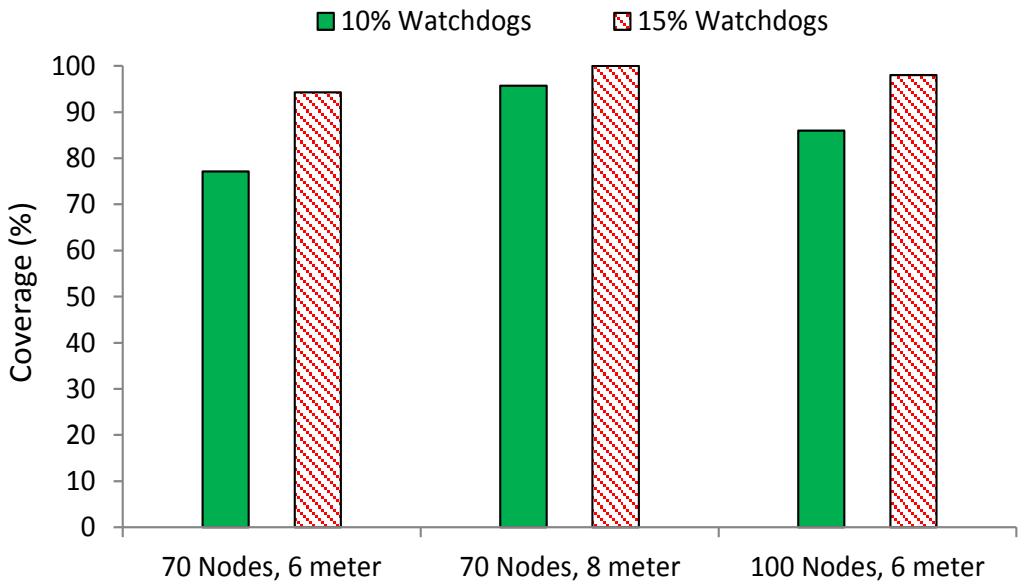


Figure 3.9: Monitoring coverages for different resource-constrained scenarios.

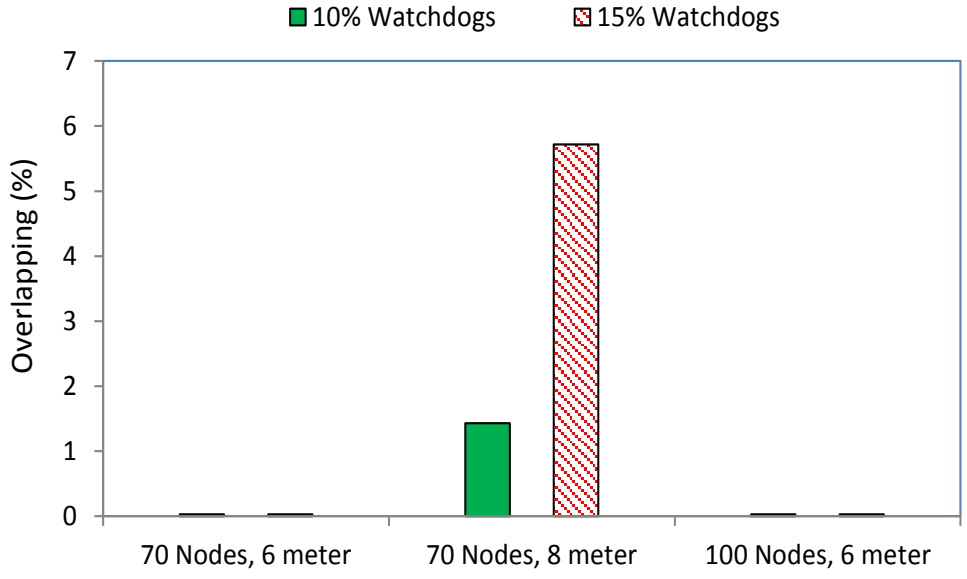


Figure 3.10: Overlapping of coverages for different resource-constrained scenarios.

3.5 Conclusion

In this chapter, we studied the watchdog selection problem from a resource management perspective. We developed and evaluated optimization models for watchdog placement in WSNs. Three of models were proposed for an unknown amount of resources: LO, FCH, and LRM. The LO model is found to be better for higher sensing ranges. It is computationally more expensive than two other models. Its full coverage can be affected by the non-circularity of sensing ranges. The LRM model is more resource intensive than two other models. The FCH model is found to be better for lower sensing ranges. It is a computationally lightweight model. It can be considered as a promising solution to watchdog selection problems. We also proposed an optimization model to maximize the monitoring coverages in resource-constrained scenarios. The resource-constrained model is an extension of the LO model. It is evaluated to observe the impact associated with budgetary limitation of resources.

Chapter 4

Segmentation-Based Trust System Placement in SCADA Networks

4.1 Introduction

A smart grid is one of the key driving forces behind smart cities. It integrates power system components with information and communication technologies (ICTs) to enhance operational efficiency. Its implementations are adding new cyber threats and vulnerabilities to energy infrastructures [Eri00, Kna13, nis10]. SCADA systems are an integral part of a modern power grid. They rely on supporting communication networks that convey measurements and control commands. A SCADA system also appears as the wide area measurement system (WAMS) in the smart grid literature [Law13]. Measurements of power system parameters are initially collected by the geographically distributed field networks. Such measurements are used in the state estimation process. State estimators are located at the control centers. Control commands are generated in accordance with the estimation of system status. Corrupted measurements can lead to inappropriate control commands. A SCADA network is used to accumulate measurements from the field networks. It is also responsible for delivering control commands to the field networks. Therefore, adversaries target SCADA networks to corrupt measurements and control commands. In a successful attack, the integrity of information is compromised and this can cause various forms of

service disruptions and catastrophic damages. The integrity of information deserves the highest priority in SCADA communications.

The vision of a smart grid includes Internet-based SCADA systems. It makes the SCADA networks more exposed to cyber-attackers. As a result, the scope of cyber security concerns becomes much wider. There are some major areas where an attack can take place: (i) field networks, (ii) communication network, and (iii) control center [Sou13]. In the field networks, RTUs exchange information with locally deployed sensors and actuators. The RTUs can be targeted for different types of DoS attacks. A DoS attack basically affects the availability of power system information. It unnecessarily consumes network resources and adds excessive delays to time-critical communications. The communication network takes place between the control center and field networks. It can be affected by two of the most possible forms of integrity attacks: (i) bad data injection that corrupts measurements, and (ii) modification of control commands through an unauthorized access. The control center can be affected by malware and hacks. In such a case, malicious control commands are generated.

Smart grid operators are required to deploy effective countermeasures to ensure trustworthy communications for SCADA systems. For this reason, continuous filtering and monitoring of SCADA traffic is necessary. Trust systems are specialized security devices that are devised for power system SCADA networks [Coa08, Coa10]. Their main functionalities include firewalling and intrusion detection. They use software agents for security analysis and response generation in a time-critical network. They offer flexible implementations and bidirectional monitoring of IP traffic. The capabilities of a trust system also include format checking. authorization checking, encryption, and authentication. All these features are utilized to intercept SCADA traffic and respond accordingly. Trust systems are responsible for initiating and distributing alert messages. They can be adopted as security resources to smart grid networks. In a SCADA network, only a selected number of nodes are equipped with trust systems due to budgetary constraints and those nodes are known as the trust nodes. Optimal placement of trust nodes is required to minimize costs and to provide better defense against cyber-attacks. An optimal placement problem

deals with the total number of trust nodes and their topological locations. Costs can be represented by the number of trust nodes. On the other hand, the quality of protection can be reflected by the number of monitored links.

In this chapter, we study the optimal trust system placement problem for smart grid SCADA networks. Such a placement problem has been reported as NP-hard in the literature [Gon11, Zha13b]. This is why heuristic solutions are commonly proposed. We also propose a heuristic placement scheme. At first, a network segmentation approach is developed based on the MST partitioning problem. Our network segmentation approach is computationally lightweight. It uses LP formulations. Once the segments are computed, the proposed scheme uses a minimization problem to select the trust nodes. The minimization problem is formulated to reduce the cost of protection. We also show that the proposed scheme is compatible with both kinds of scenarios: (i) defense planning for a network based on its requirements and (ii) defense planning for a network based on a fixed amount of resources. In our current problem, the requirements depend on the smart grid operator's preference. Smart grid operators need to decide the number of geographical divisions in their SCADA networks. The final portion of the chapter includes case studies for the IEEE test system topologies [iee15].

4.2 Segmentation for Trust System Placement

Network segmentation is an efficient method of solving trust system placement problems. In such a method, a SCADA network needs to be segmented into small pieces [Gon11, Zha13b]. Trust nodes are only selected from the bordering nodes. Bordering nodes are defined as the nodes that are connected by inter-segment links. Figure 4.1 shows an illustrative view of a segmented SCADA network where trust systems are hosted by the bordering nodes. Nodes A and B belong to segment 1; and nodes C, D, and E belong to segment 2. Three bordering nodes A, B, and D are equipped with trust systems. These bordering nodes are capable of monitoring traffic from both ingress and egress directions.

Segmentation offers two major advantages: (i) it limits the spreading of malicious

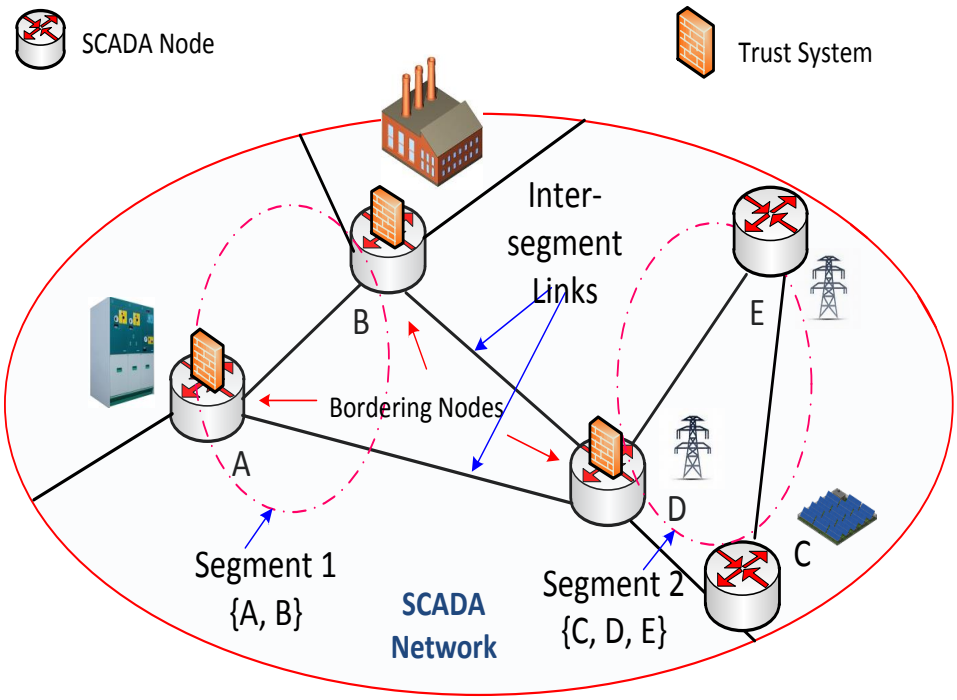


Figure 4.1: An illustrative view of a segmentation-based trust system placement.

activities to a small portion of a network and (ii) it reduces costs since only the bordering nodes are eligible for hosting trust systems. Each inter-segment link is required to be monitored by at least one trust system. This prevents the spreading of malicious activities from one segment to another. Exact solutions to network segmentation problems are computationally expensive. Previous studies in this area did not consider segmentation as a separate problem. They proposed heuristics that solve a single problem for trust system placement. Their solutions include network segments and trust nodes' locations. In contrast, the current scheme computes network segments first and then selects the trust nodes. It exploits the graph theoretic properties of MSTs to compute segments. Subsequently, it attempts to deploy a minimal number of trust systems to reduce costs.

Trust nodes are responsible for distributing time-critical messages. As SCADA networks are geographically distributed, propagation delay and segment size can affect time-critical communications. It is better to form a segment with nearby nodes. The difference in segment sizes affects the distribution of trust nodes. It is not always possible to obtain equal segment sizes due to topological constraints. Larger segments are more vulnerable

to spreading cyber-attacks since they expose more tolerance to malicious activities due to the increased number of unmonitored hops. The existing heuristics are appropriate for certain scenarios of smart grid networks: (i) a given number of trust systems with timing thresholds [Gon11], and (ii) layered architecture with tolerance levels [Zha13b]. There are still many additional scenarios that are possible for future contributions. A scenario is considered involving cyber security planning, where smart grid operators can decide the number of geographic divisions for their SCADA networks. Such divisions are termed as network segments.

There are three major factors that dominate a network segmentation procedure: topological constraints, availability of resources, and time-criticality. Topological constraints are network specific and cannot be ignored. This chapter introduces a trust system placement scheme that mainly focuses on the topology-awareness.

4.3 Problem Description

A representative communication network is assumed in the mathematical formulation for a smart grid SCADA system. In such a network, the nodes belong to power grid buses; and the links are associated with power grid branches. Such SCADA networks can be represented by weighted undirected graphs in which the links are weighted based on their propagation delays [Gon11, Zha13b].

Network segmentation is a way of solving the trust system placement problem. As SCADA nodes are geographically distributed, geographic dispersion affects the trust systems' response time. We consider the MST weight of each segment as a measure of geographic dispersion. Therefore, the ideal segmentation problem is described where the objective is to minimize the sum of MST weights of all segments in a SCADA network. The lowest sum is obtained when the MST weight of each segment is minimized. The associated constraint is the maximum segment size that also tries to keep the size as uniform as possible. In the mathematical formulation, two additional constraints are required that involve the following major sets: the SCADA node set (V) and the segment set (S). For

any segment s, τ_s^{mst}, is the MST weight and V^s, is the set of member nodes. The objective function is given in (4.1) and defined in (4.2). The weight of a link e^s in the segment s, is denoted by $w(e^s)$. The set of the MST link in a segment s, is denoted by E_s^{mst}. Constraint (4.3) ensures that the whole network will be segmented. The union of all segment node sets returns the SCADA network node set. Constraint (4.4) ensures that a node can only be located in one segment thus the intersection of any two segment node sets is an empty set. Constraint (4.5) limits the maximum number of nodes located in one segment. The cardinality of SCADA node set, segment set, and node set of the segment s, are denoted by N, K, and $|V^s|$ respectively. The exact solution to this problem requires an exhaustive search that is computationally expensive to examine all the possible combinations of segments. To meet constraint (4.5), the network graph needs to be dense enough. Even for an exhaustive search, an optimum solution may not always exist due to topological constraints.

The Ideal Segmentation Problem

$$\min \sum_{s \in S} \tau_s^{mst}, \tag{4.1}$$

where

$$\tau_s^{mst} = \sum_{e^s \in E_s^{mst}} w(e^s), \qquad \forall s, \tag{4.2}$$

subject to

$$\bigcup_{s \in S} V^s = V, \tag{4.3}$$

$$V^s \cap V^{s'} = \emptyset, \qquad \forall s \neq s' \text{ and } s, s' \in S, \tag{4.4}$$

$$|V^s| \leq \left\lceil \frac{N}{K} \right\rceil, \qquad \forall s. \tag{4.5}$$

4.4 Proposed Trust System Placement Scheme

A segmentation approach is at the heart of the proposed solution in this chapter. At first, segments are computed using a heuristic method where bordering nodes are then identified to select hosts for trust systems. Before explaining the solution approach, there are two important clarifications on terminologies used throughout this thesis: (i) to be consistent, the network theory terms 'node' and 'link' are used instead of graph theory terms 'vertex' and 'edge', and (ii) for convenience, the term 'partition' is used when it is a subtree and the term 'segment' is used when it is a subgraph.

As an exhaustive search is computationally expensive, a lightweight algorithm is developed based on the graph theoretic properties of MSTs. Inputs of the segmentation problem include the network graph and the number of segments with an output that outlines the set of segments. Such a segment is identified by the corresponding set of member nodes. This method reduces the graph into its MST and then cuts the MST into partitions. Each MST partition belongs to a particular segment and thus an MST partition can be treated as a segment skeleton. For a graph with N nodes, the MST contains $(N-1)$ number of links. To obtain K partitions, we need to eliminate $(K-1)$ links from the MST. As a result, the rest of the $(N-1)-(K-1) = N-K$, number of links form K partitions. The following proposition describes a useful property of MSTs where for a given SCADA network graph $G(V, E)$, E^{MST}, denotes the set of MST links.

Proposition 1 *Any MST partition forms the local MST for its node set.*

Proof. The MST is an optimum tree that connects all the nodes in a network. It implies that the principle of optimality is applicable to the MST [Bel57]. According to the principle of optimality, any subtree of the MST is also an optimum tree for the node set it belongs. Therefore, any partition of the MST, is actually the local MST.

Any MST partition is a connected graph. Therefore, any computed segment in the method outlined in this chapter is always a connected graph. In the segmentation problem, there is a trade-off between the MST weight (propagation delay) and segment size with

it being basically a multi-objective problem. It is computationally expensive to optimize both of them together. Moreover, uniform segment sizes are not always feasible due to topological constraints. Therefore, we relax the constraint (4.5). In the outlined MST based method, the primary consideration is given to topology-awareness where then effort is furthered to balance the segments as much as possible. To consider the impact of topology, a metric is defined named the minimum degree of a link in the MST. It is defined as follows.

$$d_{min}^{MST}(e_{u\leftrightarrow v}) = \min(d_u^{MST}, d_v^{MST}). \tag{4.6}$$

The undirected link between nodes u and v is denoted by $e_{u\leftrightarrow v}$. The degree of a node refers to the number of links connected to that node. MST degrees of nodes u and v are denoted by d_u^{MST} and d_v^{MST} respectively where they are calculated considering only MST links. The metric in (4.6) helps with the identification of leaf nodes and it is also useful in the handling of star connections. The minimum degree of a link is always one if it connects a leaf node.

The proposed solution to the trust system placement problem can be divided into three major parts: (i) SCADA network segmentation, (ii) local search for repartitioning, and (iii) Trust node selection. They are presented by Algorithm 4.1, Algorithm 4.2, and Algorithm 4.3, respectively. Algorithm 4.2 is an embedded function within Algorithm 4.1 and the output of Algorithm 4.1 is an input to Algorithm 4.3.

Two linear programming problems (LPPs) are solved as part of Algorithm 4.1 and Algorithm 4.3. It is worthwhile to describe such LPPs before outlining the algorithms. The LPP4.1 is used to create initial partitions. It is a multi-objective optimization problem. It eliminates MST links based on their minimum degrees and normalized weights. These two parameters are combined using two weighting factors: α and β. As the current priority is the topology-awareness, the weighting factors are set as follows; $\alpha = 1$ and $\beta = 0.5$, where this value of β ensures the dominance of the minimum degree. The LPP4.1 is a maximization problem. The links with a higher minimum degree and higher propagation delay are going to be eliminated so that partitions are formed. The decision variable of

LPP4.1 is $Y = (y_e)_{(N-1)\times 1}$, is a link incidence vector, such that;

$$y_e = \begin{cases} 1, & \text{if } e \in E^{MST} \text{ is selected for elimination;} \\ 0, & \text{otherwise.} \end{cases} \tag{4.7}$$

The objective is given in (4.8) and the normalized weight is defined in (4.9). Constraint (4.10) ensures the number of links to be eliminated. Constraint (4.11) confirms that there will be no singleton nodes. This means that a partition must contain at least two connected nodes.

LPP4.1: Initial Tree Partitioning

$$\max_{Y} \sum_{e \in E^{MST}} \big(\alpha d_{min}^{MST}(e) + \beta \widetilde{w}(e)\big) y_e, \tag{4.8}$$

where

$$\widetilde{w}(e) = \frac{w(e)}{\arg\max_{w(e)} w(e)}, \qquad \forall e \in E, \tag{4.9}$$

subject to

$$\sum_{e \in E^{MST}} y_e = K - 1, \tag{4.10}$$

$$y_e - d_{min}^{MST}(e) < 0, \qquad \forall e \in E^{MST}, \tag{4.11}$$

$$y_e \in \{0, 1\}, \qquad \forall e \in E^{MST}. \tag{4.12}$$

LPP4.2 is a minimization problem and is used to select trust nodes when segments are ready. It selects at least one bordering node from each inter-segment link. The value of the objective function indicates the number of trust systems required to protect the network.

The decision variable is $X_B = (x_{sb})_{\sum_{s\in S}|B(s)|\times 1}$, is a bordering node incidence vector, such that;

$$x_{sb} = \begin{cases} 1, & \text{if } b \in B(s) \text{ is selected, } s \in S; \\ 0, & \text{otherwise.} \end{cases} \tag{4.13}$$

The objective is outlined in (4.14) where the constraint (4.15) ensures at least one trust node is selected for each inter-segment link.

LPP4.2: Trust Node Computation

$$\min_{X_B} \sum_{s\in S} \sum_{b\in B(s)} x_{sb}, \tag{4.14}$$

subject to

$$\sum_{x_I \in X_I(l)} x_I \geq 1, \qquad \forall l \in L_{ss\prime}; \forall s, s\prime \in S, \tag{4.15}$$

$$x_{sb} \in \{0, 1\}, \qquad \forall s \in S \text{ and } \forall b \in B(s), \tag{4.16}$$

where

$$X_I(l) = \{x_{sb}, x_{s\prime b\prime}\}, \qquad b \in B(s), b\prime \in B(s\prime), s \neq s\prime. \tag{4.17}$$

Algorithm 4.1 is described as follows. After initialization, the MST of a SCADA network ($T(V, E^{MST})$) is computed using the Kruskal algorithm (Line 3). The minimum degree of each MST link is computed using (4.6). The weight of each MST link is normalized using (4.9). The initial partitions of the MST are obtained by solving LPP4.1. LPP4.1 computes E^I, the set of MST links to be eliminated (Line 8). The set of remaining MST links E^{SS}, is obtained (Line 9). To identify the initial partition set, the function Disjoint-set is used (Line 10) where it computes the node sets for each initial partition. To balance partition sizes, a local search is done (Line 11-27). For each pair of oversized and undersized partitions,

Algorithm 4.1 SCADA Network Segmentation

Input: $G(V, E), K$;
Output: $S = \{s_1, s_2, ..., s_K\}$;

1: **begin**
2: $E^{SS} = \varnothing, N = |V|, k_s = \left\lceil \frac{N}{K} \right\rceil, \phi = 0$; // Initialization
3: $T(V, E^{MST}) \leftarrow$ **Kruskal**$(G(V, E))$; // Computation of the MST using the Kruskal algorithm
4: **for all** $e \in E^{MST}$ **do**
5: Compute the minimum degree $d_{min}^{MST}(e)$;
6: Compute the normalized weight $\widetilde{w}(e)$;
7: **end for**
8: $E^I \leftarrow$ **Solve LPP4.1**; // Select the $(K-1)$ number of links that needs to be eliminated
9: $E^{SS} = \{E^{MST} \setminus E^I\}$; // Returns $(N - K)$ number of links
10: $S = \{s_1, s_2, ..., s_K\} \leftarrow$ **Disjoint-set**(V, E^{SS}); // Initial partition sets
11: **while** $\phi < 1$ **do**
12: Starting local search for the MST repartitioning
13: $count = 0$;
14: **for** $i = 1$ to K **do**
15: **for** $j = 1$ to K **do**
16: **if** $|s_i| > k_s$ **and** $|s_j| < k_s$ **then**
17: Compute $\triangle_{min}$ the minimum subpartition belongs to s_i that is adjacent to s_j
18: **if** $\big((|k_s - |s_i||) + (|k_s - |s_j||)\big) > \big((|k_s - |s_i| + |\triangle_{min}||) + (|k_s - |s_j| - |\triangle_{min}||)\big)$ **then**
19: $s_i = \{s_i \setminus \triangle_{min}\}$ **and** $s_j = \{s_j \cup \triangle_{min}\}$; // Updates the MST partitions
$count = count + 1$; // It checks for balancing segment sizes
20: **end if**
21: **end if**
22: **end for**
23: **end for**
24: **if** $count = 0$ **then**
25: $\phi = 1$; // Termination condition
26: **end if**
27: **end while**
28: **return** $S = \{s_1, s_2, ..., s_K\}$
29: **end**

Algorithm 4.2 Computation of $\triangle_{min}$

Input: s_i, s_j, and $T(V, E^{MST})$;
Output: $\triangle_{min}$;

1: **begin**
2: **if** $\{e_{u\leftrightarrow v} | u \in s_i, v \in s_j\} \cap E^{MST} = \varnothing$ **then**
3: $\quad \triangle_{min} = \varnothing$;
4: **else**
5: $\quad \delta_{size} = \varnothing, \delta_{set} = \varnothing, , n = 0$; //Initialization
6: $\quad E^{ps_i} = \bigcup_{\forall a,b \in s_i} \{e_{a\leftrightarrow b} | e_{a\leftrightarrow b} \in E^{MST}\}$;
7: $\quad$ **for all** $e_{u\leftrightarrow z} \in E^{ps_i}$ **do**
8: $\quad\quad E^{ps_i} = \{E^{ps_i} \setminus e_{u\leftrightarrow z}\}$; // Cutting
9: $\quad\quad \{\triangle_u, \triangle_z\} \leftarrow$ **Disjoint-set**(s_i, E^{ps_i});
10: $\quad\quad n = n + 1$;
11: $\quad\quad \delta_{set}(n) = \triangle_u$;
12: $\quad\quad \delta_{size}(n) = |\triangle_u|$;
13: $\quad\quad E^{ps_i} = \{E^{ps_i} \cup e_{u\leftrightarrow z}\}$; // Restoring
14: $\quad$ **end for**
15: $\quad n* = \arg\min_n \delta_{size}(n)$;
16: $\quad \triangle_{min} = \delta_{set}(n*)$;
17: **end if**
18: **return** $\triangle_{min}$
19: **end**

the local search computes the minimum subpartition ($\triangle_{min}$) of the oversized partition that is adjacent to the undersized partition. If no adjacent subpartition exists between them, $\triangle_{min}$ becomes an empty set. If the size of $\triangle_{min}$ meets the condition for better balancing between partitions (Line 18), an adjustment for repartitioning occurs (Line 19). The subpartition is added to the undersized partition and then removed from the oversized partition. Thus both partitions are updated. The computation of $\triangle_{min}$ is described by Algorithm 4.2. The local search continues until there is a possibility of size balancing. If none of the pairs of oversized and undersized partitions remains adjustable (Line 24), the local search is terminated (Line 25).

Algorithm 4.2 is used to compute $\triangle_{min}$. Its inputs are node set of the oversized partition (s_i), node set of the undersized partition (s_j), and the MST. At first, it checks the adjacency of the partitions (Line 2). If the partitions are not adjacent, there will be no adjustment possible (Line 3). If the partitions are adjacent, it initializes the set of subpartition sizes

Algorithm 4.3 Trust Node Selection

Input: $S = \{s_1, s_2, ..., s_K\}, G(V, E)$;
Output: V^{Trust};

1: **begin**
2: **for all** $s \in S$ **do**
3: $B(s) = \varnothing$; // Initialize bordering node sets
4: **end for**
5: **for all** $s \neq s\prime$ and $s, s\prime \in S$ **do**
6: $L_{ss\prime} = \varnothing$; // Initialize inter-segment link sets
7: **end for**
8: **for all** $e_{u \leftrightarrow v} \in E$ **do**
9: Find the segment x belongs to node u
10: Find the segment y belongs to node v
11: **if** $x \neq y$ **then**
12: $L_{xy} = \{L_{xy} \cup e\}$;
13: $B(x) = \{B(x) \cup u\}$;
14: $B(y) = \{B(y) \cup v\}$;
15: **end if**
16: **end for**
17: $V^{Trust} \leftarrow$ **Solve LPP4.2**; // This will select the trust node set
18: **return** V^{Trust}
19: **end**

(δ_{size}), the set of subpartition node sets (δ_{set}), and an indexing variable n (Line 5). This just recursively uses the Disjoint-set function and returns the minimum adjustable subpartition. The set of links belonging to the oversized partition E^{psi} is extracted (Line 6). The node u, is located in the oversized partition and adjacent to the undersized partition. Each time a link belonging to u is chosen as a cut for the oversized partition (Line 8). In a tree graph, any link can be used as a cut. A cut divides the oversized partition into two subpartitions. The Disjoint-set function identifies such subpartitions (Line 9). If one of the subpartitions contains u, it is eligible for partition adjustment. The sets δ_{size} and δ_{set} are then updated (Line 11 and 12). After updating, the cut is restored to the E^{psi} (Line 13). The same process is done for all links of E^{psi} that belong to u. Thereafter, the minimum size is identified (Line 15) and the corresponding node set $\triangle_{min}$ is computed (Line 16).

Algorithm 4.3 is carried out with inputs that include the output of Algorithm 4.1 and the network graph. It computes the trust node set V^{Trust}. At first, it initializes the bordering node set $B(s)$ for each segment (Line 3). For each pair of segments, the

inter-segment link set $L_{ss\prime}$ is initialized (Line 6). As well, with each link in the SCADA network graph, segments are identified for both nodes (Line 9 and 10). If nodes belong to different segments, the link is an inter-segment link (Line 11). The inter-segment link set and bordering node sets are then updated (Line 12-14). Once all bordering nodes are identified, LPP4.2 is solved to compute the trust node set (Line 17).

Figure 4.2 shows the complete flow diagram of the proposed trust system placement scheme. The Kruskal and Disjoint-set algorithms are well known in the literature [Cor09]. The former is used to compute MSTs and the latter is used to identify the partitions of a disconnected graph. The **while** loop is the heaviest part of the proposed scheme. It runs maximum N times for the slowest case. For each run, $\triangle_{min}$ can be computed at most N times. Therefore, the worst case computational complexity is $\sim O(N^2)$.

Figure 4.3 illustrates an example of a trust system placement problem for a small SCADA network. The figure precisely explains and outlines all the major steps of the proposed scheme. A given network graph with nine nodes and twelve undirected links, to be segmented into three segments. The average segment size is three and the MST is comprised of eight links. To create initial partitions, two MST links need to be eliminated. Link (6, 7) has the highest minimum degree in the MST which is three and is therefore eliminated. For the second link to be eliminated, there are three links with the minimum degree of two: (2, 3);(3, 6); and (4, 6). Link (2, 3) has the highest weight among them. Therefore, it is eliminated. The partitions are identified using the Disjoint-set algorithm. Node sets for partitions are: {1, 2}; {3, 4, 5, 6}; and {7, 8, 9}. As the average segment size is three, the first partition is undersized and the second partition is oversized. These two partitions are adjacent. Node 3 belongs to the oversized partition. It is adjacent to the undersized partition. It becomes the subpartition to adjusted for size balancing. As a result, node 3 is added to the undersized partition and removed from the oversized partition. The partitions are now balanced. After repartitioning, node sets for partitions become {1, 2, 3}; {4, 5, 6}; and {7, 8, 9}. These sets are the node sets for the corresponding segments. Inter-segment links and bordering nodes are identified with the help of the network graph. The inter-segment links include: (1,9); (2,9); (3,4); (3,6); (3,7); and (6,7). The bordering

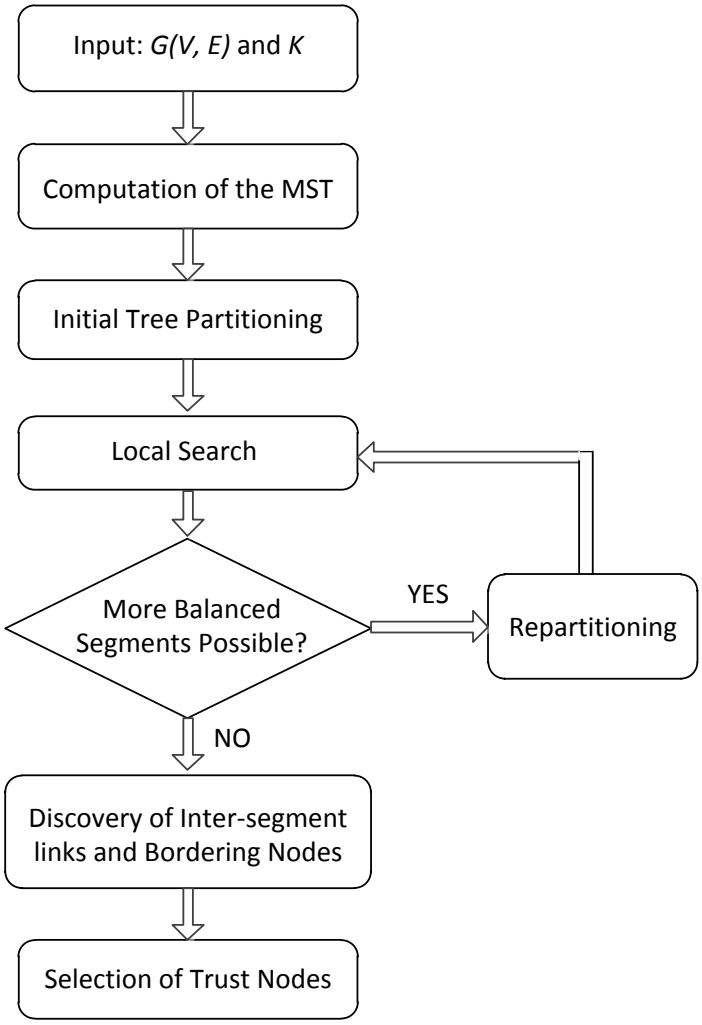
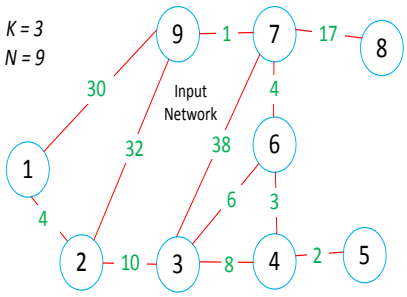


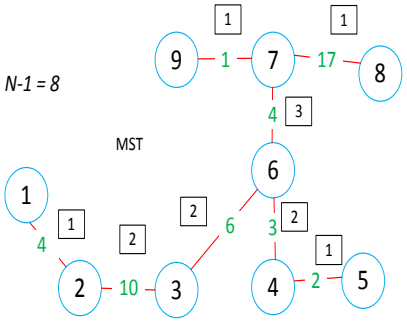
Figure 4.2: Flow diagram of the proposed placement scheme.

nodes are as follows 1, 2, 3, 4, 6, 7, and 9. Finally, the minimum number of trust nodes are selected from the bordering nodes such that each inter-segment link is monitored by at least one trust node. The trust nodes are 3, 7, and 9.

