# A Secure and Reliable Communication Platform for the Smart Grid

Vom Fachbereich Informatik der Technischen Universität Darmstadt genehmigte

Dissertation

zur Erlangung des akademischen Grades eines Doktor-Ingenieur (Dr.-Ing.)
vorgelegt von

M.Sc. Kubilay Demir

aus Ankara, Turkey

Referenten:
Prof. Neeraj Suri, Ph.D.
Prof. Dr. Abdelmajid Khelil

Datum der Einreichung: 04.10.2017
Datum der mündlichen Prüfung: 15.12.2017

Darmstadt 2017
D17

# Abstract

The increasing penetration of distributed power generation into the power distribution domain necessitates reliable and QoS-aware communication in order to safely manage the grid. The management of this complex cyber-physical system, called the Smart Grid (SG), requires responsive, scalable and high-bandwidth communication, which is often beyond the capabilities of the classical closed communication networks of the power grid. Consequently, the use of scalable public IP-based networks is increasingly being advocated. However, a direct consequence of the use of public networks is the exposure of the SG to varied reliability/security risks. In particular, the current Internet infrastructure does not support end-to-end (E2E) QoS-guaranteed communication. Furthermore, public networks' more open structure versus proprietary networks potentially exposes the SG to cyberattacks such as Denial-of-Service (DoS) and Distributed DoS (DDoS) which can compromise the high availability and responsiveness of the SG applications. Thus, there is need for new lightweight mechanisms that can provide both E2E communication guarantees along with strong DoS/DDoS attack protection.

To address this requirement, we first propose an overlay network based approach. This approach provides a QoS guarantee across the network with a dedicated QoS routing mechanism taking into account three parameters: reliability, latency and bandwidth for SG applications. To achieve the QoS guarantee, we also develop two additional mechanisms: (a) a multipath routing scheme that satisfies the critical applications' high reliability requirements by employing E2E physically-disjoint paths, and (b) an altruistic resource allocation scheme with the QoS routing mechanism targeting QoS-guaranteed communication for applications having strict QoS requirements.

Second, we propose a novel DDoS defense mechanism which leverages: (1) a semi-trusted P2P-based publish-subscribe (pub-sub) system providing a proactive countermeasure for DoS/DDoS attacks and secure group communications by aid of a group key management system, (2) a data diffusion mechanism that spreads the data packets over all the servers versus a single server to provide a robust protection against volume-based DDoS attacks that would affect some of the servers, and (3) a multi-homing-based fast recovery mechanism for detecting and requesting the dropped packets, thus paving the way for meeting the stringent latency requirements of SG applications.

Third, we develop a cloud-assisted DDoS attack resilient communication platform, built on the proposed defense mechanism discussed above. To prevent transport or application layer DDoS attacks, this platform implements a port hopping approach, switching the open port of a server over a function of both time and a secret (shared between authorized clients and server), thus efficiently dropping packets with invalid port number.

By leveraging the rapid-elasticity characteristic of the cloud, we can instantiate replica servers to take over the attacked servers without blocking the all traffic due to the data diffusion mechanism. Moreover, we propose a shuffling-based containment mechanism in order to quarantine malicious clients, which can mount a DDoS attack, exploiting the shared secret in a remarkably short time. Accordingly, the effect of a DDoS attack based on the compromised secret of the malicious clients is minimized.

Finally, to counter the transport and application layer DoS/DDoS attacks which are launched by compromised SG devices, we propose a proactive and robust extension of the Multipath-TCP (MPTCP) that mitigates such attacks by using a novel stream hopping MPTCP mechanism, termed MPTCP-H. Unlike the port hopping mechanism, MPTCP-H does not need a shared secret and time-sync between the clients. The proposed MPTCP-H hides the open port numbers of the connection from an attacker by renewing (over time) the subflows over new port numbers without perturbing the SG data traffic.

Our results demonstrate that both in the attack and attack-free scenarios, the proposed mechanisms provide a significant availability degree. The results also indicate a reasonable overhead in terms of additional latency and message for the proposed approaches.

# Kurzfassung

Die zunehmende Verbreitung dezentraler Stromerzeugung in der Energieverteilungsdomäne erfordert eine zuverlässige und QoS-fähige Kommunikation, um das Stromnetz sicher zu managen. Die Verwaltung dieses komplexen Cyber-physikalischen Systems, genannt Smart Grid (SG), erfordert eine reaktionsfähige, skalierbare und hochbandbreitige Kommunikation, die oft jenseits der Fähigkeiten der klassischen geschlossenen Kommunikationsnetze des Stromnetzes liegt. Der Einsatz skalierbarer öffentlicher IP-basierter Netzwerke wird daher zunehmend befürwortet. Eine unmittelbare Konsequenz der Nutzung öffentlicher Netze ist jedoch die Exposition des SG gegenüber vielfältigen Zuverlässigkeits- und Sicherheitsrisiken. Insbesondere unterstützt die aktuelle Internet-Infrastruktur keine Kommunikation mit Ende-zu-Ende (E2E) QoS-Garantien. Darüber hinaus setzt die offene Struktur öffentlicher Netzwerke im Vergleich zu proprietären Netzwerken das SG potenziell Cyberattacken wie Denial-of-Service (DoS) und verteilte DoS (DDoS) aus, welche die hohe Verfügbarkeit und schnelle Reaktionsfähigkeit der SG-Anwendungen beeinträchtigen können. Deswegen besteht ein Bedarf an neuen leichtgewichtigen Mechanismen, die sowohl Ende-zu-Ende-Kommunikationsgarantien als auch einen starken Schutz gegen DoS- und DDoS-Angriffe bieten.

Um diesen Anforderung gerecht zu werden, schlagen wir zunächst einen Overlay-Ne tzwerk-Ansatz vor, der QoS-Garantien im gesamten Netzwerk mittels eines dedizierten QoS-Routing-Mechanismus bereitstellt, wobei drei Parameter berücksichtigt werden: Zuverlässigkeit, Latenz und Bandbreite für SG-Anwendungen. Um die QoS-Garantie zu erreichen, entwickeln wir außerdem zwei zusätzliche Mechanismen: (a) ein Multipath-Routing-Schema, das die kritischen Anwendungen für ihre hohen Zuverlässigkeitsanforderungen durch Verwendung von physisch disjunkten Pfaden kompensiert und (b) eine altruistisches Ressourcenzuweisung mit dem QoS Routing-Mechanismus, der auf QoS-garantierte Kommunikation für Anwendungen mit strengen QoS-Anforderungen abzielt.

Zweitens schlagen wir einen neuartigen DDoS-Abwehr-Mechanismus vor, unter Ausschöpfung von: (1) einem bedingt vertrauenswürdigen P2P-basierten Publish-Subscribe (Pub-Sub) System, welches proaktive Gegenmaßnahmen für DoS/DDoS-Angriffe und sichere Gruppen-Kommunikation durch ein gruppenbasiertes Schlüsselverwaltungssystem beinhaltet, (2) einem Mechanismus zur Diffusion von Daten, der die Datenpakete über alle Server im Vergleich zu einem einzigen Server verbreitet, um einen robusten Schutz gegen volumenbasierte DDoS-Angriffe bereitzustellen, die einige der Server beeinträchtigen würden, und (3) einem schnellen Multi-Homing-basierten Wiederherstellungs Mechanismus zur Detektion und Wiederholung von verlorenen Paketen, womit der Weg geebnet wird für die Erfüllung der strengen Latenzanforderungen von SG-Anwendungen.

Drittens entwickeln wir eine Cloud-unterstÃijtzte Kommunikationsplattform, DDoS-

Angriffen widersteht und auf dem oben besprochenen Verteidigungsmechanismus aufbaut. Um DDoS-Angriffe auf Transport- oder Anwendungsebene zu verhindern, verwendet diese Plattform einen Port-Hopping-Ansatz. Dabei wird der offene Port des Servers als Funktion der Zeit und eines zwischen autorisierten Clients und Server geteiltes Geheimnis gewechselt und somit Pakete mit ungültiger Portnummer effizient gelöscht. Danke der reaktionsschnellen Elastizität der Cloud und unseres Datendiffusionsmechanismuses, können wir Replikatserver instanziieren, welche die Aufgaben der angegriffenen Server übernehmen ohne den gesamten Verkehr zu blockieren. Darüber hinaus schlagen wir einen Shuffling-basierten Containment Mechanismus vor, um bösartige Clients zu isolieren, die eine DDoS-Attacke unter Ausnutzung des gemeinsamen Geheimnisses in einer besonders kurzen Zeit ausführen können. Dementsprechend wird die Auswirkung eines DDoS-Angriffs, der auf dem kompromittierten Geheimnis der böswilligen Clients basiert, minimiert.

Schließlich, um DoS- und DDoS-Angriffen auf Transport- und Anwendungsschicht entgegenzuwirken, die von kompromittierten SG-Geräten gestartet werden, schlagen wir eine proaktive und robuste Erweiterung von Multipath-TCP (MPTCP) vor, die solche Angriffe mithilfe eines neuartigen Stream-Hopping-Mechanismus, MPTCP-H genannt, entschärft. MPTCP-H benötigt kein gemeinsames Geheimnis und Zeitsynchronisation zwischen den Clients im Gegensatz zu dem oben beschriebenen Port-Hopping-Mechanis mus. Das vorgeschlagene MPTCP-H verbirgt die offenen Portnummern der Verbindung vor einem Angreifer indem (im Laufe der Zeit) die Unterströme über neue Portnummern erneuert werden, ohne den SG-Datenverkehr zu stören.

Unsere Ergebnisse zeigen, dass die vorgeschlagenen Mechanismen sowohl in Szenarien mit Angriffen als auch in angriffsfreien Szenarien einen erheblichen Verfügbarkeitsgrad bieten.Die Ergebnisse zeigen auch einen angemessenen Zusatzaufwand hinsichtlich zusätzlicher Latenzzeiten und Nachrichten für die vorgeschlagenen Ansätze an.

# Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor, Prof Neeraj Suri, whose encouragement, guidance and support from the initial to the final stages of my work enabled me to develop a deep understanding of the subject. Without his guidance and persistent help this dissertation would not have been possible. I would very much like to acknowledge his willingness to support me beyond my studies, i.e., he regularly asks my family and personal life. I am also very grateful to Abdelmajid Khelil for accepting to be my external reviewer, and to Stefan Katzenbeisser, Guido Salvaneschi, and Michael Prade for being on my committee.

Then, I would like to thank all DEEDS group's members. Many thanks to Abdelmajid, Daniel, Heng, Hatem, Ahmed, Salman, Oliver, Habib, Stefan, Patrick, Tsweti, Hamza, Thorsten, Azad, Nico and my other friends. Also, a special thank you goes to Ute and Sabine for helping me with various paperwork and all other circumstances related to living in Germany.

Finally, I would like to thank my parents, Rustu and Guleser, and my sister, Serpil. Most importantly: Thank you, my love, Edibe, and my son, Muhammed Halid. Without your love, and support, I could not accomplish this.

# Contents

# List of Figures

# List of Tables

Dedicated to my whole family…

# Chapter 1

# Introduction and Problem Context

The traditional power grid is evolving into the Smart Grid (SG) to incorporate heterogeneous and geographically distributed energy sources for overall cost-effective power generation and distribution. However, the penetration of distributed power generation sources into the power distribution grid causes two-way electricity flows within the grid. This requires active management of the distribution grids, which were typically designed to support only one-way power distribution [WYB15].

In order to deal with such a complex system, utility companies employ management systems such as wide area monitoring, protection and control (WAMPAC) and advanced distribution automation (ADA) among others. WAMPAC and ADA (and other SG apps) need to acquire and deliver large amounts of data with latency needs of 100ms-5sec and availability/reliability needs of 99.00%-99.999% [WYB15; OK10]. To achieve this, the electric utility companies currently employ proprietary, simple centralized automation networks. However, these centralized networks invariably encounter scalability issues to deal with the (a) increasingly large and ad hoc SG structure, and (b) large amount of data traffic produced by the thousands of SG devices [Wei+10; Bud+10]. The SG requires a flexible and scalable network that can provide low-latency, high-availability communication. While an ideal solution would be an IP-based dedicated network, the cost-based implications result in the use of public networks, such as the Internet [YBG11].

An important caveat of inheriting the Internet's reliability risks and security vulnerabilities is that they can be exploited by hackers causing security and safety risks for not only the cyber-system but also for physical systems, e.g., electrical grids/appliances. To address these risks, many current IT (Information Technology) security and reliability approaches cannot be easily deployed in the SG automation network, since the SG's security and reliability requirements differ from classical IT systems on a number of dimensions, as follows: (1) IT security typically focuses on server-side protection versus client-side by deploying, for example, a powerful firewall on the server-side, while in the SG both client (SG generators, devices) and server-side require the same level of protection, (2) unlike the best-effort delivery mechanism of the Internet, the SG data traffic requires explicit assurance on timely delivery of information, and (3) most SG devices have constrained computational capacity unlike the abundant capacity found with IT systems [Wei+10]. A number of approaches to provide secure and reliable communication service for SG applications have been proposed [Kim+12; Nav+13; Hei+15; Cio+15; Bak+11]. However, none of the existing methods focus on providing secure and reliable communication over the public network for SG applications requiring high availability, especially in the case of intentional/unintentional failure of intermediary network devices, such as routers, or Distributed Denial-of-Service (DDoS) attacks on the SG devices.

## 1.1 Problem Statement

In this thesis, the focus is on designing a communication infrastructure that provides a reliable and secure communication service for SG applications in the IP-based public networks such as the Internet[1]. When utilizing the Internet infrastructure for SG applications, intentional or unintentional communication failures are the most serious concern due to the Internet's relatively open and less manageable structure, compared to proprietary networks. The loss of availability caused by intentional or unintentional communication failures can lead to safety risks in the power grid. The communication failures can be 1) unintentional or intentional failures in the intermediary communication devices such as routers, as detailed in the problem **(P1)**, and/or 2) intentional failures of the SG devices that are caused by DoS/DDoS attacks, i.e., volume-based DoS/DDoS attacks **(P2)**, transport/application-layer DoS/DDoS attacks **(P3)**, and internal DoS/DDoS attacks exploiting the elevated privileges of the compromised nodes **(P4)**. Within this context, we focus on the following problems, discussed below, to provide an assured availability for SG communication.

### 1.1.1 (P1) The Best-effort Delivery Service Provided by the Internet Violates the Strict QoS Requirements of the SG Applications.

While designing a communication platform for the SG using the Internet infrastructure, the first challenge is to ensure the provision of the required quality-of-service (QoS)[2] for the SG applications, since the Internet does not innately provide the necessary QoS guarantees for particularly the safety-critical applications requiring both low latency and high reliability. One reason for this deficiency is that the routing among Autonomous Systems (ASes) on the Internet[3] depends on commercial considerations, resulting from contracts among these ASes/ISPs. The contracts promote low cost links rather than low latency/high quality links. In addition, the BGP convergence time (i.e., the time needed for all routers to have a consistent view of the network after a failure) might take several minutes or sometimes up to 20 or 30 minutes. This can cause delays or even loss of SG data traffic, which can pose safety risks for the power grid [Wan+13].

Recently, some techniques [And+01; Sub+04; LK12] have been proposed to provide QoS-aware communication over the public network. The suggested approaches aim to improve the availability level of the communication system in a best-effort manner. These approaches, however, do not address the delivery guarantee for each message taking

[1] In this thesis, public networks refer to the Internet infrastructure. Therefore, they are interchangeably used throughout this thesis.