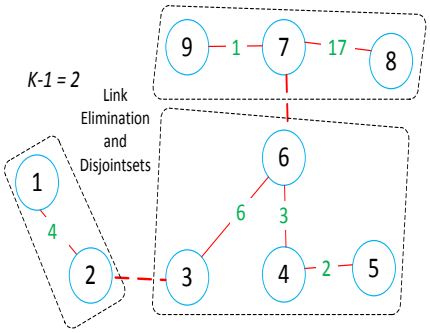
The proposed trust system placement scheme can also be applied in a different cyber security planning approach where the number of trust system (M) is given. Figure 4.4 shows the flow diagram for such convergence. The main idea is to search an appropriate number of segments through an iterative procedure. If the estimated number of trust system (Q) is greater than M then a new iteration with a lower value of K is required. This continues until $Q \leq M$ and this proves the versatility of our proposed scheme. The idea is extended in the next chapter to solve constrained placement problems.



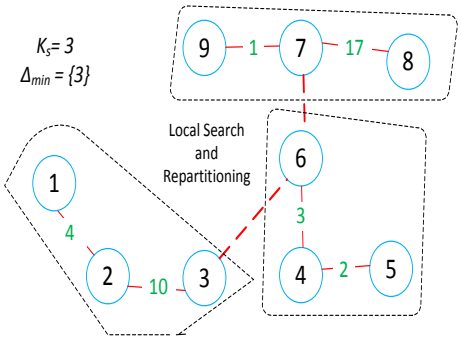
(a) A SCADA network graph to be segmented into three pieces. The link weights are representing propagation delays.



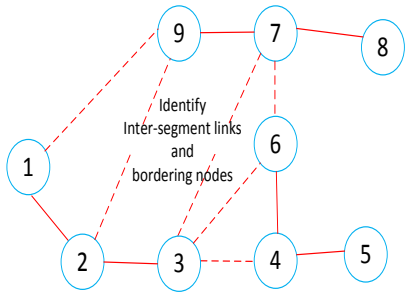
(b) The minimum spanning tree is computed using Kruskal. The minimum degree of each link is shown in the square box.



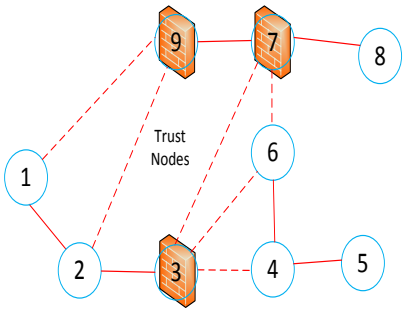
(c) Initial tree partitions from LPP4.1. Links (6,7) and (2,3) are eliminated for their higher minimum degrees and weights.



(d) Local search for segment sizes adjustment. Repartitioning strives to balance between oversized and undersized segments.



(e) Returning from the MST to the network graph outlines discovery of inter-segment links and bordering nodes. There are six inter-segment links: (1,9); (2,9); (3,4); (3,6); (3,7); and (6,7). There are seven bordering nodes: 1, 2, 3, 4, 6, 7, and 9.



(f) The required number of trust systems is minimized such that at least one trust system is placed per inter-segment link. The LPP4.2 selected trust nodes are: 3, 7, 9.

Figure 4.3: An illustrative example to explain the proposed scheme.

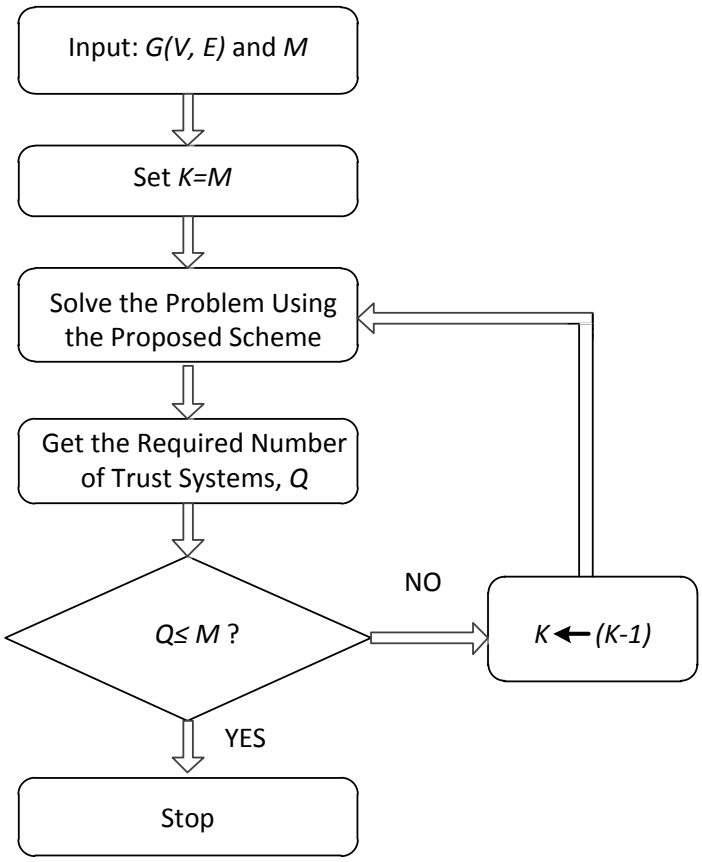


Figure 4.4: Converge with a different planning approach.

4.5 Numerical Results and Analysis

Case studies are conducted for the IEEE test system topologies [iee15]. Table 4.1 shows the summary of the experimental network parameters. Propagation delays are calculated using the methodology introduced in [Gon11] (see Appendix B for details). It assumes an optical fiber network that connects geographically distributed SCADA nodes. The proposed scheme is implemented using the MATLAB optimization toolbox. The IEEE test system topologies are used as SCADA network graphs. We categorize the topologies based on their sizes: (i) small networks and (ii) large networks. Small networks include BUS 14, BUS 30, and BUS 57; and large networks include BUS 118 and BUS 300 systems. The number of segments (K), is varied to observe the performance of the proposed scheme. For small networks, the value of K is varied from 3 to 6 with increments of 1. For large networks, the value of K is varied from 5 to 30 with increments of 5. These values are

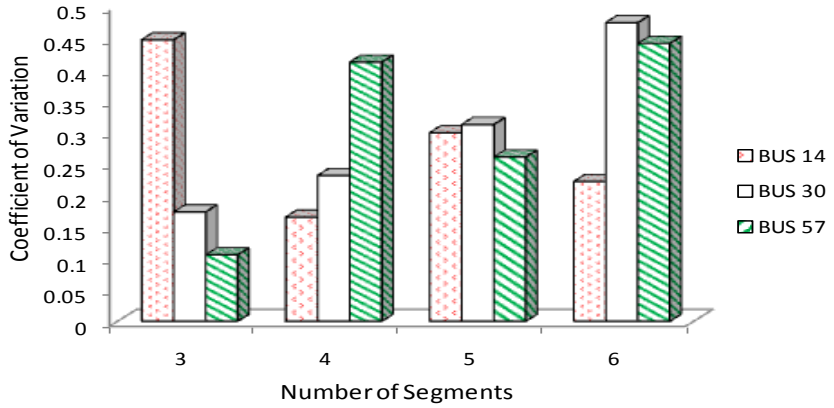
chosen considering SCADA network sizes and the feasibility of segment sizes. For small networks, the average segment size is varied between 2.33 and 19. For large networks, the average segment size is varied between 3.93 and 60. All experiments are run on a desktop machine with and Intel Core i3 3.30 GHz CPU and 4 GB RAM. Numerical results are obtained from 100 runs for each combination of inputs. For a given combination of inputs, each run generates the same results but processing time varies depending on the instance of the desktop machine. This is the reasoning for observing 100 runs to obtain an average processing time.

Table 4.1: Summary of SCADA Network Parameters

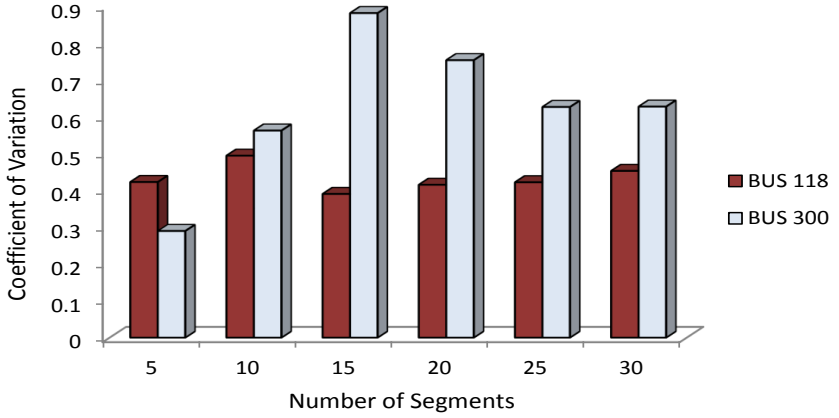
IEEE Test System Topology	**Number of Nodes (Network Size)**	**Number of Active Links**	**Link Weight Mean (μs)**	**Link Weight Standard Deviation (μs)**
BUS 14	14	20	19.55	18.84
BUS 30	30	42	24.1	24.67
BUS 57	57	78	22.33	34.41
BUS 118	118	179	8.35	6.22
BUS 300	300	409	14.59	39.21

4.5.1 Performance of the Segmentation Method

The performance of the segmentation method is evaluated based on segment sizes and geographic dispersion. For segment sizes, we choose the coefficient of variation as the metric. It refers to the ratio between standard deviation and average. It is also known as the normalized standard deviation. For a given number of segments, the average segment size in a network is a fixed number. The average remains the same for all methods. In such type of cases, the standard deviation is an important metric. As the experiments are conducted for different network sizes, normalized standard deviation is chosen. Figures 4.5(a) and 4.5(b)



(a) Small Networks

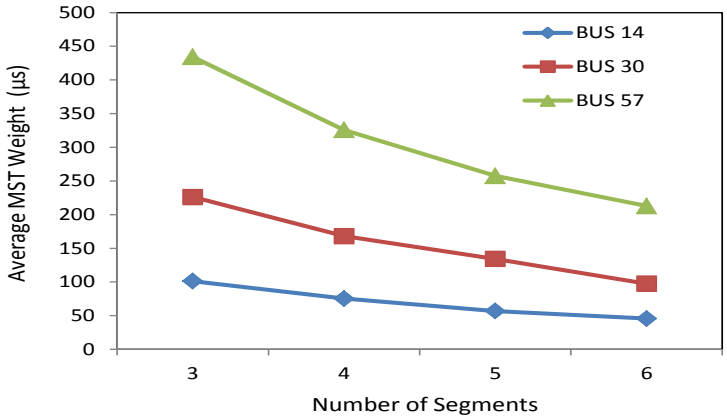


(b) Large Networks

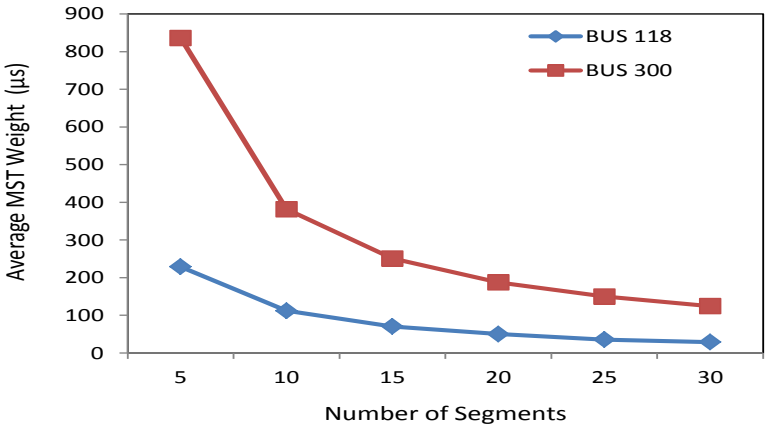
Figure 4.5: Coefficient of variation of the computed segment sizes.

show the coefficient of variation for the computed segment sizes. The former illustrates results for small networks and the latter illustrates results for large networks. In both cases, the coefficient of variation remains less than unity. Therefore, the computed segment sizes are low-variance. Large networks in Figure 4.5(b) exhibits higher coefficient of variation than small networks in Figure 4.5(a). This occurs because of the network size and the topological limitations of the MST. Segments of large networks are less balanced in size compared to that of small networks, as a result, the variation becomes larger.

For geographic dispersion, the average MST weight of the computed segments is the metric. Figures 4.6(a) and 4.6(b) show the average MST weights for small networks and large networks respectively. It is clear that in both cases the average MST weights follow a



(a) Small Networks



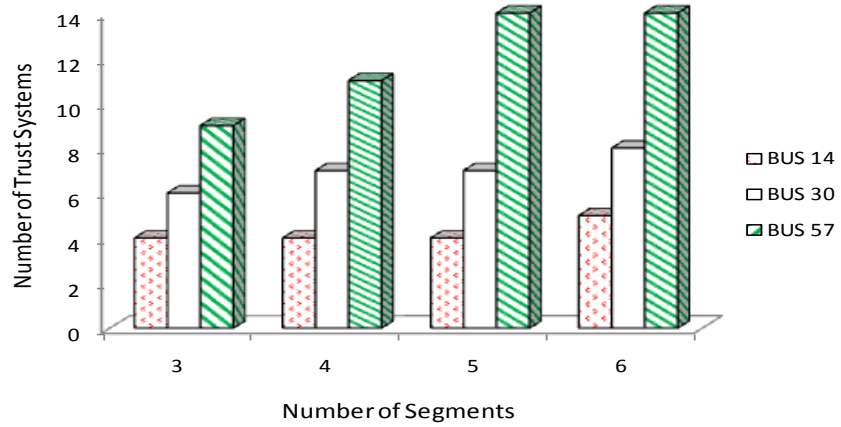
(b) Large Networks

Figure 4.6: Average MST weight of the computed segments.

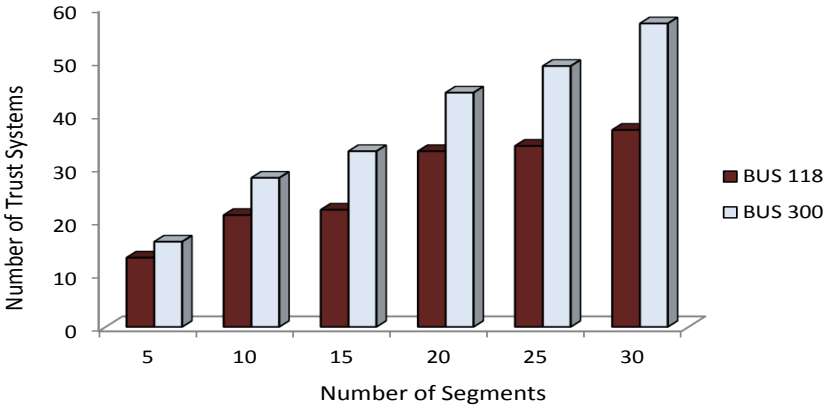
decreasing trend as the number of segments increases. Higher values of K cause decrements in segment sizes. Thus, a less number of MST links are required for a segment. As a result, the average MST weight is reduced. This reduction rate is higher for larger networks.

4.5.2 Analysis of Resource Requirements

The metric for resource requirements is the needed number of trust systems. Figures 4.7(a) and 4.7(b) show the required number of trust systems for the proposed scheme. It is observed that the required number of trust systems follow an increasing trend in accordance with increment in the number of segments. This is observed in all cases. As the number of segments increases in a SCADA network, it causes more inter-segment links. As at



(a) Small Networks



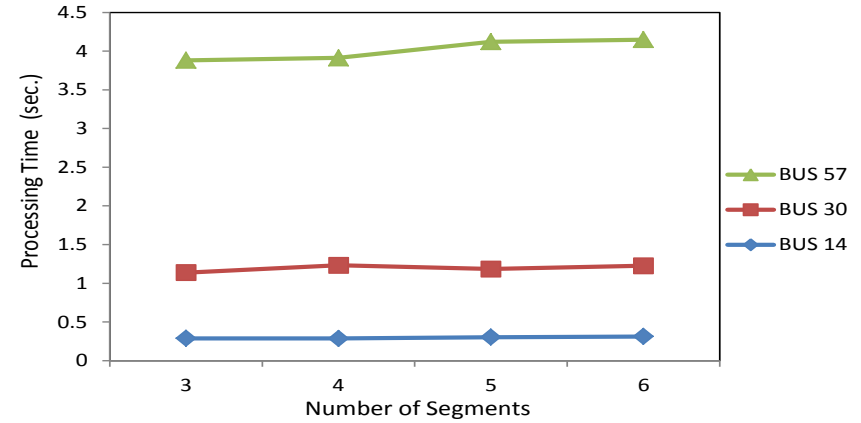
(b) Large Networks

Figure 4.7: The required number of trust systems.

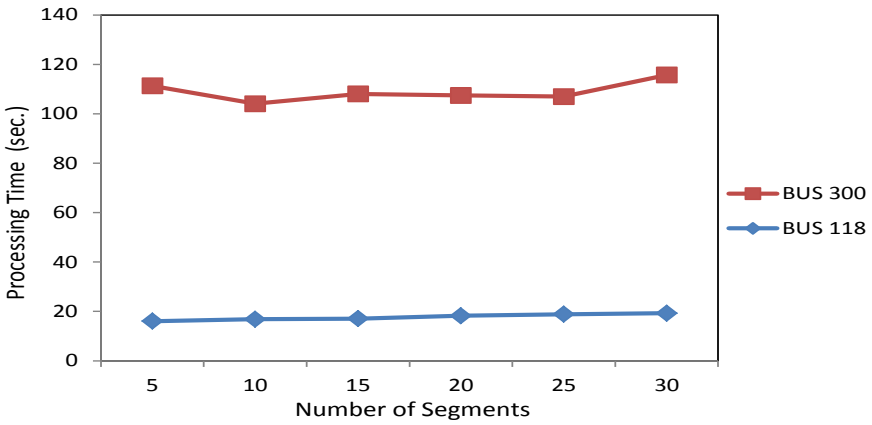
least one trust system is placed to monitor an inter-segment link, the required number is increased. Thus, increment amounts are higher for larger networks.

4.5.3 Analysis of Processing Times

Figures 4.8(a) and 4.8(b) show the average processing time for the proposed scheme. For each combination of inputs, the average is obtained after 100 runs. It is observed that the processing time mainly depends on the SCADA network size. It is clear that the number of segments has no significant impact on the processing time. The proposed scheme exhibits a small processing time. Even for large networks such as BUS 300, it can be run in a few minutes. This reveals the main advantage of the proposed scheme.



(a) Small Networks



(b) Large Networks

Figure 4.8: Average processing time of the proposed scheme.

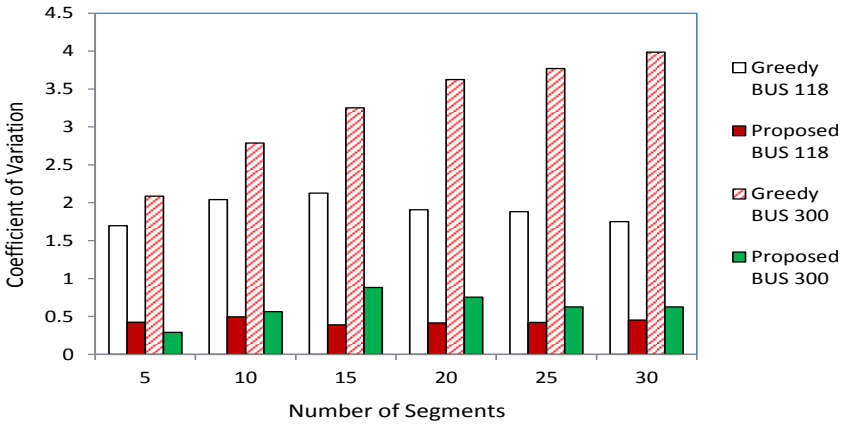
4.5.4 Impacts of Topology-Awareness: A Comparative Analysis

As the proposed scheme is a topology-aware approach to trust system placement, it is worthwhile to investigate the impacts of topology-awareness. To investigate the impacts of topology-awareness, the proposed scheme is compared with a greedy approach. The greedy approach is developed by resetting the weighting factors, α and β. For the greedy approach, $\alpha = 0$ and $\beta = 1$. This value of α nullifies the impact of the minimum degree of MST links in the segmentation method. As a result, the segments are only created based on the propagation delays. The objective of the greedy approach is to deploy the minimum number of trust systems regardless of topologies. This approach does not exert effort for balancing of the segment sizes. An additional performance metric is defined to compare

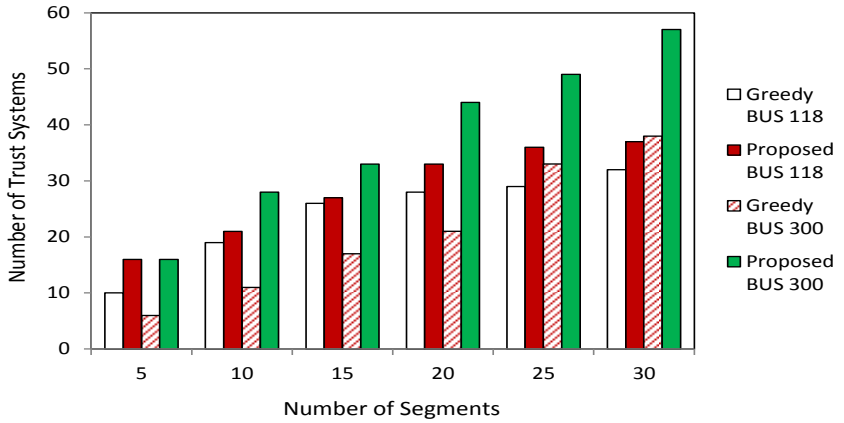
between the greedy and the proposed scheme. The metric is named tolerance factor (ρ) and it is defined as follows.

$$\rho = \frac{\text{Total number of unmonitored links in a network}}{\text{Total number of monitored links in a network}}. \tag{4.18}$$

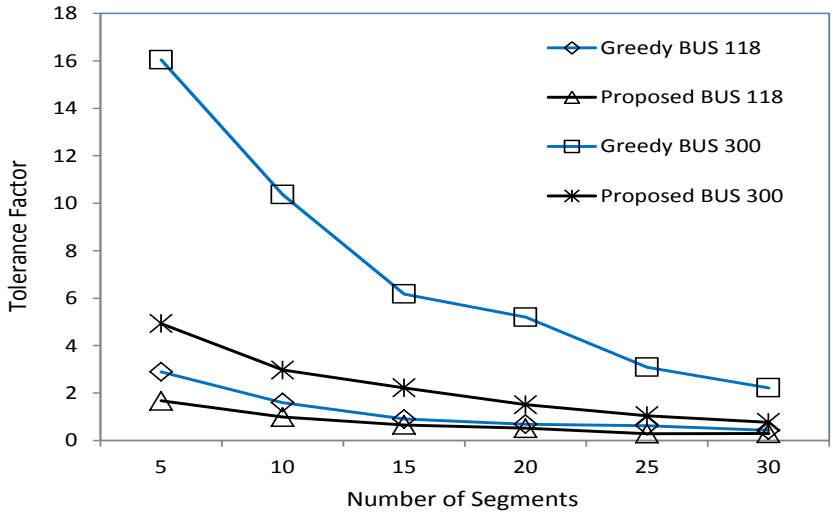
The tolerance factor is considered as the quality of protection for SCADA networks. Less protected networks expose higher tolerance and *vice versa*. To obtain comparative performance, only BUS 118 and BUS 300 topologies are considered. As well, only large networks have been chosen to demonstrate the difference between two approaches clearly. Figure 4.9(a) shows the comparative coefficient of variation for segment sizes. The greedy approach exhibits a much higher coefficient of variation in all cases. As coefficients of variation for the greedy approach are greater than unity, the computed segment sizes are high-variance. It means most of the segments are either heavily oversized or heavily undersized. Heavily oversized segments are more vulnerable to spreading cyber-attacks. Figure 4.9(b) shows the comparative number of required trust systems. It is clear that the greedy approach requires fewer trust systems in all cases. That is viewed as an advantage of the greedy approach. This happens because of having fewer of bordering nodes are caused by heavily oversized segments. There is a trade-off between the amount deployed security resources and the quality of protection. Thus it is necessary to investigate the quality of protection for both approaches. Figure 4.9(c) shows the comparative tolerance factor. In general, the tolerance factor is lower for higher values of K and *vice versa*. This is because of the number of deployed trust systems increases with the increment of K. It is observed that the greedy approach exhibits higher tolerance in all cases. In particular, for the BUS 300 topology, the greedy approach exhibits a much higher tolerance factor. It reveals that the proposed scheme offers better protection to SCADA networks. For topology-awareness, trust nodes are computed in such a way that the number of monitored links is maximized.



(a) Comparative Coefficient of Variation



(b) Comparative Number of Trust Systems



(c) Comparative Tolerance Factor

Figure 4.9: Comparative performance.

4.5.5 Gist of Results

The following important facts are extracted from the numerical results presented in this section.

The proposed trust system placement scheme computes low-variance segment sizes. Thus, it is effective in limiting the spreading of malicious activities. In particular, internal attacks from compromised nodes will be confined in a small portion of the SCADA network.

The results on the coefficient of variation do not follow any particular trend. This is because of the sparsity of MSTs and topological constraints of network graphs. The proposed scheme is a heuristic approach based on the MST. Despite this fact, consistent behavior is observed in the results on geographic dispersion of segments, resource requirements, and tolerance factor.

The proposed scheme exhibits a very small computational time. It is capable of computing trust nodes for the BUS 300 topology in a few minutes. Its computational time mainly depends on the SCADA network size. This implies that the proposed scheme can also be used as an estimation tool.

The proposed scheme uses topology-awareness to offer a better quality of protection to SCADA networks.

4.6 Conclusion

In this chapter, a lightweight trust system placement scheme has been proposed and evaluated for smart grid SCADA networks. An introduction has been provided for a heuristic algorithm based on the MST partitioning problem to segment SCADA networks in a smart grid environment. The proposed scheme offers a quality of protection using a topology-aware trust node selection. Additionally, it has shown that the scheme is compatible with both types of cyber security planning approaches: (i) optimal location for a given number of segments with the number of trust systems is unknown and (ii) optimal placement for a given number of trust systems with an unknown number of segments. The scheme can

also be used in developing an interactive cyber security planning tool.

Our proposed scheme is well-suited for the IEEE test system topologies. It is able to avoid the segments comprising stand-alone or singleton nodes. For other types of topologies, the feasibility of solutions depends on the concentration of star connections. As star-connected portions cannot be segmented, each star connected portion becomes a segment [Las05]. If a network is heavily star-connected, the desired number of segments may not be computed without having singleton nodes. In such a case, constraint (4.11) needs to be relaxed.

Chapter 5

Constrained Cases of Trust System Placement in SCADA Networks

5.1 Introduction

In the previous chapter, we developed an MST partitioning heuristic to solve the trust system placement problem. In this chapter, further study is given for two constrained cases of trust system placement in energy SCADA networks. Such study includes adoption and adaptation of the proposed MST partitioning heuristic to two different scenarios. The first case pertains to a resource-constrained scenario, where the number of trust systems is limited. The second case pertains to a latency-constrained scenario, where a latency threshold is considered for the network segmentation problem. Two iterative versions of the MST partitioning heuristic are developed for those cases to compute the trust nodes. In addition to the aforementioned constraints, our current solutions are designed for a distributive placement. It means that at least one trust node is selected from each segment. This distributiveness can be beneficial to power system islanding situations. The power system islanding is a method of preventing a wide area blackout [Tro13, Yan06]. It isolates faulty parts of a power system from the rest. The islands are created to minimize the impact of a cascaded failure. A segment can be partially or fully located in a faulty portion. It is important to have trustworthy communications between segments for restoration of

the normal operation. Figure 5.1 shows a simple example of a distributive trust system placement. The inter-segment links are shown by the dotted lines and each of them is connected to at least one trust node. Three out of seven bordering nodes are selected as trust nodes. Each segment contains at least one trust node.

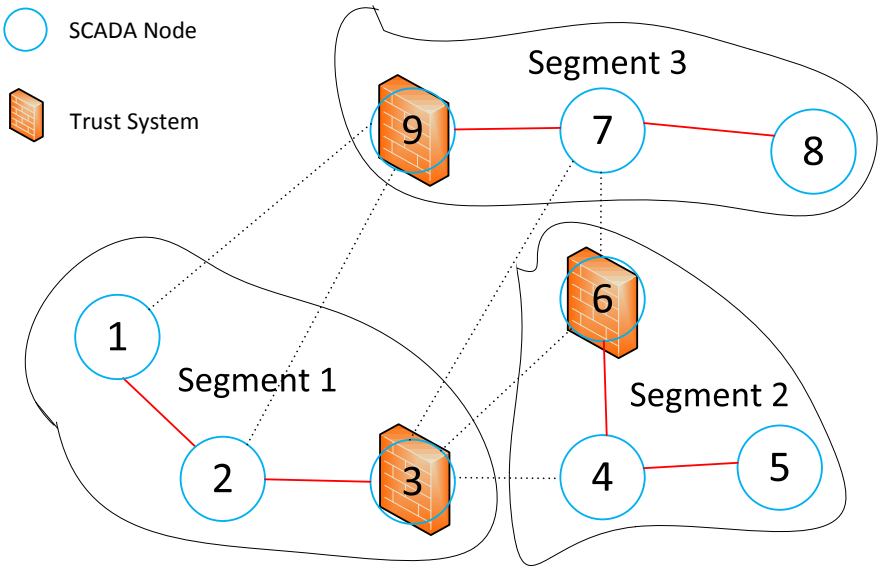


Figure 5.1: An example of a distributive trust system placement.

5.2 Adoption of the MST Partitioning Heuristic

In a smart grid environment, an electric power grid is assumed to be accompanied by a representative SCADA communication network. Such a network is represented by an undirected graph whose links are weighted by propagation delays. Only a selected number of nodes host trust systems due to budgetary constraints. Network segmentation is an efficient way of solving the trust system placement problem. Its advantages are twofold: (i) limiting the spreading of cyber-attacks and (ii) reduction of costs. Cyber-attacks are confined in a segment since all inter-segment traffic are monitored by trust systems. Costs are reduced since only bordering nodes are eligible to be the trust nodes.

The proposed MST partitioning heuristic was designed for a given number of segments. Where the number of segments is not known in advance, it would depend on the constraint that is of focus in the problem formulation. As it varies with the value of the constraint and thus the reasoning for the necessary iterations to obtain the optimal number of segments. Advantages of the heuristic are threefold: (i) it simplifies the SCADA network segmentation problem, (ii) it computes the MST route for each segment, and (iii) it provides a measure of geographic dispersion when weighted by propagation delays. The MST routing is a very useful strategy for delivering messages to multiple destinations [Bha83, Wu04].

In this chapter, the same terminologies are used as the previous chapter for consistency. It is recalled that the term 'partition' refers to a subtree and the term 'segment' refers to a subgraph. Each partition is the MST for the particular segment that it belongs. The only difference between them is the number of links. They can have the same number of links only when the segment is a star-connected one.

There are two basic tasks associated with the proposed segmentation-based trust system placement scheme: (i) the identification of the segments and (ii) the selection of trust nodes from the bordering node set. These two tasks are performed using LPP formulations. For convenience, decision variables for those LPPs are recalled here. The first LPP is solved to create MST partitions and its decision variable is $Y = (y_e)_{(N-1)\times 1}$. It is a link incidence vector, such that,

$$y_e = \begin{cases} 1, & \text{if } e \in E^{MST} \text{ is selected for elimination;} \\ 0, & \text{otherwise.} \end{cases} \tag{5.1}$$

The second LPP is solved to select trust nodes from the bordering node set and its decision variable is $X_B = (x_{sb})_{\sum_{s\in S}|B(s)|\times 1}$. It is a bordering node incidence vector, such that,

$$x_{sb} = \begin{cases} 1, & \text{if } b \in B(s) \text{ is selected, } s \in S; \\ 0, & \text{otherwise.} \end{cases} \tag{5.2}$$

These two variables are the common features of our segmentation-based trust system place-

ment schemes. They reappear in the formulations of both of outlined constrained problems.

5.3 Trust System Placement in a Resource-Constrained Scenario

In the resource-constrained scenario, trust systems are considered as a security resource. A centrality-based scheme is proposed to solve the placement problem. In general, centrality measurement is a method of ranking the nodes in a network. The ranking reflects the importance of the nodes depending on various criteria. It is now adopted for vulnerability assessment of critical infrastructures [Ste15]. In particular, it can also be used in cyber-physical vulnerability assessment of smart grid networks [Vel15, Sri13]. Centrality is measured using graph theoretic analysis of networks. The simplest form of centrality is the node degree centrality. It refers to the number of links connected to a node. In [Ern12], the usefulness of centrality measures in power system contingency analysis has been studied. Centrality measures can also be applied in the security design of communication networks [Sat16]. It can be a basis for security deployment strategies for resource-constrained cases. Energy SCADA system is associated with a supporting communication network. The current formulation exploits the degree centrality in trust system placement to offer better cyber protection with accompanying advantages. If a high degree centrality node is equipped with a trust system, a higher number of links can be monitored. A high degree centrality node has more chance to get affected by malicious activities. If a high degree centrality node is compromised, it can affect a higher number of nodes.

5.3.1 Problem Statement

To maximize cyber security, a SCADA network can be segmented in such a way that the most important nodes become the bordering nodes. Centrality measurement is used to rank the nodes in a network depending on their importance. For a resource constrained-scenario, the degree centrality of each node is considered. Another important consideration

is to place at least one trust system in a segment. This is to enhance a distributiveness in trust system placement. To act on power system islanding, trustworthy communications are required. This distributiveness reduces the range of unmonitored areas. The developed resource-constrained trust system placement problem is outlined in the following sentence. Select a given number of trust nodes so that the number of monitored links is maximized and inter-segment communications become trustworthy.