[2] In this thesis, we consider two factors for QoS, i.e., latency and availability, unless otherwise stated. Moreover, availability and reliability of SG communication network are interchangeable [OK10].

[3] BGP:Border Gateway Protocol is typically used for routing among Autonomous Systems (ASes)/Internet Service Providers(ISPs)

into account each application's availability requirement, particularly in cases where long-duration underlay failures occur. Therefore, providing a high availability, particularly for safety-critical SG applications even in the case of intentional/unintentional failures of intermediary communication devices is a serious problem that has not been addressed.

### 1.1.2 (P2) Volume-based DDoS Attacks on a Large Amount of SG Devices, Launched by Adversaries Controlling a Large Botnet.

While decreasing the cost of operation, employing public networks naturally makes the power grids vulnerable to cyber attacks. An adversary, whose aim is to render critical devices inaccessible, can mount a DDoS attack against either intermediary network devices or the SG devices directly. The former is covered by problem **(P1)**. However, the latter, particularly volume-based DDoS attacks on the SG devices in the Internet, also represents a major threat to the SG applications, considering SG applications' stringent availability and latency requirements [AP15]. IT-based cyber security solutions, e.g., firewalls and intrusion detection systems (IDS), are known to be effective in securing the IT infrastructure. However, the resource constraints (computational, memory and bandwidth) of SG devices often preclude the direct applicability of such IT solutions [Wei+10].

Moreover, as availability constitutes a safety property for SG applications (especially for control functions), deploying proactive defense mechanisms becomes indispensable for SG communication. Proactive defense mechanisms are introduced as countermeasures, e.g., [Sta+05; Jia+14] constantly move/hide the attack surface of the system to boost the cost of an successful attack for the attacker. However, since these proactive defense mechanisms are mainly designed to mitigate DDoS attacks in typical web applications requiring low availability, they are not suitable for safety critical SG applications requiring high availability and responsiveness [SK05].

### 1.1.3 (P3) Application and Transport Layer DoS/DDoS Attacks.

Transport and application layer DoS/DDoS attacks, such as DNS and HTTP flooding attacks, exploit network protocol or application vulnerabilities of the target in order to deplete the target's resources [ZJT13; Mor+11]. Whereas in this problem **(P3)** the attackers need to discover the vulnerabilities in the transport and application layers to launch such attacks, in the problem **(P2)** the adversaries require a large enough Botnet to mount volume-based DoS/DDoS attacks. An attacker that manages a large Botnet and discovers the vulnerabilities can bring down many more target devices, which can cause catastrophic damage on the power grid [AP15].

To address the application and transport layer attacks, SG communication networks need to have lightweight security mechanisms for preventive/proactive defenses to such DoS/DDoS attacks due to the constrained resources of the SG devices. The featured approaches to address such attacks are port hopping based DoS/DDoS defense mechanisms [LT04; FPT12], which periodically switch an open port of a service in a pseudorandom manner and confuse potential intruders. The port hopping mechanism facilitates both the detection and filtering of unauthenticated packets with a lower cost and does not require changes in the existing systems and protocols [LWC14]. However, these mechanisms, as in all capability-based defense mechanisms, use a key shared among each communicating party for determination of the current open port, posing a high security risk for the SG communication.

### 1.1.4 (P4) DoS/DDoS Attacks Launched by Compromised SG Devices Against the Other SG Devices.

The deployment of SG devices in a wide geographical area renders it difficult to protect them from being physically compromised [AP17]. The attackers can use the compromised SG devices inside a wide area network (WAN) to launch DoS/DDoS attacks on critical SG devices [Mor+11; AP17]. The attacker exploits the elevated privileges of the compromised device(s) that are common with target devices in WAN, such as secret keys or specific protocol/application knowledge that enable the attacker to bring down the target at low cost [AP15; Mor+11]. Since SG applications heavily depend on the availability of the communication network, such an attack originating from internal-devices and targeting some critical SG devices in WAN should be proactively mitigated.

## 1.2 Research Questions

Given the problems associated with using the public network for SG applications, we have formulated the following four research questions that guide the research presented in this thesis:

### 1.2.1 (R1) How to Maintain QoS of the SG Applications Between Each Node in the case of Failure or Degradation of Intermediary Network Devices in the Internet?

This research question **(R1)** targets problem **(P1)**. As mentioned in problem **(P1)** (Section 1.1.1), in instances of intentional/unintentional failure/degradation of intermediary network devices in the Internet, the SG applications experience unacceptable loss of the availability. However, the high path redundancy of the Internet infrastructure can be used to bypass the failed or degraded paths and to meet the QoS requirements of the SG applications. This research question **(R1)** is addressed by a substantial technical contribution **(C1)**.

### 1.2.2 (R2) How to Mitigate Volume-based DoS/DDoS Attacks over the Public Network Considering the Constrained Resource of SG Devices?

Research question **(R1)** focuses on QoS-assurance only when the failures or degradations occur in the intermediary network devices. This leaves open the question as to how to defend the SG devices against DoS/DDoS attacks that also perturb the QoS requirements.

Using the public networks increases concerns about the effect of volume-based DDoS attacks that congest the link to the target by sending a high volume of traffic. To counter such an attack before it saturates the access link of the target, proactive light-weight defense mechanisms are needed for the constrained SG devices. We address the research question **(R2)**, which targets problem **(P2)**, by the technical contribution **(C2)**.

### 1.2.3 (R3) How to Counter the Application and Transport Layer DoS/DDoS Attacks on the Constrained SG Devices?

The other DoS/DDoS related threats to the SG communication are application and transport layer DoS/DDoS attacks where low-rate forged data packets are sent to saturate the target's resources by exploiting the vulnerabilities of the target system. These attacks also can render critical SG devices inaccessible which causes instability in the power grid or power blackout. To address this issue, a defense mechanism that is light-weight and compatible with the current devices is needed. The research question **(R3)**, which targets the problem **(P3)**, is addressed by the third contribution **(C3)**.

### 1.2.4 (R4) How to Protect SG Devices from DoS/DDoS Attack by Compromised SG Devices?

As the compromised devices act similar to the normal devices and have elevated privileges, such "internal" DoS/DDoS attacks also pose a high threat to significantly damage the SG communication network and the control system of power network. Therefore, to protect SG devices from the internal DoS/DDoS attacks, a defense mechanism that requires fewer common "secrets" between SG devices for protection is needed. The contribution **(C4)** addresses this research question **(R4)** targeting problem **(P4)**.

### 1.2.5 Summary

To provide the required QoS of SG applications in the Internet in the case of intentional/ unintentional failures of intermediary communication devices **(R1)**, we propose an ov erlay-based network architecture **(C1)**. To counter the volume-based DDoS attacks in question **(R2)**, we develop a pub-sub[4]-based DoS/DDoS defense mechanism, as a second contribution **(C2)**. In addition, to address the application and transport DoS and DDoS attacks in question **(R3)**, we develop a cloud-assisted DoS/DDoS defense mechanism, built on **(C2)**, as the third contribution **(C3)**. Moreover, to counter DoS/DDoS attacks launched by compromised SG devices **(R3)** against the other SG devices by exploiting common secrets, we develop a Multipath-TCP extension as the forth contribution **(C4)**, which is used instead of UDP, employed in all the previous techniques, **(C1)**, **(C2)**, and **(C3)**.

[4] Publish-Subscribe

## 1.3 Scientific Contributions

We address the questions **(R1)**, **(R2)**, **(R3)** and **(R4)** by the following substantial contributions. First, the contributions of this thesis are detailed (Sections 1.3.1/1.3.2/1.3.3/1.3.4), and then the problems and the contributions addressing those problems are reviewed (Section 1.3.5).

### 1.3.1 (C1) Robust QoS-aware Communication in the Smart Grid

To provide the required latency and reliability of SG applications in the Internet infrastructure, we propose a novel overlay network architecture, termed HetGrid. HetGrid provides reliable and QoS-aware communication on heterogeneous[5] networks by overcoming the degraded paths that are detected by monitoring of the underlay network. HetGrid selects and employs the "strongest"[6] overlay nodes to manage inter-AS communication rather than placing dedicated servers into each domain. This mechanism only needs local underlay knowledge to enable reliable communication across the network. To provide reliable and QoS-aware communication, HetGrid also employs the following mechanisms in a self-adaptive manner: (1) Source Routing-based QoS Routing (SRQR) finds the "best" path considering bandwidth, latency, and reliability requirements of the applications. It also uses altruistic flow allocation (AFA) to reserve the "best" path for high critical applications. (2) To obtain fault-tolerant communications for high priority applications, CMR employs adequate paths for multipath routing depending on the reliability requirement of the application. The details of HetGrid approach [DGS14; DGS15] are presented in Chapter 4.

The simulation result demonstrates that even for BGP router failures or heavy Internet congestion, HetGrid provides practical QoS-satisfaction rates by employing the above mechanisms in an adaptive manner. In addition, HetGrid provides a significantly higher QoS-satisfaction rate for each application compared to direct TCP connections between pairs. Thus, HetGrid demonstrates both the feasibility of using a heterogeneous network for SG applications and an architecture to provide a robust QoS-aware communication.

### 1.3.2 (C2) A Secure and Reliable Communication Platform for the Smart Grid.

To counter the volume-based attacks on SG devices in public networks, which is not addressed by HetGrid (C1), we introduce in Chapter 5, a pub-sub-based proactive DDoS attack defense mechanism as well as its lightweight security mechanism [DS17b], called SeReCP. By taking into account the constrained SG devices resources and the security requirements of SG applications, SeReCP uses geographically dispersed pub-sub brokers to proactively counter DDoS attacks that cannot be handled by the constrained SG devices. Targeted or blindly sweeping DDoS attacks against the pub-sub brokers can easily

[5] In this work, heterogeneous networks refer to the combination of public (i.e., the Internet) and private networks. In addition, without loss of generality we interchangeably use public network, the Internet and heterogeneous network.

[6] In terms of computational power, memory, bandwidth, and multihoming feature

render some of the critical devices inaccessible and pose safety risks. To address this issue, SeReCP also employs a data diffusion approach which enables spreading of the consecutive data packets across the pub-sub brokers by using a token-based stateless authentication mechanism.

In addition, we develop a multihoming-based fast "recovery" mechanism for SeReCP to meet the stringent availability and latency requirements of SG applications in the case of several pub-sub brokers being simultaneously attacked. This mechanism allows each publisher to transmit every two consecutive data packets to two different network interfaces[7] of each pub-sub broker during the diffusion of the publication data over all brokers. If one of the network interfaces of any broker is under attack, that broker can request a missing packet after a short waiting time using the remaining functional network interface[8]. This allows for fast packet "recovery" compared to classical ACK-based mechanisms such as TCP's cumulative ACK. On the other hand, to protect end-to-end (E2E) confidentiality and integrity of the data, we propose a group key management system. This system provides role-based access rights for both publisher and subscriber in addition to protection from replay attacks.

To assess the effectiveness of our approach against DDoS attacks in the Internet infrastructure, we employed the NorNet Testbed [Dre15] that contains multihomed nodes spread over all of Norway. The results show that SeReCP introduces an acceptably low latency overhead of 40 ms for the SG applications tolerating maximum 200ms [OK10]. We compare our system with the reference work of Angelos et al. [SK05], which also employs data diffusing mechanisms for real-time applications. [SK05] shows stable performance for up to 5% of pub-sub brokers being attacked. Once over 5% of the brokers have been attacked, the TCP connection stalls due to the large amount of dropped packets. In contrast, SeReCP shows stable performance up to 30% of pub-sub nodes being compromised. Overall, these results evidence that SeReCP provides the required security for SG applications during volume-based attacks in the public networks.

### 1.3.3 (C3) Securing the Cloud-Assisted Smart Grid

To provide protection from application and transport layer DoS/DDoS attacks for SG communication on SeReCP **(C2)**, we develop a new cloud-assisted defense mechanism [DS17a], built upon SeReCP. To this end, we first propose a hybrid hierarchical cloud-extension concept (HHCEC), which is a SG-relevant cloud-assisted architecture. HHCEC provides ultra-high responsiveness and security with its (a) hybrid and geographically dispersed structure, and (b) specialized broker-based publish-subscribe communication system. Second, we propose a novel approach termed Port Hopping Spread Spectrum (PHSS), which acts as a strong defense against transport and application layer DoS/DDoS attacks, as well as the volume-based DoS/DDoS attacks, against the broker servers. PHSS

[7] For simplicity, we explain the idea using a broker with two network interfaces card (NIC). However, SeReCP transmits the consecutive data depending on the number of NICs of each broker

[8] In this work we consider an Akamai-sized pub-sub broker network (∼2500) [SK05]. Each broker in this network has multiple IP addresses/NICs connected to different carriers (multihomed). Whereas the IP addresses of brokers are publicly known, information about the relationships between IP-addresses and their corresponding brokers is confidential. In such a network, it is very unlikely that all NICs of any one broker will be simultaneously compromised by an attacker.

is equipped with two distinctive features: (1) *port hopping*, changing the open port of the broker server as a function of the time and a secret shared between the broker server and the publishers[9], and (2) *packet spreading*, diffusing consecutive data packets over a number of broker servers in the cloud versus a single broker server. This data spreading approach enables PHSS to instantiate replica broker servers to take over the attacked broker servers without blocking all traffic by taking advantage of the cloud's rapid-elasticity.

Lastly, to minimize the impact of a probable compromising the secret, we introduce (1) *a token-based authentication method* that allows for a light-weight periodic transmission of the secret to each client (publisher), and (2) *a shuffling-based containment mechanism* that quarantines malicious clients, without rendering the broker server inaccessible. To do this, the containment mechanism repositions/shuffles the clients over the ports of the broker server with a negligible overhead.

To assess the efficiency of the proposed approach, we construct a proof-of-concept prototype using EC2-micro instances [Ama16] and the PlanetLab (http://planet-lab.org) test-bed. We evaluate PHSS's effectiveness in providing network availability by using the *shuffling-based containment mechanism* against DDoS attacks that use the compromised secret. We also compare our approach with the public key-based re-keying method used by the existing port hopping mechanisms [LT04]. Our results show that by containing the impact of the DDoS attack in a notably shorter time period, PHSS provides a high network availability of over 98% during the attack versus the typical 60% availability achieved by the public key-based re-keying method. Furthermore, the experimental results show that our proposed mechanism causes neither significant throughput degradation (i.e. <0.01% throughput degradation) nor additional latency. We detail the contribution **(C3)** in Chapter 6.

### 1.3.4 (C4) Towards DDoS Attack Resilient Wide Area Monitoring Systems.

To counter DoS/DDoS attacks on the SG devices, which are launched by compromised other SG devices, we extend upon the Multipath-TCP (MPTCP) approach for SG applications using long-lived connection such as the phasor measurement applications. The basic MPTCP provides long-duration communication connections [Paa+14] and provides reactive mitigation against attacks with its diverse multi-path functionality. However, in order to achieve proactive and robust protection of the transport and application layer DoS/DDoS attacks as in PHSS, we introduce a novel stream hopping mechanism, termed MPTCP-H, that is directly integrated into MPTCP. MPTCP-H does not need a shared secret and time-sync between the clients unlike the port hopping mechanism of PHSS **(C3)**, thus avoiding the secret key disclosure employed by the compromised SG devices. The proposed hopping mechanism hides open port numbers by refreshing sub-flows over time with new port numbers, without causing data traffic interruptions. This approach of hiding port numbers is shown to provide high coverage against transport and application layer DoS/DDoS attacks. The results from MPTCP-H demonstrate that the proposed

[9] The terms client/publisher and server/broker are used interchangeably in the rest of the thesis. In addition, while every SG device/application server can be a publisher and/or a subscriber, the brokers are dedicated servers for their respective roles.

approach indeed secures the system with minimal additional latency and message overhead. The details regarding MPTCP-H mechanism [DS17c] are introduced in Chapter 7.

### 1.3.5 Overview

In this thesis the main problem of focus is the loss of availability of SG systems using the public network in the case of unintentional and intentional communication failures. We consider two different types of failures 1) unintentional or intentional failures in the intermediary communication devices such as routers, and/or 2) intentional failures of the SG devices that are caused by DoS/DDoS attacks. For the second type of failure we consider three different types of DoS/DDoS attacks that can compromise the SG communications, i.e., volume-based DoS/DDoS attacks, transport/application-layer DoS/DDoS attacks, and internal DoS/DDoS attacks exploiting the elevated privileges of the compromised nodes.

To address the first failure type, we introduce HetGrid **(C1)**, providing QoS assurance for SG applications over the public network. To mitigate the second type of failure (DoS/DDoS attacks) we propose three mechanisms i.e., SeReCP **(C2)**, (PHSS + HHCEC) **(C3)** and MPTCP-H **(C4)**, to target all three aforementioned types of attack.

The mechanisms proposed in this thesis are integrated with each other to provide high availability assurance for the SG against communication failures. For example, HHCEC + PHSS **(C3)** is developed by enhancing SeReCP **(C2)** to address the application and transport layer DoS/DDoS attacks in addition to the volume-based attacks by redesigning the system of SeReCP **(C2)** using cloud assistance. In our integrated system, it can be shown that using HHCEC + PHSS **(C3)** versus SeReCP **(C2)** provides better protection against the DoS/DDoS attacks. However, to address the authorization issues between the publishers and brokers/subscribers, we also employ the scheme of SeReCP, which utilizes a token-based authentication scheme between publishers and brokers to provide a "stateless" connection and a strong replay attack protection. In addition, for E2E integrity, we also deploy the cryptography-based group key management system of SeReCP **(C2)**, which enables the application of role-based access rights for both publishers and subscribers as well as the protection from replay attacks. Furthermore, HetGrid **(C1)** is employed, which enables the SG communication system to benefit from the path redundancy in the Internet by using an overlay network in order to assure the transmission of messages with ultra-high availability and ultra-low latency between the publisher and the brokers of SeReCP **(C2)** or HHCEC **(C3)**.

Finally, **(C4)** replaces UDP in the other proposed mechanisms, to mitigate the secret key discloser-based internal DoS/DDoS attacks which are caused by a vulnerability in the port hopping mechanism of PHSS **(C3)**.

Overall, by building a system using the proposed mechanisms in this thesis, we derive a communication platform that provides secure and reliable communication for SG applications over a public network.

## 1.4 Publications Resulting from the Thesis

- Demir, K., Germanus, D., & Suri, N. (2014) "Robust and Real-time Communication on Heterogeneous Networks for Smart Distribution Grid" In Proc. of IEEE International Conference on Smart Grid Communications (SmartGridComm), pp. 386-391.
- Demir, K., Germanus, D., & Suri, N. (2017) "Robust QoS-aware communication in the smart distribution grid" In Journal of Peer-to-Peer Networking and Applications, Springer, vol. 10, no. 1, pp.193-207.
- Demir, K., & Suri, N. (2017) "SeReCP: A Secure and Reliable Communication Platform for the Smart Grid" In Proc. of IEEE Pacific Rim International Symposium on Dependable Computing (PRDC), pp.175-184.
- Demir, K., & Suri, N. (2017) "Towards DDoS Attack Resilient Wide Area Monitoring Systems" In Proc. of Security in Critical Infrastructures (S-CI) Workshop @ARES Article No: 99.
- Demir, K., Ismail, H., Vateva-Gurova, T., & Suri, N. (2017) "Securing the Cloud-Assisted Smart Grid" In International Journal of Critical Infrastructure Protection, Elsevier, submitted.
- Demir, K., Nayyer, F., & Suri, N. (2017) "A Defense mechanism against DoS attacks in Phasor Measurement Traffic" In Journal of Technology and Economics of Smart Grids and Sustainable Energy, Springer, submitted.
- Demir, K., Nayyer, F., & Suri, N. (2017) "MPTCP-H: A DDoS Attack Resilient Transport Protocol to secure Wide Area Measurement Systems" In International Journal of Critical Infrastructure Protection, Elsevier, submitted.

## 1.5 Thesis Structure

To begin, the state of the art, with respect to the work presented in the thesis, is discussed (Chapter 2). We further detail a secure and reliable platform system model along with our attack model (Chapter 3). As contribution **(C1)**, in Chapter 4 we introduce a novel overlay network, providing reliable communication for the SG in the Internet. Chapter 5 presents a secure and reliable platform for the SG, which corresponds to our second technical contribution **(C2)**. Subsequently, in Chapter 6, we introduce a defense mechanism for the cloud-assisted SG, as contribution **(C3)**. In Chapter 7, we propose a DDoS attack resilient transport protocol for WAMS, as contribution **(C4)**. Finally, the summary and conclusion of the thesis is presented in Chapter 8.

# Chapter 2

# Related Work

In this chapter we discuss the state of the art approaches related to the work in this thesis by categorizing them into the following relevant groups:

- Approaches providing QoS-aware communication (Section 2.1)
- Proactive DoS/DDoS defense approaches (Section 2.2)
- Cloud computing approaches for SG (Section 2.3)

## 2.1 QoS-aware Communication

The existing approaches providing QoS-aware communication for the SG span two distinct subjects fields: (i) Reliable and QoS-aware communication systems specific for SG (Section 2.1.1), and (ii) Systems providing reliable and QoS-aware communication for web applications (Section 2.1.2).

### 2.1.1 Resilient and QoS-aware communication systems for smart grid

There are limited works proposing new QoS techniques specific for SG applications [Riz+14; Kho+13; Ali+13], which utilize various features of different networking technologies, such as Multi-protocol Label Switching (MPLS) and VLAN. However, the proposed techniques can be functional in the case where the SG utilizes a proprietary network that is constituted from those networking technologies.

Recently, many middleware/overlay-based approaches simplifying application development on a variety of platforms, operating systems, networking technologies have been proposed to provide reliable SG operation on the heterogeneous networks and systems. We discuss the most prominent ones below.

The INTEGRIS [Nav+13] project proposes a novel information and communication technologies (ICT) infrastructure based on mixing heterogeneous OSI layer 2 technologies (PLC, wireless, etc.) integrated through a middleware. It suggests the use of a QoS broker device to enhance the QoS in SG by employing a centralized QoS management. Since they offer a QoS management mechanism for a dedicated heterogeneous network, this proposal can be implemented for only utility owned communication networks.

The GridStat [Bak+11] project proposes a middleware framework which manages network resources to timely and reliably deliver the messages acquired anywhere on the network and transmitted to multiple other points. In addition, GridStat utilizes a pub-sub network of message routers controlled by a hierarchical QoS management plane to satisfy the NASPInet QoS requirements. However, GridStat assumes that it receives certain QoS guarantees from the underlay network and the underlay network topology is fully known. GridStat also uses static routing to avoid the overhead of dynamic link-state-based routing. This proposal cannot obtain end-to-end (E2E) guaranteed delivery in the use of public carriers since it is not designed to use the best-effort Internet infrastructure and rather it requires dedicated networks.

The SmartC2Net [Cio+15] project aims to develop, implement and validate robust solutions that facilitate Smart Grid operation on top of heterogeneous off-the-shelf communication infrastructures with diverse properties. The functions of the proposed new middleware include: (1) adaptive network and grid monitoring, (2) control methodologies for communication network configurations and QoS settings, and (3) models of the extended information and procedures for adaptive information management. SUNSEED [Ste+14] proposes an evolutionary approach to usage of the existing communication networks from both energy and telecom operators by improving their robustness/reliability. The project proposes an exposed application programming interfaces (API) based on open standards (W3C) to enable third-party creation of new businesses associated with energy and communication sectors (e.g., virtual power plant operators). Although these two projects do not propose any mechanisms that assure the required QoS provisioning of critical SG applications, they introduce promising network monitoring and interoperability mechanisms for the SG utilizing public networks.

Albano, et al. [Alb+15] review varied categories of communication middleware focusing on message oriented middleware (MOM). They particularly address data distribution services (DDS) targeting distributed real-time systems (for smart grid applications) with complex distributed applications, where prioritization requirements have to be assured. Furthermore, Prodejev, et al. [Pre+14] devise a working architecture that relies on the ETSI M2M components (upgraded by CoAP and Websockets), and is mapped to the Smart Grid. The authors analyze whether the heterogeneous solution is able to meet the communication requirements of the diverse Smart Grid applications. Due to the lack of underlay topology-awareness of these approaches, the critical application's reliability and latency requirements cannot be met by these approaches alone.

SeDAX [Kim+12] proposes a data-centric communication method on a secure overlay network on top of the existing TCP/IP network. This method provides good routing performance and self-configurable group communication. However, it is inapplicable for real-time applications, e.g, ADA, and distributed generation, since it does not guarantee E2E latency.

Deconinck, et al. [Dec+10] propose a dependable infrastructure for autonomous decentralized microgrid control. It enables power devices to interact over a self-organized and semantic peer to peer overlay network on top of the existing TCP/IP network, called Agora. Since this work concentrates on non-time-critical applications (secondary and tertiary voltage control), it cannot cope with the strict timeliness requirements of SG applications.

### 2.1.2 Systems Providing Reliable and QoS-aware Communication for Web Applications

Although there have been advances in the QoS provisioning in network-level approaches, models such as DiffServ [Bla+98], IntServ [Wro97] and MPLS [RVC00] are still far from deployment across the Internet due to the changing requirements in the networking infrastructure or the configuration differences among the domains. Although MPLS/VPN [EBB08] is introduced as a QoS-guaranteed communication protocol, its QoS-guarantee does not guarantee *inter*-AS connections, only *within* AS connections.

As the Internet is increasingly used for mission critical applications, connection reliability and latency are becoming significant challenges. To address these challenges, service overlay networks managed by third party providers are advocated. The providers target to offer QoS-guaranteed service for multiple applications and clients on the Internet, as proposed in RON [And+01], OverQoS [Sub+04], and NGSON [LK12]. RON and NGSON are well-defined, recognized service overlay network approaches. They provide reliable and timely communication on wide area networks for distributed applications. However, they do not offer timely delivery guarantee per message for safety-mission critical applications, e.g., the islanding protection in SG. In addition, no adaptive QoS and reliability mechanisms depending on application criticality are introduced in those proposals. For safety critical applications, even short-lived failures of the Internet infrastructure can pose a significant risk of damage on the grid. As a potential solution to these problems, Han et al. [HWJ08] propose a topology-aware overlay framework to maximize path independence for better availability and performance of E2E communication in the Internet. They do not introduce any traffic prioritization or resource allocation mechanisms in their work. Yang et al. [Yan+09] propose a message-oriented middleware with QoS awareness which provides QoS assurance depending on the requirements of each application. Their proposed middleware [Yan+09] , however, has the following shortcomings with regards to the stringent QoS requirements of SG applications: 1) a lack of fault tolerance due to the assumption of no failure in the local brokers, and 2) no assured resource allocation for the traffic of the critical application, although a simple delay-based QoS mechanism is in place.

Other than [DGS15; DGS14], all of the above works lack at least one of the following criteria: (i) high fault tolerance, (ii) scalability, (iii) adaptive QoS management, or (iv) support for the heterogeneous network, as depicted in Table 2.1.

## 2.2 DoS/DDoS Resilient Communication

The existing techniques providing DoS/DDoS resilient communication fall in two main categories: (i) Secure and reliable communication for the SG (Section 2.2.1), and (ii) Proactive DDoS defense for web applications (Section 2.2.2).

### 2.2.1 Secure and Reliable Communication for the SG

Some existing IDSes developed for SG [Zha+11; BS11] aim to detect DoS/DDoS attacks and then subsequently investigate them. These approaches introduce promising features

TABLE 2.1: Existing Works Comparison regarding QoS Requirements of SG Applications

| **Requirements/ Existing Works** | GridStad | Integris | MPLS-VPN | ATM | RON | NGSON | HetGrid [DGS14; DGS15] |
|---|---|---|---|---|---|---|---|
| High Fault Tolerance | Y | N | Y | N | P | P | **Y** |
| Scalability | Y | P | Y | N | N | Y | **Y** |
| Adaptive QoS/Reliability management | N | N | P | N | P | N | **Y** |
| Heterogeneous Network | P | N | N | Y | Y | Y | **Y** |
| **Symbols** | | | **N: No, Y: Yes, P: Partially** | | | | |

to incorporate with the technique [DS17c] proposed in this thesis that is focused on mitigating the internal-attacks mounted by the compromised devices. Karthikeyan et al. [Kar14] employ three methods, i.e., Marking Scheme, TTL Value analysis and MAC value analysis, to detect and isolate DDoS attacks in routers of the SG network. While the aforementioned approaches requires a proprietary network for their deployment, there are also some approaches that do not need (or only need to some extent) a dedicated network, as follows.

GridStat [Bak+11] proposes a pub-sub network of message routers controlled by a hierarchical management plane to meet the NASPInet's QoS and security requirements. In addition to its lack of QoS assurance on the public network, GridStat does not provide a DoS/DDoS resilience particularly for internal attacks exploiting the elevated privileges, which poses a security risk for the particularly critical SG application. SeDAX [Kim+12] proposes a data-centric communication method on a secure overlay network. This approach involves trusted authentication servers allowing parties to periodically obtain topic-based group keys to assure E2E confidentiality and integrity. While SeDAX provides secure and efficient communication for SG applications allowing latency-tolerance and relatively low availability, it does not support SG applications requiring low latency and high availability due to its message passing technique and lack of a mechanism providing DoS/DDoS attack resilience. SmartC2Net [Cio+15] aims to develop resilient solutions that facilitate SG operations on top of heterogeneous off-the-shelf communication infrastructures. C-DAX [Hei+15] employs a pub-sub paradigm to decouple communication parties in space, time, and synchronization. C-DAX enables topic access control, end-to-end integrity and end-to-end confidentiality of data, and authentication of nodes. Despite their lack of countermeasures for DoS/DDoS attacks, SmartC2Net and C-DAX offer promising features which can be incorporated with our work to provide secure and reliable communication.

### 2.2.2 Proactive DDoS Attack Defense

For security of IT infrastructures, traditional security solutions, e.g, firewalls, intrusion detection systems (IDS), or Virtual Private Networks (VPN), are both widespread and effective. However, since the SG devices typically have constrained computational, bandwidth, and memory resources, the direct use of these traditional security mechanisms is often not possible [DS17b; ZJT13]. Hence, for providing the required security for SG communication systems, security solutions that proactively counter the attacks should be implemented. Within this context, we develop our approaches based on the following proactive techniques.

Overlay networks can offer an Internet-wide network of nodes to create a first-level firewall that proactively counters DDoS attacks. In this scenario, the requests first need to pass through the nodes of Overlay Network before getting to the target server. [Sta+05; Jia+14; KMR02; NSS10] are overlay-based proactive DDoS attack defense mechanisms which aim at hiding or moving the position of the application sites to prevent DDoS attacks based on available information about their locations. Secure Overlay Services (SOS) [KMR02] architecture consists of a three-layer hierarchy of overlay nodes to control access to the protected target server. The goal is to ensure that any client can find a path to the target server under DDoS attacks; maintaining a small probability of compromising all available paths between clients and the target server. Although SOS can protect against blind DDoS attacks, it is ineffective against sophisticated and targeted DDoS attacks on a given overlay network. Such attacks can disturb latency-sensitive applications communicating over the attacked overlay node until the connection is established over a non-attacked overlay node. SIEVE [FP12] offers a lightweight distributed filtering protocol that intends to expand the filtering and receiving capacity of the protected target. In this architecture, the server needs to provide some kind of secret to the client that can help it to pass through the filter. Since SIEVE isolates the protected server in IP level by deploying it in a private network in order to protect the server from direct flooding attacks, it is not deployable in a network that contains large amount critical nodes/servers spread over a large-scale geographical area.