5.3.2 Proposed Solution

The MST partitioning heuristic is adopted for network segmentation to avoid a computationally expensive exhaustive search. It was developed for a known number of segments. For resource-constrained scenarios, the number of segments depends on the availability of trust systems. This is why an iterative version of the heuristic is necessary for our current solution. In addition, a centrality-measure is included to distinguish links at the time of partitioning. Unlike the previous chapter, repartitioning of segments is not included. This is because our current solution focuses on the centrality of bordering nodes rather than the balancing of segment sizes. To consider the impact of the centrality, a metric of link centrality is defined. For an undirected link between nodes u and v, it is defined by,

$$\psi(e_{u\leftrightarrow v}) = \frac{c_u + c_v}{2}. \tag{5.3}$$

The degree centralities of u and v are denoted by c_u and c_v respectively. The centrality metric in (5.3) is also used to compute a centrality-based weight for MST links. At first, the SCADA network graph is reduced to its MST; and then the MST is cut into partitions. Each MST partition is basically a segment skeleton. For a graph with N nodes, the MST contains $(N-1)$ links. To obtain K partitions, it requires the elimination of $(K-1)$ links from the MST. As a result, the rest of the $(N-1)-(K-1) = N-K$, links to form K partitions.

Algorithm 5.1 is iteratively applied in our proposed trust system placement scheme. It contains two LPPs where the solving of such LPPs are important parts of Algorithm 5.1.

This needs to be introduced for a better understanding of Algorithm 5.1. The LPP5.1 is a maximization problem that is used to create MST partitions. It eliminates the MST links based on their centrality metrics. The links that connect the higher degree centrality nodes are going to be eliminated so that partitions are formed. The decision variables of LPP5.1 is a link incidence vector, $Y = (y_e)_{(N-1)\times 1}$.

The objective is given in (5.4). The centrality metric based weight is defined in (5.5). It minimizes the chance of forming singleton nodes and helps the handling of star connections. Constraint (5.6) ensures the number of links to be eliminated.

LPP5.1: Tree Partitioning

$$\max_{Y} \sum_{e \in E^{MST}} c_e^L y_e, \tag{5.4}$$

where

$$c_e^L = \begin{cases} 1, & \text{if } e \in E^{MST} \text{ connects an MST leaf;} \\ \psi(e), & \text{otherwise.} \end{cases} \tag{5.5}$$

subject to

$$\sum_{e \in E^{MST}} y_e = K - 1, \tag{5.6}$$

$$y_e \in \{0, 1\}, \qquad \forall e \in E^{MST}. \tag{5.7}$$

The LPP5.2 is a minimization problem that is used to select trust nodes when segments are identified. The objective is given in (5.8). The fractional term in the objective ensures the priority of higher centrality. The degree centrality of a bordering node b located in a segment s is c_{sb}. Constraint (5.9) ensures at least one trust node is located in each segment. Constraint (5.10) ensures at least one trust node is selected for each inter-segment link. The decision variable of is a bordering node incidence vector, $X_B = (x_{sb})_{\sum_{s \in S} |B(s)| \times 1}$.

LPP5.2: Trust Node Computation

$$\min_{X_B} \sum_{s \in S} \sum_{b \in B(s)} \frac{1}{c_{sb}} x_{sb}, \tag{5.8}$$

subject to

$$\sum_{b \in B(s)} x_{sb} \geq 1, \qquad \forall s \in S, \tag{5.9}$$

$$\sum_{x_I \in X_I(l)} x_I \geq 1, \qquad \forall l \in L_{ss\prime}; \forall s, s\prime \in S, \tag{5.10}$$

$$x_{sb} \in \{0, 1\}, \qquad \forall s \in S \text{ and } \forall b \in B(s), \tag{5.11}$$

where

$$X_I(l) = \{x_{sb}, x_{s\prime b\prime}\}, \qquad b \in B(s), b\prime \in B(s\prime), s \neq s\prime. \tag{5.12}$$

Algorithm 5.1 takes the network graph and the number of segments as inputs. It provides the set of trust nodes as output. After initialization, the centrality metric of each link is computed using (5.3) (Line 4). The Kruskal algorithm is used to compute the MST of the given SCADA network ($T(V, E^{MST})$) (Line 6). The centrality based weight of each MST link is computed using (5.5) (Line 8). The MST partitions are obtained from the solution of LPP5.1. LPP5.1 returns E^I, the set of MST links to be eliminated (Line 10). E^{SS} is the set of remaining MST links and it is obtained from the difference of set E^{MST} and E^I (Line 11). The Disjoint-set algorithm is used to identify the partition set (Line 12). It computes the node set for each segment. Once the segments are defined, the next task is to identify the bordering node sets. For each segment, the bordering node set $B(s)$ is initialized as the empty set (Line 14). For each pair of segments, the inter-segment link set $L_{ss\prime}$ is also initialized as the empty set (Line 17). Segments are identified on a link-by-link basis. For each link in the network graph, both nodes are checked (Line 20 and 21). If

both nodes do not belong to the same segment, the link is an inter-segment link (Line 22). Thus, the inter-segment link set and bordering node sets are updated (Line 23-25). Once all bordering nodes are identified, LPP5.2 is solved to compute the trust node set (Line 28).

Algorithm 5.1 Centrality-Based Trust System Placement

Input: $G(V, E), K$;
Output: V^{Trust};

1: **begin**
2: $E^{SS} = \varnothing, N = |V|$; // Initialization
3: **for all** $e \in E$ **do**
4: Compute the centrality metric $\psi(e)$
5: **end for**
6: $T(V, E^{MST}) \leftarrow$ **Kruskal**$(G(V, E))$; // Computation of the MST using the Kruskal algorithm
7: **for all** $e \in E^{MST}$ **do**
8: Compute the centrality based weight c_e^L;
9: **end for**
10: $E^I \leftarrow$ **Solve LPP5.1**; // This will select the $(K-1)$ number of links that needs to be eliminated to creat the initial tree partitions
11: $E^{SS} = \{E^{MST} \setminus E^I\}$; // Returns $(N-K)$ number of links
12: $S = \{s_1, s_2, ..., s_K\} \leftarrow$ **Disjoint-set**(V, E^{SS}); // Identify node set for each segment
13: **for all** $s \in S$ **do**
14: $B(s) = \varnothing$; // Initialize bordering node sets
15: **end for**
16: **for all** $s \neq s\prime$ and $s, s\prime \in S$ **do**
17: $L_{ss\prime} = \varnothing$; // Initialize inter-segment link sets
18: **end for**
19: **for all** $e_{u \leftrightarrow v} \in E$ **do**
20: Find the segment x belongs to node u
21: Find the segment y belongs to node v
22: **if** $x \neq y$ **then**
23: $L_{xy} = \{L_{xy} \cup e\}$;
24: $B(x) = \{B(x) \cup u\}$;
25: $B(y) = \{B(y) \cup v\}$;
26: **end if**
27: **end for**
28: $V^{Trust} \leftarrow$ **Solve LPP5.2**; // This will select the trust node set
29: **return** V^{Trust}
30: **end**

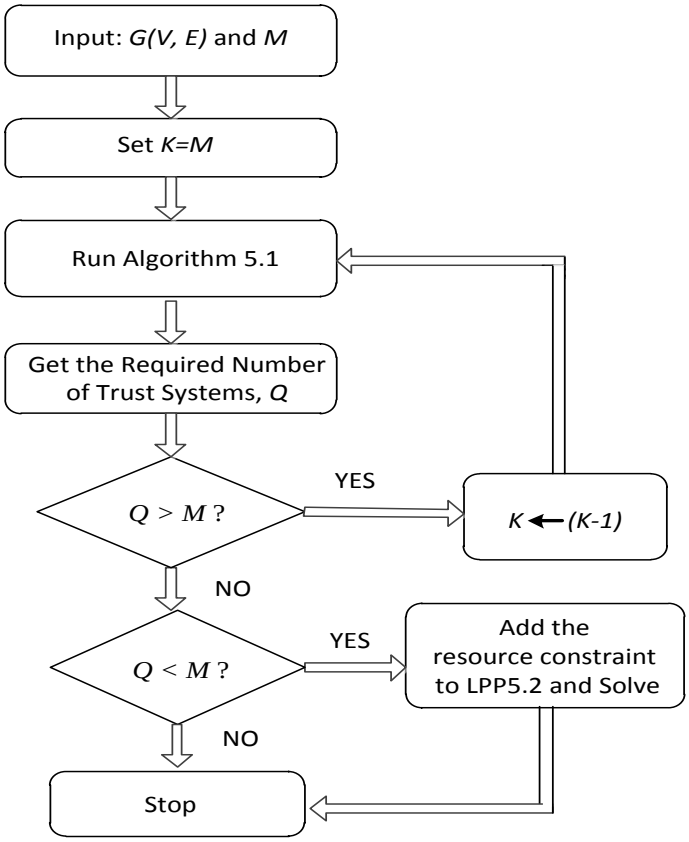


Figure 5.2: Flow diagram of the resource-constrained placement scheme.

Figure 5.2 shows the complete flow diagram of the proposed trust system placement scheme. The main idea is to search an appropriate number of segments from an iterative procedure. If the estimated number of trust systems (Q) is greater than M, then a new iteration with a lower value of K is required. This continues until $Q \leq M$. If $Q < M$, this additional constraint is added to LPP5.2:

$$\sum_{s \in S} \sum_{b \in B(s)} x_{sb} = M. \tag{5.13}$$

Figure 5.3 shows an illustrative example of a centrality-based solution using the proposed scheme where the dotted lines indicate inter-segment links. It shows that the original graph is decomposed to its MST. As the MST Links $(4, 5)$ and $(4, 9)$ have the highest centrality, they are eliminated to obtain three partitions. Finally, bordering nodes 4, 6, 9, and 13 are selected as trust nodes. Each segment contains at least one trust node.

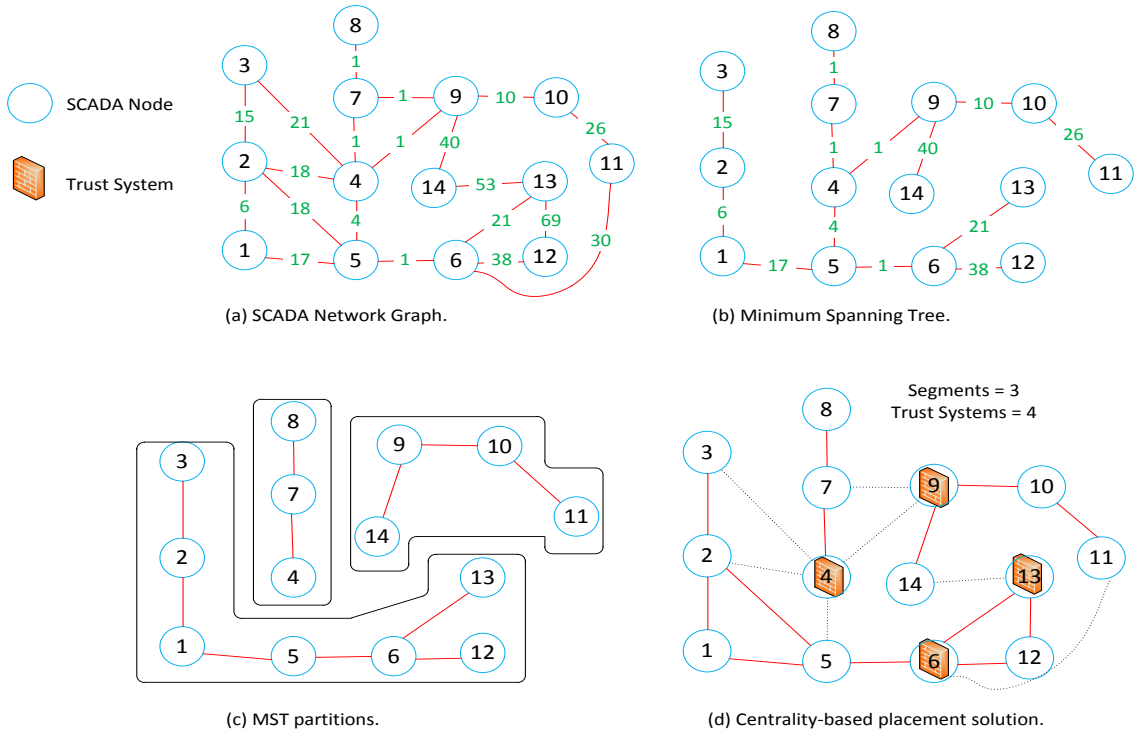


Figure 5.3: An illustrative example of a centrality-based placement solution.

5.3.3 Numerical Results and Analysis

Case studies are conducted for the IEEE BUS 30 and BUS 57 test system topologies [iee15]. Table 5.1 shows the summary of experimental parameters. The methodology described in [Gon11] is used to calculate propagation delays. It assumed an optical fiber network for SCADA communications. The proposed scheme is implemented using the MATLAB optimization toolbox. The aforementioned IEEE test system topologies are used as SCADA network graphs. The number of trust systems (M) is varied to observe the performance of the proposed scheme. For the BUS 30 topology, the value of M is varied from 4 to 12 with increments of 1. For the BUS 57 topology, the value of M is varied from 5 to 20 with increments of 1. These values are chosen with consideration of the SCADA network sizes. For the BUS 30 topology, the average number of nodes per trust system is varied between 2.5 and 7.5. For the BUS 57 topology, the average number of nodes per trust system is varied between 2.85 and 11.4. All experiments are run on a desktop machine with Intel Core i3 3.30 GHz CPU and 4 GB RAM. As our proposed

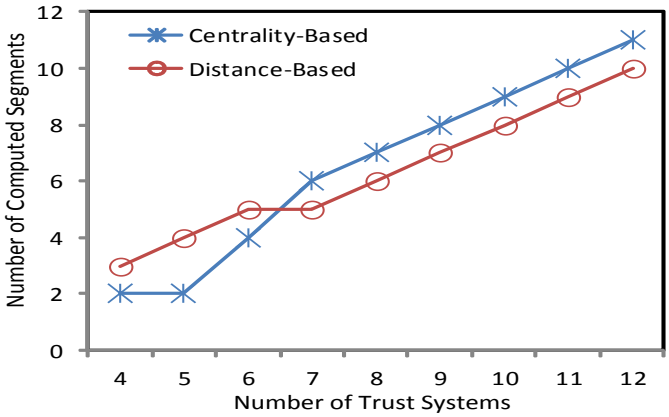
trust system placement scheme is a centrality-based approach, the impacts of centrality are mainly investigated. The proposed scheme is compared with a distance-based trust system placement approach. The distance-based approach eliminates links based on their propagation delays as it tries to minimize each segment's MST. However, it does not consider the centrality of nodes while still having similar steps as the proposed scheme. Degree centrality is considered for the proposed scheme. The comparisons are made based on the following two metrics: (i) number of computed segments and (ii) tolerance factor. The former is related to the spreading of an unmonitored cyber-attack. In general, a higher number of segments indicates lower spreading and *vice versa*. The latter is related to the chance of being monitored by a trust system. Lower values of tolerance indicate a higher chance of being monitored and *vice versa*.

Table 5.1: Experimental Network Parameters for the Resource-Constrained Case

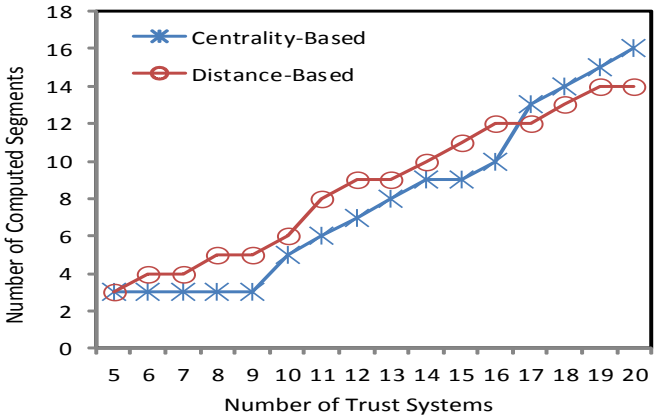
IEEE Test System Topology	**Number of Nodes**	**Number of Links**	**Mean Link Weight (μs)**	**Maximum Degree**	**Mean Degree**
BUS 30	30	42	24.1	7	2.7333
BUS 57	57	78	22.33	6	2.7368

Figures 5.4(a) and 5.4(b) show the comparative number of computed segments. Figure 5.4(a) shows the results for BUS 30 and Figure 5.4(b) illustrates the results for BUS 57. In both cases, curves can be divided into two regions: (i) lower region where M takes small values, and (ii) upper region where M takes larger values. The upper region starts at $M = 7$ and 16 respectively. In this region, the proposed scheme exhibits a higher number of segments than the distance-based approach. On the other hand, the distance-based approach exhibits a higher number of segments in the lower region. This happens due to the fact that the proposed scheme computes a higher number of inter-segment links. Each of such links is monitored by at least one trust system.

The tolerance factor is investigated (ρ) as a quality of cyber protection for SCADA



(a) BUS 30 Topology



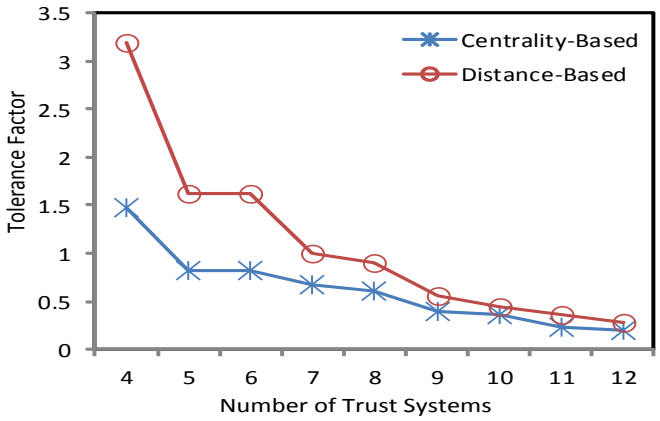
(b) BUS 57 Topology

Figure 5.4: Comparative number of computed segments.

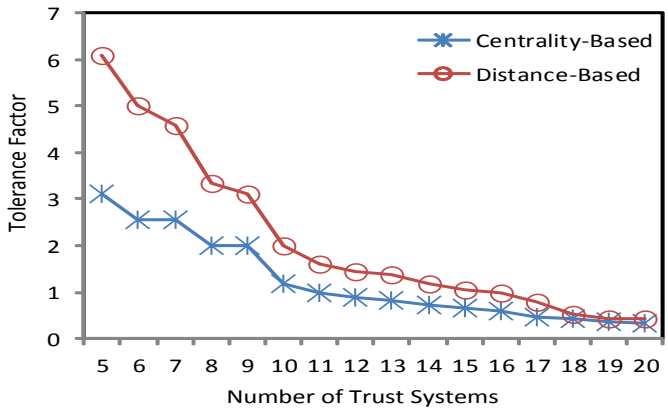
networks. A lower tolerance is an indication of more protected networks and *vice versa*. It is given by;

$$\rho = \frac{\text{Total number of unmonitored links in a network}}{\text{Total number of monitored links in a network}}. \tag{5.14}$$

Figures 5.5(a) and 5.5(b) show the comparative tolerance factor. They belong to BUS 30 and BUS 57, respectively. It is observed that the proposed scheme exhibits lower tolerance in all cases. In particular, it exhibits lower tolerance for the lower amount of resources. This implies that the proposed scheme outperforms the distance-based approach due to the selection of trust nodes preferring higher degree centrality. The gap between



(a) BUS 30 Topology



(b) BUS 57 Topology

Figure 5.5: Comparative tolerance factor.

the two approaches reduces as the number of trust systems increases. It means that the proposed scheme offers better utilization of the limited resources.

5.4 Trust System Placement in a Latency-Constrained Scenario

Trust systems are deployed for continuous monitoring of SCADA traffic. They are responsible for initiating and delivering alert messages in a timely manner. The delivery of alert messages is a time critical task. Any excessive delay between an observation of a malicious activity and the message delivery makes the network more vulnerable. This is why

latency-aware trust system placement is an important deployment strategy. As the trust systems use specialized hardware and dynamic placement is not feasible for a geographically distributed network. It requires a long-term deployment plan. The simultaneous consideration of time and resource constraints can only provide a best effort solution. It may or may not meet the time criticality depending on the amount of resource. For this reason, our latency-constrained problem does not include a resource constraint. The distributiveness in a trust system placement plan is beneficial to time criticality. It reduces latency since each segment contains at least one trust system.

5.4.1 Problem Statement

To deliver alert messages in a timely manner, a SCADA network can be segmented in such a way that each segment exhibits minimal geographic dispersion. For the latency-constrained case, the MST weight of each segment is considered as a measure of geographic dispersion. An MST weight also represents the total propagation delay of the MST route (shortest tree path) that connects all the nodes in a segment. Therefore, the MST weight can be used as an upper bound for the latency in a segment when at least one trust system is placed in a segment. Our latency-aware trust system placement problem can be stated as follows. Create segments so that the MST weight of each segment remains below a threshold.

5.4.2 Proposed Solution

It is recalled that the term 'partition' to refer a subtree and the term 'segment' to refer a subgraph. Each partition is the MST of a particular segment. There are two types of partitions: star and non-star. A star-connected partition may correspond to both star and non-star segments. On the other hand, non-star partitions always correspond to non-star segments. For the latency-constrained case, the number of segments depends on the latency threshold. Therefore, an iterative version of the previously proposed MST heuristic is developed for our current problem. In the previous chapter, segments were repartitioned

to balance sizes. In our current scheme, segments are repartitioned to meet the latency threshold. We start with a multi-objective optimization model and then demonstrates the necessity of a single objective for the latency-awareness. The following metric is defined to consider the impact of network topology. It is called the minimum degree of an MST link.

$$d_{min}^{MST}(e_{u\leftrightarrow v}) = \min(d_u^{MST}, d_v^{MST}). \tag{5.15}$$

The undirected link between nodes u and v is denoted by $e_{u\leftrightarrow v}$. The degree of a node is defined by the number of connections it has. The MST degree of a node is calculated by only considering MST links; d_u^{MST} and d_v^{MST}, that are the MST degrees of nodes u and v respectively. For the links that connect leaf nodes, the metric in (5.15) is unity. A star connection is formed when a number of leaf nodes are connected to a central node. Thus, the metric is used to identify star connections.

The segmentation approach works as outlined in the following paragraph. At first, the SCADA network graph is reduced to its MST. Thereafter the MST is cut into partitions. There are $(N-1)$ number of MST links in a graph of N nodes. It requires the elimination of $(K-1)$ number of MST links to form K partitions. The value of K depends on the latency threshold θ. In the iterative procedure, a reasonable starting point for K is required. It starts with a small value of K and then increases it to meet the latency threshold. The minimum value of K is K_{min}; it is obtained as follows.

The latency threshold is denoted by θ, while the total MST weight of a given network is denoted by W, and the maximum MST link weight is given by w_{max}^{MST}. To find the minimum feasible value of K, its lower bound is required to be set first. Note that our problem is feasible only when $\theta \leq W$. Let, $f(K)$ is the sum of the eliminated MST link weights that create K partitions. As K is always a positive integer, its lower bound can be expressed as,

$$K \geq \frac{W - f(K)}{\theta}, \qquad \forall K \geq 1. \tag{5.16}$$

From (5.16), it is clear that K decreases as $f(K)$ increases. The upper bound of $f(K)$

is given by,

$$f(K) \leq (K-1)w_{max}^{MST}, \qquad \forall K \geq 1. \tag{5.17}$$

Combining (5.16) and (5.17), the following lower bound is obtained for K,

$$K \geq \frac{W-(K-1)w_{max}^{MST}}{\theta}, \tag{5.18}$$

$$\text{Finally, } K \geq \frac{W+w_{max}^{MST}}{\theta+w_{max}^{MST}}, \qquad \forall K \geq 1. \tag{5.19}$$

To gives:

$$K_{min} = \left\lceil \frac{W+w_{max}^{MST}}{\theta+w_{max}^{MST}} \right\rceil. \tag{5.20}$$

LPP5.3 and LPP5.4 are two LPPs that are solved as part of the proposed scheme. LPP5.3 is a multi-objective maximization problem that creates MST partitions. It selects the set of MST links to be eliminated where the selection depends on the following two criteria: minimum degree and normalized weight. A linear combination of these criteria is formed using weighting factors α and β. Settings of these factors dominate the solutions regarding LPP5.3. As the current priority is the latency-awareness that is always $\beta = 1$ along with chosen lower values for α. At $\alpha = 0$, LPP5.3 becomes purely latency-aware. The decision variables of LPP5.3 is a link incidence vector $Y = (y_e)_{(N-1)\times 1}$.

LPP5.3: Tree Partitioning

$$\max_{Y} \sum_{e \in E^{MST}} \big(\alpha d_{min}^{MST}(e) + \beta \widetilde{w}(e)\big) y_e, \tag{5.21}$$

where

$$\widetilde{w}(e) = \frac{w(e)}{\arg\max_{w(e)} w(e)}, \qquad \forall e \in E, \tag{5.22}$$

subject to

$$\sum_{e \in E^{MST}} y_e = K - 1, \tag{5.23}$$

$$y_e - d_{min}^{MST}(e) < 0, \qquad \forall e \in E^{MST}, \tag{5.24}$$

$$y_e \in \{0, 1\}, \qquad \forall e \in E^{MST}. \tag{5.25}$$

The objective is expressed in (5.21) with the normalized weight is given in (5.22). Constraint (5.23) addresses the number of MST links to be eliminated and constraint (5.24) prevents the chance of forming singleton nodes. Constraint (5.24) also helps the handling of star connections.

LPP5.4 is a single objective minimization problem that selects trust nodes when segments are finalized. It minimizes the number of required trust systems under multiple constraints. The decision variable of LPP5.4 is a bordering node incidence vector, $X_B = (x_{sb})_{\sum_{s \in S} |B(s)| \times 1}$.

LPP5.4: Trust Node Computation

$$\min_{X_B} \sum_{s \in S} \sum_{b \in B(s)} x_{sb}, \tag{5.26}$$

subject to

$$\sum_{b \in B(s)} x_{sb} \geq 1, \qquad \forall s \in S, \tag{5.27}$$

$$\sum_{x_I \in X_I(l)} x_I \geq 1, \qquad \forall l \in L_{ss\prime}; \forall s, s\prime \in S, \tag{5.28}$$

$$x_{sb} \in \{0,1\}, \qquad \forall s \in S \text{ and } \forall b \in B(s), \tag{5.29}$$

where

$$X_I(l) = \{x_{sb}, x_{s\prime b\prime}\}, \qquad b \in B(s), b\prime \in B(s\prime), s \neq s\prime. \tag{5.30}$$

The objective is expressed in (5.26). Constraint (5.27) ensures that at least one trust node is selected for each inter-segment link. Constraint (5.28) ensures at least one trust node is selected from each segment.

The latency-aware trust system placement scheme is presented using Algorithm 5.2 and it is described as follows. The inputs to Algorithm 5.2 are a network graph and a latency threshold. Its output is the set of trust nodes. After initialization, the Kruskal algorithm is applied to obtain the MST of the SCADA network ($T(V, E^{MST})$) (Line 3).

The minimum degree and the normalized weight of each MST link are computed using (5.15) and (5.22) respectively (Line 5-6). The while loop iteratively computes segments (Line 8-20). LPP5.3 computes E^I, the set of MST links to be eliminated (Line 9). The set of remaining MST links E^{SS} is obtained from the difference of set E^{MST} and E^I (Line 10). The Disjoint-set algorithm is applied to identify the node set for each segment (Line 11). The while loop checks for the maximum MST weight of the non-star partitions τ_{max} (Line 12). There is a reference value τ_{ref} to observe the improvement in latency and is initially set to a large value (W). If τ_{max} does not satisfy the latency threshold and remains lower than the reference, then K is increased by one and the reference is updated. This continues until the latency threshold is met by all non-star partitions or until no successive improvement occurs in (Line 13-19). As star-connected partitions are not reducible, only non-star partitions are taken for latency improvement.

Once the segments are identified, parameters of LPP5.4 are required to be computed. For each segment, the bordering node set $B(s)$, is initialized as the empty set (Line 22). For each pair of segments, the inter-segment link set $L_{ss\prime}$ is also initialized as the empty set (Line 25). For each link in the network graph, segments are identified for both of its nodes (Line 29 and 30). If both of them are not located in the same segment, it is identified as an

Algorithm 5.2 Latency-Aware Trust System Placement

Input: $G(V, E), \theta$;
Output: V^{Trust};

begin
$E^{SS} = \varnothing, N = |V|, K = K_{min}, \tau_{ref} = W, \phi = 0$; // Initialization
$T(V, E^{MST}) \leftarrow$ **Kruskal**$(G(V, E))$; // MST computation
for all $e \in E^{MST}$ **do**
  Compute the minimum degree $d_{min}^{MST}(e)$;
  Compute the normalized weight $\widetilde{w}(e)$;
end for
while $\phi < 1$ **do**
  $E^{I} \leftarrow$ **Solve LPP5.3**; // Selection of $(K-1)$ number of links for elimination
  $E^{SS} = \{E^{MST} \setminus E^{I}\}$; // Set of remaining $(N-K)$ number of links
  $S = \{s_1, s_2, ..., s_K\} \leftarrow$ **Disjoint-set**(V, E^{SS}); // Node set identification for each segment
  Compute τ_{max}
  if $\tau_{max} < \theta$ or $\tau_{max} \geq \tau_{ref}$ **then**
    $\phi = 1$ // Termination condition
  end if
  if $\tau_{max} > \theta$ and $\tau_{max} < \tau_{ref}$ **then**
    Set $K = K + 1$
    Set $\tau_{ref} = \tau_{max}$
  end if
end while
for all $s \in S$ **do**
  $B(s) = \varnothing$; // Initializing bordering node sets
end for
for all $s \neq s\prime$ and $s, s\prime \in S$ **do**
  $L_{ss\prime} = \varnothing$; // Initializing inter-segment link sets
end for
for all $e_{u \leftrightarrow v} \in E$ **do**
  Identify the segment x belongs to node u
  Identify the segment y belongs to node v
  if $x \neq y$ **then**
    $L_{xy} = \{L_{xy} \cup e\}$;
    $B(x) = \{B(x) \cup u\}$;
    $B(y) = \{B(y) \cup v\}$;
  end if
end for
$V^{Trust} \leftarrow$ **Solve LPP5.4**; // Selection of the trust node set
return V^{Trust}
end

inter-segment link (Line 30). This outlines how the inter-segment link set and bordering node sets are updated (Line 31-33). Once these sets are finalized, LPP5.4 is solved to obtain the trust node set (Line 36).

In Algorithm 5.2, input parameters for LPP5.4 are dependent on the output of LPP5.3. Thus, the overall solution is governed by the settings of α and β. Figure 5.7 shows an illustrative example of the solutions obtained using different settings. This example considers the IEEE BUS 14 test system topology.

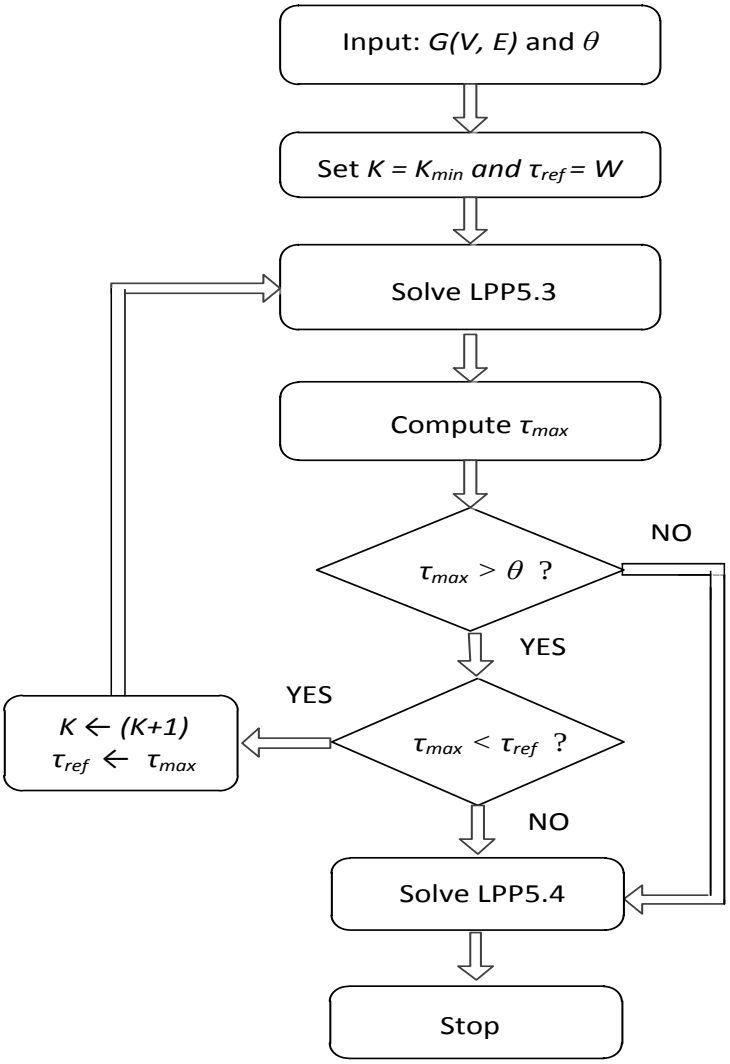


Figure 5.6: Flow diagram of the latency-constrained placement scheme.