An overlay-based target hiding technique aiming at addressing the shortcomings of the technique in [KMR02] is proposed in [SK05], where the authors propose to spread the duplicated data packets across the overlay nodes between the client and the target. This ensures robust protection against targeted DDoS attacks that bring down some of the overlay nodes at the expense of latency and packet overheads.

Further examples of moving/hiding target defense are port and address hopping techniques. Lee et al. [LT04] present a random port hopping (RPH) technique where the server uses time-varying UDP/TCP port number as well as a shared secret among the server and clients. Fu et al. [FPT12] state that the RPH in [LT04] undergoes time differences due to the local clock drift. In order to address the time-synchronization issue in [LT04], Fu et al. [FPT12] propose two algorithms, BiGWheel and HoPerAA, which enable the RPH for multiple servers and clients in the presence of clock-drift. In this approach, the secret is used by the clients without a restricted time duration, which poses the risk of compromising the secret. With a compromised secret, the communication will be interrupted for a certain amount of time duration because of the direct attack against the ports.

The time synchronization issue is also addressed by [BHK07a] through an acknowledgment based port hoping strategy. In cases where the acknowledgment packet is lost in the network, this arrangement can force the two sides to communicate on a common port for a longer time period. This enables the attacker to obtain the port number to start a directed attack and disrupt communication. Moreover, this approach may not be a practical scheme for communication when there are multiple users.

A shuffling-based moving target defense mechanism is proposed by [Jia+14] to reduce the level of large-scale DDoS attacks with the help of cloud computing properties. By replacing attacked servers with newly instantiated replica servers and optimally shuffling client-to-server assignments, their solution can gradually isolate DDoS attacks on network and computation resources, and restore quality of service for benign-but-affected clients. This method is actually a reactive method and not convenient for applications requiring high availability.

Based on the above discussion of the related work, the techniques [DS17b; DS17a; DS17c] proposed in this thesis are developed by addressing the existing works' shortcomings concerning the SG security threats and requirements associated with our network model, as shown in table 2.2.

## 2.3 Cloud Computing for th SG

Multiple features of cloud computing, such as on-demand service, flexibility, pay-for-use and instant network access, are continuously attracting the attention of researchers working on the system development for potential future power grids [BMR15].

GridCloud [BI14] was proposed in order to design a prototype and present a well-defined software platform with the aim of meeting the requirements of the future power grid in the cloud. GridCloud develops a cloud architectural model for monitoring, managing and controlling the power systems, which is achieved by integrating some of the technologies such as GridStat, Isis 2, TCP-R and GridSim [BI14].

A contemporary approach for power system frequency monitoring system (FNET) [Zha+10] is proposed as a wide-area monitoring system. The main architecture of FNET includes a broadly deployed network of frequency disturbance recorders (FDR) which returns phasor readings to either a local central point or a remote data center with Ethernet. Handling the data of the FNET application with diverse configuration requirements (number of CPU, memory, etc.) by using in-house infrastructures is not a cost-effective solution for the power grid. Rather, leveraging the cloud computation for the FNET applications would be the most feasible solution [BI14].

[Mah+13] proposed a framework, Grid-Cloud, which enables PMU-based state estimation applications on a cloud infrastructure. To identify the limitations of the current standard cloud infrastructures, the authors carry out a real-world implementation using the Red-Cloud and PlanetLab infrastructures [Mah+13]. Based on the results, the authors infer that a best effort state estimation can be fulfilled when using the in-time data. Otherwise, the outdated data can be used for historical analysis.

TABLE 2.2: Existing Works Comparison regarding Security Requirements of SG Applications

| **Requirements/ Existing Works** | GridStad | SeDAX | C-DAX | SOS | Cloud-based e.g., Akamai [NSS10] | Angelos at al. [SK05] | SeReCP [DS17b] | PHSS [DS17a] |
|---|---|---|---|---|---|---|---|---|
| Scalability | Y | Y | N | N | N | N | **Y** | **Y** |
| Key Management | N | N | Y | N | N | Y | **Y** | **Y** |
| DDoS Defense (volume-based) | N | N | N | Y | Y | Y | **Y** | **Y** |
| DDoS Defense (transport and application layer) | N | N | N | P | P | N | **N** | **Y** |
| Targeted or sophisticate DoS/DDoS attacks | N | N | N | N | N | N | **Y** | **Y** |
| **Symbols** | **N: No, Y: Yes, P: Partially** | | | | | | | |

[Bae+15] introduced a smart-frame, which consists of three hierarchical levels, i.e., top, regional and end user, for the SG applications based on cloud computing. This framework is designed to provide scalable, flexible and secure information management for those applications. In addition, to address information security issues, a security solution based on identity-based encryption and signature, and identity-based proxy re-encryption are proposed.

The aforementioned existing work provides the basic inspiration behind the SG-relevant cloud-assisted architecture, HHCEC, proposed in this thesis (cf. Chapter 6, [DS17a]). HHCEC, however, is a dispersed and hybrid design architecture focused on providing secure and high responsiveness for the SG applications.

# Chapter 3

# System & Threat Model

In this section, we introduce the basic system and threat models that are common across the proposed approaches in this thesis. The differences and details are discussed in the corresponding chapters.

## 3.1 System Model

Similar to contemporary SG models, we consider SG communication nodes (e.g., intelligent electrical devices (IED), substations and control centers) to span a large geographical area and connect to the wide area network (WAN) of the utility. The WAN is built using both public (ISPes) and private networks taking into account availability requirements of the applications and their relative cost-effectiveness. Hence, the SG communication network comprises many Autonomous Systems (AS) and domains. In addition, some SG communication nodes such as transmission/distribution substations and control centers are multihomed, i.e., multiple ISP connections, and have direct fiber optic links between them. As a result, the SG communication network is considered to be a heterogeneous network.

In this section, we present the underlay and overlay models that are foundational to the development of HetGrid **(C1)**, which provides a base for the other contributions in this thesis, i.e., **(C2)**, **(C3)**, and **(C4)**. The system models of the other contributions are also gradually introduced throughout this section.

### 3.1.1 Underlay Model

We model the underlay topology, which corresponds to the SG communication network, as a directed graph $G_u = (V_u, E_u)$ where $V_u$ and $E_u$ are the set of vertices and edges. The vertices refer to ASes, or private domains, and the edges represent the peerings between them. Although there are many internal routers inside an AS and a private domain, we consider them as underlay nodes to simplify the route calculation between pairs.

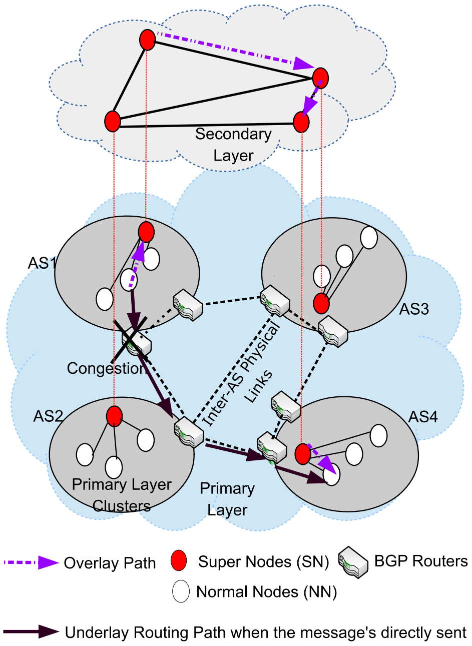


FIGURE 3.1: Basic HetGrid Architecture

### 3.1.2 Overlay Model

To pave the way to obtaining E2E physically-disjoint paths, containing no common underlay router and overlay node, and providing QoS-aware routing in a lightweight manner, the bootstrap node[1] clusters the communication nodes depending on their autonomous system (AS) and domains and selects the nodes with the highest computational capacity from each cluster as a supernode (SN) (a master SN and $d-1$ redundant SN, cf. Chapter 4). This results in a two layer overlay for HetGrid as illustrated in Fig. 3.1.

We define a primary layer overlay as a directed graph $G_p = (V_p, E_p)$. Vertices refer to all overlay nodes (including SNs and normal nodes (NNs), which are SG communication nodes (e.g., IEDs)). Edge set $E_p$ represents virtual links between NNs and their SNs (each NN is connected to only SNs in the same cluster).

A secondary layer is defined as a directed graph $G_s = (V_s, E_s)$. Vertices in $V_s$ only consist of SNs that participate in both layers, i.e., SN is a gateway, and therefore, $V_s \subset V_p$. In addition, if a physical link (the peering) $e_u \in E_u$ exists between two ASes or the private domains, there exists a secondary layer edge $e_s \in E_s$.

To support end-to-end (E2E) QoS-guaranteed communication in a heterogeneous network that includes the Internet, HetGrid is equipped with a dedicated QoS routing mechanism. This mechanism provides QoS guarantees across the network, taking into account three parameters: reliability, latency and bandwidth for SG applications. To achieve this, we develop two elements, namely (a) a multipath routing mechanism compensating the critical applications for their high reliability requirements by employing E2E physically-disjoint paths, and (b) an altruistic resource allocation with the QoS routing

[1] A node in the overlay network that provides initial configuration information to newly joining nodes

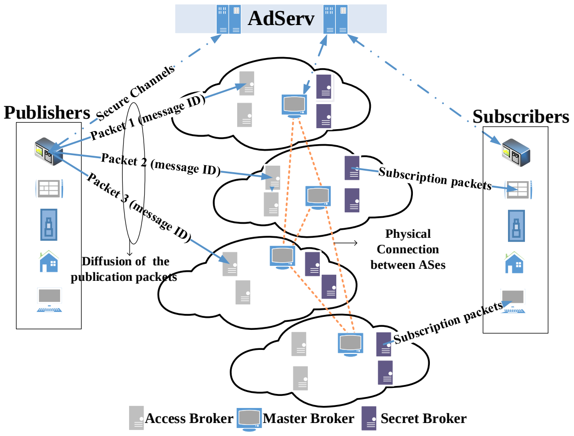


FIGURE 3.2: Basic Architecture of SeReCP

mechanism targeting QoS-guaranteed communication for applications having strict QoS requirements. The details of HetGrid are presented in Chapter 4.

### 3.1.3 Model Assumptions

In this work, the link state between two overlay nodes $(u, v)$ is denoted by bandwidth $B_{uv}$, latency $L_{uv}$, and reliability (simply loss rate) $R_{uv}$. We assume that each primary layer node $p$ regularly derives its available computation capacity, $C_p$, and link state information between itself and its SNs and transmits them to its SNs. On the other hand, each secondary layer node $u$ obtains $C_u$ and its adjacent links' available link state information, and disseminates them to all $V_s$ (SNs). We consider that the bootstrap node broadcasts the updated membership list over the secondary layer, only upon change of membership of the clusters. We expect that the overlay network has a low churn rate given the operational characteristics of the SG communication nodes. Moreover, when a NN sends a packet to its SN in order to deliver the packet to its destination over QoS-satisfied path(s), the SN employs a Bloom Filter model to find the SN to which the destination node belongs (inspired by [Zha+02]).

### 3.1.4 Pub-Sub Model

To counter volume-based DoS/DDoS attacks against SG devices, we propose a pub-sub-based defense mechanism, called SeReCP. We employ a broker (P2P)-based pub-sub system, as pub-sub systems inherently provide scalability and proactive DDoS attack defense, particularly for constrained devices. The broker (P2P)-based pub-sub system of SeReCP selects the ”strongest” nodes (in terms of computation capacity, multihoming feature and trust-level), as brokers. These brokers are clustered depending on their autonomous system (AS) and geographical proximity, as illustrated in Fig. 3.2. By doing so, we aim at overlapping the brokers from SeReCP and $SNs$ of HetGrid. SeReCP is devised for a messaging paradigm where, upon reception of a publication from a publisher, the

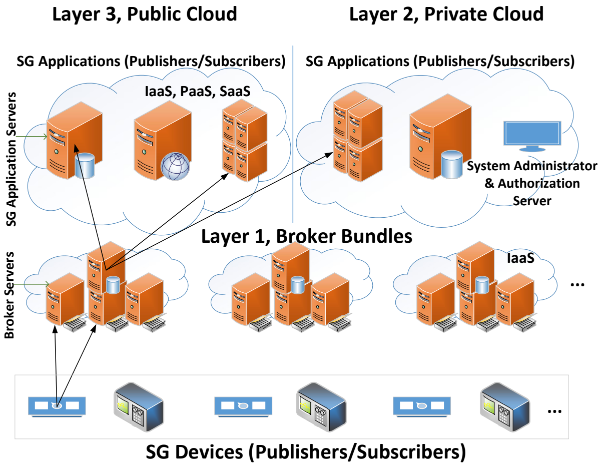

Layer 3, Public Cloud
Layer 2, Private Cloud
SG Applications (Publishers/Subscribers)
SG Applications (Publishers/Subscribers)
IaaS, PaaS, SaaS
SG Application Servers
System Administrator & Authorization Server
Layer 1, Broker Bundles
IaaS
Broker Servers
...
...
SG Devices (Publishers/Subscribers)

FIGURE 3.3: Hybrid Hierarchical Cloud Concept (HHCEC)

broker transmits the data to brokers responsible for delivering the data to the corresponding subscribers. Moreover, to provide QoS assurance between the brokers or between the brokers and publishers/subscribers of SeReCP, we utilize the HetGrid, which provides a scalable QoS-aware routing.

The main role of SeReCP in combating DDoS attacks is to distinguish between authorized and unauthorized traffic in the brokers and then to either enable the traffic to access to the destination (subscriber) or to drop/filter it, respectively. Thus, the system provides the functionality of a firewall scattered over the wide area network to prevent any congested link to the target(s). We detail the defense mechanism of SeReCP in Chapter 5.

### 3.1.5 Cloud-assisted Pub-Sub Model

To manage millions of SG devices and to handle large amounts of data in a reliable, scalable, and cost-effective way, the SG utilities increasingly extend their communication-based management system to the (herein advocated) cloud computing platforms to enable reliable and on-demand access to varied computing resources [BI14]. Therefore, we propose a hybrid hierarchical cloud-extension concept (HHCEC), which is a SG-relevant cloud-assisted architecture. HHCEC provides ultra-high responsiveness and security with its (a) hybrid and geographically dispersed structure, and (b) specialized broker-based publish-subscribe communication system. Providing the specific SG requirements is the driver behind proposing a 3-layer HHCEC cloud-assisted architecture, as depicted in Fig. 3.3. *The first layer* is composed of *Broker Bundles*, which are dispersed based on the grid topology throughout the utility territory. *The second layer* is an in-house private cloud infrastructure comprised of application servers that process data requiring high availability and/or confidentiality. *The third layer* consists of public cloud infrastructure(s). This layer communicates and shares corresponding data with third parties.

In HHCEC, a system administrator, that considers the geographical distance and the latency between the brokers on the cloud and publishers, assigns each publisher to a broker bundle. Furthermore, the system administrator monitors/maintains the latency between the broker bundles and the publishers/subscribers to re-assign the publishers/subscribers to a new broker bundle in cases where intolerable latency is detected. In order to transmit the messages between the brokers and the publishers/subscribers while preserving the ultra-high availability and ultra-low latency requirements, the overlay of HetGrid is employed as in SeReCP. This enables the SG communication system to benefit from the path redundancy in the Internet, i.e., by using the overlay network in addition to providing QoS-aware routing.

In addition, we propose a novel approach termed Port Hopping Spread Spectrum (PHSS), which acts as a strong defense against transport and application layer DDoS attacks, as well as high-volume DoS/DDoS attacks, against the broker servers of HHCEC. The details of HHCEC and PHSS are introduced in Chapter 6.

### 3.1.6 The Transport Protocol for SG Applications Requiring Long Duration Communication

The effectiveness of such cyber-control systems is determined by achieving real-time, accurate state information as obtained from an efficient and reliable communication schema.

In practice, this state assessment is achieved using Wide Area Monitoring Systems (WAMS) that use Phasor Measurement Units (PMUs, and also known as Synchrophasors) for data acquisition to monitor real-time power transmission and to detect grid instabilities [Mar+14]. Furthermore, in addition to high reliability, WAMS require long-duration connections for phasor measurement traffic. Since the Multipath-TCP (MPTCP) allows long-duration connections as well as path redundancy for resilient communication, we propose use of MPTCP particularly for SG applications requiring long-duration connections (such as the phasor measurement applications of WAMS).

Furthermore, to counter internal-attacks mounted using elevated privilege against the WAMS devices, we develop a proactive and robust extension of the Multipath-TCP (MPTCP) transportation protocol, termed as MPTCP-H, that mitigates such attacks by using a novel stream hopping mechanism. The proposed stream hopping mechanism periodically renews the subflows over new port numbers to hide the open port numbers of the connection from an attacker. The connections and the structure of MPTCP is depicted in Fig. 3.4. The details of the proposed extension, MPTCP-H, are presented in Chapter 7.

## 3.2 Threat Model

We introduce threat models which are common for the contributions **(C2)**, **(C3)**, and **(C4)**. The differences in threat models/attack types are presented in Chapter 5-6-7, respectively.

### 3.2.1 Security Goals

Our security objective is to guarantee delivery of the publisher data to the corresponding subscribers within the time window specified in the application requirements. To achieve this, DoS/DDoS attacks must be proactively prevented or at least mitigated to meet the high availability requirements of the critical devices. Moreover, any lossy or outdated

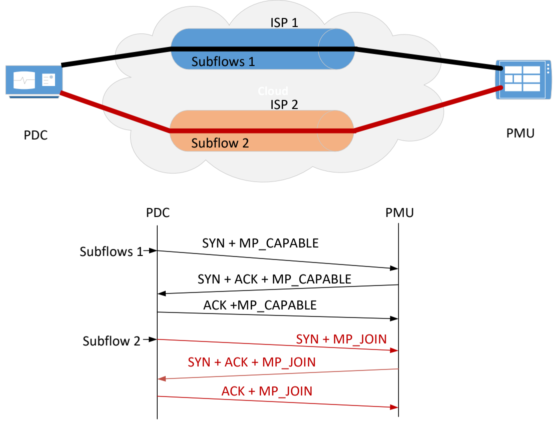


FIGURE 3.4: MPTCP Connection

data needs to be detected by the brokers and subscribers. The origin of the data should be identifiable within the group communication paradigm. Also, data requiring confidentiality can be decrypted by the corresponding subscriber but neither by the broker nor by intruders.

### 3.2.2 Attack/Threat Model

An adversary whose aim is to render critical devices (publishers) inaccessible can mount a DDoS attack against subscribers passing through broker(s) or direct broker(s) that maintain the communication between the publishers and the corresponding subscribers. The attacks can be mounted for a short time, which forces the peers to reset their communication and authentication, introducing an unacceptable loss of availability for the critical applications. For critical applications, replay and repudiation attacks can pose a high risk e.g., receiving an outdated measurements can result in a wrong decision for field devices. An adversary can obtain an elevated privilege by compromising secrets to resend/delay some of the data.

We consider a strong threat model where (1) An adversary can have access rights to some underlay routers to eavesdrop, capture, drop, resend, or alter the data traffic to carry out a replay or DDoS attack against brokers. By exploiting elevated privileges, a large amount of zombies can launch an attack on brokers to deny the service. (2) The secrets of some publishers, subscribers and brokers can be compromised by the intruders to attack brokers and subscribers (if the attacker gains a right to pass the authentication of the brokers).

### 3.2.3 Assumptions

As in contemporary attack models, we assume that (a) publishers obtain only the IP addresses of the broker servers and (b) valid certificates are issued by a Certification Authority to all brokers/publishers/subscribers and to Authorization Servers in a secure way.

Since we focus on the broker defense against DDoS attacks, the protection of the Authorization Server is beyond the scope of the paper. It is worth mentioning that the pathological case of attackers that can fully saturate the Internet backbone links in HHCEC is also beyond the scope of this approach.

### 3.2.4 Security Vulnerabilities of SG Systems

To highlight security vulnerabilities of SG devices regarding the threats above, we outline a study conducted on a testbed using real SG devices. Morris et al. [Mor+11] conducted tests to evaluate the vulnerability of PMUs and PDCs in the context of attacks originating from inside a WAN by building a testbed consisting of PMUs, PDCs, a router and a Network Analyzer [Mor+11]. They launched TCP flooding (SYN and FIN) and UDP garbage flooding attacks on the devices for both specific and random ports. The test results showed that all devices under flooding attacks are eventually overwhelmed and start denying service when the traffic volume increases beyond the data processing ability of the device. Based on these results, in order to mitigate these issues the authors suggest that utilities to monitor the volume of the network traffic should be enabled in order to detect and/or limit transmission of the traffic to the devices. Moreover, the fuzzing tests conducted in [Mor+11] show that even individual packets can result in device failures i.e., resetting the devices. These test results indicate that DoS/DDoS attacks can be a serious threat to the safety and reliability of the power network. Such DoS/DDoS attacks can lead to partial loss of availability, and thus can lead to wrong state estimation of the power network or delay in removing power system failures [Mor+11]. For these reasons, a proactive defense mechanism needs to be employed to mitigate the DoS/DDoS attacks for SG communication.

# Chapter 4

# Reliable Communication over Public Networks

To support the communication requirements of the SG, utilities typically prefer dedicated, private E2E communication networks. However, this may not always be accomplished due to cost and technical restrictions. Therefore, the SG communication network could become a heterogeneous network consisting of multiple private networks and Internet service providers (ISPs). Furthermore, the scale of SG communication networks might span a territory (e.g., a state or metropolitan area) with millions of nodes. While some SG communication nodes (e.g., substations) have high computation capacity and multihoming with high outgoing bandwidths, other nodes (e.g., smart meters, sensors and circuit breakers) may only have basic functionalities [Gun+13; Bud+10; KK13].

Irrespective of the type of SG schema, the common element across them is the need for reliable, timely and responsive communication to facilitate effective sensing and control. Most of these applications have stringent latency (in the range of 100 ms to 5 s) and reliability (99.00%-99.9999%) requirements [US 10; WYB15]. Unfortunately, the present Internet infrastructure does not innately provide the necessary QoS guarantees for such safety-critical applications, which essentially require both low latency and high reliability. One reason for this deficiency is that the routing, utilizing BGP:Border Gateway Protocol, among ASes of the Internet, depends on commercial considerations resulting from contracts among these ASes. These contracts promote low cost links rather than low latency links, although there are better paths that BGP can accommodate in the Internet infrastructure. In addition, the BGP convergence time might take several minutes or even sometimes up to 20 or 30 minutes. This can cause delays or even loss of traffic [Wan+13; RGZ06]. Furthermore, in such heterogeneous networks, E2E QoS cannot be guaranteed by employing the current underlying QoS approach (e.g., DiffServ [Bla+98], IntServ [Nav+13], MPLS [RVC00]), due to administration configuration differences in each domain [Bud+10].

To meet the QoS requirements for SG applications, multiple approaches have been proposed such as INTEGRIS [Nav+13], GridStat [Bak+11], and CRUTIAL [DBC09]. While INTEGRIS and GridStat mainly focus on dedicated private networks, CRUTIAL targets reliable communication between control centers and substations (but not all SG devices) by employing multihoming techniques. In addition to these approaches, overlay networks have emerged as an effective way to improve the performance of real-time Internet applications [And+01; LM04; Sub+04; Vul+12; LK12]. The overlay routing solutions use overlay nodes to bypass performance degradation on Internet paths without requiring changes in the underlying network layer. They can provide timely and relatively reliable Internet services. Nevertheless, these solutions do not target (1) delivery guarantee for each message (high reliability) even in case of permanent underlay failures, in addition to (2) application-adaptive and criticality aware resource allocation.

Building on this background, in this chapter, we propose HetGrid as an overlay based communication infrastructure that provides the following capabilities and contributions:

1. *High reliability in the heterogeneous networks:* HetGrid strives to build physically-disjoint multipaths, and meets the strict QoS requirements of SG applications via a light-weight low-overhead communication architecture. To achieve high reliability, it employs Source Routing-based QoS Routing (SRQR) and Compensative Multi-Routing (CMR) mechanisms.

2. *Application-adaptive and criticality aware resource allocation:* SG applications not only need flow-based (periodic) data acquisition, they also need a periodic data accusation (e.g., alert messages) with diverse QoS requirements. This necessitates a smart resource allocation on the overlay network. Thus, HetGrid employs Altruistic Flow Allocation (AFA) in order to reserve/allocate the "best" paths (in terms of QoS metrics) for high priority (critical) applications in a distributed manner.

Overall, HetGrid's overlay network obtains QoS-satisfied paths for each SG application by using SRQR and AFA on the heterogeneous networks. In addition, CMR strives to maintain the communication at the required latency and reliability level. Our evaluation focuses on QoS-satisfaction of each application with the diverse reliability, latency and bandwidth requirements.

Since TCP provides a reliable connection between end hosts, the current IP-based industrial critical applications typically rely on TCP connections. This is a pragmatic solution and a widely used approach in industrial systems if the TCP overhead is small compared to the application payload size. Unfortunately, the SG applications message payloads are usually less than 1000 bytes [KK13]. Consequently, the TCP's additional overhead (e.g., larger header, session establishment messages etc.) entails an unacceptably large protocol overhead that is not viable for SG applications. Hence, HetGrid advocates and employs a lower-overhead UDP connection between peers.

We evaluate HetGrid by comparing the direct TCP Vegas[1] [BOP94] connection (between the end hosts) based on the applications' QoS-satisfaction rate. The evaluation is performed in four perturbation scenarios: 1. dynamic link state changes in the underlay network, 2. failure in the underlay routers, 3. heavy congestion in the underlay routers, and 4. bursty traffic on the overlay network. We note that in existing works, although there are many direct TCP connection and overlay based QoS enhancement comparisons [And+01; Sub+04; Vul+12], they focus on performance improvement in terms of latency and loss rate but not on delivery guarantee in time for each message (high reliability) in case of large-scale failures.

[1] In this chapter, TCP is used to refer to TCP Vegas.

## 4.1 QoS Requirements and Challenges for SG Applications over the Heterogeneous Network

As highlighted in the previous section, E2E ownership of the network for the SG may not always be possible because of cost considerations, spectrum availability etc. While heterogeneous networks are proposed as potential solutions, detailing the challenges that SG applications will encounter is very much an open issue. Before introducing our proposal, the major communication requirements and challenges for SG applications communicating over the heterogeneous networks are discussed in the rest of this section.

### 4.1.1 E2E Latency Guarantee

Many types of information delivery between power devices are only meaningful if they arrive within a predefined time frame (i.e., deadline). Delayed information is of limited value, and in the worst case, damage might occur in the grid. For example, the islanding protection actions must be made within a time window of 150-300 ms [Kan+09]. As public networks such as the Internet typically provide best-effort services, time-sensitive applications running on such a network potentially result in damage to the grid. In cases where the public network is used for SG communication, the communication system should be supported by additional measures e.g., increasing the number of end hosts with multihoming feature [KK13].

### 4.1.2 Reliability

From [OK10], SG reliability is defined as the degree to which a communication system must remain operational. The operability of SG devices relies on the communication infrastructure in order to maintain the stability of the grid in their respective domains. Hence, the communication infrastructure must be fault-tolerant, especially for safety critical applications, to protect the grid and ensure efficient operation. In particular, the Internet's communication reliability is affected by a number of possible failures. For example, BGP failure and congestion introduce high reliability risks due to BGP's convergence time and its policy based routing approach [Wan+13]. To cope with this, a self-organizing overlay network, supported by multi-homing for critical end points, is a potential alternative to satisfy the communication requirements on the heterogeneous networks [RGZ06; KK13].

Furthermore, in cellular networks the failure of core network resources potentially leads to disconnection of many power devices, which poses safety risks in the grid. To mitigate this, ad-hoc connections with physically-near devices, which connect to different ISPs, can be provided in order to obtain fault tolerant communication for the critical applications [KK13].

### 4.1.3 Scalability

Due to the continual growth of the SGs, an *a priori estimation* of the network scale is difficult to ascertain. In addition, the workload of SG applications can increase rapidly depending on conditions such as the weather or the electricity price [KK13]. As the workload of the network increases, meeting latency and reliability requirements of safety critical applications becomes more challenging. Hence, the communication network for the SG applications should be scalable and adaptive to the changing network dynamics [KK13].

### 4.1.4 End-to-End QoS Guarantee

SG applications have diverse QoS requirements. Moreover, some of SG applications may need different priorities for their messages under different conditions depending on the function of that data. For example, periodic metering measurement traffic typically has a lower priority, whereas these metering measurements may necessitate higher priority when they are required in active demand response applications [Bud+10]. Moreover, after a power outage, if a large number of meters must be registered in a short time, the meter registration traffic may be considered higher priority and critical. Hence the need is of a QoS mechanism that discovers the resources across the network and allocates them in a distributed manner depending on the applications' real-time QoS requirements. Moreover, it should not be centralized in order to avoid a single point failure [KK13].

## 4.2 System Model

As SG communication nodes span a large geographical area and connect to WAN of the SG via diverse ISPs and private links, the SG communication network contains many Autonomous Systems (AS) and domains. In addition, some the utility owned powerful nodes are multihomed by connecting to multiple ISPes or other nodes via direct links such as transmission/distribution substations and control centers. As a result, the SG communication network is considered as a heterogeneous network. The underlay and overlay models are presented in Chapter 3.1.1-3.1.2, as they provides a base for the all contributions in this thesis. We now present the data types and application models that underlie the development of HetGrid.

### 4.2.1 Data Type and Delivery Model of SG Applications

There are three types of data traffic in SG applications: (D1) Sensing traffic, (D2) Control traffic and (D3) Coordination traffic. All data traffic types can be periodic or aperiodic, and their sizes are typically less then 1 Kb [YBG11].