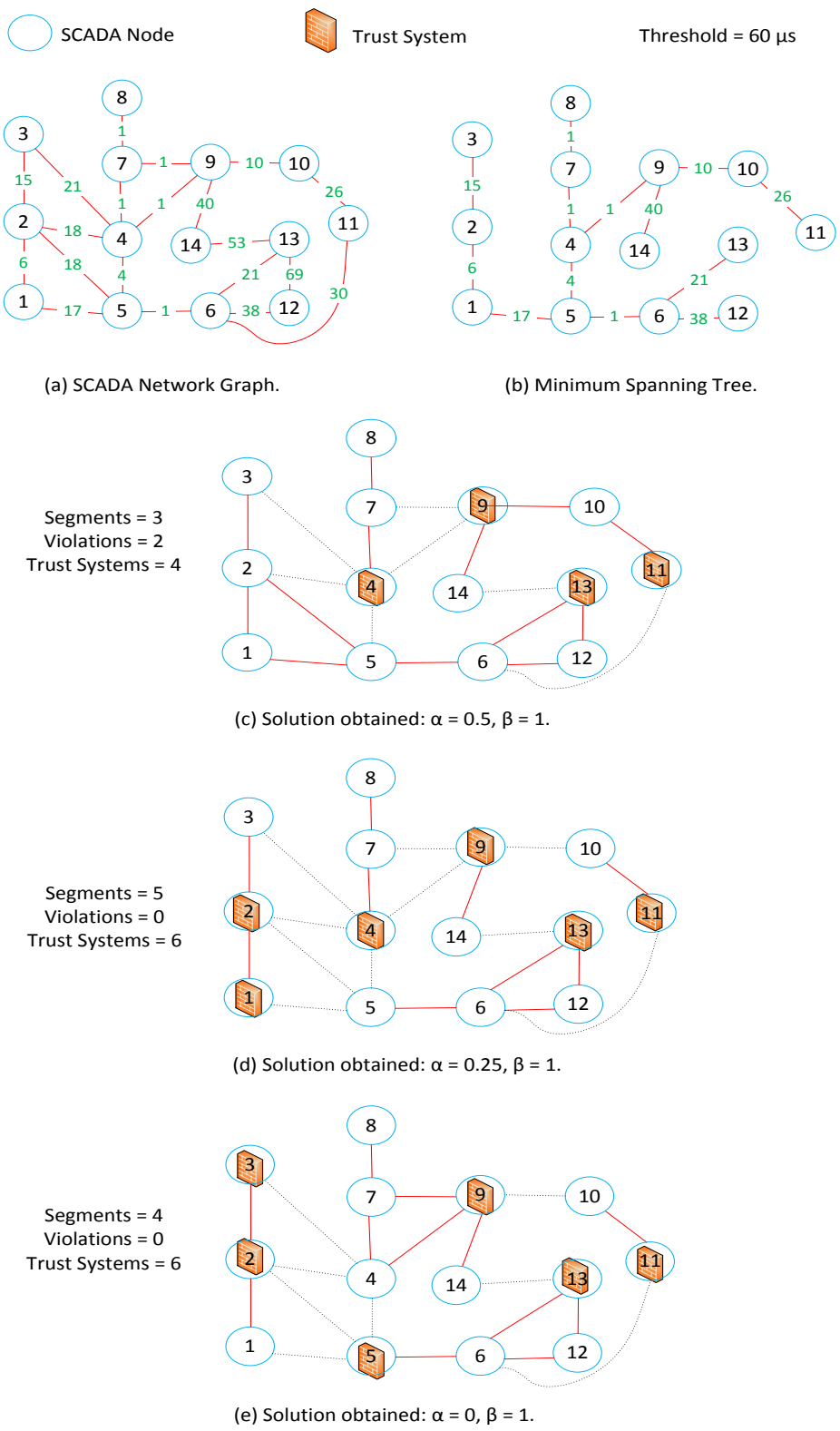


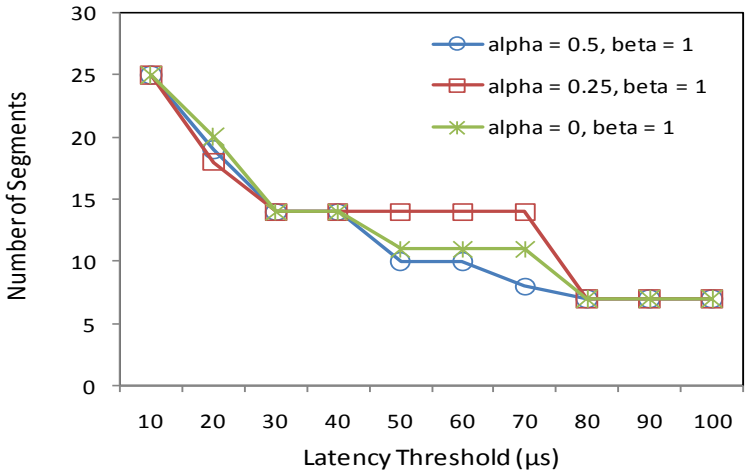
Figure 5.7: An illustrative example using the IEEE BUS 14 topology.

5.4.3 Numerical Results and Analysis

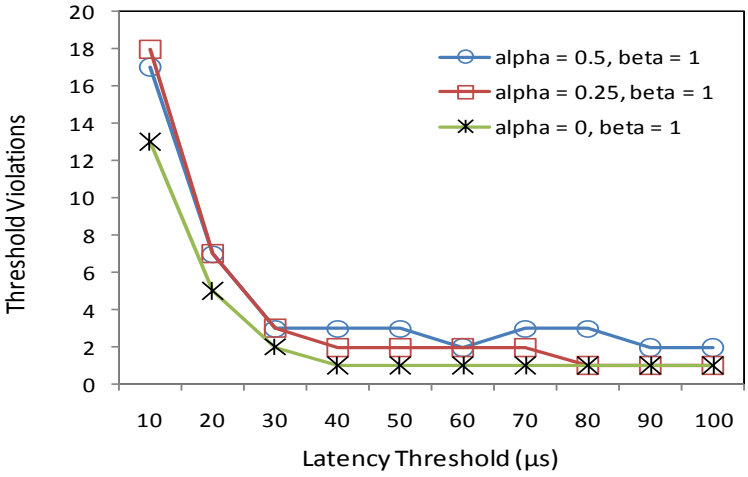
Table 5.2: Experimental Parameters for the Latency-Constrained Case

IEEE Test System Topology	**Number of Nodes (Network Size)**	**Number of Active Links**	**Link Weight Mean (μs)**	**Link Weight Standard Deviation (μs)**
BUS 118	118	179	8.35	6.22

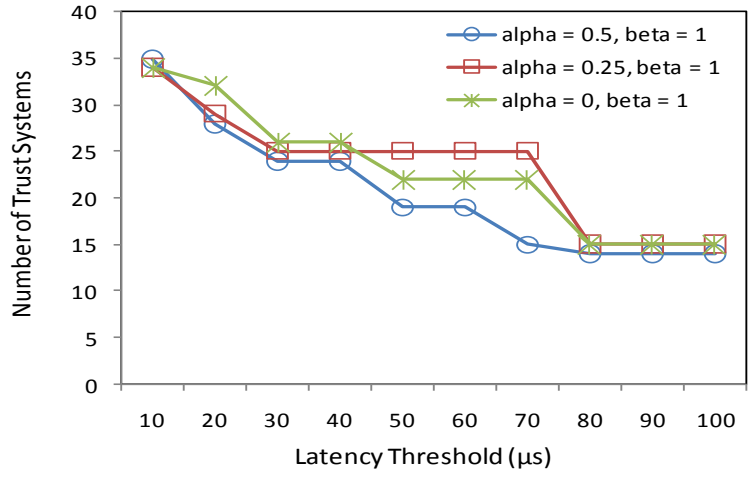
The IEEE BUS 118 test system topology is considered for conducting case studies [iee15]. Experimental parameters are summarized in Table 5.2. An optical fiber network is assumed to obtain propagation delays. The methodology of obtaining propagation delays was introduced in [Gon11]. The MATLAB optimization toolbox is used to implement the proposed scheme where the latency threshold for MST weights (θ) is varied to observe performance. The value of θ is varied from 10 to 100 μs with increments of 10 μs. These values are chosen with consideration to the propagation delay parameters of the IEEE BUS 118 topology. A desktop machine with Intel Core i3 3.30 GHz CPU and 4 GB RAM is used for running the experiments. The performance is evaluated using three different settings for α and β. Since the scheme is latency-aware, we always keep $\beta = 1$. The values of α are 0.5, 0.25, and 0. The proposed scheme is purely latency-aware when $\alpha = 0$. Two other settings for α can be considered as partially latency-aware. The comparison criteria include three metrics: (i) number of computed segments, (ii) number of threshold violations, and (iii) number of trust systems. Figures 5.8(a), (b), and (c) depict comparative performance of different settings. In general, a lower threshold involves computing a higher number of segments that result in a higher resource requirement. It can lead to an increased number of violations.



(a) Number of computed segments.



(b) Number of threshold violations.



(c) Number of required trust systems.

Figure 5.8: Comparative performance of different settings.

Figure 5.8(a) shows that in the lower threshold region (up to 40 μs), a slightly higher number of segments are computed for $\alpha = 0$. Two other settings show very similar comparative results. For the region in Figure 5.8(b) where $\alpha = 0$, a much lower threshold violation is exhibited. Two other settings are still very close regarding violations. Figure 5.8(c) shows that the required number of trust systems is a little bit higher for $\alpha = 0$ in this region. In the mid-latency region (above 40 and below 80 μs), Figure 5.8(a) and (c) show much larger gaps between $\alpha = 0.5$ and $\alpha = 0.25$. The $\alpha = 0$ stays in the middle of them. For the higher threshold region (above 80 and up to 100 μs), three settings compute the same number of segments but the trust system requirement is slightly lower for $\alpha = 0.5$. It also shows higher violations than two others. Figure 5.8(b) shows the same number of violations for $\alpha = 0.25$ and $\alpha = 0$ in the higher threshold region. The best performance in terms of threshold violation is observed for $\alpha = 0$. Therefore, our setting for pure latency-awareness is effective in network segmentations.

5.5 Conclusion

In this chapter, an introduction and an evaluation of two trust system placement schemes were completed for smart grid SCADA networks. The first scheme was proposed for a resource-constrained case. It exploits the degree centrality of bordering nodes to provide more efficient utilization of resources. The second scheme was proposed for a latency-constrained case where it uses MST weights as a measure of latency. It tries to keep the MST weight of each segment below a latency threshold. In both of the proposed schemes, the trust system placement problem is divided into two major parts: (i) network segmentation and (ii) trust node selection. The first part is required to be solved to formulate the second part. This procedure greatly simplifies the placement problems.

Chapter 6

Trust System Placement for Distributed Monitoring in SCADA Networks

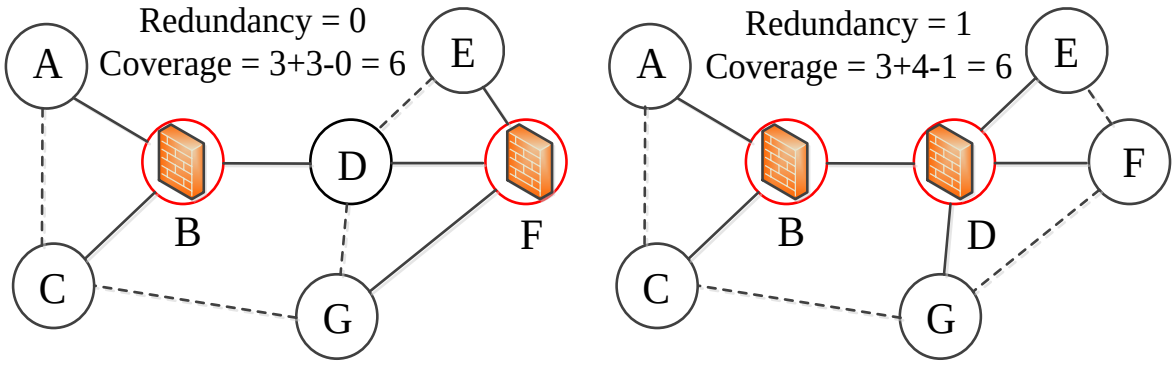
6.1 Introduction

In the previous two chapters, we studied several segmentation-based trust system placement approaches. In those placement approaches, trust systems are operated in the tunnel/gateway mode [Coa10]. This only guarantees the monitoring of inter-segment traffic. A number of successive hops remain unmonitored inside a large segment. It is worthwhile to investigate other modes of operation such as the active/router mode. In the router mode, a trust system is placed as an inline hardware to block cyber-attacks. It also works as a blocking device for potentially harmful packets and thus it can be utilized for securing routes in smart grid networks [Zha13b]. The router mode is not an alternative to the gateway mode but rather they are considered complementary to each other. The best practice is to deploy trust systems in a distributed manner [Coa08]. This includes both types of distributions, functional and geographical. To enhance security, those distributed trust systems require coordination and collaboration. A gateway trust system can be deployed as a coordinator for a set of router trust systems.

In this chapter, the trust system placement problem is studied from a distributed monitoring perspective. Two trust system placement schemes are proposed for enhancing distributed monitoring in SCADA networks. Such schemes are developed targeting two different quality metrics for distributed monitoring: (i) link coverage and (ii) path tolerance. They aim to improve the quality of distributed monitoring in resource-constrained scenarios.

6.2 Link Coverage Maximization

Trust systems are deployed to bring SCADA links under security monitoring. They provide better security when the higher number of links are monitored. In a resource-constrained network, there are a limited number of trust nodes. The link coverage refers to the number of the monitored links in a network where a higher number improves the security coverage. The degree centrality-based trust node selection is a way of increasing this number. However, the redundancy occurs when two trust nodes are directly connected. In such a case, the same link is monitored by two trust systems. As a result, the overall coverage is no longer the sum of their degree centralities.



(a) No redundancy.

(b) One redundancy.

Figure 6.1: Link coverage calculation examples.

Figure 6.1 shows two examples that explain how to calculate the coverage. In the first

example, there is no redundancy as two trust nodes B and F are not directly connected. Thus, the total coverage is the sum of their degree centralities. The second example illustrates a redundancy due to the direct connection between two trust nodes B and D. The redundant link is discounted to obtain the actual coverage. The monitored links are shown by the solid lines, while the unmonitored ones are displayed by the dashed lines.

A quadratic assignment model named link coverage maximization (LCM) is developed to take the redundancy into account. It is necessary to have an exact expression for the link coverage so that an objective function can be set. An undirected graph $G(V, E)$ is considered for a SCADA network. It consists of N nodes where M is the number of trust systems that are going to be placed. The degree centrality of the node i is c_i. The decision variable of LCM is $X = (x_i)_{N\times 1}$. It is a trust node assignment vector, such that,

$$x_i = \begin{cases} 1, & \text{if } i \text{ is a trust node;} \\ 0, & \text{otherwise.} \end{cases} \tag{6.1}$$

In the LCM model, a problem is solved that maximizes the link coverage for a given number of trust systems. It is formulated as follows.

$$\text{(LCM)} \quad \max_X \sum_{i=1}^{N} c_i x_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} e_{ij} x_i x_j, \tag{6.2}$$

subject to

$$\sum_{i=1}^{N} x_i = M, \tag{6.3}$$

where

$$e_{ij} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ are directly connected;} \\ 0, & \text{otherwise.} \end{cases} \qquad \forall i, j \in V, i \neq j; \tag{6.4}$$

$$x_i \in \{0, 1\}. \tag{6.5}$$

The objective that involves maximizing the link coverage is given by (6.2). It contains a quadratic term that represents the redundancy. The resource constraint is given by (6.3) and the redundancy is determined in (6.4). To reduce the complexity of LCM, the linearization technique proposed in [She07] is applied. The linearization introduces a vector of auxiliary variables, $\widetilde{X} = (\chi_{ij})_{(\sum_i^N \sum_{j=1,j\neq i}^N \mathbf{1})\times 1}$. The linearized objective is given by,

$$\max_{X,\widetilde{X}} \sum_{i=1}^{N} c_i x_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1,j\neq i}^{N} e_{ij}\chi_{ij}. \tag{6.6}$$

The following additional constraints describe the relationship between the decision and the auxiliary variables,

$$\chi_{ij} \geq x_i + x_j - 1, \qquad \forall i,j \in V, i \neq j, \tag{6.7}$$

$$\chi_{ij} \geq 0, \qquad \forall i,j \in V, i \neq j. \tag{6.8}$$

To verify the impact of redundancy consideration, LCM is compared with a linear assignment problem (LAP) approach. LAP only considers degree centrality since redundancy incurs quadratic terms. Figure 6.2 shows a comparison of the link coverage for the IEEE BUS 118 and BUS 300 test system topologies. LCM reaches the full coverage much earlier than LAP for both topologies. It reaches the full coverage at ∼48% and ∼45% trust nodes, respectively. It is observed that LAP requires ∼30% and ∼25% additional trust nodes to achieve the full coverage. Figure 6.3 shows a comparison of the redundancy for the IEEE BUS 118 and BUS 300 test system topologies. This time the redundancy is plotted versus the link coverage. The gap between LCM and LAP becomes much wider as the link coverage is increased. For LCM, ∼28% and ∼20% links are redundant at the full coverage. The reduced number of redundantly monitored links cause higher coverage. For LAP, the redundancy is ranging from $\sim 75\%$ to $\sim 85\%$ at the full coverage.

From the discussion above, it is clear that the proposed LCM is a resource-efficient trust system placement model. Its average time complexity is $\sim O(|E|\log(|E|+|V|))$.

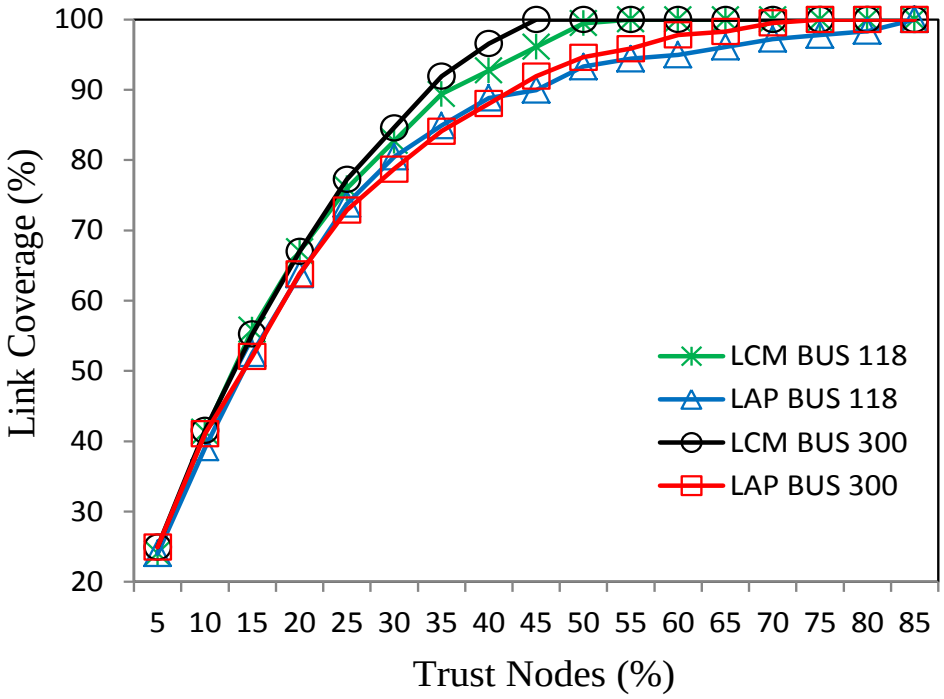


Figure 6.2: Coverage comparison between LCM and LAP.

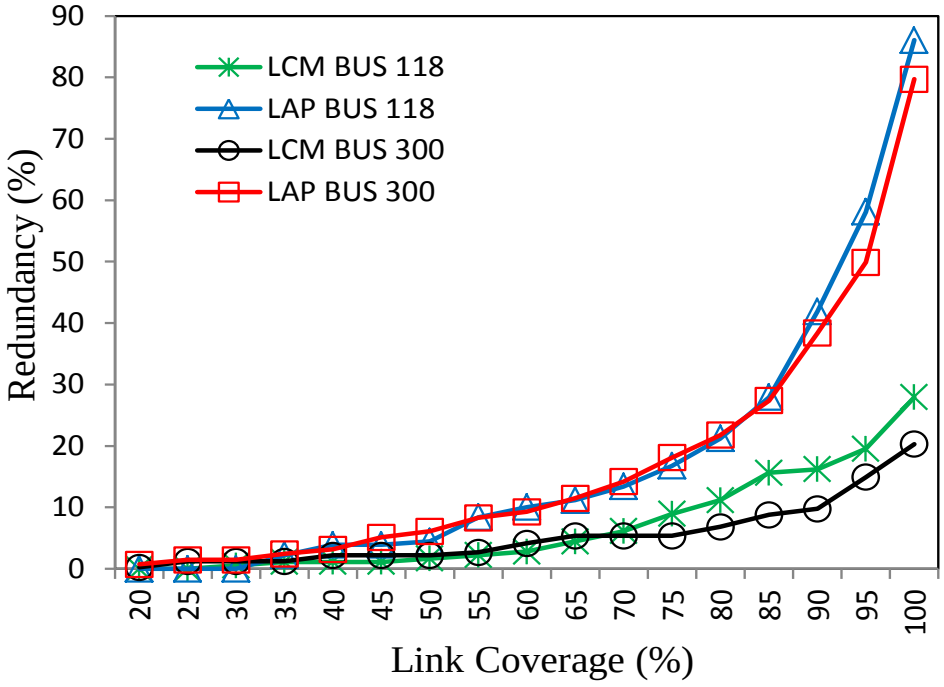


Figure 6.3: Redundancy comparison between LCM and LAP.

6.3 Minimal Path Tolerance

LCM maximizes the number of monitored links for a resource-constrained case. However, the amount of traffic through a node also depends on routing policies. In addition, LCM does not guarantee the protected routing paths. A protected route is not always feasible due to the resource constraints and latency concerns. For example, in Figure 6.1, the shortest routing path between A and G, is A-C-G. In Figure 6.1(a), the most secure route between A and G, is A-B-D-F-G. In Figure 6.1(b), the most secure route between A and G, is A-B-D-G. This implies the difference between a shortest path and a most secure path. Thus, a secure path becomes longer than the shortest one. Trust systems are required to be placed in a distributed manner to improve this situation. A model is developed that distributes trust systems based on the all pair shortest path routing policy. It aims to map secure routes on shortest paths. It is conceptually related to the betweenness centrality, which is computed using shortest paths [Hol03]. The key difference is that we are more interested in the hop distance than the node centrality.

To address the distributiveness, we define a new metric called path tolerance. It is defined as the maximum number of successive non-trust nodes on a routing path. It is related to the longest unmonitored subpath on a route, whose length is measured by the hop count. This is because SCADA nodes are not located in a fixed interval of distance. A regular SCADA node is a non-trust node unless it is equipped with a trust system. In other words, if a node is not a trust node, it is called a non-trust node. There are two key motivations behind the introduction of the path tolerance: (i) reduction of hop distance between trust nodes and (ii) limiting the propagation of unwanted traffic through shortest paths. The reduction of hop distance between trust nodes to improve collaboration. The limited propagation to stop malicious packets and cyber-attacks. The path tolerance also reduces the maximum hop distance between a node and its nearest trust node.

Figure 6.4 shows some simple examples that clarify the concept of the path tolerance. It is clear that the relative position of trust nodes is the key factor in this calculation. For the same number of trust nodes, different path tolerances are observed for different placement policies. A lower path tolerance is observed when the trust nodes are located

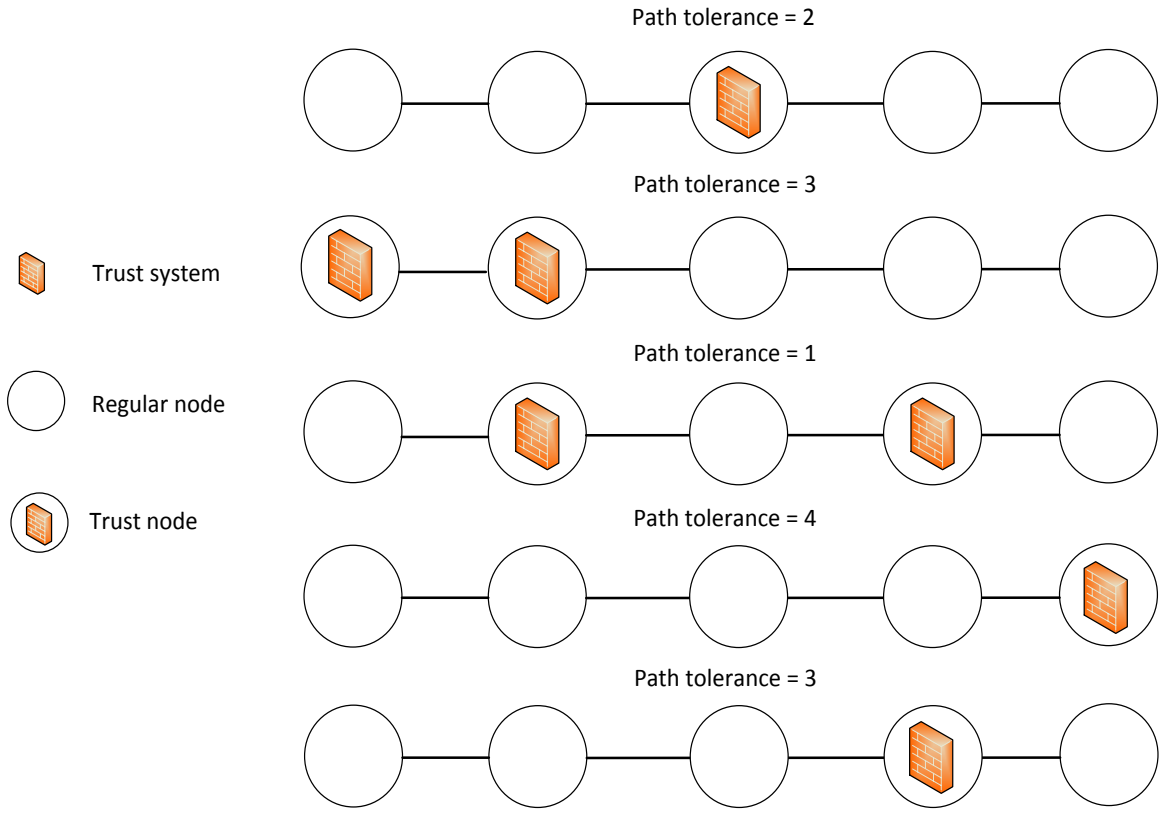


Figure 6.4: Path tolerance calculation examples.

in a more distributed manner. For a resource-constrained network, our target is to keep the path tolerance as low as possible. Therefore, a path-by-path search for minimal path tolerance is required to be completed. The computation of all pair shortest paths in a network is very useful for conducting such a search. Note that the interest of the proposed method is the hop count, not the distance. If a path does not have a trust node at all, its path tolerance is the hop count plus one. The path tolerance is zero when a path only comprises trust nodes.

An important consideration includes that some of the strategically important SCADA nodes are required to be equipped with the trust systems. This can be assured by a reservation for such nodes. Those nodes are assumed to be the preassigned trust nodes.

To address the facts discussed in this section, an optimization model named minimal path tolerance (MPT) is derived. The main idea is to set a limit for the path tolerance so that all pair shortest paths in a network satisfy it. At the same time, the model includes

a reservation for the preassigned trust nodes. The model is formulated as follows,

$$\text{(MPT)} \quad \max_{X,\tilde{X}} \sum_{i=1}^{N} c_i x_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} e_{ij}\chi_{ij}, \tag{6.9}$$

subject to

$$\sum_{i=1}^{N} x_i = M, \tag{6.10}$$

$$1 + a_{ij} - \sum_{k \in P_{ij}} x_k a_{ij} \leq \gamma_M, \qquad \forall i, j \in V, i \neq j, \tag{6.11}$$

$$x_i - g(x_i) \geq 0, \qquad \forall i \in V, \tag{6.12}$$

$$\chi_{ij} \geq x_i + x_j - 1, \qquad \forall i, j \in V, i \neq j, \tag{6.13}$$

$$\chi_{ij} \geq 0, \qquad \forall i, j \in V, i \neq j, \tag{6.14}$$

$$x_i \in \{0, 1\}. \tag{6.15}$$

The main features of MPT are constraints (6.11) and (6.12). Constraint (6.11) keeps the path tolerance below a maximum value γ_M. The shortest hop distance between i and j is denoted by a_{ij}. The node set for the shortest path between i and j is denoted by P_{ij}. It includes i, j, and all intermediary nodes between them. Constraint (6.12) reserves the preassigned trust nodes. The set of the preassigned trust nodes is denoted by V^{rsv}. The function $g(.)$ is given as,

$$g(x_i) = \begin{cases} 1, & \text{if } i \in V^{rsv}; \\ 0, & \text{otherwise.} \end{cases} \qquad V^{rsv} \subset V; \tag{6.16}$$

The solution to the proposed MPT model cannot be directly obtained since the maximum path tolerance γ_M is not known in advance. It depends on M, and on the preassigned trust nodes. A heuristic algorithm is developed that uses two integer linear programs (ILPs). The first ILP is used in an iterative procedure that finds the lowest feasible value of γ_M. The second ILP attempts to improve the objective. The ILPs are as follows.

$$\text{(ILP6.1)} \quad \min_{X} \sum_{i=1}^{N} x_i, \tag{6.17}$$

subject to

$$1 + a_{ij} - \sum_{k \in P_{ij}} x_k a_{ij} \leq \gamma, \qquad \forall i, j \in V, i \neq j, \tag{6.18}$$

$$x_i - g(x_i) \geq 0, \qquad \forall i \in V, \tag{6.19}$$

$$x_i \in \{0, 1\}. \tag{6.20}$$

ILP6.1 minimizes the resource requirement for a given path tolerance γ. It iteratively computes the initial trust node set V^{Init}. The following function is defined to include V^{Init} in the final solution.

$$\widetilde{g}(x_i) = \begin{cases} 1, & \text{if } i \in V^{Init}\ ; \\ 0, & \text{otherwise.} \end{cases} \qquad V^{Init} \subset V; \tag{6.21}$$

ILP6.2 is solved to utilize the unassigned resources (if any) for the link coverage maximization. The number of trust nodes assigned by ILP6.1 is denoted by Q.

$$\text{(ILP6.2)} \quad \max_{X, \widetilde{X}} \sum_{i=1}^{N} c_i x_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} e_{ij} \chi_{ij}, \tag{6.22}$$

subject to

$$\sum_{i=1}^{N} x_i = M, \tag{6.23}$$

$$x_i - \widetilde{g}(x_i) \geq 0, \qquad \forall i \in V, \tag{6.24}$$

$$\chi_{ij} \geq x_i + x_j - 1, \qquad \forall i, j \in V, i \neq j, \tag{6.25}$$

$$\chi_{ij} \geq 0, \qquad \forall i, j \in V, i \neq j, \tag{6.26}$$

$$x_i \in \{0, 1\}. \tag{6.27}$$

The final set of trust nodes is denoted by V^{Trust} and it is observed that $V^{Trust} \supseteq V^{Init}$. Algorithm 6.1 presents the MPT heuristic. The iteration starts with $\gamma = 1$. If it requires more resources than available amount, γ is increased by 1. This iteration process continues

Algorithm 6.1 MPT Heuristic

Input: $G(V, E), M, V^{rsv}$;
Output: V^{Trust};

begin
$V^{Init} = \varnothing, N = |V|, \phi = 0, \gamma = 1$; // Initialization
while $\phi < 1$ **do**
  $V^{Init} \leftarrow$**Solve ILP6.1**; // Solving the first ILP
  $Q = |V^{Init}|$; // Calculation of resource requirement
  if $Q > M$ **then**
    $\gamma = \gamma + 1$;
  else
    $\gamma_M = \gamma$;
    $\phi = 1$; // Termination condition
  end if
end while
if $Q < M$ **then**
  $V^{Trust} \leftarrow$**Solve ILP6.2**; // Solving the second ILP
else
  $V^{Trust} = V^{Init}$;
end if
return V^{Trust};
end

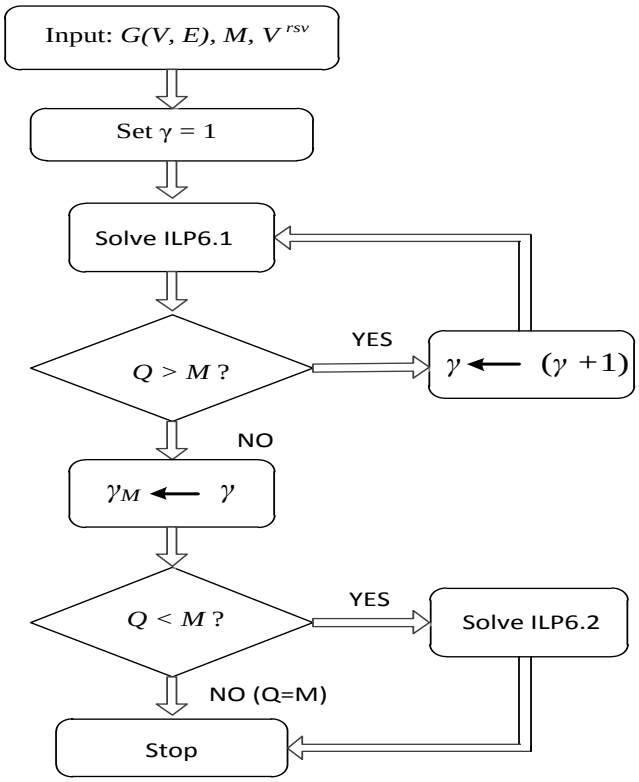


Figure 6.5: Flow diagram for the MPT solution.

until a feasible amount is attained. If there are still unassigned resources, the second ILP is solved. Figure 6.5 illustrates the corresponding flow diagram. In the worst cast, MPT's average complexity is $\sim O(|P^*_{ij}||V|^2 \log(|V|))$, where $|P^*_{ij}|$ is the cardinality of P^*_{ij}; the node set of a shortest path that has the highest hop count in a given network.