We classify data delivery requirements of SG applications into four different modes: (M1) guarantee, (M2) no-guarantee, (M3) in-time, and (M4) best-effort time. To illustrate, while a protection application requires delivery in M1 and M3, a video surveillance application can be based on M2 and M3. In all modes, the packets are formed by employing the UDP protocol. In order to address UDP's reliability shortcomings and guarantee data delivery, we make use of the adaptable Ack mechanism (AAM) in M1. To obtain timely data delivery in M3, a source routing-based QoS routing (SRQR) mechanism is employed considering latency constraints. Furthermore, for an application that needs both M1 and

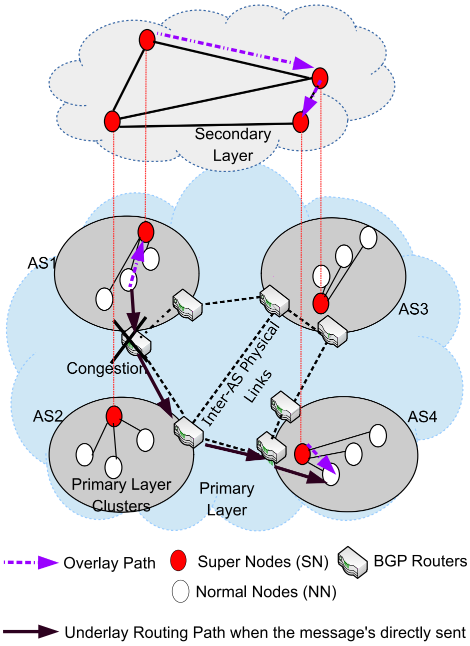


FIGURE 4.1: Basic HetGrid Architecture

M3, in addition to SRQR, the data is routed over multiple redundant paths to exploit physical path diversity using a compensative multipath routing (CMR) mechanism.

We use a priority scale ranging from high to low priority (high, medium, and low priority). We assume that safety or mission critical applications with low latency and high reliability requirements are allocated as a high priority.

### 4.2.2 Application Model

We consider diverse SG applications to comprehensively evaluate our proposal. Hence, we employ the following application characterizations, according to the above data traffic types: we firstly categorize the SG applications into two classes, sensing $A_s$, and controlling $A_c$ applications. While $A_s$ transmits its data in a periodic manner (flow), $A_c$ delivers data aperiodically (occasional). Furthermore, these applications are assigned to different classes based on their priorities, i.e., $A_{shigh}$, $A_{smedium}$, and $A_{slow}$ and $A_{chigh}$, $A_{cmedium}$, and $A_{clow}$.

### 4.2.3 Assumptions

We assume that each application has a unique ID. For each application ID, priority and QoS requirements information exists in all overlay nodes. The application ID is a part of the packet payload and written by the sending power application.

## 4.3 Construction of a Supernode-based Two-layer Overlay Network

The objectives of the HetGrid's overlay network design are twofold: (1) To mitigate the overhead of the QoS routing that probes the underlying network to find paths satisfying the QoS requirements, and (2) To obtain physically disjoint redundant paths for the multipath routing mechanism, which provides fault-tolerant communication for the critical applications.

### 4.3.1 Clustering of Nodes and SN Selection

The bootstrap node clusters nodes depending on their AS, and then selects SNs from each cluster depending on their resources (i.e., computation capacity, outgoing bandwidth, and multihoming), as illustrated in Fig. 4.1. The purpose of this clustering is to mitigate the overhead of the overlay-based link state routing and to improve routing scalability and performance on the overlay network. This results in a two layer overlay design. The secondary layer only consists of the SNs that are interconnected with each other. The primary layer, on the other hand, consists of all NNs and SNs in their respective clusters. The primary layer clusters are structured using a star topology, whereas secondary layer links are constructed according to the physical links that connect ASes or the private networks.

In order to cluster nodes according to their AS, we can use the approach from Ren et al. [RGZ06]. Their procedure is as follows: BGP updates can be regularly accessed by the bootstrap node. By using these updates, IP prefixes of ASes and the AS-AS connection relationships can be obtained. AS-based clusters are then constructed by the bootstrap node, matching the IP-prefixes of ASes with node IPs.

The bootstrap node defines a master SN (mSN) and $d-1$ redundant SNs (rSN) from the selected SNs for each cluster. Together, the mSN and the rSNs provide at least $d$ redundant disjoint paths for each pair in the overlay network. In case of mSN failures, the rSNs can take over the mSN tasks. This is possible as rSN regularly checks the mSN by using heartbeats, in addition to periodically synchronizing the required data with mSN. By an overseeing rSN, the transition is accomplished by disseminating a leadership message, "I'm the new-mSN" to all overlay nodes of its cluster. This takes less than 10s in our implementation.

While the selection criterion for mSN is that it is the ″strongest″ node in the cluster, the criteria for $d-1$ rSNs is the provision of the ″highest″ path diversity (in terms of underlay routers) between each pair within the cluster, in addition to a sufficiently high computation capacity. Selecting the overlay nodes in a cluster providing physically-disjoint paths between each pair, however, is a difficult task. To cope with this, each peer (including SNs and NNs) runs the traceroute tool towards the others within its cluster to obtain the underlay routers pattern between them. They relay the traceroute data to the mSN, which then heuristically chooses $d-1$ rSN that statistically provide "the least" correlated paths (in terms of underlay routers patterns) between each pair in their cluster (cf. Han [HWJ08]). Since the underlay topology changes only infrequently (monthly or at a longer interval), this traceroute operation adds only limited additional overhead [HWJ08].

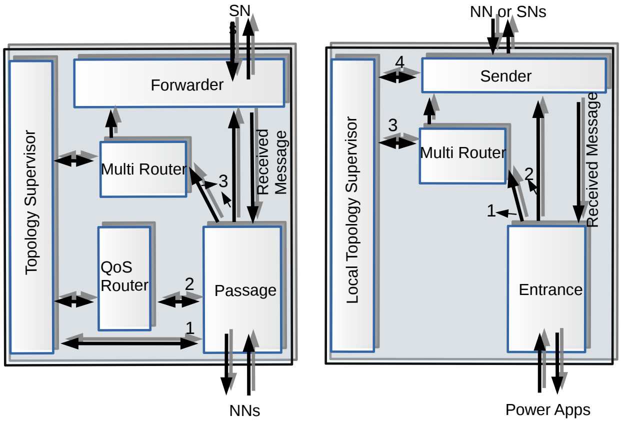


FIGURE 4.2: The architecture of software on a SN and NN, respectively

### 4.3.2 Obtaining Disjoint Redundant Paths

In order to obtain a fault-tolerant communication system for the critical applications, HetGrid aims to determine E2E physically-disjoint redundant paths between each pair (of NNs) by using the constructed overlay network.

The connection between a pair can be intra-AS or inter-AS. Thus, the determination of redundant disjoint paths on the network requires different approaches for each scenario. For intra-AS connections, HetGrid enables NNs to transmit their data through mSN and $d-1$ rSN, providing the "highest" path diversity, to any other NN as explained above. Thus, the NN can send its data, replicated $d$ times, over their respective SNs (1 mSN and $d-1$ rSN) to any other NN in the same AS in order to obtain reliable and timely data delivery guarantees.

In case of inter-AS connections, mSN calculates multiple disjoint paths towards each destination of a given application. This is unlike intra-AS connections which provide adequate path diversity by simply sending the replicated data over their respective SNs. Since the secondary layer is based on physical connectivity of the underlying network, mSN can readily define inter-AS disjoint redundant paths towards any other SN by omitting the path(s) which have the same overlay nodes (SNs) with already selected path(s). However, some ASes have single upstream (i.e., single BGP router to connect to another AS). This disturbs the E2E path disjointness. To cope with this, mSN takes advantage of multihoming features of its cluster's $d$ SNs to obtain disjoint upstreams for each redundant path (we assume that SNs have multiple network connections, e.g., different carrier connections in a substation, a meter concentrator etc.). Furthermore, the mSN organizes the $d$ SNs to provide different upstream networks when multiple disjoint paths towards a destination of an application are defined. Thus, HetGrid ensures an E2E physically-disjoint path for the critical applications in the Internet infrastructure.

## 4.4 The Software Architecture of HetGrid

Fig. 4.2 depicts the conceptual software architecture (stack) for the SN (left side) and the NN (right side). We first detail the NN operations.

The *Entrance* component serves as a gateway for exchanging data between power applications and the NN stack. After receiving a packet from a power application, the *Entrance* component determines the priority of the packet by using the application ID. If the packet is of high priority, the *Entrance* relays it to the *Multi Router* (#1 in Fig. 4.2), otherwise it is relayed to the *Sender* (#2). *Multi Router* replicates the packet $d$ times and then hands the packets over to the *Sender* (#3) to be delivered to $d$ SNs (the details below). *Sender* relays medium and low priority packets depending on their destination domain. Packets that are destined to extra-AS are sent to mSN and packets that are destined to intra-AS are directly transmitted to the destination NN[2]. In addition, the *Local Topology Supervisor*, which maintains local network discovery, gives information about the addresses where packets are destined (#4).

Let us assume that a packet destined to an extra-AS is received by a mSN's *Passage* component which is functioning as a gateway for exchanging packets between ingress SNs and NNs. As described above, intra-AS deliveries are fulfilled either directly (medium or low priority) or over $d$ SNs (high priority).

The *Passage* component first determines the QoS requirements for the packet using the application ID tag. Subsequently, the *Passage* component consults the *Topology Supervisor*, which is responsible for network discovery and maintenance, about the destination NN's mSN (#1). Then, the *QoS Router* is queried to find a suitable routing path between itself and the mSN, which has to satisfy the QoS requirements (#2).

If the packet is of high priority, it is first processed by the *Multi Router* before being delivered to the *Forwarder*. If not, the packet is handed over to the *Forwarder* (#3) after tagging it with the route information. If it is of high priority, implying $d-1$ rSN also received the replicated packet, *Multi Router* in the mSN identifies SNs (among $d$ SN) which should relay the replicated packet (cf. Section 4.5.4) and their routes over the secondary layer. This information is then transmitted to the defined rSNs and, finally, the defined SN's *Multi Router* emits the packet, tagged with the respective route information, to the *Forwarder*. The *Forwarder*'s task is to send the packet to the next address in the packet header irrespective of its priority.

Once the QoS routing path(s) are determined for the destinations of a given application in the ingress mSN, that information can be stored and reused if the application needs a periodic data flow towards the destinations. Hence, packets with already known application IDs can be directly handed over to the *Forwarder*. However, in case of significant network state changes, the *Topology Supervisor* causes a reset of the stored information to allow the system to adapt to the new network state. In addition, all aperiodic packets are simply relayed to $d$ SNs to route over their the ”best” path towards the destination, which is reserved (cf. Section 4.5.3).

[2] As Internet service providers (ISP) can assure re-convergence time in the range of a few seconds by employing MPLS within AS, we do not need to use any overlay routing inside the cluster. However, for high priority applications, HetGrid still sends the messages over $d$ SN's that provide disjoint paths over a given AS' routers.

## 4.5 Routing

The fundamental differences of SG applications from current Internet-based applications are their stringent QoS requirements and needs pertaining to the timely delivery guarantee for each message (e.g., islanding protection messages). However, the current Internet infrastructure mainly provides a best-effort delivery. Hence, any communication system that is proposed for the SG should: 1) provide QoS-satisfied paths for each application and 2) be fault-tolerant to support the timely delivery guarantee for each message of the critical applications by employing multipath routing and smartly allocating the resources. To address these requirements, the following mechanisms are employed on the **secondary layer** overlay network in an application-adaptive manner: HetGrid provides the QoS-satisfied paths for each application by employing a Source Routing-based QoS Routing (SRQR). Furthermore, to obtain timely delivery guarantee for each message of the critical applications, HetGrid takes advantage of Compensative Multi-Routing (CMR) in addition to the Altruistic Flow Allocation (AFA) mechanism. We introduce these mechanisms and detail how they cooperate in a self-adaptive way.

### 4.5.1 Source Routing-based QoS Routing (SRQR)

Essentially, SRQR takes advantage of the shortest path algorithm to make routing decisions on the secondary layer, considering QoS metrics, i.e., reliability, latency and bandwidth. Moreover, SRQR employs *source routing* in order to facilitate multihop routing by speeding up the transmitting path at overlay nodes. It also helps bind a packet flow to a selected path (barring significant link state changes sensed by the heartbeat mechanism), making performance more predictable, and providing support for multipath routing in HetGrid. When the strict QoS requirements of SG applications are considered, using the *source routing* to obtain predictable network performance can be an efficient method.

The paths are constructed using the shortest path algorithm with hop normalized path weights for bandwidth, reliability, and latency to result in Equation 4.1 as:

$$PathWeight = \alpha_b \sum_{i=0}^{n} (\frac{B_{i,i+1}}{B_{i,i+1} - R_B})/n * \\ * \alpha_r \frac{-\sum_{i=0}^{n} logR_{i,i+1}}{\sum_{i=0}^{n} logR_{i,i+1} - logR_R} * \alpha_l \frac{R_L}{R_L - \sum_{i=0}^{n} L_{i,i+1}} \tag{4.1}$$

In the Equation 4.1, $n$ is the number of hops in the path. While $B_{i,i+1}$, $R_{i,i+1}$ and $L_{i,i+1}$ are residual bandwidth, current reliability, and latency of the link, respectively. $R_B$, $R_R$ and $R_L$ denote required bandwidth, reliability and latency, respectively. The alpha coefficients enable tuning of the influence that the individual components have on the overall path weight.

This formula combines the influence of the multiplicative (reliability), concave (bandwidth) and additive (latency) metrics in a proportional manner to calculate the path weights while assuring that only paths which satisfy all metrics are selected. It ensures the best available path in terms of the metrics. The derivation of this formula is presented in Appendix A [DGS14].

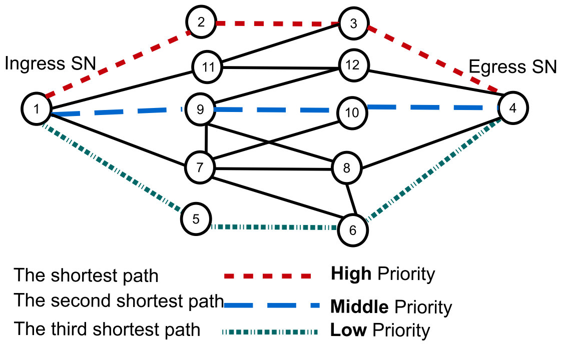


FIGURE 4.3: Basic illustration of AFA

The main goal of SRQR is to bypass performance degradation on the Internet path by having multihop routing on the overlay network. However, a drawback of this approach is that it entails additional hops leading to performance degradation in the overlay network. To circumvent this, we employ the following approach: After the start of the flow over the path (which is allocated by mSN), each overlay node on the path probes the destination to decide whether direct communication meets the QoS requirements. If it does, then the overlay node skips the rest of overlay nodes on the path and sends directly to the destination. When a significant link state change is reported, the overlay node probes the direct link whether it still satisfies the flow's requirements, skipping the rest of path. This approach provides a significant improvement in performance for SRQR.

### 4.5.2 Resource Monitoring

To obtain QoS-satisfied communications by using SRQR, available resources of the links (i.e., link bandwidth, reliability, and latency) need to be monitored. Thus, HetGrid employs pinging and direct bandwidth measurement methods in each mSN to obtain its adjacent links' states, as in Li et al. [LM04]. Each mSN disseminates the gathered link state information to the other mSNs when significant changes occur on the links. However, these measurements might be noisy, and this leads to oscillation or the wrong selection in the path allocation. To avoid this, HetGrid applies a 5% hysteresis bonus to the "last good" measurements for the three metrics, thus providing a reasonable trade-off between responsiveness to the link state change and the oscillations.

### 4.5.3 Altruistic Flow Allocation (AFA)

SG applications have both periodic (flow) and aperiodic data traffic. Assuring availability of the resources for critical/high priority data traffic in such a network is a difficult task. Existing works try to cope with resource allocation by building different virtual networks for QoS requirement classes on top of an overlay network and smartly allocate the resources, as done with policy routing in [And+01]. However, these methods base on static resource allocation for the applications that need static QoS requirements and introduce best-effort performance but not predictable. For SG applications with changeable and strict QoS requirements [US 10], these are not efficient approaches.