6.4 Numerical Results

To evaluate the proposed placement schemes, case studies are conducted on the IEEE BUS 118 and BUS 300 test system topologies [iee15]. These topologies are used as the SCADA network graphs. The computation of all pair shortest paths was one of the most important parts in the study of path tolerance. The Floyd-Warshall algorithm is used to compute all pair shortest paths. The hop count is used instead of the distance in that computation. Table 6.1 summarizes the statistics of the shortest paths. The maximum hop count for BUS 118 and BUS 300 are 14 and 24, respectively. It means that the highest possible values

for path tolerances could be 15 and 25. For the preassigned trust nodes, the eigenvector centrality is used to rank the nodes' importance [New10]. A reservation is made for high-ranked nodes. The number of trust nodes is gradually increased to observe the link coverage and the path tolerance. All experiments are implemented using an integrated MATLAB-CPLEX solver on a desktop machine with Intel Core i3 3.30 GHz and 8 GB RAM. It took a total of 164 hours (∼7 days) to obtain the results presented in this section.

Figures 6.6 and 6.7 show comparisons between LCM and MPT for a 0% reservation. LCM exhibits a better link coverage while MPT displays a much lower path tolerance. Thus, the motives behind their formulations are justified. LCM gradually reaches the full coverage, whereas MPT fluctuates a little bit in the middle. The opposite characteristics are observed for the path tolerance. MPT gradually decreases as the number of trust nodes is increased. LCM fluctuates in the middle and drops suddenly to the lowest value.

Figure 6.8 shows the impact of reservations on the link coverage for BUS 118. With LCM the coverage initially degrades. The gap is reduced as the number of trust nodes is increased. For MPT, reservations initially add fluctuations and then get improved coverages. The higher number of reservations is not necessarily following a particular trend in the coverage. Figure 6.9 shows the impact of reservations on the link coverage regarding BUS 300. For LCM, the degradation is clearly observed. MPT shows a lot of fluctuations but a clear trend of degradation is observed.

Figure 6.10 shows the impact of reservations on the path tolerance for BUS 118. LCM exhibits no regular trend, however, fluctuations are noted. For MPT, it is clear that reservations degrade the performance on a regular basis. Figure 6.11 shows the similar MPT characteristics for BUS 300. For LCM, a much higher path tolerance is observed for BUS 300 that is due to larger network size. No regular relation can be assumed between reservations and LCM's fluctuating path tolerance as it is much steeper than that of BUS 118.

So far our discussion on the path tolerance only includes the maximum path tolerance for a network. It does not describe the whole scenario as the distribution of the path tolerance is also an important metric. To provide a comprehensive view, Figure 6.12 shows

Table 6.1: Summary of All Pair Shortest Paths

Hop Count	**No. of Paths in BUS 118**	**No. of Paths in BUS 300**
1	179	409
2	397	890
3	626	1332
4	795	1844
5	878	2367
6	899	2800
7	839	3286
8	745	3704
9	573	3976
10	398	4102
11	268	4096
12	186	3783
13	96	3433
14	24	2793
15	0	2080
16	0	1508
17	0	1006
18	0	652
19	0	403
20	0	223
21	0	105
22	0	41
23	0	15
24	0	2

the number of paths for LCM whose tolerances exceed MPT's maximum. It is clear that the shortest paths are much more vulnerable in the LCM strategy.

According to the results discussed here, it can be said that the MPT strategy is a promising solution to distributed monitoring in SCADA networks. It enhances distributiveness at the cost of coverage and offers an improved QoSS for the lower number of trust nodes. On the other hand, the performance of MPT is approached by LCM when the number of trust nodes exceeds 45%. The MPT scheme can be used as an expansion planning tool for a segmented network, where the monitoring is being extended to intra-segment traffic.

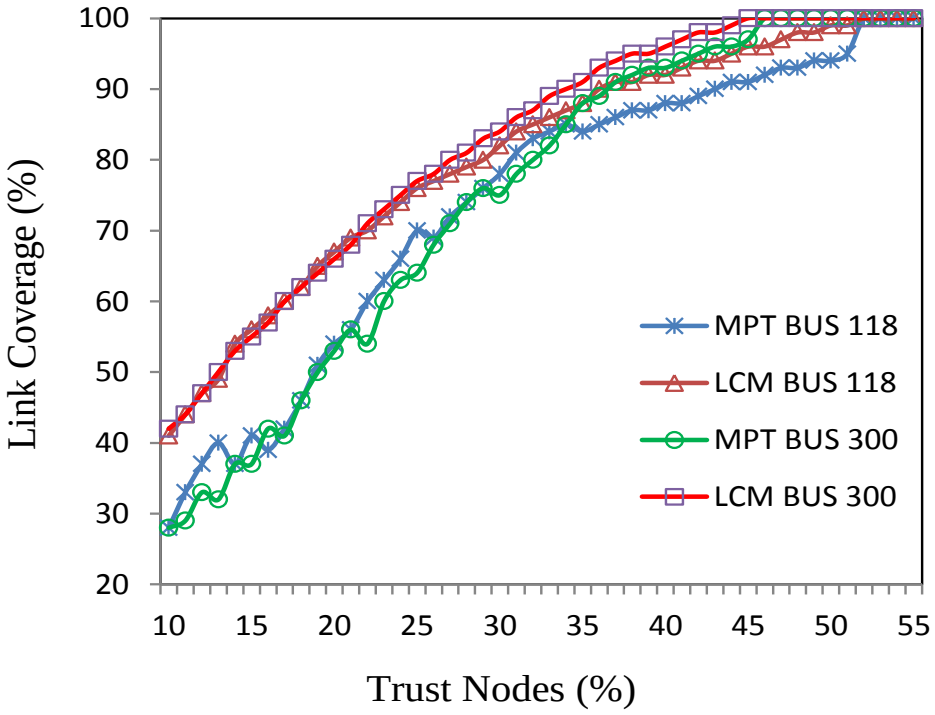


Figure 6.6: Comparative coverage for a 0% reservation.

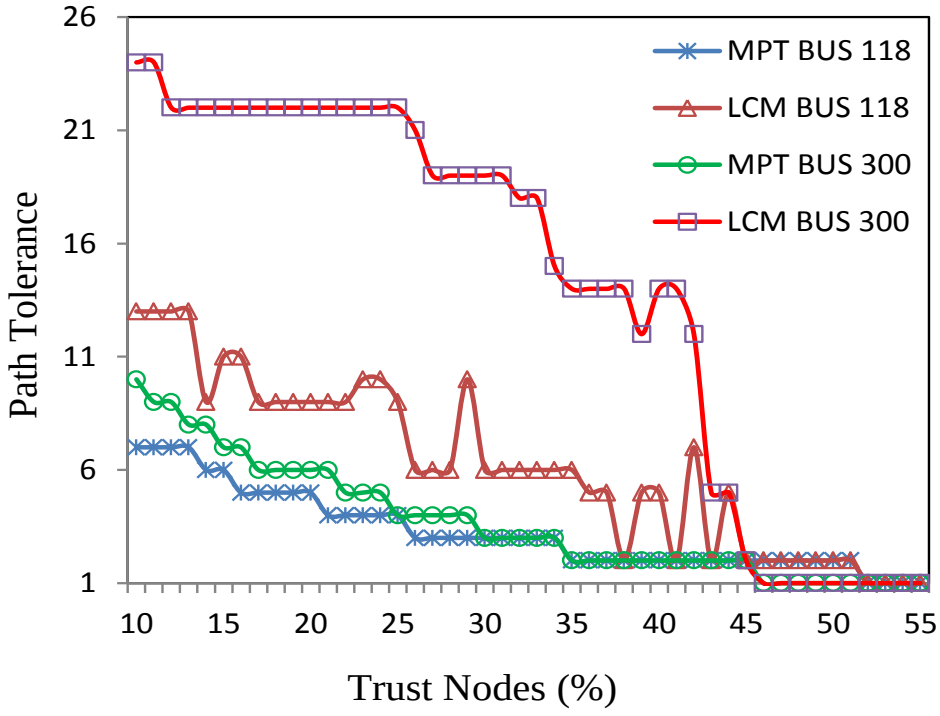
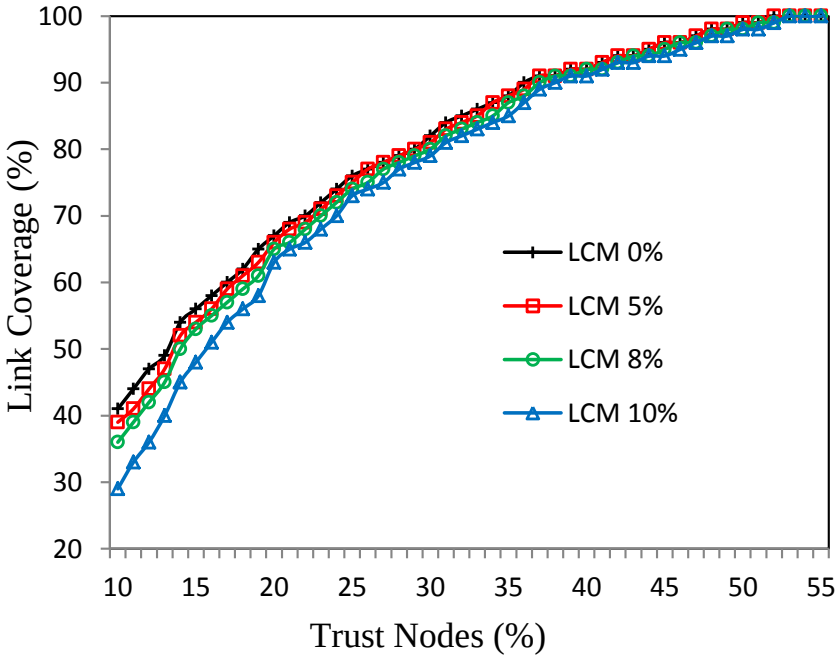
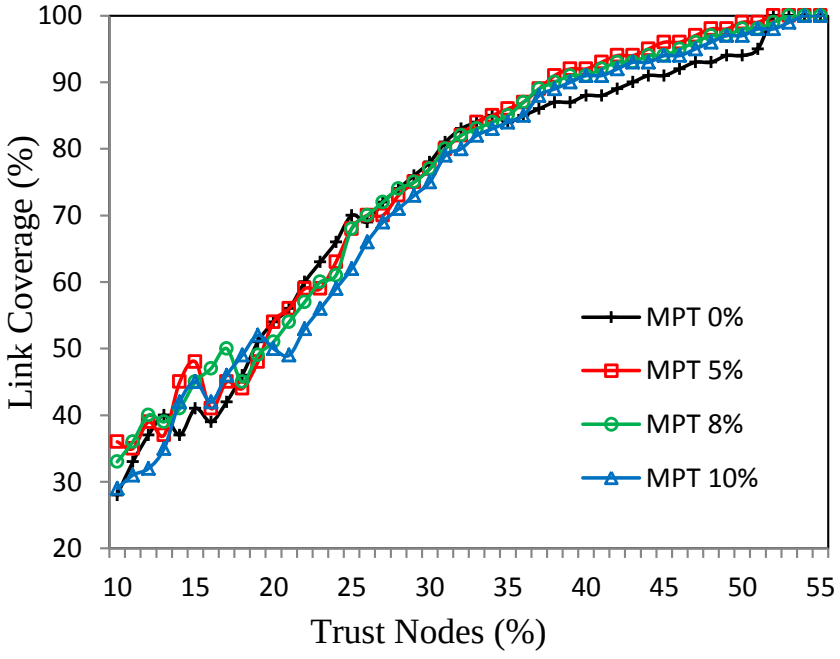


Figure 6.7: Comparative path tolerance for a 0% reservation.

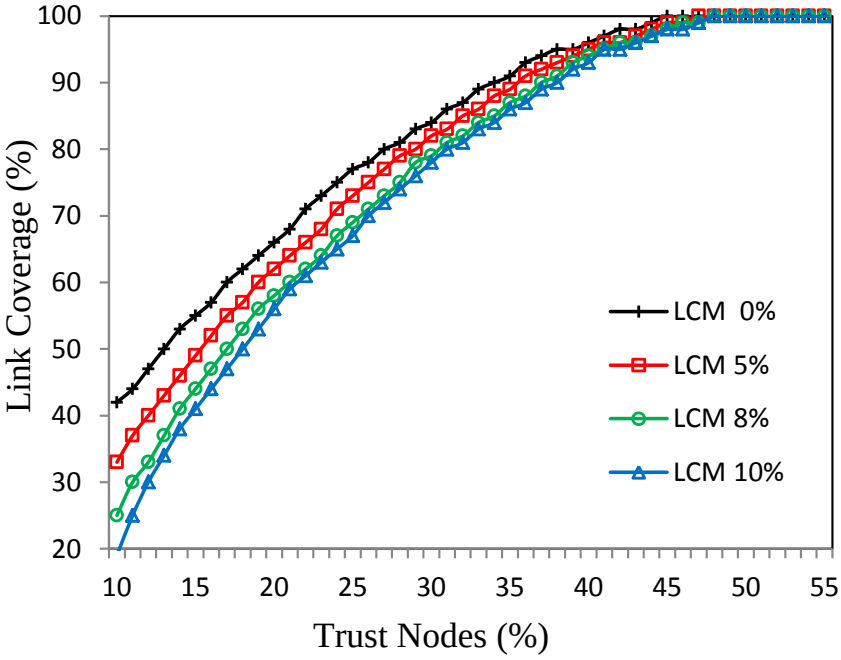


(a) LCM

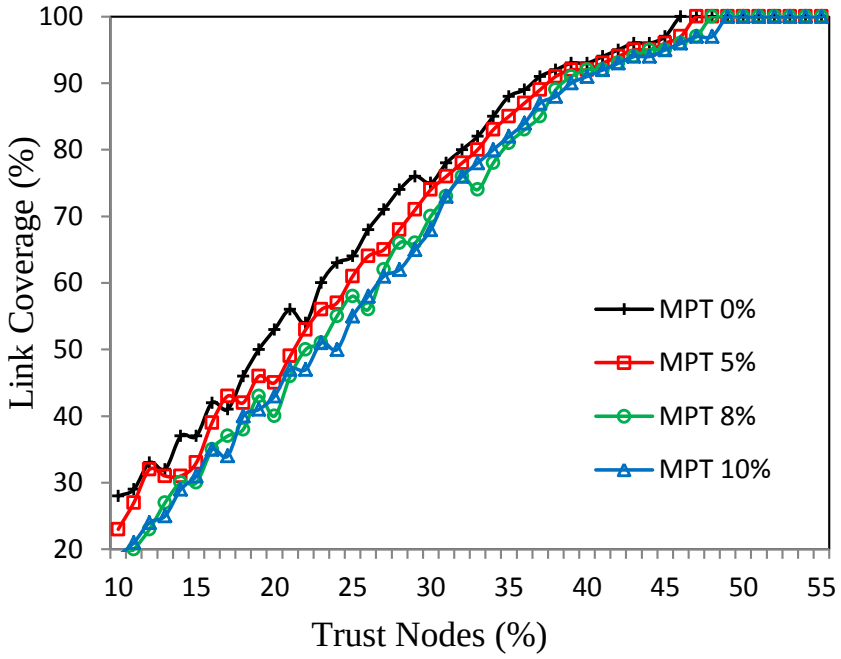


(b) MPT

Figure 6.8: Impact of reservations on the link coverage for BUS 118.

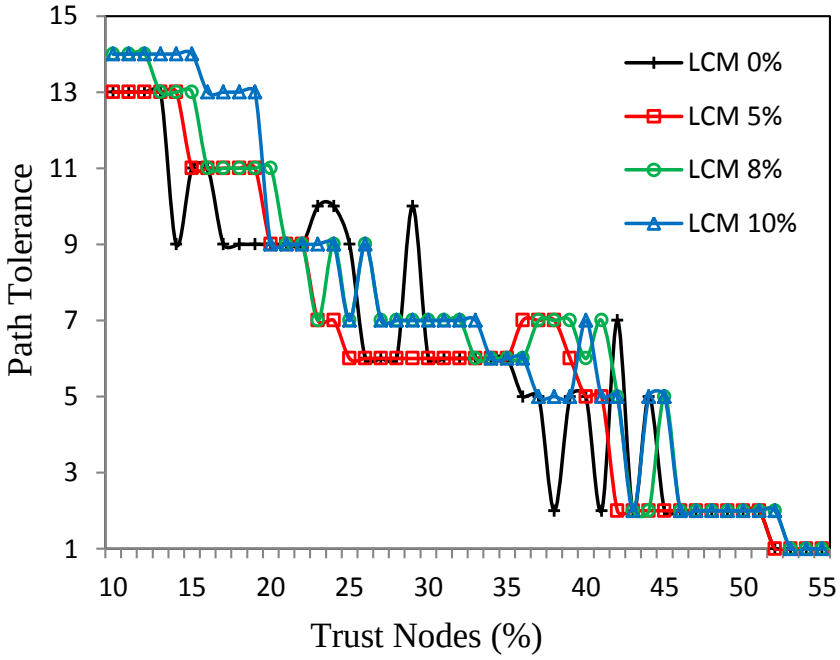


(a) LCM

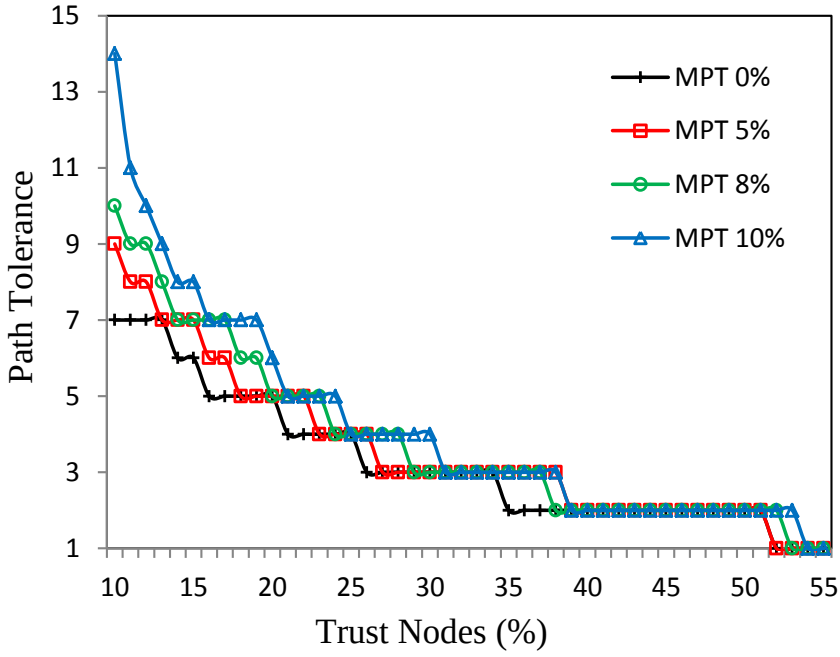


(b) MPT

Figure 6.9: Impact of reservations on the link coverage for BUS 300.

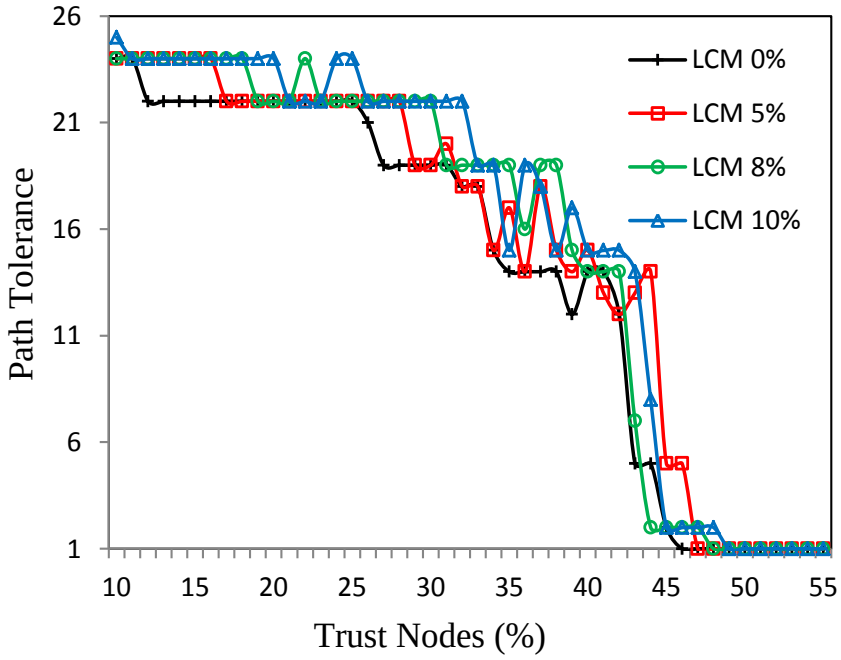


(a) LCM

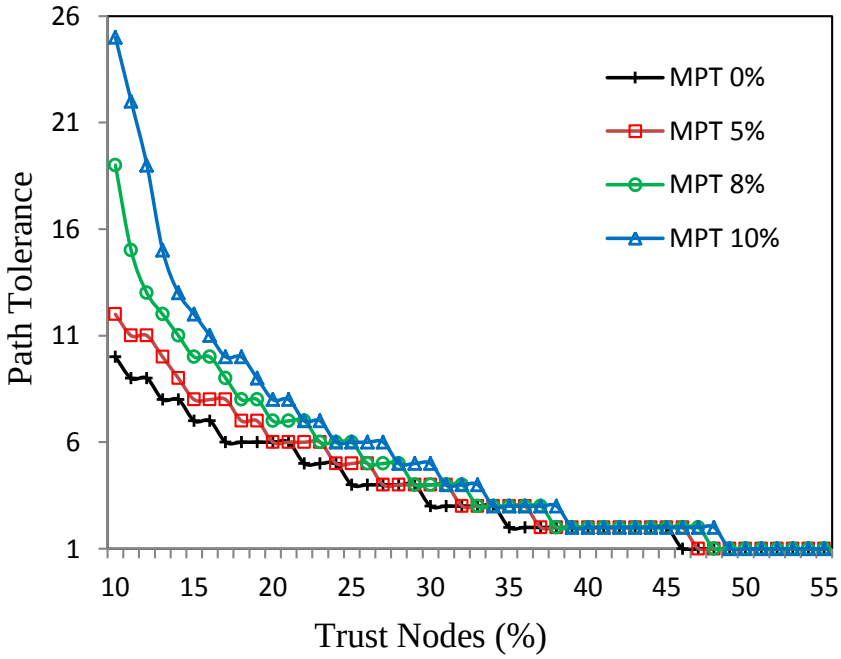


(b) MPT

Figure 6.10: Impact of reservations on the path tolerance for BUS 118.

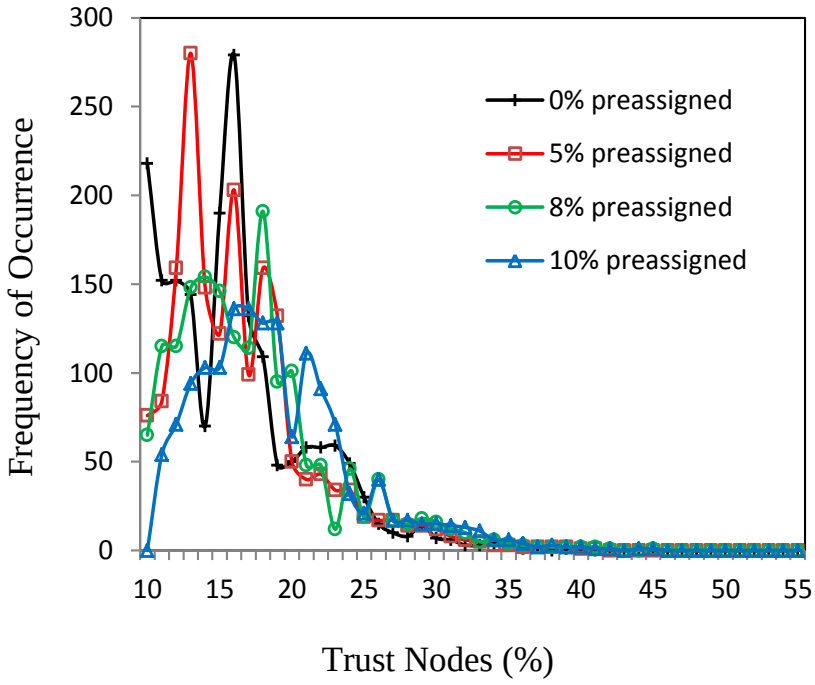


(a) LCM

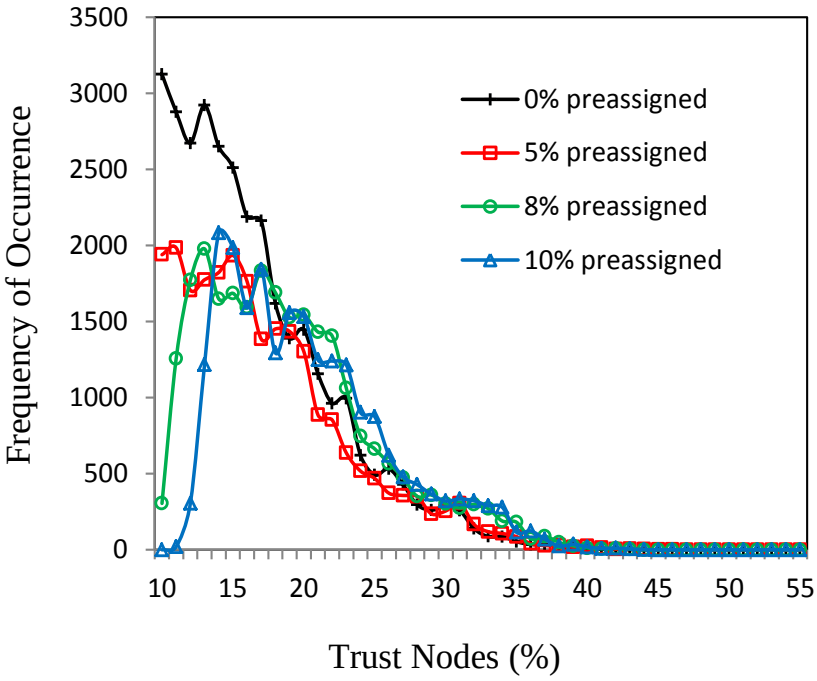


(b) MPT

Figure 6.11: Impact of reservations on the path tolerance for BUS 300.



(a) BUS 118



(b) BUS 300

Figure 6.12: No. of paths for LCM that exceed MPT's γ_M.

6.5 Conclusion

In this chapter, the trust system placement problem has been studied for enhancing distributed monitoring in smart grid SCADA networks. The study developed solutions for the active/router mode of operation, where trust systems are placed as inline hardware. It proposed two metrics for trust system placement schemes: (i) link coverage and (ii) path tolerance. It also introduced a quadratic objective function that provides the exact link coverage information for a trust system deployment scenario. Two different placement schemes were developed to emphasize each of the metrics separately. Such schemes solve resource-constrained optimization problems and were evaluated as well as compared for the IEEE test system topologies. The path tolerance is a promising metric for the enhancement of distributed monitoring in resource-limited scenarios.

Chapter 7

Security Service Placement for Advanced Metering Infrastructures

7.1 Introduction

In smart cities, an AMI network is responsible for the metering and billing of main household utilities such as electricity, water, and gas. The AMI network is devised also to deliver market information to smart grid participants. Its operations are mainly dependent on the deployment of smart meters. It is expected that the smart meters would support two-way communications to facilitate load control functionalities. Smart meters measure, store, and transmit energy consumption data to the AMI network. They are located at customer premises where such areas are typically considered low-trust environments. An estimation shows that the number of deployed smart meters will exceed one billion globally by 2022 [oci15]. The integration of such physically unprotected devices will add millions of highly vulnerable access points to an AMI network. Such access points can be exploited to conduct various malicious activities in both the upstream and downstream directions. AMI concentrators are the gateway to a wide area mesh network that includes the smart grid control center. They locally collect the upstream data that are initially transmitted by smart meters. The data are then delivered to the control center through the mesh network. Thereafter the collected data are used to perform a number of vital tasks for the

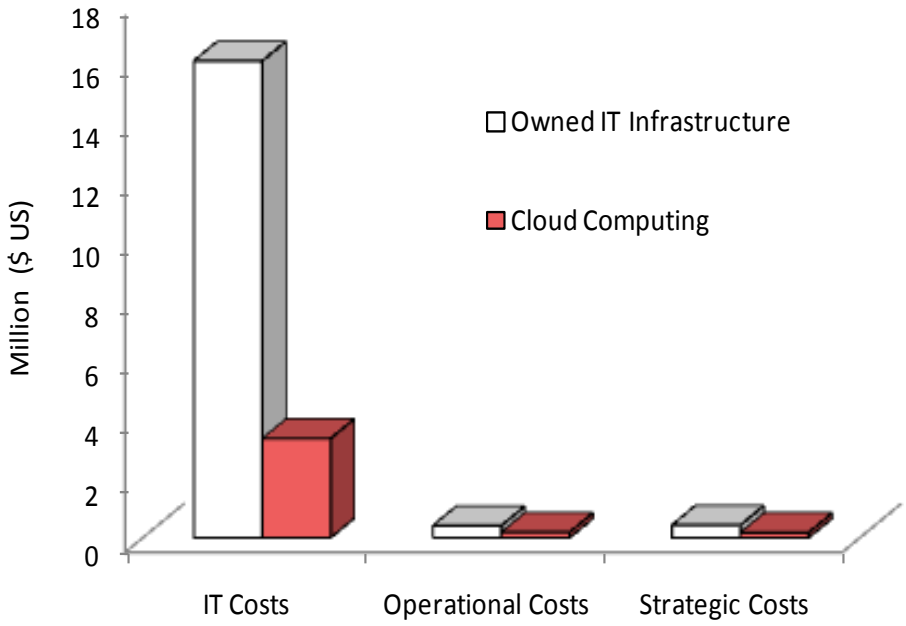


Figure 7.1: Cost comparison in smart grid management.

grid, such as billing, controlling, forecasting, and planning. Each of these tasks is directly or indirectly involved with revenue and the continuity of the service. Thus, corrupted data from compromised smart meters can have a big impact on the grid. The upstream traffic from smart meters are required to be continuously monitored at each AMI concentrator.

It is anticipated that future grids will be powered by the advancement of cloud computing [iee13]. There will be a strong interdependency between smart grids and cloud computing [Gre13, Yig14, Ber15]. Cloud data centers are going to be one of the highest energy consumption spots. On the other hand, smart grid applications are moving towards cloudification. The key areas where cloud computing can greatly contribute include: energy management, information management, and security. Monitoring, data storage, and data analytics are three basic ingredients of cloud services that can be utilized in smart grid management.

Cloud-based services are an emerging trend in IT. Their key offerings are better management, cost-effectiveness, and faster response. Such services can also be adopted to support smart grid operations. In particular, managed cyber security can be an attrac-

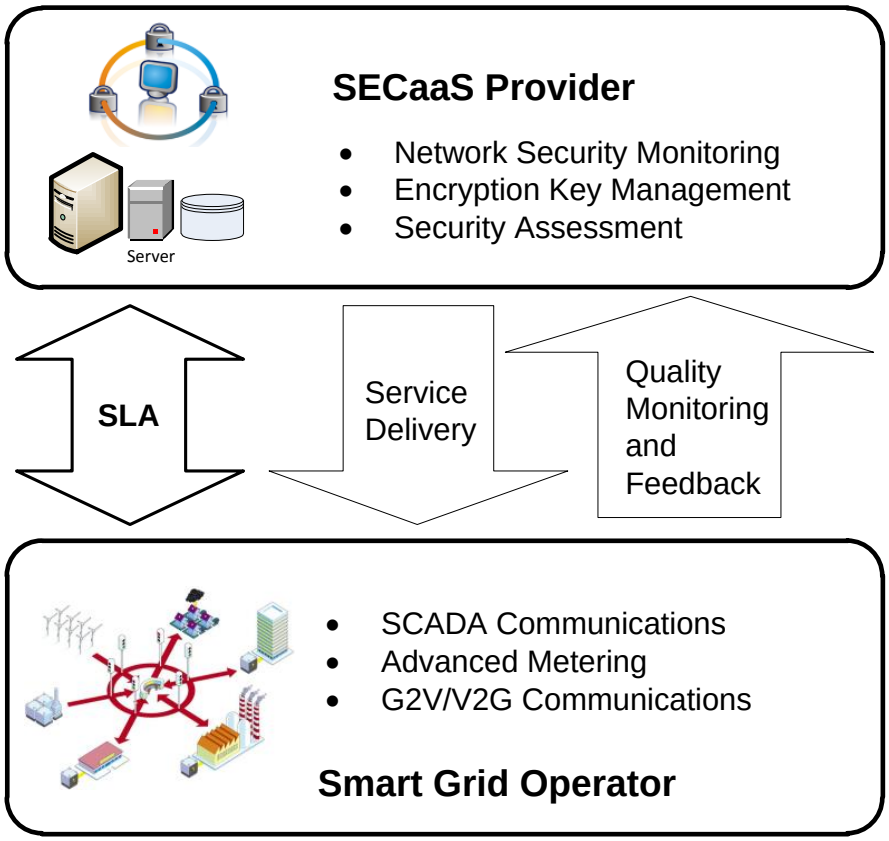


Figure 7.2: A SECaaS architecture for smart grid operations.

tive option for smart grid operators. The SECaaS model is devised to provide managed security to clients from both cloud and non-cloud environments [csa11]. It can also be adopted to serve enterprise customers such as smart grid operators [Has13a, Has15]. The extension of SECaaS to non-cloud environments avails the potential of cloud computing to many novel applications. Cloud-based smart grid management solutions can offer huge savings over conventional IT infrastructure. Figure 7.1 provides a comparison of estimated costs in smart grid management. It shows three years TCA for a facility with thirty Itanium 16-core processors [int15a]. These costs are estimated using IBM's smart computing workload simulator [ibm15]. Figure 7.2 illustrates a SECaaS architecture for smart grid operations that outlines the two major participants: a SECaaS provider and a smart grid operator. Their interactions are governed by an SLA. The SECaaS provider offers three major types of services: (i) network security monitoring, (ii) encryption key management, and (iii) security assessment. The smart grid operator owns three communication net-

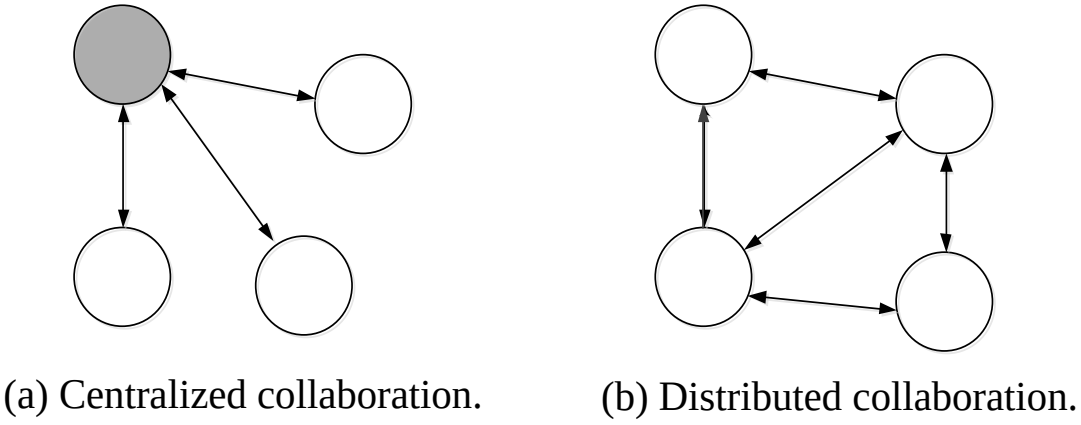
works: (i) SCADA, (ii) AMI, and (iii) grid-to-vehicle/vehicle-to-grid (G2V/V2G). The quality of delivered security services are monitored with periodic feedback is sent to the provider. The focus of this chapter is the security monitoring service for AMIs. New elements or smart meters are continuously integrated with an AMI. Such customer-owned devices are highly vulnerable to cyber-attacks and are geographically dispersed regarding the service area of the grid. The smart grid operator requires a pervasive solution for the network intrusion management. As cloud services are delivered through a separate access network, the security monitoring offers an out-of-band intrusion management. In addition, a cloud-centric collaborative monitoring can be achieved without leveraging the AMI network. Collaborative detection mechanisms are considered to be more efficient for large-scale distributed networks [Zhu12, int15b, Pat17]. On the other hand, it can be a worthwhile business opportunity for CSPs.