AFA introduces an implicit allocation mechanism for quick adaptation to the dynamic background traffic of the overlay network. The implicit allocation fundamentally relies on binding a flow to a specific path by utilizing source routing. To make this happen, the other nodes in the overlay network also refrain from using the resources on that path by following the restrictions from resource monitoring mechanism. Moreover, to assure availability of the resources for critical/high priority data traffic, AFA selects a path from $k$ paths[3] depending on the application's priority, as illustrated in Fig. 4.3. Thus, the low priority applications sacrifice the "best" resources (but their requirements are still satisfied with the path allocated to them) in an exponential manner, for the sake of the critical applications in a distributed manner. Our AFA approach implicitly provides a resource reservation for aperiodic and critical messages. For a pair belonging to an application, a corresponding path (indicated by $z$) is chosen by the ingress mSN between the first/shortest path and $k_{th}$ path by using the following equation:

$$\lambda = k(\frac{\mathrm{e}^{\rho} - 1}{\mathrm{e} - 1}) \tag{4.2}$$

$$z \leftarrow \lfloor \lambda \rceil$$

$z$ is the integer value where $\lfloor \rceil$ indicates rounding $\lambda$ to the nearest integer and $\rho$ represents priorities in the range [0-1], e.g., 0, 0.5, and 1 represent low, medium and high priority respectively.

Following the equation, the $z_{th}$ path is identified by the ingress mSN for the flow allocation. If $k$ equals to 0, the equation 4.2 cannot find a path and multiple paths are required to compensate for the reliability requirement of the application. To do this, HetGrid employs the following multi-routing CMR mechanism.

### 4.5.4 Compensative Multi Routing (CMR)

In case that SRQR fails to find any path that satisfies the QoS requirements ($k = 0$), the CMR mechanism, which is running on mSNs, tries to find multiple paths whose total reliability (packet loss rate) satisfies the application. Once CMR has found the paths, it relays redundant replicated data over them. In aiming to cope with permanent failures, CMR also strives to discover disjoint paths rather than transmitting multiple packets over a single path that has the highest reliability degree. To do this, mSN running CMR organizes the $d$ SN in its cluster for the assignment of disjoint redundant paths, thus achieving E2E disjointness. However, the open question is how to determine the number of SNs (among $d$ SNs) whose paths provide the required reliability degree. To address it, we propose the following approach:

$$PR_i^D = \prod_{j}^{N} RL_j \tag{4.3}$$

[3] $k$ paths are found by using the k shortest path algorithm and equation 1. We do not put a limit on $k$ to pave the way for obtaining more disjoint paths. In our implementation, $k$ is observed between 5-20

$$R_R \leq 1 - \prod_{i}^{M}(1 - PR_i) \tag{4.4}$$

where $PR_i^D$ is the reliability degree of path $i$ towards the destination $D$, and $RL_j$ is the reliability degree of the link on the path (containing $N$ links). While equation (3) calculates path reliability, equation 4.4 computes total reliability of $M$ parallel paths and compares the total reliability with the required reliability degree. CMR employs an iterative algorithm for finding the number of parallel paths ($M$), which compensate for the required reliability degree $R_R$. Moreover, while selecting the parallel paths, this algorithm tries to eliminate the path whose similarity (in terms of SNs) on the already selected path(s) are higher than a threshold value, heuristically defined.

### 4.5.5 Adaptable Ack Mechanism (AAM)

For an application that needs feedback or delivery guarantee, HetGrid introduces AAM. It supports an adaptable acknowledgment mechanism by allowing applications to adjust the delay of sending acknowledgment depending on their latency requirements. Thus, several ACK responses may be combined together into a single response (by combining the number of network updates), therefore reducing protocol overhead. Since many SG applications have small payloads (e.g., 100-200 bytes) [KK13], it is clear that AAM obtains efficient data transmission by minimizing the Ack traffic. We build AAM inspired by TCP's delayed acknowledgment technique.

### 4.5.6 Putting It All Together

The SRQR protocol strives to find the QoS-satisfied path for each application according to their QoS requirements (e.g., bandwidth, latency, reliability). Additionally, the protocol seeks to balance traffic over the secondary layer of the overlay network. However, as SRQR employs the shortest path algorithm that tends to use a considerable amount of resources, this can lead to a lack of resources for the critical application. AFA provides a solution by reserving the "best" resources for high priority applications. If SRQR cannot find a path that satisfies the application's reliability requirements, CMR can compensate by employing multipath (adequate number) routing for the affected applications. Finally, AAM guarantees data delivery and reductions in the protocol overhead in the network. This will be done by employing an adaptive mechanism which configures the delay of Ack messages depending on time-sensitivity of applications. HetGrid provides QoS-satisfied and fault tolerant communication without producing expensive overhead, as it is able to employ these mechanisms depending on the application requirements.

## 4.6 Evaluation

HetGrid is implemented by using the OverSim [BHK07b] and the INET framework that run on OMNeT++ [Pon93]. This simulation setup paves the way for our perturbation scenarios. The main purpose of our simulation-based evaluation is to assess HetGrid

TABLE 4.1: Performance evaluation parameters

| **App.** | **Msg. size** | **Para.** | **Prio.(p)** | $R_R$ | $R_L$ | **Deli. Mode** |
|---|---|---|---|---|---|---|
| Ash | 32 B | 1 event/15s | high (1) | high (99.90%) | low (<150ms) | M1, M3 |
| Asm | 32 B | 1 event/15s | medium (0.5) | medium (99%) | medium (>150ms, <2s) | M3 |
| Asl | 32 B | 1 event/15s | low (0.1) | low (97%) | high (>2s) | M2, M4 |
| Ach | 32 B | 1 event/120s | high (1) | high (99.90%) | low (<150ms) | M1, M3 |
| Acm | 32 B | 1 event/120s | medium (0.5) | medium (99%) | medium (>150ms ,<2s) | M3 |
| Acl | 32 B | 1 event/120s | low (0.1) | low (97%) | high (>2s) | M2, M4 |

against direct TCP connection according to (1) QoS satisfaction of each application, and (2) fault-tolerance in the system, e.g., the effect of failures on the critical applications. We deploy TCP Vegas from among TCP flavors in our implementation since it can address the heterogeneous networks better than the other flavors [Alb+15] and obtains between 40 and 70% better throughput, with $1/5^{th}$ to $1/2$ the losses compared to the TCP Reno [Pre+14].

We first present underlay topology and background traffic model, followed by overlay network, traffic demands and metrics.

### 4.6.1 Underlay Network Topology

We randomly produce a hierarchical topology using BRITE [Med+01] in order to construct an Internet-like underlay topology. The topology includes 20 nodes (we consider that each node denotes an AS's BGP router) for the AS level and 10 nodes (i.e, interior gateway protocol (IGP) routers) under each BGP router with an edge density changing from 2 to 5. For inter-AS and intra-AS networks, two bandwidth configurations are used: all links are either OC3 (i.e., 51.84 Mbps) or OC48 (i.e., 155.52 Mbps). The propagation delay of each link is randomly chosen between 0-10 ms subject to a uniform distribution.

### 4.6.2 Background Traffic

A dynamic background traffic load across the network is generated during simulation in order to assess HetGrid's success in the case of dynamic latency and bandwidth in the underlay network. To produce this background traffic, we deploy 200 servers (each one connects to a given edge router) that relays a packet (1-100kb) per second to a random server.

### 4.6.3 Overlay Network and Traffic Demands

In the simulation, $|V_o|$=1000, including (|mSN|=20 and |NN|=980) and these overlay nodes are randomly deployed to 20 ASes. The AS-based clusters have two $rSN$ ($d-1$ = 2, total |rSN| = 40) and their computation capacities are adequate and larger than the other NNs. The outgoing bandwidths of SNs and NNs are 10 Mbps and 1 Mbps, respectively.

According to data traffic requirements of SG applications [YBG11], six application models are employed, as shown in Table 4.1. Each overlay node randomly runs one of the six applications and chooses a destination node to relay the application's messages. In the simulation, each node measures its adjacent links' latency, bandwidth and loss rate every 5 seconds and if there is more than 5% change [And+01], e.g., a significant alternation of the three metrics or an outage in the network, it broadcasts the measured values of the link(s).

### 4.6.4 Metrics

HetGrid is assessed based on the QoS-satisfaction of each application and the fault-to lerance in the system. QoS-satisfaction of a given application is computed based on its data delivery monitoring results, i.e., latency and loss rate. A dropped or timed out message is specified as an unsuccessful message delivery. To quantify the satisfaction of a communication, QoS-satisfaction rate (QSR) is formalized as:

$$QSR = 1 - \frac{DroppedOrTimedoutMessages}{SentMessages} \tag{4.5}$$

### 4.6.5 Simulation Methodology

Our evaluation is carried out in four different scenarios. The first three scenarios aim to realize the most common Internet perturbations in order to assess HetGrid's QoS-satisfaction performance on the Internet. The final one's aim is to investigate the scalability of HetGrid if the overlay traffic sharply increases. Lastly, we compare the network overhead of the both approaches to highlight the caveats in each approach.

**Dynamic Link State**: To realize the dynamic behavior of the Internet, we periodically (every 10 sec.) and randomly switch bit error rate (BER) of links from the range of (1e-10, 1e-7) to (1e-10, 1e-3) as well as the background traffic produced by the servers. In this scenario, we aim to evaluate whether HetGrid provides an adequate QoS-satisfaction level for SG applications on a best-effort network like the Internet (considered as the Internet provides unstable performance).

**2% Underlay Router Failure**: As mentioned in Section 4.2, BGP router failures and their re-convergence time are severe problem for the applications which have stringent QoS requirements. To investigate the effectiveness of HetGrid on these failure types, 2% of the underlay routers fail around 10 min (like typical BGP re-convergence time [5]). These router failures repeat every 10 min in randomly selected routers during the simulations. 2% of the routers are fixed as 1 BGP and 1 IGP router in our simulation. This scenario helps assess HetGrid's failure recovery mechanism, as well as the fault tolerance efficiency of CMR for critical applications, in case resource failures occur in the Internet.

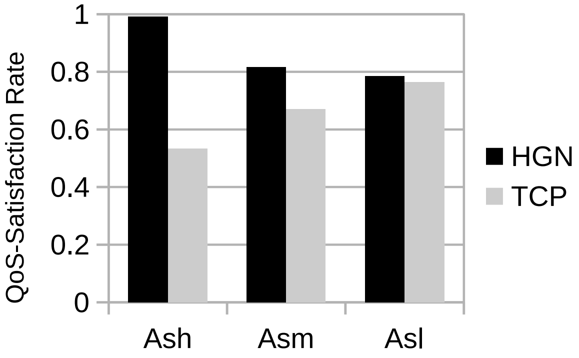


FIGURE 4.4: Dynamic link state scenario: Sensing applications

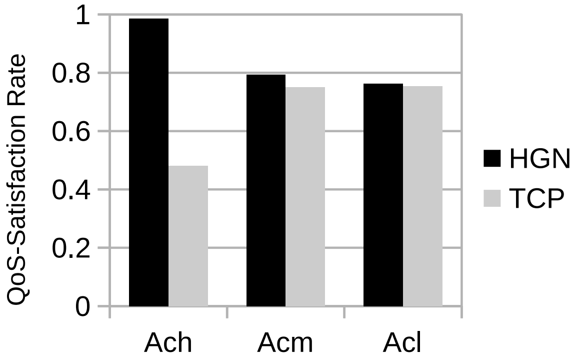


FIGURE 4.5: Dynamic link state scenario: Control applications

**Heavy Congestion Scenario**: When the majority of Internet users are concurrently online, traffic congestion leads to long lag time for the users. To recreate this perturbation, the delivery rate of the background traffic is increased by sending a packet every 100 ms instead of every 1 s, by 40% of the servers (random selection). Employing this scenario, we assess the ″best″ path selection efficiency of SRQR.

**Bursty Traffic on the Overlay Network**: To assess the scalability of HetGrid in case of bursty traffic on the overlay network, we increase delivery rate of the sensing applications in three step, i.e., 1 msg. /15 sec, 1 msg. /10 sec, and 1 msg. /5 sec (SG applications′ traffic volume can change depending on certain conditions [Kan+09]). The main aim of this scenario is to investigate whether AFA smartly allocates resources for critical applications even in bursty overlay traffic.

### 4.6.6 Simulation Results and Discussion

In our simulations, we consider that while the sensing applications require periodic data delivery, e.g., periodic voltage measurements, the control applications need aperiodic data delivery, e.g., command messages. However, while these applications share the overlay network, AFA allocates resources for the sensing applications only, not for the control applications. Hence, we compare HetGrid (HGN) directly with the TCP connection differing in both the control and the sensing applications to assess their performance

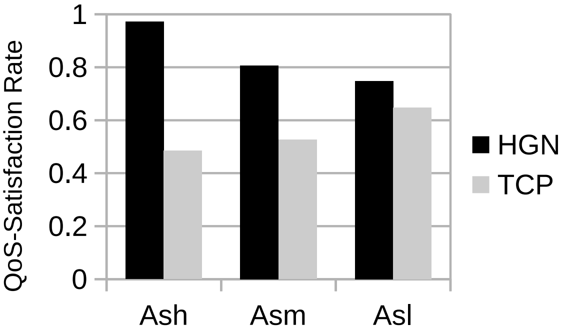


FIGURE 4.6: 2% Underlay Router Failure: Sensing applications

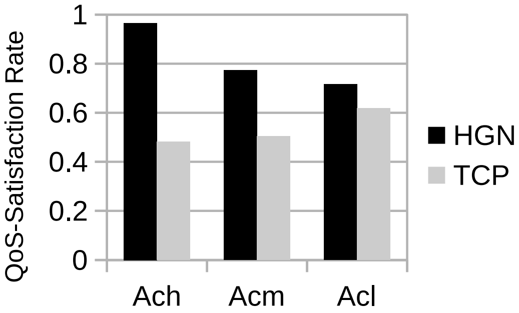


FIGURE 4.7: 2% Underlay Router Failure: Control applications

for both traffic types. Simulation results are first investigated for each specific scenario and then discussed holistically.

### Dynamic Link State

Fig. 4.4 depicts QSR of HGN and direct TCP connections for the sensing applications with three different priorities, i.e., high, medium, and low priority. Remarkably, it shows that HGN provides higher QSR in each priority level in comparison to direct TCP connections between pairs, due to SRQR. In addition, although the high priority applications have stringent QoS requirements, HGN provides significant QSRs for higher priority applications in contrast to direct TCP connections, owing to AFA's priority-based flow allocation mechanism. On the other hand, Fig. 4.5 shows that HGN presents a performance near sensing applications for control applications due to AFA's selection of the "best" resource reservation for aperiodic messages (considered as high priority). However, TCP also provides a performance close to that of the sensing applications (middle/low priority), but not with a consistent behavior. TCP's inconsistency is due to its lack of adaptability to the link state change in the inter-AS connections.

### 2% Underlay Router Failure

Fig. 4.6 shows QSR of HGN and direct TCP connections for the sensing applications when 2% of the underlay routers fail. In the Fig. 4.6, whereas HGN provides QSR with slight degradation for each priority, TCP connections present a remarkable QSR degradation in comparison to their dynamic link state scenario results. Owing to HGN's fast recovery system, the sensing applications using HGN experience slight degradation in comparison to TCP connections. In particular, the high priority applications experience lower degradation than the others in HGN thanks to CMR's sufficient multipath routing mechanism. Moreover, Fig. 4.7 depicts HGN nearly maintaining its QSR performance also for aperiodic application.