In this chapter, a cloud-centric collaborative monitoring architecture is proposed for AMI networks. Subsequently, a service placement model is developed that aims to minimize the monitoring latency for the architecture. The model considers access latency and internal latency for geographically distributed private clouds. The access latency pertains to cloud access networks and occurs at the data gathering stage. The internal latency pertains to intra- and inter- data center delays in the cloud domain and occurs at the collaboration stage.

7.2 Collaborative Security Basics

To describe the proposed system architecture, it is worthwhile to discuss the concept of a collaborative security first. The major building blocks of a collaborative security system include: monitoring entity, decision entity, and collaboration entity [Men15]. A monitoring entity is responsible for an initial screening of traffic with a set of pre-defined criteria is reviewed throughout the process. The acquired information is then transferred to a computationally more powerful entity for confirmation. The powerful entity is known as the decision entity that applies advanced analytical tools to detect anomalies. The collabo-

ration entity assists the decision entity to exchange information between peering systems. It shares locally computed results and collects results from peering systems. It is obvious that collaborative security systems require additional communication efforts. They have to be associated with a collaboration network. Two basic types of collaboration networks are considered in the outlined system architecture: (i) centralized and (ii) distributed. For centralized collaboration, every node directly collaborates with a central server. For distributed collaboration, there is no central server thus every node participate with the same role. Each node collaborates with a set of peering nodes depending on its neighborhood. Figure 7.3 illustrates two types of collaboration networks.



(a) Centralized collaboration. (b) Distributed collaboration.

Figure 7.3: Collaboration networks.

7.3 System Architecture

A system is considered that delivers a collaborative security monitoring service to an AMI WAN. The system is devised considering the aforementioned SECaaS architecture for smart grid operations. A SECaaS provider delivers the service to a smart grid operator through access networks. The SECaaS provider owns and operates a geographically distributed private cloud that includes a set of data centers and a correspondent backbone network. The smart grid operator owns a geographically distributed AMI network that is comprised of a layered architecture. Its topmost layer consists a WAN of AMI concentrators and a control center. Each AMI concentrator is a gateway to the WAN for upstream traffic.

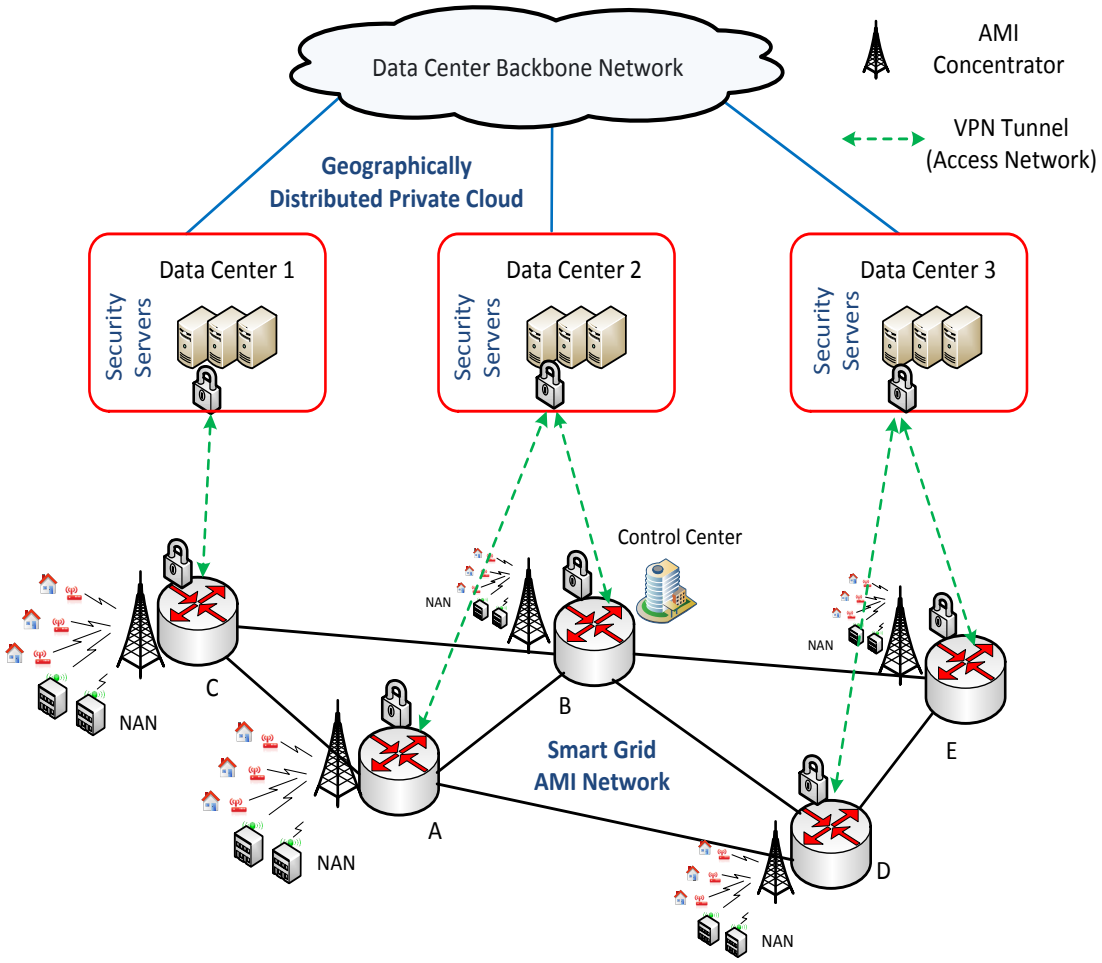


Figure 7.4: Cloud security service placement for an AMI network.

It forms a NAN with nearby metering devices to collect upstream traffic. Such metering devices are located at homes and buildings. The security service for an AMI concentrator is placed in a particular data center. An illustration of the service placement is provided in Figure 7.4 that outlines an AMI WAN consisting of five AMI concentrators. Three data centers of a private cloud host servers for the AMI WAN.

Figure 7.5 shows a functional view of our system architecture where the two major participants include: (i) AMI manager and (ii) cloud resource manager. The AMI manager is responsible for AMI concentrators while the cloud resource manager is responsible for data centers. The cloud resource manager solves the service placement problem as it has the complete knowledge about resources in data centers, backbone network, and access

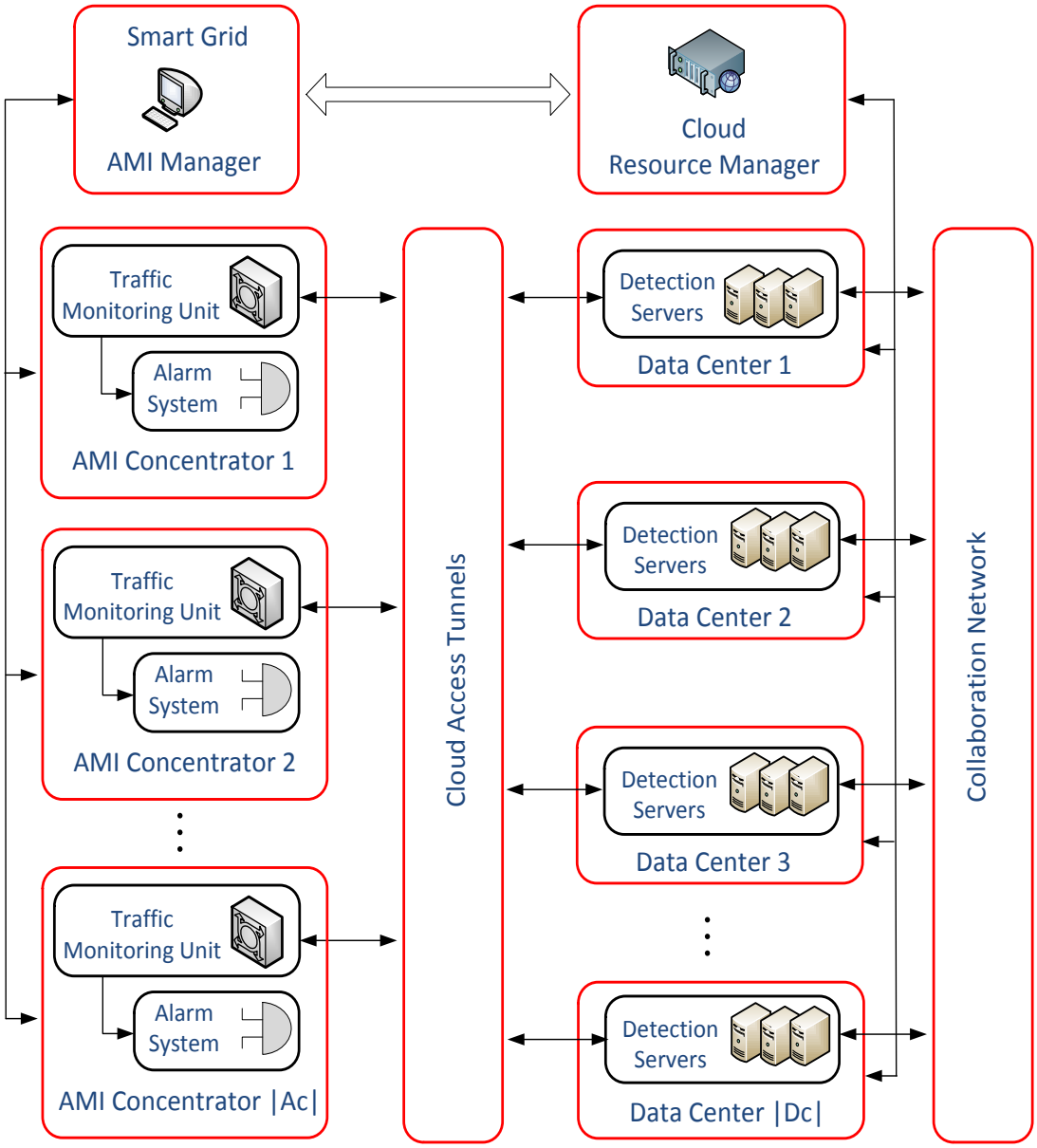


Figure 7.5: System architecture for providing cloud-centric collaborative security to an AMI network.

networks. Each AMI concentrator is equipped with a traffic monitoring unit (TMU). TMUs interface between the AMI network and the cloud. The role of a TMU is to perform an initial assessment of upstream traffic. It requires a pre-defined set of criteria to detect possible forms of anomalies. The assessment results are then transferred to a powerful detection server. In the cloud, there exists a dedicated detection server for every TMU.

Each TMU has a maximum traffic handling capacity as it performs admission control for the detection server. Detection servers are run on VMs at the data centers. The communication between a TMU and a detection server takes place in a secure cloud access tunnel. The detection server deploys advanced analytical tools and collaborates with other detection servers. A collaboration network is formed in the cloud domain and thus a cloud-centric collaboration is achieved. It consists of logical links from both intra-data center and data center backbone networks. An intra-data center logical link is defined when it is a collaboration between two servers located in the same data center. A backbone logical link is defined when it is a collaboration between two servers located in two different data centers. The final result is sent to the TMU. Depending on the result, alarms and alert systems are activated at the AMI concentrator. The AMI manager negotiates with the cloud resource manager where it provides quality monitoring report, performance feedback, as well as network updates. The information on quality and performance are used to determine SLA violations as well as requirements for service upgrades. Network updates include any major change in AMI WAN such as the addition of new nodes and the removal of existing nodes.

7.4 Problem description

Latency is one of the major concerns in security monitoring. It is related to the time difference between a detected event and its appropriate response. When a malicious activity is detected, an action should be taken as soon as possible. The primary focus here is the minimization of latency in cloud security services. The geographical dispersion between service hosts and their clients contributes towards excessive latency in cloud-based services. The availability of resources can also affect the latency in service deliveries. Here it is presumed that the SECaaS provider has sufficient resources to serve the AMI network. This means that data centers are collectively capable of handling the total workload from the AMI network. However, individual data centers are resource-constrained. At some instance, the capacity of a data center refers to the number of TMU it can serve. In the outlined service placement problem, there are two sources of latencies: access link latency

and internal link latency. The former belongs to the virtual private network (VPN) access tunnels between TMUs and detection servers. The latter belongs to the collaboration network. It can be either intra- or inter- data center link latency depending on the hosts of collaborating pairs. The overall latency belongs to two different networks. The outlined service placement problem is simply stated as follows. Place the security services in such a way that the overall latency is minimized.

7.5 Service Placement Scheme

The outlined service placement scheme solves an optimization problem that minimizes the overall latency. At first, an optimization model is developed for the overall latency. The original model is computationally expensive. Consequently, a simplification technique is applied to reduce the complexity without compromising the exactness of solutions. The preliminary assumptions for the service placement are as follows. There exists an SLA between the smart grid operator and the SECaaS provider. TMUs are readily installed to all AMI concentrators and the service capacity of a data center is calculated based on the number of detection servers it can host. As the maximum capacity of a TMU is a known parameter, CPU and RAM requirements can be estimated for a detection server. Detection servers are assumed to be of identical capacity. The access and backbone networks are available with sufficient bandwidths for support of the security service.

An AMI WAN is considered along with a geographically distributed private cloud. The set of AMI concentrators in the WAN is Ac while the set of data centers in the private cloud is Dc. Let t_{mn} be the access latency between the AMI concentrator $m \in Ac$, and the data center $n \in Dc$; $t_{nn'}$, be the latency between data centers n and n'. At any instance of time, r_n is the capacity of the data center n. The collaboration indicator between two detection servers corresponding to m and m' is $\lambda_{mm'}$. It depends on the collaboration network and is given by,

$$\lambda_{mm'} = \begin{cases} 1, & \text{if there is a direct collaboration between } m \text{ and } m'; \\ 0, & \text{otherwise;} \end{cases} \qquad \forall m, m' \in A_c, m \neq m'. \tag{7.1}$$

The collaboration level between two detection servers belongs to AMI concentrators m and m' is $\mu_{mm'}$. This depends on how frequent they need to communicate each other with respect to the data gathering frequency. It is given by,

$$\mu_{mm'} = \min(\widetilde{\mu}_{mm'}, \lambda_{mm'}), \quad \text{where,} \quad \widetilde{\mu}_{mm'} \in (0, 1]\,; \quad \forall m, m' \in A_c, m \neq m'. \tag{7.2}$$

The decision variable for the optimization model is $\Omega = (\omega_{mn})_{(\sum_{m=1}^{|A_c|} \sum_{n=1}^{|D_c|} \mathbf{1}) \times 1}$. It is a server incidence vector such that,

$$\omega_{mn} = \begin{cases} 1, & \text{if the server for } m \text{ is hosted by } n; \\ 0, & \text{otherwise.} \end{cases} \tag{7.3}$$

The service placement problem is only feasible when the following condition is satisfied by the private cloud,

$$\sum_{m=1}^{|A_c|} \sum_{n=1}^{|D_c|} \omega_{mn} \leq \sum_{n=1}^{|D_c|} r_n. \tag{7.4}$$

As the model includes the collaboration latency, it can be termed as the collaboration-aware (CoA) service placement model. It is formulated as follows.

$$\text{(CoA)} \quad \min_{\Omega} \sum_{m=1}^{|A_c|} \sum_{n=1}^{|D_c|} t_{mn} \omega_{mn} + \frac{1}{2} \sum_{m=1}^{|A_c|} \sum_{\substack{m'=1 \\ m' \neq m}}^{|A_c|} \sum_{n=1}^{|D_c|} \sum_{n'=1}^{|D_c|} \mu_{mm'} \lambda_{mm'} t_{nn'} \omega_{mn} \omega_{m'n'}, \tag{7.5}$$

$$\sum_{n=1}^{|D_c|} \omega_{mn} = 1, \qquad \forall m \in A_c, \tag{7.6}$$

$$\sum_{m=1}^{|A_c|} \omega_{mn} \leq r_n, \qquad \forall n \in D_c, \tag{7.7}$$

$$\omega_{mn} \in \{0,1\}, \qquad \forall m \in A_c, \forall n \in D_c. \tag{7.8}$$

The objective in (7.5) is called cycle latency. It considers full-duplex communications for the collaboration network. It combines latencies from different networks. The linear term corresponds to access link latencies and the quadratic term corresponds to cloud's internal latency. It occurs due to the collaboration among servers. Constraint (7.6) ensures that only one detection server will be placed for each AMI concentrator. Constraint (7.7) ensures that the capacity of every data center will be respected. The allocated load to a data center will not exceed its capacity.

The current formulation is identified as a QAP which is computationally expensive. To reduce the complexity, the linearization technique described in [She07] is used. The linearized objective function is given by,

$$\min_{\Omega,\Xi} \sum_{m=1}^{|A_c|}\sum_{n=1}^{|D_c|} t_{mn}\omega_{mn} + \frac{1}{2}\sum_{m=1}^{|A_c|}\sum_{\substack{m'=1 \\ m'\neq m}}^{|A_c|}\sum_{n=1}^{|D_c|}\sum_{n'=1}^{|D_c|} \mu_{mm'}\lambda_{mm'}t_{nn'}\xi_{mnm'n'}. \tag{7.9}$$

The following set of auxiliary variables are incurred due to the linearization,

$$\Xi = (\xi_{mnm'n'})_{(\sum_{m=1}^{|A_c|}\sum_{m'=1,m'\neq m}^{|A_c|}\sum_{n=1}^{|D_c|}\sum_{n'=1}^{|D_c|}\mathbf{1})\times 1} \quad \text{where,} \quad \xi_{mnm'n'} \in \{0,1\}. \tag{7.10}$$

The relationship between the decision and auxiliary variables is defined by the following additional constraints,

$$\xi_{mnm'n'} \geq \omega_{mn} + \omega_{m'n'} - 1, \qquad \forall m, m' \in A_c, m' \neq m; \quad \forall n, n' \in D_c, \tag{7.11}$$

$$\xi_{mnm'n'} \geq 0, \qquad \forall m, m' \in A_c, m' \neq m; \quad \forall n, n' \in D_c. \tag{7.12}$$

The average computational complexity of solving the linearized CoA is $\sim O((|A_c||D_c| + 0.5|A_c|(|A_c|-1)|D_c||D_c|)\log(|A_c||D_c| + 0.25|A_c|(|A_c|-1)|D_c||D_c|))$ [Chv83].

7.6 Numerical Evaluation and Discussion

An evaluation is completed for the proposed service placement scheme considering two different settings for the data center backbone network. The first setting is a partial-mesh topology and the second one is an augmented Abilene topology. In both settings, a 50 unit $\times$ 50 unit geographic area is considered where data centers and AMI concentrators are distributed. Since there is no well-accepted topology for AMI WANs, one is generated using the IEEE BUS 57 test system topology and GENSEN [iee15], [Cam07]. The network size and connection density are set based on the BUS 57 topology. Positions of AMI concentrators are generated using GENSEN. The partial-mesh topology consists of five data centers and seven backbone links. The augmented Abilene topology consists of ten data centers and thirteen backbone links. It is generated considering the Abilene topology, which is commonly used in the evaluation of data center backbones [abi12, Gho13]. The experimental topologies are depicted in Figures 7.6 and 7.7. Details of the experimental topologies are given in Appendix C and Appendix D. One of the major challenges in the evaluation setup was the generation of delay parameters for access links and backbone paths. As randomly generated latencies do not preserve the geographic relevance of the topologies, we use the Euclidean distance method [Hua15]. Preservation of geographical relevance is an important part of this current study. In the recent literature regarding cloud service placement, the Euclidean distance method is frequently used for evaluation [Yan16, Su16]. Latencies are assumed to be directly proportional to the Euclidean distances. For the backbone network, the Floyd-Warshall all pair shortest path algorithm is used to compute the inter-data center latencies [Cor09]. As the main feature of our pro-

posed scheme is the collaboration-awareness, the aim is to compare its performance with collaboration-unaware ones. Previous works such as [Bed12] and [Zha13a] only consider access latencies since no collaboration among servers existed. Therefore, such as arrangement is set as the common criterion for collaboration-unaware schemes. In other words, collaboration-unaware schemes only consider access latencies. The general solution to such types of service placement problems is bipartite matching (BM). A BM is a linear formulation that locates an optimal host for each server. The developed CoA scheme is compared with the BM. The normalized latency (t_{NL}) is defined as a metric for the proposed scheme. It is given by,

$$t_{NL} = \frac{\text{Cycle latency for the solution obtained in CoA}}{\text{Cycle latency for the solution obtained in BM}}. \tag{7.13}$$

When t_{NL} takes a value less than unity, it is an indication that CoA is outperforming BM.

Evaluation scenarios include two major investigations: (i) resource-unconstrained cases and (ii) resource-constrained cases. To create a resource-unconstrained case, the following condition is set,

$$r_n \geq |A_c|, \qquad \forall n \in D_c. \tag{7.14}$$

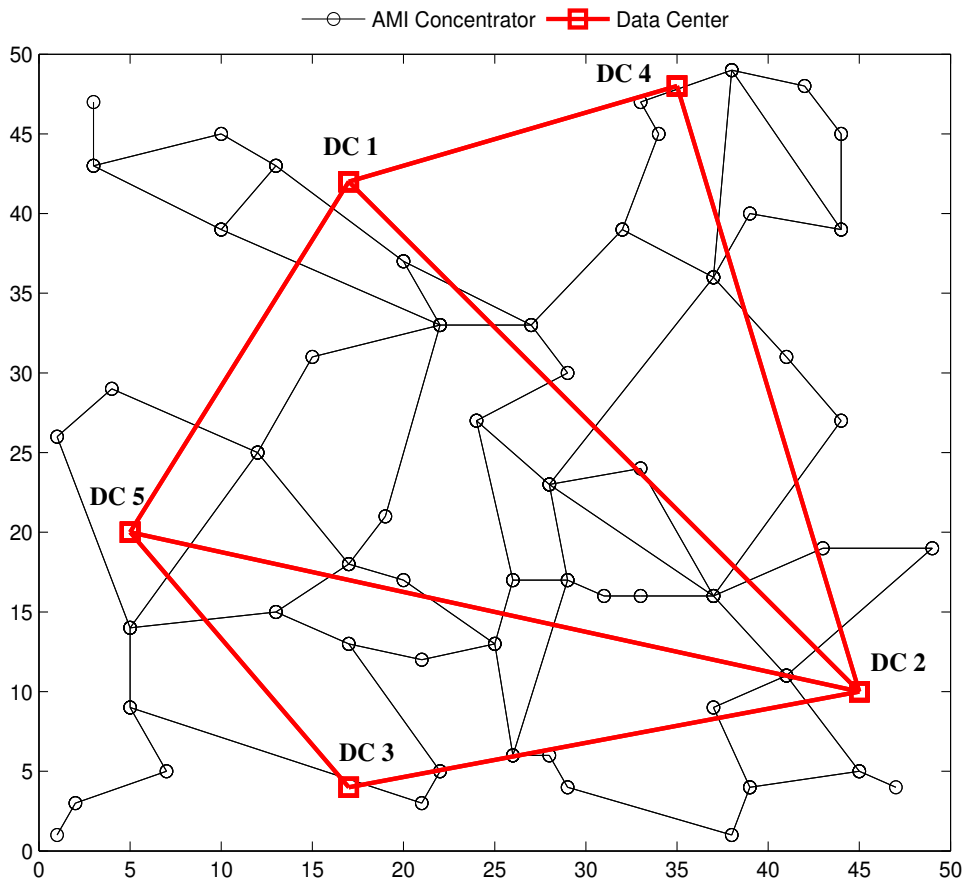


Figure 7.6: Experimental topologies from Part I: (i) an AMI wide area mesh network of 57 nodes and (ii) a partial-mesh backbone network of 5 data centers.

The investigation of a resource-unconstrained case is useful in the latency-aware resource planning for data centers as it determines the ideal capacity for each data center. Therefore, the workload distribution among data centers is also observed in addition to the normalized latency.

The investigation of the resource-constrained cases provides an idea of the performance in the dynamic service placement. In a data center, the availability of resources varies due to a number of factors such as scheduled maintenance, failures, and catastrophic reasons. To create the resource-constrained cases, 1000, different combinations of data center capacities are generated for each backbone topology. For the partial-mesh backbone, data center capacities are uniformly distributed between 10 to 25. For the augmented Abilene, data

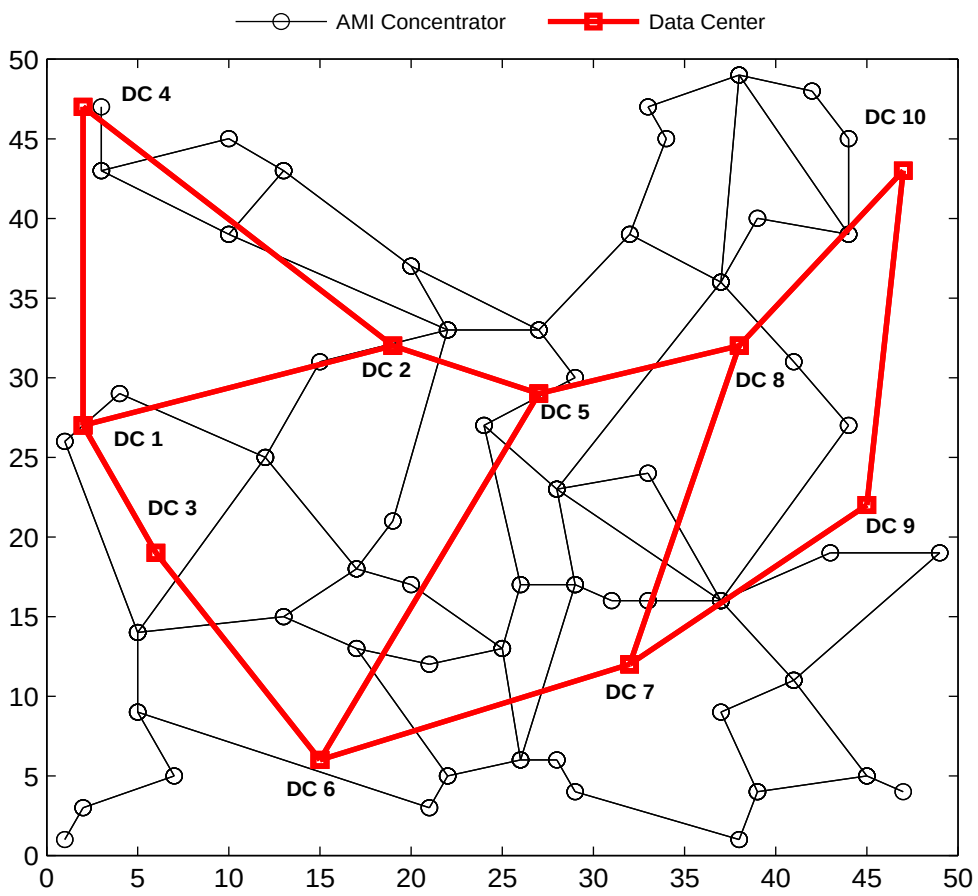


Figure 7.7: Experimental topologies from Part II: (i) an AMI wide area mesh network of 57 nodes and (ii) an augmented Abilene backbone network of 10 data centers.

center capacities are uniformly distributed between 4 to 12. These ranges are chosen based on the network sizes and the total workload. In a resource-constrained case, the capacity of a data center is varied between above and below the average load. However, constraint (7.4) is satisfied by every single combination. For the constrained cases, we only observed the normalized latency since the workload distribution varies with capacities.

In each case of the investigation, results are studied for both types of collaboration, distributed and centralized. For the distributed case, detection servers that belong to single hop neighbors collaborate with each other. It means that collaboration network is virtually the same as the AMI WAN. For the centralized collaboration, all detection servers only collaborate with the control center's detection server. The control center is supposed to be

the most important node in the AMI WAN. It is selected based on the eigenvector centrality [New10]. For fairness, the collaboration level is kept the same between all collaborating parties. Thus, $\widetilde{\mu}_{mm'}$ is set to a fixed value μ for a given experiment. We use five different collaboration levels for performance comparisons, $\mu = 0.1$, 0.25, 0.4, 0.5, and 1.

All experiments are implemented using an integrated MATLAB-CPLEX solver on a desktop machine with Intel Core i3 3.30 GHz and 8 GB RAM. A total of 28 hours were required for the experiments on the partial-mesh backbone. Likewise, a total of ∼1250 hours (∼52 days) for the experiments on the augmented Abilene backbone.

7.6.1 Resource-Unconstrained Cases

Figures 7.8, 7.9, 7.10, and 7.11 show normalized latency for different experimental settings under resource-unconstrained cases. The key observations are as follows. The CoA scheme always outperforms BM. The distributed collaboration exhibits much lower latencies than the centralized one. The collaboration level makes big differences in latencies. The proposed CoA scheme performs much better as μ increased. The augmented Abilene backbone exhibits lower latencies since it is more distributed than the partial-mesh.

Figures 7.12, 7.13, 7.14, and 7.15 show the workload distribution among data centers under resource-unconstrained cases. The bars on their x-axis follow the same sequence as the legends. The bars go from left to right, while the legends go from top to bottom. The following facts are observed. The same amount of workload but a different combination of hosted server can make a big difference in latencies. For the centralized collaboration the workload distribution for CoA has small difference with that of BM.

7.6.2 Resource-Constrained Cases

Figures 7.16, 7.17, 7.18, and 7.19 show normalized latencies for different experimental settings under resource-constrained cases. The same trend as the resource-unconstrained cases have been observed for the collaboration types and backbone networks. The collaboration level dominates the latency characteristics. The summary of results is provided

in Tables 7.1 and 7.2. Their confidence intervals are summarized in Tables 7.3, 7.4, 7.5, and 7.6. These intervals correspond to a two-sided 95% confidence level [Bra87] (see Appendix E).

Table 7.1: Normalized Latency: Partial-Mesh.

	Distributed				**Centralized**			
Coll. Level (μ)	max	min	mean	Std. Dev.	max	min	mean	Std. Dev.
0.1	0.8746	0.8455	0.8668	0.0041	0.9816	0.9756	0.9778	0.0008
0.25	0.7261	0.6963	0.7106	0.0034	0.9486	0.9414	0.9466	0.0018
0.4	0.6119	0.58	0.5939	0.0032	0.9211	0.9082	0.9177	0.0036
0.5	0.5513	0.5177	0.5323	0.0041	0.904	0.8854	0.8981	0.005
1	0.3597	0.3406	0.3484	0.0022	0.8091	0.7733	0.7943	0.0087

Table 7.2: Normalized Latency: Augmented Abilene.

	Distributed				**Centralized**			
Coll. Level (μ)	max	min	mean	Std. Dev.	max	min	mean	Std. Dev.
0.1	0.8114	0.7701	0.7843	0.0064	0.9803	0.9548	0.9671	0.0039
0.25	0.6359	0.5882	0.6067	0.0085	0.9378	0.89	0.9164	0.0089
0.4	0.5328	0.4735	0.4965	0.0105	0.8966	0.8335	0.8683	0.0131
0.5	0.4838	0.4188	0.4426	0.0115	0.8719	0.7996	0.8382	0.0154
1	0.3196	0.2659	0.2847	0.0094	0.7665	0.6645	0.7148	0.0225

Table 7.3: Confidence Interval: Dist. Coll. Partial-Mesh.

Coll. Level (μ)	mean	Std. Dev.	Upper Bound	Lower Bound	Interval
0.1	0.8668	0.0041	0.8671	0.8665	0.0005
0.25	0.7106	0.0034	0.7108	0.7104	0.0004
0.4	0.5939	0.0032	0.5941	0.5937	0.0004
0.5	0.5323	0.0041	0.5326	0.532	0.0005
1	0.3484	0.0022	0.3485	0.3483	0.0002

Table 7.4: Confidence Interval: Cent. Coll. Partial-Mesh.

Coll. Level (μ)	mean	Std. Dev.	Upper Bound	Lower Bound	Interval
0.1	0.9778	0.0008	0.9778	0.9778	0.0001
0.25	0.9466	0.0018	0.9467	0.9465	0.0002
0.4	0.9177	0.0036	0.9179	0.9175	0.0004
0.5	0.8981	0.005	0.8984	0.8978	0.0006
1	0.7943	0.0087	0.7948	0.7938	0.0011

Table 7.5: Confidence Interval: Dist. Coll. Augmented Abilene.

Coll. Level (μ)	mean	Std. Dev.	Upper Bound	Lower Bound	Interval
0.1	0.7843	0.0064	0.7847	0.7839	0.0008
0.25	0.6067	0.0085	0.6072	0.6062	0.0011
0.4	0.4965	0.0105	0.4972	0.4958	0.0013
0.5	0.4426	0.0115	0.4433	0.4419	0.0014
1	0.2847	0.0094	0.2853	0.2841	0.0012

Table 7.6: Confidence Interval: Cent. Coll. Augmented Abilene.

Coll. Level (μ)	mean	Std. Dev.	Upper Bound	Lower Bound	Interval
0.1	0.9671	0.0039	0.9673	0.9669	0.0005
0.25	0.9164	0.0089	0.917	0.9158	0.0011
0.4	0.8683	0.0131	0.8691	0.8675	0.0016
0.5	0.8382	0.0154	0.8392	0.8372	0.0019
1	0.7148	0.0225	0.7162	0.7134	0.0028

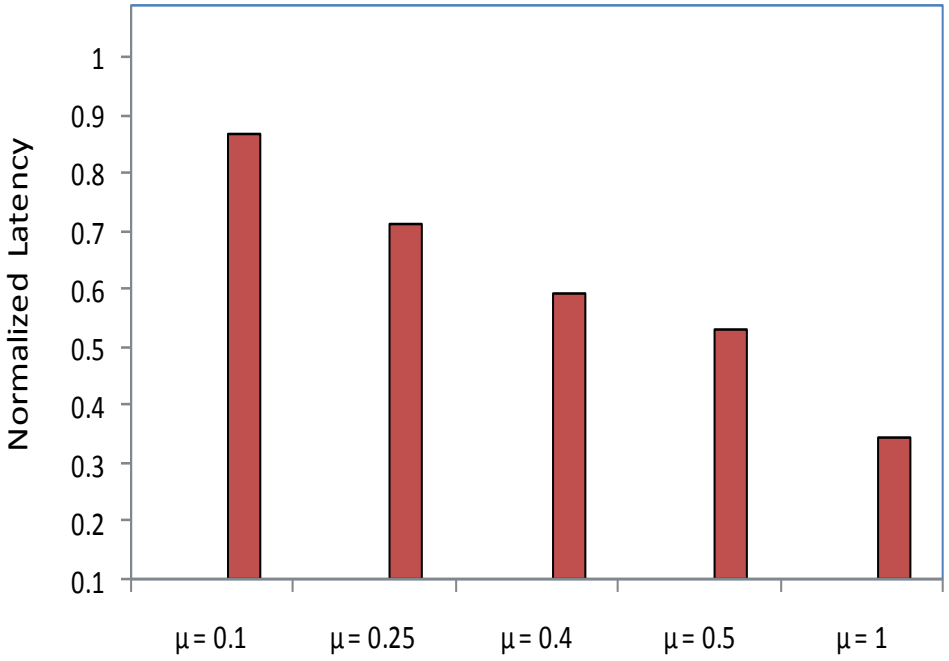


Figure 7.8: Normalized latency in a distributed collaboration for the partial-mesh backbone and resource-unconstrained data centers.

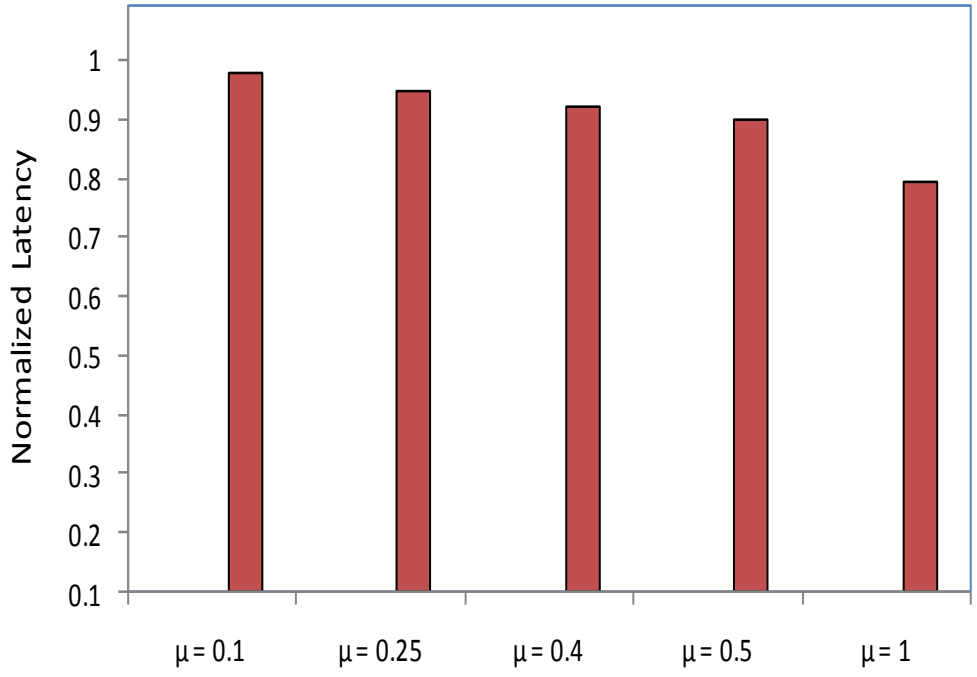


Figure 7.9: Normalized latency in a centralized collaboration for the partial-mesh backbone and resource-unconstrained data centers.

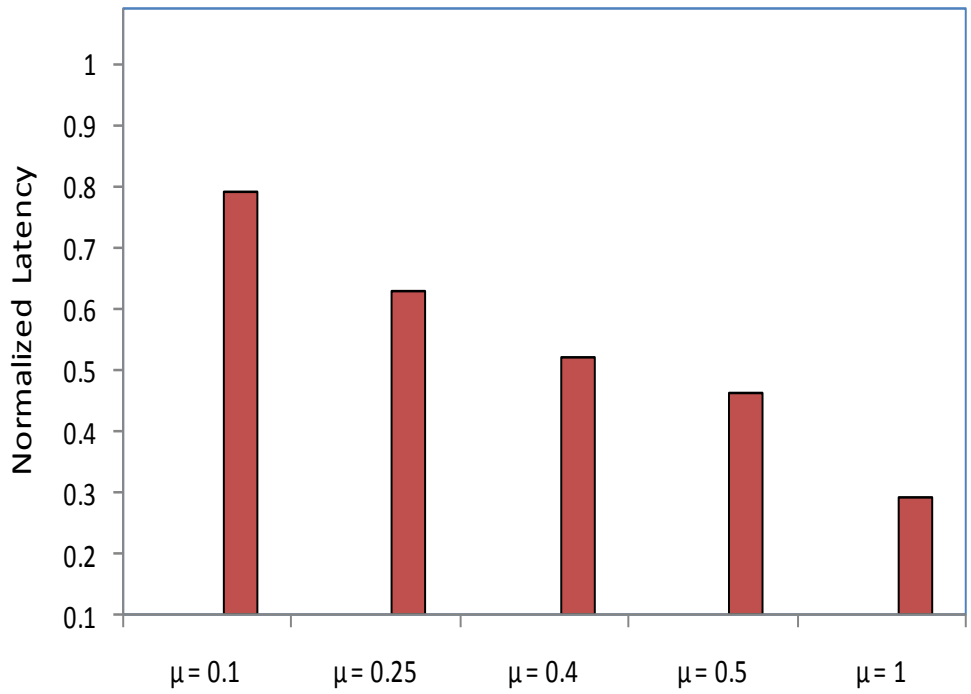


Figure 7.10: Normalized latency in a distributed collaboration for the augmented Abilene backbone and resource-unconstrained data centers.

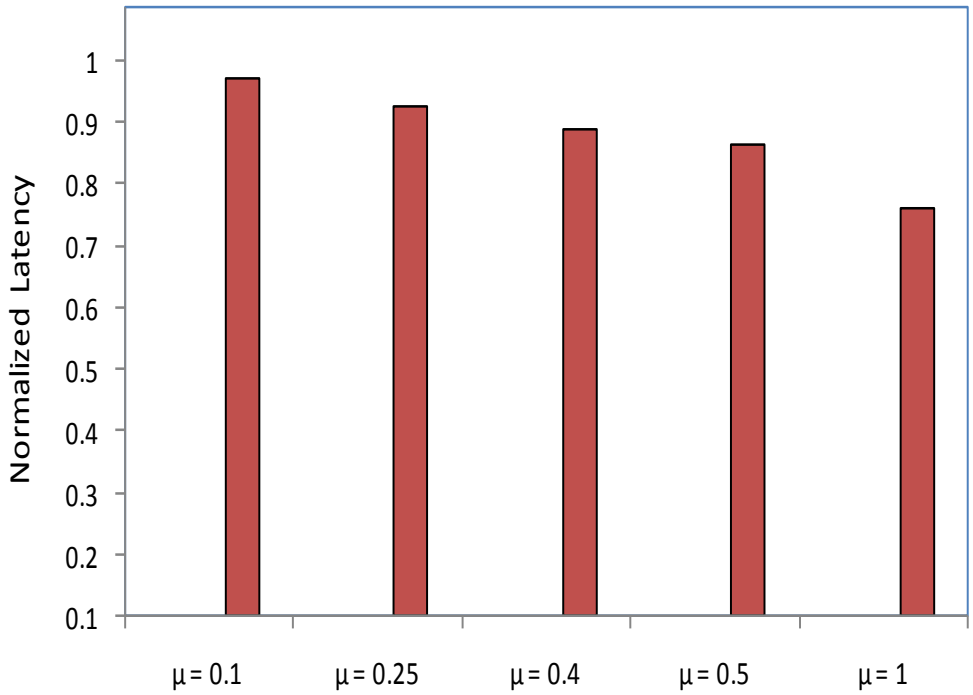


Figure 7.11: Normalized latency in a centralized collaboration for the augmented Abilene backbone and resource-unconstrained data centers.

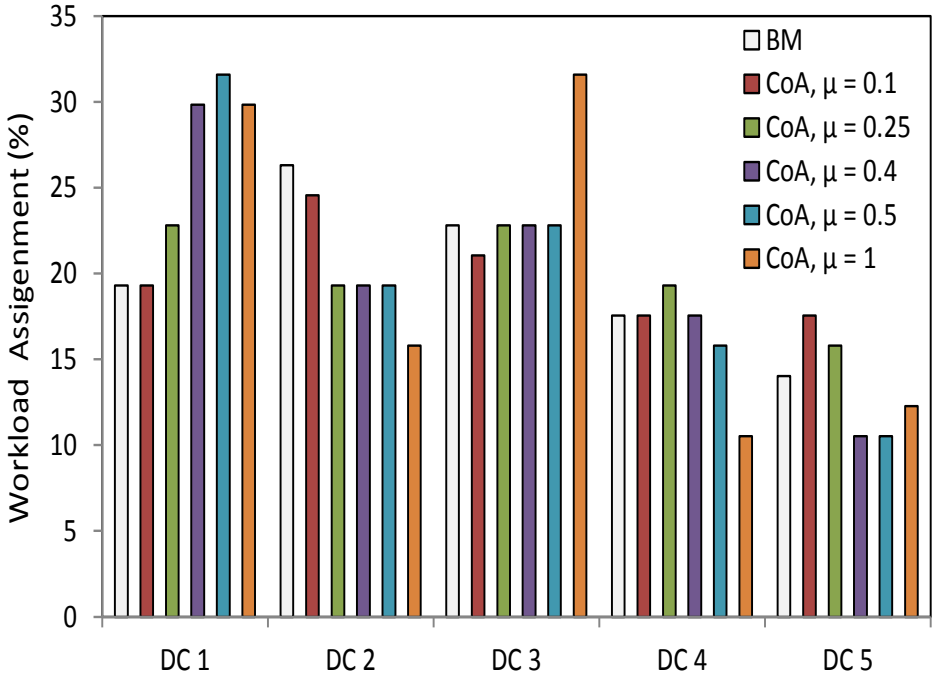


Figure 7.12: Workload distribution in a distributed collaboration for the partial-mesh backbone and resource-unconstrained data centers.

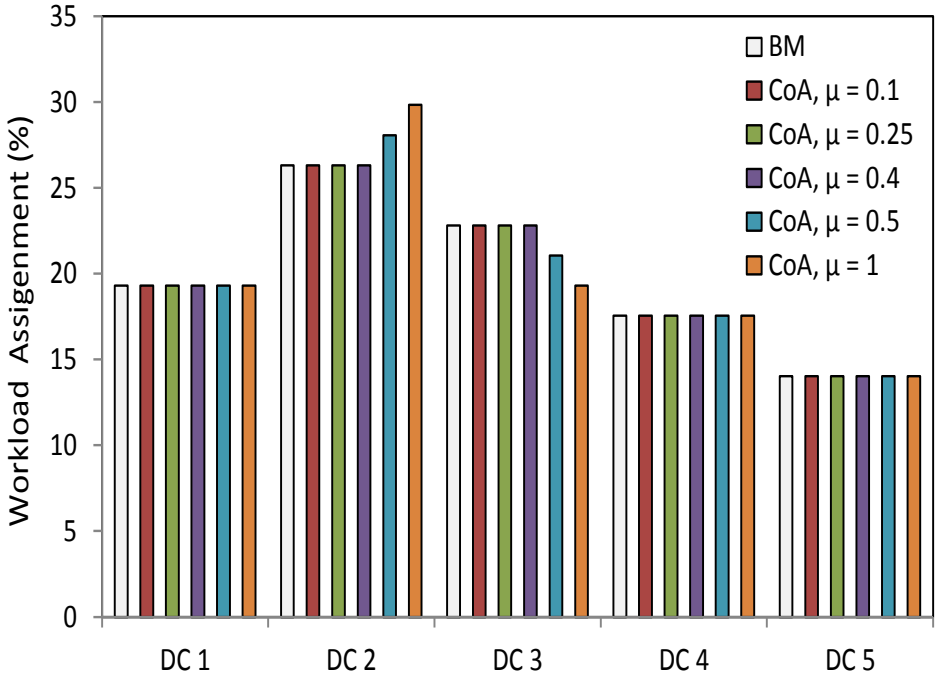


Figure 7.13: Workload distribution in a centralized collaboration for the partial-mesh backbone and resource-unconstrained data centers.

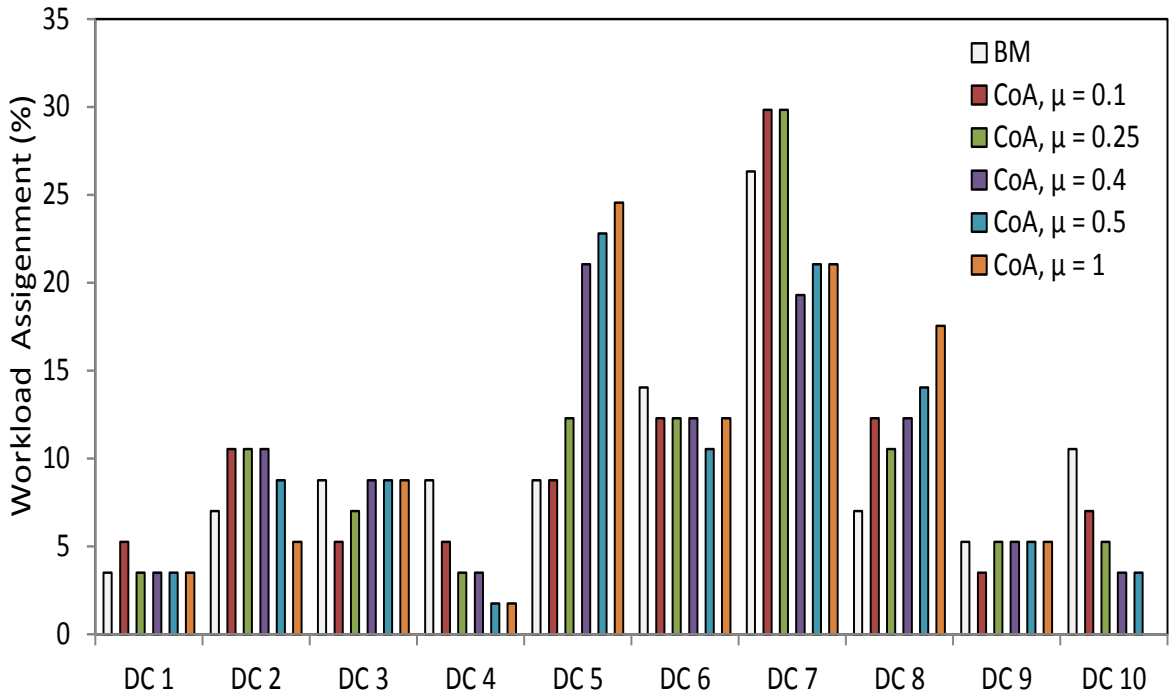


Figure 7.14: Workload distribution in a distributed collaboration in the augmented Abilene backbone and resource-unconstrained data centers.

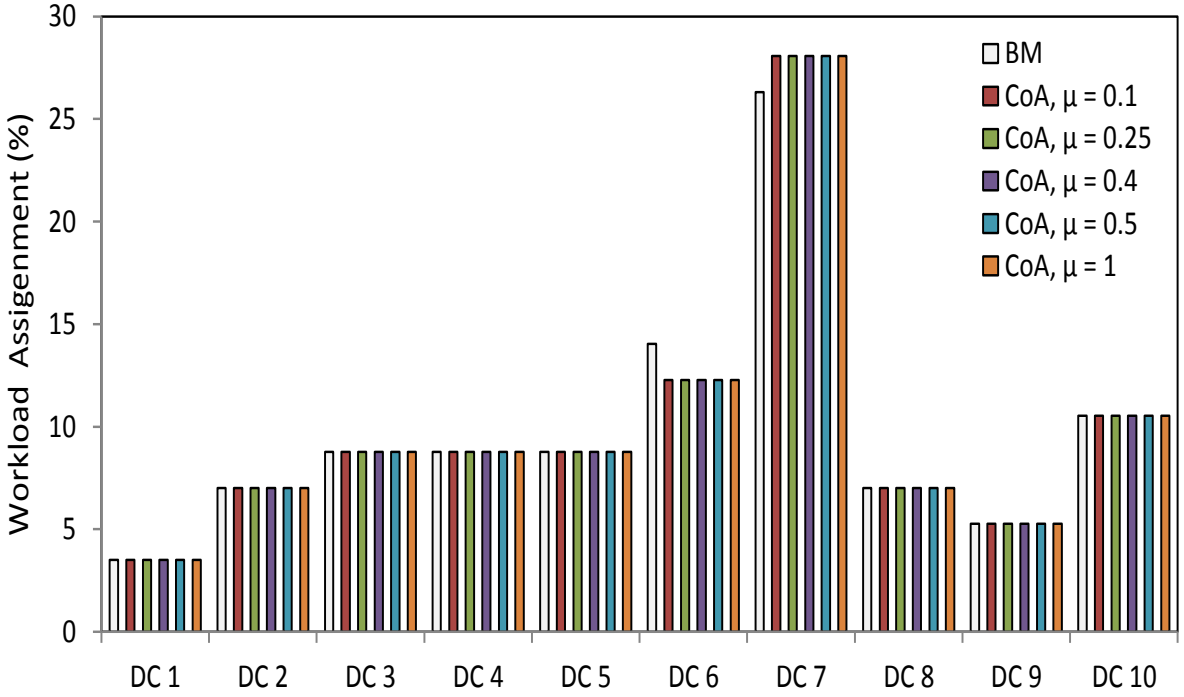


Figure 7.15: Workload distribution in a centralized collaboration for the augmented Abilene backbone and resource-unconstrained data centers.

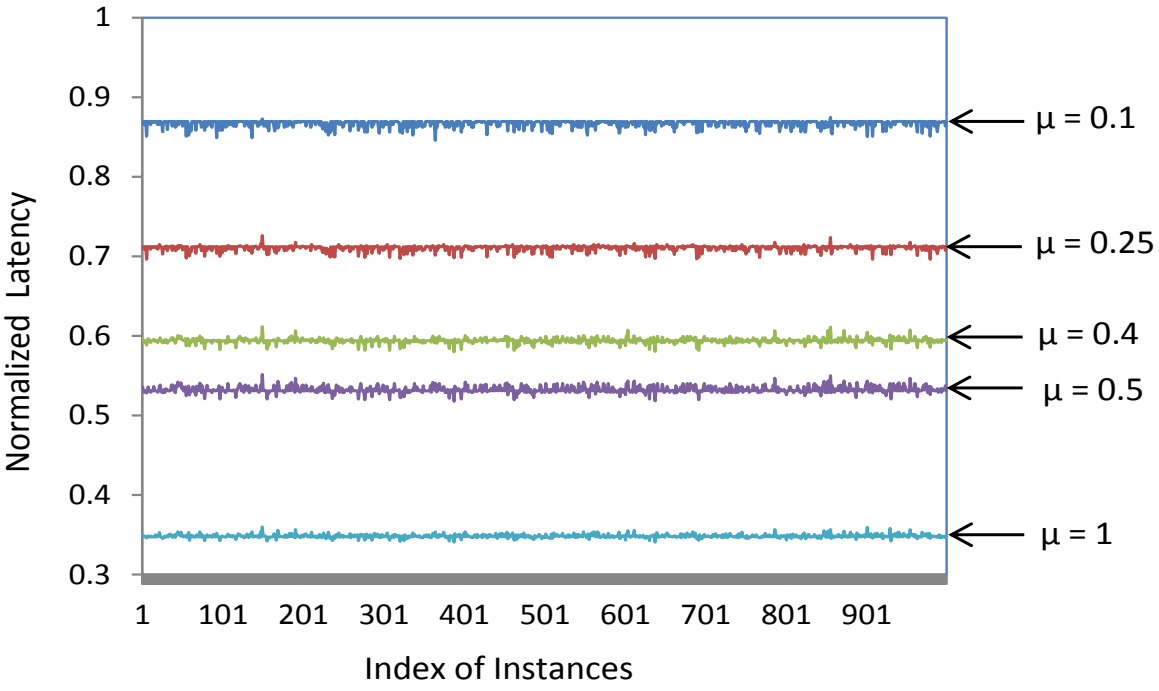


Figure 7.16: Normalized latency in a distributed collaboration for the partial-mesh backbone and resource-constrained data centers.

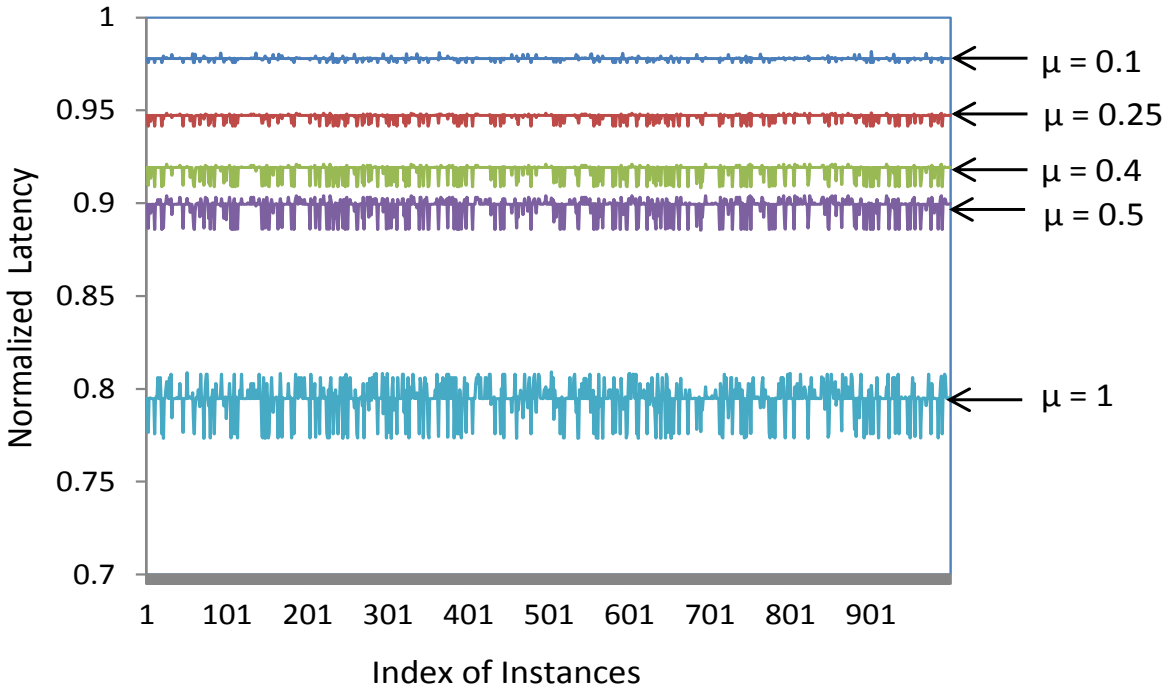


Figure 7.17: Normalized latency in a centralized collaboration for the partial-mesh backbone and resource-constrained data centers.

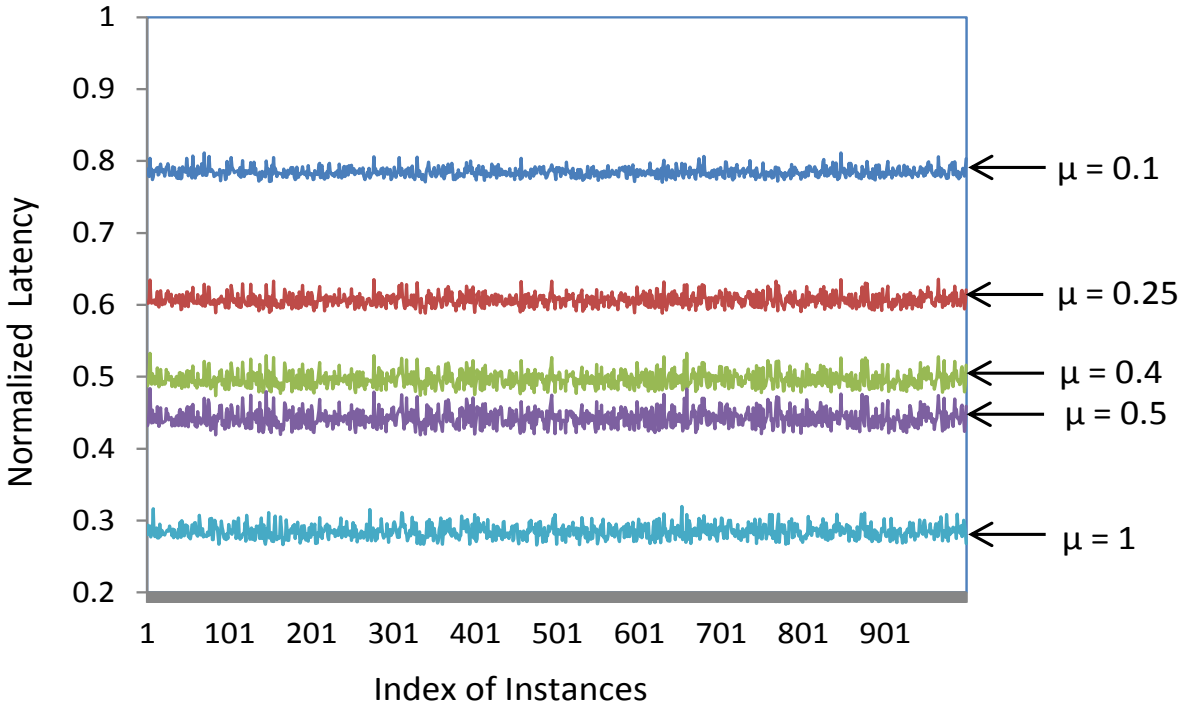


Figure 7.18: Normalized latency in a distributed collaboration for the augmented Abilene backbone and resource-constrained data centers.

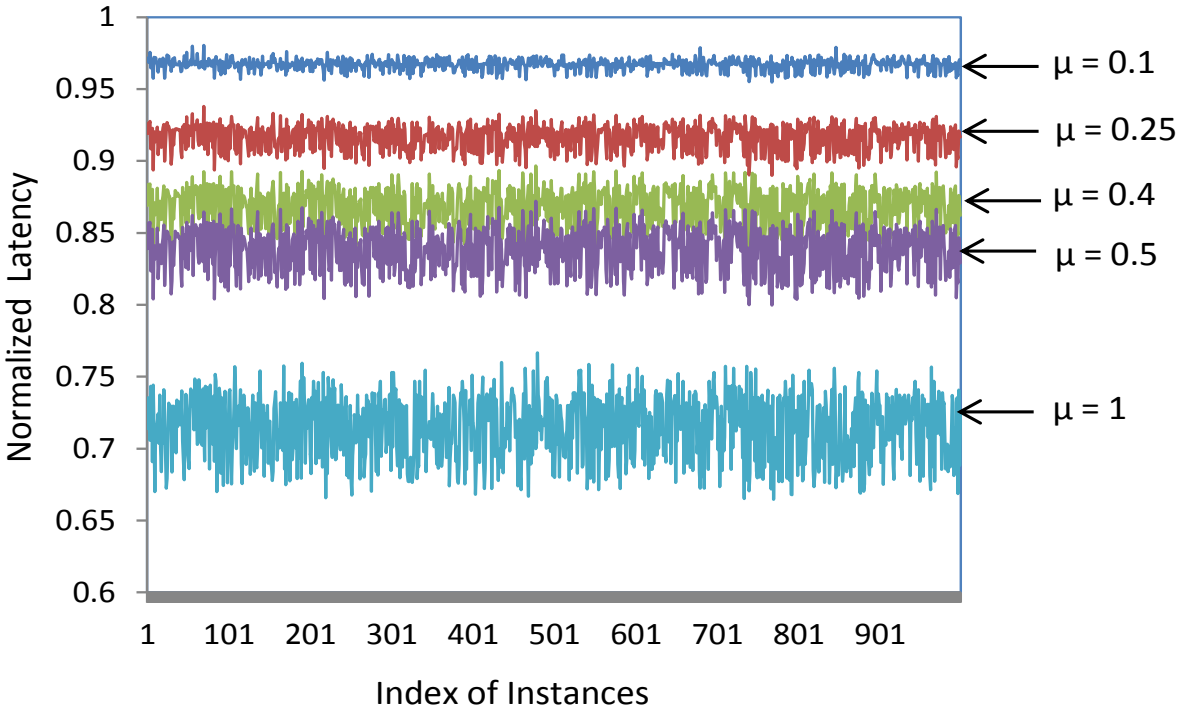


Figure 7.19: Normalized latency in a centralized collaboration for the augmented Abilene backbone and resource-constrained data centers.

7.7 Conclusion

In this chapter, an architectural framework was introduced for a cloud-centric collaborative security service that is dedicated to smart grid AMI networks. The architecture considers a geographically distributed private cloud and an AMI WAN. It aims to perform security monitoring on upstream traffic whose initial source is the customer owned smart meters. A service placement scheme is also proposed that to minimize the overall latency between clients and servers. It includes both the impact of types and amounts of collaboration. It was evaluated for two different backbone network topologies. It exhibits an improvement of latency in all experimental setups.

Chapter 8

Conclusion and Future Work

In this chapter, we conclude this thesis based on a review of its major contributions. In addition, we discuss some research directions for future work.

8.1 Concluding Remarks

This thesis includes a number of cyber security placement schemes that were developed for three different smart city infrastructures. Such infrastructures include WSNs, smart grid SCADA networks, and AMIs. The security placement problems have been studied from different optimization aspects depending on the context of each infrastructure. For WSNs, the watchdog placement problems have been studied from a resource management perspective. Regarding SCADA networks, the trust system placement problems have been studied considering different goals and scenarios. Regarding AMIs, a latency-aware cloud-based security service placement problem has been studied. Our concluding remarks based on the major contributions of this thesis are as follows.

We have developed four optimization models to study the watchdog selection problem in WSNs. In this study, watchdogs are the security resources that perform behavioral monitoring of sensor nodes. There are two major concerns: monitoring coverage and overlapping between coverages. The ideal goal is to maximize the coverage while keeping

the overlapping minimal. The proposed models have been evaluated to identify their best-suited scenarios.

We have developed a computationally lightweight segmentation approach to solving the trust system placement problem in smart grid SCADA networks. Our approach uses the MST of a given network to create segments. It provides efficient handling of star connections along with offering a better balance between segment sizes. A new metric named minimum link degree is used to handle star connections and to prevent singleton segments. For the IEEE test system topologies, its computed segment sizes are low-variance. Thus, it helps limit the impact of a successful cyber-attack. Subsequently, we have extended the segmentation approach to solving the constrained trust system placement problems while separately studying the resource and latency constraints. For the first case, a placement scheme is proposed that maximizes the number of monitored links for a given number of trust systems. The second case addressed the time criticality issue in alert propagations. We proposed a latency threshold-based scheme that reduces the geographic dispersion of segments.

We have formulated two optimization models that solve trust system placement problems for distributed monitoring in smart grid SCADA networks. Those models are appropriate for the active/router mode, whereas segmentation methods focus on the tunnel/gateway mode of operation. The router mode is an appropriate option for monitoring intra-segment traffic. We established a quadratic objective function that exactly represents the total number of links monitored by the deployed trust systems in a given network. Our objective function is much more efficient than the linear one in maximizing link coverages. In addition, we have introduced a new metric named path tolerance for distributed monitoring in SCADA networks. It is a measure of a maximum number of consecutive unmonitored links in a path. Its lower value indicates a better security. We proposed a scheme that optimizes the path tolerance for a given number of trust systems. The proposed scheme can be used as an expansion planning tool for a segmented network.

We have proposed a cloud-centric collaborative security monitoring architecture for the future AMI WANs. It aims to exploit the potential of cloud computing to deliver secu-

rity service to clients' geographically distributed enterprise networks. We have proposed and evaluated a collaborative security service placement scheme that reduces latency. Our scheme solves an optimization problem considering both kinds of latencies for geographically distributed private clouds, access and internal. It has been evaluated for distributed and centralized collaborations. It shows a significant reduction in the latency, especially for the distributed ones.

8.2 Future Work

Cyber security enhancement in smart city infrastructures is an emerging research area. Its scope and diversity are getting wider day by day. Only a subset of its deployment issues has been studied in this thesis. The proposed schemes in this thesis can be extended to many other possible directions for further investigations. In particular, the following directions are noteworthy.

• In this thesis, the proposed watchdog placement schemes only provide a static solution. The energy consumption rate of a watchdog is much higher than that of an ordinary sensor node. It depends on two factors: (i) communication frequency between watchdogs and the BS and (ii) the distance between watchdogs and the BS. A watchdog is required to perform more frequent communications and the distance is much longer. An extension to a dynamic and periodic scheduling of watchdogs can be an interesting direction. It can address a fairness issue based on the energy consumption.

• In this thesis, the proposed trust system placement schemes mainly exploit graph theoretic properties of SCADA networks. In the future, the volume of SCADA traffic is expected to be much higher. This is due to the massive implementation of smart grid and IoT technologies. The consideration of traffic handling capacities can add a new dimension to the trust system placement problem.

• The extensive volume of SCADA traffic is more likely to cause excessive queueing delay at trust nodes. It can affect the time criticality regarding alert propagations. A stochastic model can be developed to address this excessive delay. The model can be a

useful tool for the expanded trust system placement plans.

• Cyber vulnerability assessment of individual smart grid components is an important task. Each node in a SCADA network is required to be assessed based on its cyber-physical attributes. In the trust system placement problem, a priority can be set for the most vulnerable nodes.

• In this thesis, the proposed security service placement scheme relaxed bandwidth constraints in both access and cloud domains. For the access domain, the bandwidth requirement can be estimated using an AMI traffic model. For the cloud domain, the bandwidth requirement can be estimated by the amount of collaborative information to be exchanged between peers. The consideration of bandwidth can be an interesting and challenging direction.

References

[Abd13] A. Abduvaliyev et al. "On the vital areas of intrusion detection systems in wireless sensor networks". *IEEE Communications Surveys and Tutorials*, Vol. 15, No. 3, pp. 1223-1237, 2013.

[abi12] "Abilene Backbone". http://abilene.internet2.edu, 2012. [Online, accessed Oct. 2016].

[Al15] A. Al-Fuqaha et al. "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications". *IEEE Communications Surveys and Tutorials*, Vol. 17, No. 4, pp. 2347-2376, May 2015.

[Alh12] T. Alharkan and P. Martin. "IDSaaS: intrusion detection system as a service in public clouds". In *Proceedings of the 3rd Cloud Computing 2012*, Ottawa, ON, Canada, pp. 11-17, 2012.

[Alm16] H. Almohri et al. "Security optimization of dynamic networks with probabilistic graph modeling and linear programming". *IEEE Transactions on Dependable and Secure Computing*, Vol. 13, No. 4, pp. 474-457, July/Aug. 2016.

[Bat12] M. Batty et al. "Smart cities of the future". *European Physical Journal Special Topics*, No. 214, pp. 481-518, Nov. 2012.

[Bed12] I. L. Bedhiaf, R. B. Ali, and O. Cherkaoui. "On the problem of mapping virtual machines to physical machines for delay sensitive services". In *Proceedings of the GLOBECOM2012-Symposium on Next Generation Networking and Internet*, Anaheim, CA, USA, pp. 2628-2633, 2012.

[Bel57] R. E. Bellman. *Dynamic Programming*. Princeton University Press, NJ, USA, 1957.

[Ben08] Z. Beneson et al. "Vulnerabilities and Attacks in Wireless Sensor Networks". In *Wireless Sensor Network Security*. IOS Press, Amsterdam, Netherlands, 2008.

[Ber15] S. Bera et al. "Cloud computing applications for smart grid: a survey". *IEEE Transactions on Parallel and Distributed Systems*, Vol. 18, No. 5, pp. 1477-1494, May 2015.

[Bha83] K. Bharath-Kumar and J. M. Jaffe. "Routing to multiple destinations in computer networks". *IEEE Transactions on Communications*, Vol. 31, No. 3, pp. 343-351, March 1983.

[Bra87] Paul Bratley. *A Guide To Simulation*. second edition, Springer Verlag, New York, NY, USA, 1987.

[Bro11] J. Brown and P. Robinson. "PKI reborn in the cloud". In *RSA Conference Europe 2011*, London, UK, 2011.

[But14] I. Butun I. S. Morgera, and R. Sankar. "A survey of intrusion detection systems in wireless sensor networks". *IEEE Communications Surveys and Tutorials*, Vol. 16, No. 1, pp. 266-282, 2014.

[Cam07] T. Camilo et al. "GENSEN: A topology generator for real wireless sensor networks deployments". In *Proceedings of IFIP SEUS*, Santorini Island, Greece, pp. 436-445, 2007.

[Car14] A. A. Cardenas et al. "A framework for evaluating intrusion detection architectures in advanced metering infrastructures". *IEEE Transactions on Smart Grid*, Vol 5, No. 2, pp. 906-915, March 2014.

[Cav09] A. Cavoukian, J. Polonetsky, and C. Wolf. *Smart Privacy for the Smart Grid: Embedding Privacy into the Design of Electricity Conservation*. Information and Privacy Commissioner, Toronto, ON, Canada, 2009.

[Chv83] V. Chvatal. *Linear Programming*. Freeman, New York, NY, USA, 1983.

[Coa08] G. M. Coates et al. "Collaborative, trust-based security mechanisms for a regional utility intranet". *IEEE Transactions on Power Systems*, Vol. 23, No. 3, pp. 831-844, Aug. 2008.

[Coa10] G. M. Coates et al. "A trust system architecture for SCADA network security". *IEEE Transactions on Power Delivery*, Vol. 25, No. 1, pp. 158-169, Jan. 2010.

[Cor09] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, USA, third edition, 2009.

[cpl] https://www.ibm.com/software/commerce/optimization/cplex-optimizer.

[csa11] "Defined categories of service". Technical Report SECaaS Version 1.0, Cloud Security Alliance, 2011.

[Dua14] J. Duan et al. "An energy-aware trust derivation scheme with game theoretic approach in wireless sensor networks for IoT applications". *IEEE IoT Journal*, Vol. 1, No. 1, pp. 58-69, Feb. 2014.

[Eri00] G. Ericsson. "Cyber security and power system communication- essential parts of a smart grid infrastructure". *IEEE Transactions on Power Delivery*, Vol. 25, No. 3, pp. 1501-1507, July 2000.

[Ern12] T. A. Ernster and A. Srivastava. "Power system vulnerability analysis- towards validation of centrality measures". In *Proceedings of IEEE PES T&D Conference and Exposition*, Orlando, FL, USA, pp. 1-6, 2012.

[eu007] "Smart cities: ranking of European medium sized cities". Technical report, The European Union, Vienna, Austria, 2007. [Final report].

[Fai15] M. A. Faisal et al. "Data-stream-based intrusion detection system for advanced metering infrastructure in smart grid: a feasibility study". *IEEE Systems Journal*, Vol. 9, No. 1, pp. 31-44, March 2015.

[fao09] "How to Feed the world in 2050". Technical report, FAO, 2009. [Online, accessed April 2013], available: http://www.fao.org/wsfs/forum2050.

[Fou12] M. Fouda et al. "A lightweight message authentication scheme for smart grid communications". *IEEE Transactions on Smart Grid*, Vol. 2, No. 4, pp. 675-685, Dec. 2012.

[Get12] V. Getov. "Security as a service in smart clouds- opportunities and concerns". In *Proceedings of 36th IEEE ICCSA2012*, Izmir, Turkey, pp. 373-378, 2012.

[Gho13] A. Ghosh et al. "Scalable multi-class traffic management in data center backbone networks". *IEEE Journal on Selected Areas in Communications*, Vol. 31, No. 12, pp. 1-12, Dec. 2013.

[Gia13] A. Giani el al. "Smart grid data integrity attacks". *IEEE Transactions on Smart Grid*, Vol. 4, No. 3, pp. 1244-1253, Sept. 2013.

[Gon11] J. Gonzalez et al. "Optimization of trust system placement for power grid security and compartmentalization". *IEEE Transactions on Power Systems*, Vol. 26, No. 2, pp. 550-563, May 2011.

[Gre13] R. C. Green II, L. Wang, and M. Alam. "Applications and trends of high performance computing for electric power systems: focusing on smart grid". *IEEE Transactions on Smart Grid*, Vol. 4, No. 2, pp. 922-931, June 2013.

[Has13a] M. M. Hasan and H. T. Mouftah. "Cloud-based security services for the smart grid". In *Proceedings of the CASCON2013*, Markham, ON, Canada, pp. 388-391, 2013.

[Has13b] M. M. Hasan and H. T. Mouftah. "Security and Privacy Challenges in a Smart City". In *Proceedings of the Fifth WiSENSE2013*, Ottawa, ON, Canada, pp. 8-12, 2013.

[Has15] M. M. Hasan and H. T. Mouftah. "Encryption as a service for smart grid advanced metering infrastructure". In *Proceedings of the ISCC2015*, Larnaca, Cyprus, pp. 216-221, 2015.

[Has16a] M. M. Hasan and H. T. Mouftah. "A study of resource-constrained cyber security planning for smart grid networks". In *Proceedings of the EPEC2016*, Ottawa, ON, Canada, pp. 1-6, 2016.

[Has16b] M. M. Hasan and H. T. Mouftah. "Latency-aware segmentation and trust system placement in smart grid SCADA networks". In *Proceedings of the CAMAD2016*, Toronto, ON, Canada, pp. 37-42, 2016.

[Has16c] M. M. Hasan and H. T. Mouftah. "Optimal trust system placement in smart grid SCADA networks". *IEEE Access*, Vol. 4, pp. 2907-2919, May 2016.

[Has17a] M. M. Hasan and H. T. Mouftah. "Cloud-centric collaborative security service placement for advanced metering infrastructures". *IEEE Transactions on Smart Grid*, 2017. submitted.

[Has17b] M. M. Hasan and H. T. Mouftah. "Cyber-Physical Vulnerabilities of Wireless Sensor Networks in Smart Cities". In *Security and Privacy in Cyber-Physical Systems: Foundations and Applications*. John Wiley, London, UK, pp. 263-280, 2017.

[Has17c] M. M. Hasan and H. T. Mouftah. "Optimization of watchdog selection in wireless sensor networks". *IEEE Wireless Communications Letters*, Vol. 6, No. 1, pp. 94-97, Feb. 2017.

[Hol03] P. Holme. "Congestion and centrality in traffic flow on complex networks". *Advances in Complex Systems*, Vol. 6, No. 2, pp. 163-176, July 2003.

[Hua05] C. Huang and Y. Tseng. "The coverage problem in a wireless sensor network". *Springer Mobile Networks and Applications*, Vol. 10, No. 4, pp. 519-528, Feb. 2005.

[Hua11] J. Huang et al. "A business model for cloud computing based on a separate encryption and decryption service". In *Proceedings of the ICISA 2011*, Jeju Island, South Korea, pp. 1-7, 2011.

[Hua12] G. Huang et al. "Measurement-aware monitor placement and routing: a joint optimization approach for network-wide measurements". *IEEE Transactions on Network Service Management*, Vol. 9, No. 1, pp. 48-59, March 2012.

[Hua15] H. Huang et al. "Network distance prediction for enabling service-oriented applications over large-scale networks". *IEEE Communications Magazine*, Vol. 53, No. 8, pp. 166-174, Aug. 2015.

[Hus11] M. Hussain and H. Abdulsalam. "SECaaS: Security as a service for cloud based application". In *Proceedings of the 2nd Kuwait Conference on E-Services and E-Systems, 2011*, Kuwait, pp. 1-4, 2011.

[Hwa10a] K. Hwang and D. Li. "Trusted cloud computing with secure resources and data coloring. *IEEE Internet Computing*, Vol 14, No. 5, pp. 14-22, Sept. 2010.

[Hwa10b] J. Hwang, T. He, and Y. Kim. "Exploring in-situ sensing irregularity in wireless sensor networks". *IEEE Transactions on Parallel and Distributed Systems*, Vol. 21, No. 4, pp. 547-561, April 2010.

[ibm15] http://www.ibm.com/common/sc/simulator, 2015. [Online, accessed Feb. 2015].

[iee13] "IEEE vision for smart grid communications: 2030 and beyond". Technical Report STDV98261, IEEE Standards Association, New York, NY, USA, 2013.

[iee15] "Power Syst. Test Case Arch.". http://www.ee.washington.edu/research/pstca/, 2015. [Online, accessed April 2015].

[int15a] http://www.intel.com, 2015. [Online, accessed Feb. 2015].

[int15b] "Collaborative security: an approach to tackling Internet security issues". Technical report, Internet Society, Geneva, Switzerland, 2015. [Online, accessed Jan. 2017], available: http://www.internetsociety.org.

[Kan14] S. Kang, B. Veeravalli, and K. Aung. "ESPRESSO: An encryption as a service model for cloud storage systems". In *Proceedings of the AIMS2014*, Brno, Czech Republic, pp. 15-28, 2014.

[Khu12] R. Khune and J. Thangakumar. "A cloud-based intruder detection system for Android smartphones". In *Proceedings of the ICRCC 2012*, pp. 180-184, 2012.

[Kna13] E. D. Knapp and R. Samani. *Applied Cyber Security and the Smart Grid*. Elsevier, Watham, MA, USA, 2013.

[Koc11] A. Kochut et al. "Evolution of the IBM cloud: enabling an enterprise cloud services ecosystem". *IBM Journal of Research and Development*, Vol. 55, No. 6, pp. 1-7, Nov.-Dec. 2011.

[Las05] M. Laszio and S. Mukherjee. "Minimum spanning tree partitioning algorithm for microaggregation". *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 7, pp. 902-911, July 2005.

[Law13] Y. Law, G. Kounga, and A. Lo. "WAKE: Key management scheme for wide-area measurement systems in smart grid". *IEEE Communications Magazine*, Vol. 51, No. 1, pp. 34-41, Jan. 2013.

[Lia10] G. Liang, R. Agarwal, and N. Vaidya. "When watchdog meets coding". In *Proceedings of IEEE INFOCOM*, San Diego, CA, USA, pp. 2267-2275, 2010.

[Lu12] R. Lu et al. "EPPA: An efficient and privacy preserving aggregation scheme for secure smart grid communications". *IEEE Transactions on Parallel and Distributed Systems*, Vol. 23, No. 9, pp. 1621-1631, May 2012.

[Lu15] R. Lu et al. "Recent Advances in Industrial Wireless Sensor Networks Toward Efficient Management in IoT". *IEEE Access*, Vol. 3, pp. 622-637, May 2015.

[mat] http://www.mathworks.com.

[Men13] S. Meng and L. Liu. "Enhanced monitoring-as-a service for effective cloud management". *IEEE Transactions on Computers*, Vol. 62, No. 9, pp. 1705-1720, Sept. 2013.

[Men15] G. Meng et al. "Collaborative security: a survey and taxonomy". *ACM Computing Surveys*, Vol. 48, No. 1, pp. 1-42, July 2015.

[Mes12] J. Meszaros. "Towards security management in clouds utilizing SECaaS". In *Proceedings of the WSEAS Latest Trends in IT2012*, Vienna, Austria, pp. 449-455, 2012.

[mit12] http://www.cities.mit.edu, 2012. [Online; accessed April 2013].

[Moh11] A. Mohsenian-Rad and A. Leon-Garcia. "Distributed Internet-based load alerting attacks against smart power grids". *IEEE Transactions on Smart Grid*, Vol. 2, No. 4, pp. 667-674, Dec. 2011.

[Mon15] Q. Monnet et al. "Fair election of monitoring nodes in WSNs". In *Proceedings of GLOBECOM2015-Symposium on Adhoc and Sensor Networks*, San Diego, CA, USA, pp. 1-6, 2015.

[New10] Mark Newman. *Networks An Introduction.* Oxford University Press, Oxford, UK, 2010.

[nis10] The smart grid interoperability panel a cyber security working group, guidelines for smart grid cyber security. Technical report, NISTIR 7628, NIST, 2010.

[oci15] "The Future of Smart Cities: Cyber-Physical Infrastructure Risk". Technical Report OMB 1670-0027, U.S. DHS/OCIA, 2015.

[ora15] http://www.oracle.com/us/products, 2015. [Online, accessed Jan. 2015].

[Pan16] X. Pan et al. "HogMap: Using SDNs to incentivize collaborative security monitoring". In *Proceedings of the SDN-NFVSec2016*, New Orleans, LA, USA, pp. 7-12, 2016.

[Pas15] S. Pastrana et al. "DEFIDNET: A framework for optimal allocation of cyberdefenses in intrusion detection networks". *Elsevier Computer Networks*, Vol. 80, pp. 66-88, 2015.

[Pat17] A. Patel et al. "A nifty collaborative intrusion detection and prevention architecture for smart grid ecosystems". *Elsevier Computers & Security*, Vol 64, pp. 92-109, 2017.

[Per14] C. Perera et al. "A Survey on Internet of things From industrial market perspective". *IEEE Access*, Vol. 2, pp. 1660-1679, Jan. 2014.

[Rak13] M. Rak et al. "Security as a service using an SLA-based approach via SPECS". In *Proceedings of the IEEE ICCTS2013*, Bristol, UK, pp. 1-6, 2013.

[Ren14] Y. Ren et al. "A novel approach to trust management in unattended wireless sensor networks". *IEEE Transactions on Mobile Computing*, Vol. 13, No. 7, pp. 1409-1423, July 2014.

[Ren16] J. Ren et al. "Adaptive and channel-aware detection of selective forwarding attacks in wireless sensor networks". *IEEE Transactions on Wireless Communications*, Vol. 15, No, 5, pp. 3718-3731, May 2016.

[Saj16] A. Sajid, H. Abbas, and K. Saleem. "Cloud-assisted IoT-based SCADA systems security: a review of the state of the art and future challenges". *IEEE Access*, Vol. 4, pp. 1375-1384, March 2016.

[Sat16] A. V. Sathanur and D. J. Haglin. "A novel centrality measure for network-wide cyber vulnerability assessment". In *Proceedings of IEEE HST*, Waltham, MA, USA, pp. 1-5, 2016.

[Sch12] H. Schaffers et al. "Smart cities and the future Internet: towards cooperation frameworks for innovation". *Springer LNCS*, No. 6656, pp. 431-446, 2012.

[Sha16] A. Shameli-sendi et al. "Efficient provisioning of security service function chaining using network security defense patterns". *IEEE Transactions on Service Computing*, pp(99), Vol. pp, No. 99, pp. 1-14, 2016.

[She07] H. D. Sherali, and J. C. Smith. "An improved linearization strategy for zero-one quadratic programming problems". *Optimization Letters*, Vol. 1, No. 1, pp. 33-47, Jan. 2007.

[Soh07] K. Sohraby, D. Minoli, and T. Znati. *Wireless Sensor Networks- Technology, Protocols, and Applications.* John Wiley and Sons Inc., Hoboken, NJ, USA, 2007.

[Som10] T. Sommestad, G. Ericsson, and J. Nordlander. "SCADA system cyber security-a comparison of standards". In *IEEE Power and Energy Society General Meeting*, Minneapolis, MN, USA, pp. 1-8, 2010.

[Sou13] K. Sou, H. Sandberg, and K. Johansson. "On the exact solution to a smart grid cyber-security analysis problem". *IEEE Transactions on Smart Grid*, June Vol. 4, No. 2, pp. 856-865, June 2013.

[Sri12] S. Sridhar, A. Hahn, and M. Govindarasu. "Cyber-physical system security for the electric power grid". *IEEE Proceedings*, Vol. 100, No. 1, pp. 210-224, Jan. 2012.

[Sri13] A. Srivastava et al. "Modeling cyber-physical vulnerability of the smart grid with incomplete information". *IEEE Transactions on Smart Grid*, Vol. 4, No. 1, pp. 235-244, March 2013.

[Ste15] G. Stergiopoulosa et al. "Risk mitigation strategies for critical infrastructures based on graph centrality analysis". *Elseveir International Journal of Critical Infrastructure Protection*, No. 10, pp. 34-44, 2015.

[Su16] M. Su et al. "Systematic data placement optimization in multi-cloud storage for complex requirements". *IEEE Transactions on Computers*, Vol. 65, No. 6, pp. 1964-1977, May 2016.

[Sub11] L. Subramanian et al. "An architecture to provide cloud-based security services to smartphones". In *Proceedings of the 27th WWRF Meeting*, Dusseldorf, Germany, pp. 1-8, 2011.

[Sun07] B. Sun et al. "Intrusion detection techniques in mobile ad-hoc and wireless sensor networks". *IEEE Wireless Communications Magazine*, Vol. 14, No. 5, pp. 56-63, Oct. 2007.

[Tha16] U. Thakore, G. A. Weaver, and W. H. Sanders. "A quantitative methodology for security monitor deployment". In *Proceedings of the IEEE/IFIP DSN2016*, Tolouse, France, pp. 1-12, 2016.

[Tit13] D. Titze, P. Stepahnow, and J. Schutte. "A configurable and extensible security service architecture for smartphones". In *Proceedings of the 27th WAINA 2013*, pp. 1057-1063, 2013.

[Tro13] P. A. Trodden et al. "MILP formulation for controlled islanding of power networks". *Elsevier Electrical Power and Energy Systems*, Vol. 45, pp. 501-508, 2013.

[ucl12] "Smart cities study: international study on the situation of ICT, innovation, and knowledge in cities". Technical report, The Committee of Digital and Knowledge-based Cities of UCLG, 2012.

[Vel15] C. Vellaithurai et al. "CPIndex: Cyber-physical vulnerability assessment for power-grid infrastructures". *IEEE Transactions on Smart Grid*, Vol. 6, No. 2, pp. 566-575, March 2015.

[Wal06] J. Walters et al. "Wireless Sensor Network Security: a Survey". In *Security in Distributed, Grid, and Pervasive Computing*. CRC Press, Boca Raton, FL, USA, 2006.

[Wan13] W. Wang and Z. Lu. "Cyber security in the smart grid: survey and challenges". *Elsevier Computer Networks*, Vol. 57, pp. 1344-1371, 2013.

[Wu04] Bang Ye Wu and Kun-Mao Chao. *Spanning Trees and Optimization Problems*. CRC Press, Boca Raton, Florida, 2004.

[Wu11] F. Wu, Y. Kao, and Y. Tseng. "From wireless sensor networks towards cyber physical systems". *Elsevier Pervasive and Mobile Computing*, Vol. 7, pp. 397-413, 2011.

[Wu16] J. Wu et al. "A hierarchical security framework for defending against sophisticated attacks on wireless sensor network in smart cities". *IEEE Access*, Vol. 4, pp. 416-424, Jan. 2016.

[Xia13] Z. Xiao, Y. Xiao, and D. Du. "Non-repudiation in neighborhood area networks for smart grid". *IEEE Comm. Magazine*, Vol. 51, No. 1, pp. 18-26, Jan. 2013.

[Yan06] B. Yang, V. Vittal, and G. T. Heydt. "Slow-coherency-based controlled islanding-a demonstration of the approach on the August 14, 2003 blackout scenario". *IEEE Transactions on Power Systems*, Vol. 21, No. 4, pp. 1840-1847, Nov. 2006.

[Yan16] L. Yang et al. "Cost aware service placement and load dispatching in mobile cloud systems". *IEEE Transactions on Computers*, Vol. 65, No. 5, pp. 1440-1452, May 2016.

[Yas12] W. Yassin et al. "A cloud-based intrusion detection service framework". In *Proceedings of the CyberSec2012*, Kualalampur, Malaysia, pp. 213-218, 2012.

[Yig14] M. Yigit et al. "Cloud computing for smart grid applications". *Elsevier Computer Networks*, Vol. 70, pp. 312-329, 2014.

[Zan14] A. Zanella et al. "Internet of Things for Smart Cities". *IEEE IoT Journal*, Vol. 1, No. 1, pp. 22-32, Feb. 2014.

[Zha11] Y. Zhang et al. "Distributed intrusion detection system in a multi-layer network architecture of smart grid". *IEEE Transactions on Smart Grid*, Vol. 2, No. 4, pp. 796-808, Dec. 2011.

[Zha13a] Q. Zhang et al. "Dynamic service placement in geographically distributed clouds". *IEEE Journal on Selected Areas in Communications*, pp. Vol. 31, No. 10 1-11, Oct. 2013.

[Zha13b] Y. Zhang et al. "Trust system design optimization in smart grid network infrastructure". *IEEE Transactions on Smart Grid*, Vol. 4, No. 1, pp. 184-195, March 2013.

[Zha16] Y. Zhang et al. "Inclusion of SCADA cyber vulnerability in power system reliability assessment considering optimal resources allocation". *IEEE Transactions on Power Systems*, Vol. 31, No. 6, pp. 4379-4394, Nov. 2016.

[Zho15] P. Zhou et al. "Toward energy efficient trust system through watchdog optimization for WSNs". *IEEE Transactions on Information Forensics and Security*, Vol. 10, No. 3, pp. 613-625, March 2015.

[Zhu12] Q. Zhu et al. "GUIDEX: A game-theoretic incentive-based mechanism for intrusion detection networks". *IEEE Journal on Selected Areas in Communications*, Vol. 30, No. 11, pp. 2220-2230, Dec. 2012.

APPENDICES

Appendix A

Topology Files for WSNs

Two random WSN topologies have been generated using [Cam07], a realistic topology generator. Each of them was set for a 40m×40m deployment area. The first topology includes 70 sensor nodes and has its coordinates shown in Table A.1.

Table A.1: Parameters of the 70 node WSN topology.

Node ID	**x**	**y**	**Node ID**	**x**	**y**	**Node ID**	**x**	**y**	**Node ID**	**x**	**y**
1	10	34	19	10	24	37	28	10	55	12	5
2	20	28	20	32	28	38	38	8	56	32	16
3	30	24	21	6	26	39	14	28	57	36	12
4	34	12	22	28	28	40	24	34	58	6	36
5	10	2	23	6	20	41	4	32	59	36	11
6	22	14	24	22	16	42	28	30	60	16	26
7	18	30	25	24	38	43	10	10	61	18	16
8	24	36	26	16	14	44	4	18	62	34	6
9	8	2	27	30	10	45	38	6	63	12	8
10	20	38	28	10	14	46	18	28	64	20	8
11	30	20	29	22	8	47	22	32	65	30	34
12	18	12	30	34	24	48	30	23	66	7	14
13	12	30	31	22	28	49	4	4	67	4	2
14	12	26	32	28	18	50	16	24	68	12	18
15	10	28	33	12	6	51	28	20	69	30	14
16	26	32	34	26	24	52	16	36	70	2	32
17	18	38	35	6	14	53	24	26			
18	36	30	36	20	20	54	28	26			

The second topology includes 100 sensor nodes and has its coordinates shown in Table A.2.

Table A.2: Parameters of the 100 node topology.

Node ID	**x**	**y**	**Node ID**	**x**	**y**	**Node ID**	**x**	**y**	**Node ID**	**x**	**y**
1	4	32	26	8	30	51	28	5	76	18	10
2	22	20	27	20	6	52	12	32	77	26	36
3	20	16	28	24	30	53	12	20	78	28	8
4	20	38	29	21	38	54	6	20	79	32	20
5	22	40	30	28	32	55	19	28	80	40	32
6	30	34	31	20	30	56	6	2	81	28	24
7	2	30	32	2	31	57	10	36	82	8	14
8	40	34	33	14	4	58	30	20	83	8	14
9	38	32	34	12	14	59	2	40	84	16	2
10	22	38	35	3	30	60	16	2	85	30	18
11	28	40	36	28	26	61	26	26	86	20	36
12	26	2	37	38	4	62	8	16	87	26	28
13	16	38	38	38	40	63	16	18	88	10	18
14	20	28	39	4	38	64	27	6	89	34	10
15	6	12	40	2	26	65	20	24	90	22	22
16	16	12	41	12	40	66	26	20	91	14	8
17	26	14	42	28	12	67	18	34	92	32	20
18	40	16	43	8	40	68	22	34	93	4	2
19	22	28	44	8	10	69	2	14	94	8	30
20	10	40	45	16	16	70	26	18	95	18	36
21	16	36	46	34	6	71	34	30	96	32	20
22	28	6	47	34	38	72	24	22	97	4	6
23	8	6	48	22	30	73	12	20	98	38	14
24	12	10	49	6	30	74	28	4	99	8	18
25	16	6	50	8	34	75	24	4	100	4	40

Appendix B

Calculation of Propagation Delays for the IEEE Test Systems

This thesis has adopted the procedure described in [Gon11] to calculate propagation delays for the IEEE test system topologies. It assumes an optical fiber network for power grid SCADA backbones. In [iee15], branch resistance parameters are specified. At first, the line length is calculated using the following relation.

$$\text{line length} = \frac{\text{branch resistance} \times \text{cross-sectional area}}{\text{static resistivity}} \tag{B.1}$$

The value of static resistivity for the aluminum wire is 2.50188×10^{-8} Ohm-meter. The cross-sectional area was set to 0.00080642 square meters. Finally, the propagation delay for optical fiber networks is calculated using the following relation.

$$\text{propagation delay} = \frac{\text{line length}}{\text{speed of light}} \tag{B.2}$$

Appendix C

Topology Files for the AMI WAN

The AMI topology of 57 nodes was generated based on the BUS 57 test system [iee15]. Table C.1 shows the location for each node.

Table C.1: Node location for the 57 node AMI topology

Node ID	**x**	**y**	**Node ID**	**x**	**y**	**Node ID**	**x**	**y**
1	29	17	20	32	39	39	44	45
2	37	36	21	33	16	40	31	16
3	13	15	22	21	12	41	39	4
4	3	43	23	43	19	42	28	23
5	37	9	24	3	47	43	5	14
6	33	47	25	42	48	44	10	45
7	25	13	26	41	31	45	5	9
8	26	6	27	28	6	46	44	27
9	47	4	28	49	19	47	29	4
10	21	3	29	15	31	48	37	16
11	39	40	30	24	27	49	22	5
12	41	11	31	33	24	50	34	45
13	20	17	32	22	33	51	1	26
14	13	43	33	19	21	52	1	1
15	38	1	34	12	25	53	2	3
16	4	29	35	7	5	54	44	39
17	17	18	36	27	33	55	20	37
18	29	30	37	45	5	56	17	13
19	10	39	38	26	17	57	38	49

Each link corresponds to a particular pair of nodes. Table C.2 shows the specification for each link.

Table C.2: Link specification for the 57 node AMI topology

Link ID	**from/to**	**from/to**	**Link ID**	**from/to**	**from/to**	**Link ID**	**from/to**	**from/to**
1	4	18	27	12	13	53	34	35
2	20	21	28	12	16	54	4	5
3	24	25	29	22	38	55	27	28
4	24	26	30	47	48	56	44	45
5	7	29	31	1	17	57	9	12
6	32	34	32	11	13	58	37	38
7	11	41	33	37	39	59	21	22
8	41	43	34	46	47	60	52	53
9	15	45	35	9	11	61	49	50
10	14	46	36	13	15	62	48	49
11	10	51	37	2	3	63	38	49
12	13	49	38	5	6	64	25	30
13	11	43	39	10	12	65	50	51
14	40	56	40	36	37	66	29	52
15	39	57	41	36	40	67	26	27
16	9	55	42	38	44	68	23	24
17	1	2	43	38	48	69	54	55
18	3	4	44	6	8	70	56	57
19	8	9	45	9	10	71	53	54
20	22	23	46	12	17	72	41	42
21	13	14	47	32	33	73	42	56
22	7	8	48	4	6	74	19	20
23	3	15	49	28	29	75	30	31
24	14	15	50	35	36	76	18	19
25	6	7	51	1	16	77	31	32
26	1	15	52	9	13	78	41	56

Appendix D

Topology Files for Data Center Backbone Networks

Part I The partial-mesh backbone network of 5 data centers is specified by Table D.1 and Table D.2.

Table D.1: Data center location.

Data Center	**x**	**y**
DC1	17	42
DC2	45	10
DC3	17	4
DC4	35	48
DC5	5	20

Table D.2: Backbone link specification.

Link ID	**from/to**	**from/to**
1	DC1	DC2
2	DC1	DC4
3	DC1	DC5
4	DC2	DC3
5	DC2	DC4
6	DC2	DC5
7	DC3	DC5

Part II The augmented Abilene backbone network of 10 data centers is specified by Table D.3 and Table D.4.

Table D.3: Data center location.

Data Center	**x**	**y**	**Data Center**	**x**	**y**
DC1	2	27	DC6	15	6
DC2	19	32	DC7	32	12
DC3	6	9	DC8	38	32
DC4	2	47	DC9	45	22
DC5	27	29	DC10	47	43

Table D.4: Backbone link specification.

Link ID	**from/to**	**from/to**	**Link ID**	**from/to**	**from/to**
1	DC1	DC2	8	DC5	DC8
2	DC1	DC4	9	DC6	DC7
3	DC1	DC3	10	DC7	DC8
4	DC2	DC4	11	DC7	DC9
5	DC2	DC5	12	DC8	DC10
6	DC3	DC6	13	DC9	DC10
7	DC5	DC6			

Appendix E

Calculation of Confidence Intervals

The confidence interval is the difference between two bounds namely lower and upper bounds. Such bounds are calculated as follows.

$$\text{Lower Bound} = m_{sample} - \frac{\sqrt{v}_{sample} \times \text{z-value}}{\sqrt{n}_{obs}} \tag{E.1}$$

$$\text{Upper Bound} = m_{sample} + \frac{\sqrt{v}_{sample} \times \text{z-value}}{\sqrt{n}_{obs}} \tag{E.2}$$

Where m_{sample} is the sample mean, v_{sample} is the sample variance, n_{obs} is the number of observations (sample size), and z-value is a value that corresponds to a particular confidence level [Bra87]. The sample standard deviation is $\sqrt{v}_{sample}$.

Table E.1: Two-tailed z-distribution chart

Confidence Level (%)	**z-value**
90	1.645
95	1.96
98	2.33
99	2.58

Table E.1 shows the z-distribution chart that is used to obtain z-value for a given confidence level. The experimental results presented in Chapter 7 has the sample size of 1000. This size is large enough and suitable for the z-distribution. As the two-tailed 95% confidence level is considered, the value 1.96 is set for z-value. The confidence interval of the mean is given by,

$$\text{Confidence Interval} = \text{Upper Bound} - \text{Lower Bound}. \tag{E.3}$$