#### Heavy Congestion Scenario

The effects of heavy congestion on the QoS-satisfaction of the sensing applications are shown in Fig. 4.8. HGN provides significant QSRs for high and medium priority messages in comparison to TCP connections. However, both HGN and TCP presents almost the same QSR performance for the low priority since AFA allocates the limited resources for higher priority applications in such a heavy congested network. In addition, Fig. 4.9 depicts that HGN provides a similar performance to the sensing applications for the control applications even under heavy congestion.

#### Bursty Traffic on the Overlay Network

Fig. 4.10 shows the efficiency of HGN while increasing the work load on the overlay network. We can observe that HGN saves QSR of high priority applications compared with medium and low priority applications. This provides relatively an adequate QSR for the high priority/critical applications if the bursty traffic is occasionally experienced by the overlay network.

#### Network Overhead Comparison

Fig. 4.11 shows the network overhead comparison of the both approaches in different failure scenarios: D.L.S., 2% F. and H.C. denote Dynamic Link State, 2% Underlay Router Failure, and Heavy Congestion scenarios, respectively. As seen in Fig. 11, TCP Vegas produces more overhead than HGN in D.L.S. scenario, despite HGN's additional source routing overhead. This is because TCP Vegas employs fast retransmission for each applications, whereas AAM of HGN uses an adaptive acknowledge mechanism in addition to UDP transport protocol. Moreover, as TCP's a higher header size (20 bytes), this yields additional protocol overhead for SG applications requiring a small size packet delivery (e.g., 100-200 bytes)[YBG11], TCP is not convenient transport protocol. On the other hand, as shown in Fig.4.11, when the failure/congestion area expands in the network, HGN network's overhead can surpass TCP's, since it must disseminate more link state information. However, the infrequent occurrence of long failures or heavy congestions in the Internet offer a pragmatic basis for the additional overhead for HGN.

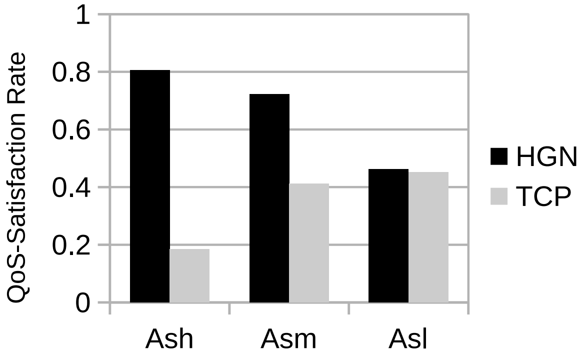


FIGURE 4.8: Heavy Congestion Scenario: Sensing applications

**Discussion**

In our evaluation, we assessed QSR performances of HGN and direct TCP connection in common Internet perturbations as well as in overlay bursty traffic. We separately evaluate their QSR performances for periodic and aperiodic traffic by producing the sensing and the control SG applications, respectively. The results show that HGN presents a significant QSR for SG applications on the Internet-like network in scalable manner thanks to its clustering mechanism. In particular, its QSR for high priority applications shows that employing HGN enables the usage of the heterogeneous network for SG applications. The maintained QSR for high priority applications, in even the underlay failures or heavy congestions, is also a notable feature of SG applications. Furthermore, although HGN saves the resources for the sake of high priority applications by sacrificing the QoS of medium and low priority applications, HGN's QSR performances for medium and low priority applications still outperform TCP connection. HGN also shows that if bursty traffic happens on the overlay traffic, it does not allow significant QSR degradation for high priority applications. In the simulation experiments, since HetGrid has a reactive link state dissemination mechanism and a low overhead transport mechanism (UDP + AAM), we do not observe a remarkable overhead rise in comparison to TCP Vegas. Finally, despite a significant decline in the number of the unsatisfied, high priority messages in the use of HetGrid, the unsatisfied messages could cause severe problems in the grid. This can be handled by investing for more multihoming and direct fiber optic links between SNs.

## 4.7 Conclusion

We have shown in this chapter that HetGrid provides reliable and QoS-aware communication on heterogeneous network, considering the SG applications' requirements. HetGrid selects and employs overlay nodes with the most resources to manage inter-AS communication rather than place dedicated servers into each domain. Also, it requires only local underlay knowledge to obtain reliable communication across the network. To provide reliable and QoS-aware communication, HetGrid uses the following mechanisms in a self-adaptive manner: (1) SRQR discovers the "best" paths meeting bandwidth, latency,

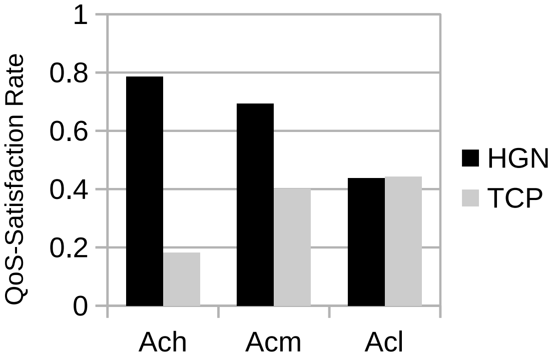


FIGURE 4.9: Heavy Congestion Scenario: Control applications

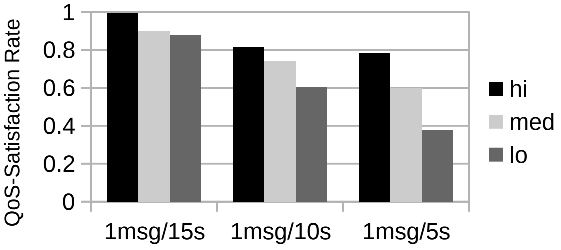


FIGURE 4.10: Bursty Traffic on the overlay network: Sensing applications

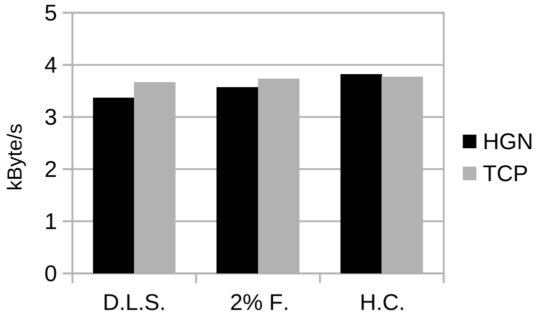


FIGURE 4.11: Overhead comparison in different failure scenarios: D.L.S., 2% F. and H.C. denote Dynamic Link State, 2% Underlay Router Failure and Heavy Congestion scenarios respectively.

and reliability requirements of the applications. To reserve the ”best” path for high critical applications, SRQR also takes advantage of altruistic flow allocation (AFA), and (2) To provide fault tolerant communications for the high priority applications, CMR uses adequate paths for multipath routing to meet the reliability requirement of the applications.

The simulation results show that HetGrid provides a significantly higher QoS-satisf action rate for each application compared with direct TCP connection between pairs. In addition, even for BGP router failures or heavy Internet congestions, HetGrid provides practical QoS-satisfaction rates by employing the above mechanisms in an adaptive manner. Thus, HetGrid shows both the feasibility of using a heterogeneous network for SG applications and also the architecture to achieve the robust QoS-aware communication.

# Chapter 5

# A Secure and Reliable Communication Platform

To monitor and control the power grid, the utilities currently employ proprietary and closed automation networks. However, these networks invariably encounter scalability issues to deal with the (a) increasingly large and *ad hoc* SG structure, and (b) large data traffic produced by the thousands of SG devices. As a result, the grid requires a flexible and scalable network that can provide low-latency, high-availability, secure and reliable communication. While an ideal solution would be a dedicated network, the financial reality results in the use of IP-based public networks such as the Internet [Bud+10; RGZ06; KK13]. The caveat is the inheriting of the Internet's reliability risks and security vulnerabilities, that can be exploited by hackers causing security and safety risks for not only the cyber-system but also for physical-systems, e.g., electrical grid/appliances [WL13].

Hence, SG communication networks need to have lightweight security mechanisms for preventive/proactive defenses to DDoS attacks in the SG's distributed and composite communication-control cyber-physical environment. As pub-sub approaches inherently provide a proactive DDoS attack protection, a number of approaches based on them have been proposed for the SG. GridStat [Bak+11] employs a pub-sub system and long-term security key pads to provide secure and scalable communication between the parties. However, long-term security keys can potentially introduce severe security vulnerabilities, e.g., compromised keys can be distributed to a large number of zombies to access/attack the network. SeDAX [Kim+12] also introduces a pub-sub system which contains trusted authentication servers allowing the parties to periodically obtain topic-based group keys. This assures E2E confidentiality and integrity. However, SeDAX does not introduce any authentication mechanism between the publisher and pub-sub brokers and this paves the way for DDoS attacks against both the brokers and subscribers. Moreover, none of the existing approaches [Bak+11; Kim+12; SK05; Hei+15; KMR02; FP12] focus on addressing the high availability requirements of the SG devices/data traffic in case of a targeted or blindly sweeping DDoS attack against pub-sub brokers to sustain communication between the critical SG entities.

SeReCP introduces a novel pub-sub-based proactive DDoS attack defense mechanism as well as a lightweight security mechanism. In SeReCP, taking into account the requirements for SG data traffic, device resources and security, we propose a pub-sub system proactively countering DDoS attacks that cannot be handled by the constrained SG devices. However, to render inaccessible some of the critical devices, targeted or blindly sweeping DDoS attacks against pub-sub brokers can be launched, which poses safety risks for the grid. To cope with this issue, we employ a data diffusion approach which makes possible spreading the data packets across the pub-sub brokers thanks to its token-based stateless authentication mechanism. Moreover, to address the stringent availability and latency requirements of SG applications in the case of a DoS/DDoS attack, we propose a multihoming-based fast "recovery" mechanism. We transmit every two consecutive data packets to two different network interfaces of each pub-sub broker during spreading data packet across the brokers. If one of the network interface of any brokers is under attack, the broker(s) request the missing packet after a relatively short waiting time using the remaining functional network interface. This provides a fast packet "recovery" compared to classical ACK-based mechanisms such as TCP's cumulative ACK. Moreover, to protect end-to-end (E2E) confidentiality and integrity of the data, we introduce a group key management system which provides role-based access rights for both publisher and subscriber, in addition to protection from replay attacks.

We assess our approach evaluating: (1) network availability for SG applications over targeted or blindly sweeping DDoS attacks on the pub-sub brokers. For the SG, availability is not only successful data delivery but also a delivery meeting the application's latency requirements (2) overhead in terms of resource usage and additional transmission delay produced by the proposed security mechanism. The results show that SeReCP introduces an acceptably low latency overhead of 40 ms for the SG applications requiring latency less than 200ms [OK10]. We compare our approach with the reference work of Angelos et al. [SK05], which also utilizes data diffusing mechanisms for real-time applications. The approach in[SK05] demonstrates stable performance for up to 5% of pub-sub brokers being attacked. Over 5% failure of brokers causes the TCP connection to break. In contrast, SeReCP shows stable performance for up to 30% of pub-sub nodes being compromised. This demonstrates SeRECP's highly promising capability to effectively build safety critical SG applications utilizing public networks. To summarize, our contributions in this chapter are:

- We define the security requirements and threats for the SG. Based on this, we propose a novel pub-sub approach which provides secure/reliable communication in case of DDoS attacks and for link/node failures.

- Considering the high availability requirements of the SG traffic, we propose a mu ltihoming-based fast "recovery" mechanism in addition to the data diffusion approach, which provides minimum drop/ack/re-transmission over attacks on the intermediate pub-sub brokers.

- Considering the constraints of SG devices and their group communication requirements, we introduce a novel group key management mechanism, which provides replay and repudiation attack protection in addition to confidentiality and integrity assurance.

- The evaluation of SeReCP is performed on a real test-bed NorNet [Dre15], providing multihomed nodes distributed all over Norway. The evaluation validates the effectiveness of SeReCP in terms of availability under attack and for its low overhead.

## 5.1 SG Network and Security Requirements

Traditionally, power grid communication systems have been physically isolated from public networks. This has been changing due to the cost effectiveness of utilizing public networks and the technical features offered by them in terms of bandwidth, latency, stability and availability. While decreasing the cost of operation, employing public networks naturally makes the power grids vulnerable to cyber attacks. We survey some differences of SG communication security requirements from classical IT systems (e.g., Internet, Web) and introduce the features of our approach that address the corresponding requirements.

### 5.1.1 SG security requirements

In SG communication networks, the security objective is to defend the data from unauthorized acts with the prioritized concerns (driven by safety implications) being: 1) data availability, 2) data integrity, and 3) data confidentiality.

For availability requirements, SG applications require timely and reliable access to information. Lossy or delayed information can result in an inaccurate system state estimation. Correspondingly, incorrect control decisions can occur, resulting in damage to the grid. For integrity, the unauthorized modification of information can result in wrong decisions on power management. For confidentiality, to protect personal and proprietary information, unauthorized information access and disclosure need to be prevented. For system reliability, confidentiality might not be critical, yet for systems involving interactions with customers, such as demand response and advanced metering infrastructure (AMI) applications, it is important [Wei+10].

A unicast delivery of a time-critical command by a constrained SG device to multiple entities inevitably results in large delays/congestion in the network and the potential for damage to power equipment. The more efficient approach is multicast, to deliver a time-critical message to all related entities belonging to the same group [KK13]. Hence, authentication/confidentiality schemes for SG security must be able to efficiently support multicast communication (*Requirement 1*).

### 5.1.2 Differences from typical IT security

IT-based cyber security solutions, e.g, firewalls, intrusion detection systems (IDS), and Virtual Private Networks (VPN), are known to be effective in securing the IT infrastructure. However, the resource constraints (computational, memory and bandwidth) of SG devices often preclude the direct applicability of such IT solutions [Wei+10].

In a typical IT system, the application servers are often more secure than the edge/ client nodes. In SG networks, the edges require the same level of security as the control center servers, as the edge devices (such as relays, circuit breakers,...) can cause harm to human life, damage equipment or power lines. Furthermore, SG communication nodes offer limited functionality given their resource constraints. Hence, directly employing sophisticated IT-based DDoS defense/authentication mechanisms has limited applicability to the SG, resulting in the need for lightweight and proactive DDoS protection mechanism to be employed (*Requirement 2*) [Wei+10]. We advocate broker-based pub-sub systems to provide for proactive DDoS mechanisms, as well as multicast communication.

In the case of failures in IT networks, a simple solution might be to reboot using a node or an application. However, in many SG control applications, this is not admissible from a control stability viewpoint. Moreover, the DDoS attacks leading to violation of the timing requirements or loss of control messages data can result in imbalance of the grid due to the improper control. Therefore, SG communication networks are required to avoid single-point failures regarding physical network infrastructure, routing protocol and security mechanisms (*Requirement 3*) [Wei+10]. To cope with this, we introduce a data diffusion approach enabling delivery of the scattered data packets over multipath. This ensures minimum packet drop in the case of pub-sub broker failures. In addition, we propose multihoming based fast "recovery" mechanism in order to resend the dropped packets in the fastest way. To address authentication, the use of high-overhead public key based authentication is of limited usability in the resource-constrained SG devices (*Requirement 4*). Therefore, we propose a token-base mechanism providing a stateless light-weight authentication between brokers and publishers, in addition to an efficient group key management system for E2E security.

## 5.2 Goals, Models and Assumptions

### 5.2.1 Security Goals

Our security goal is to ensure delivery of the publishers' data to the corresponding subscribers within the deadline[1] stipulated in the application requirements. To achieve this goal, DDoS attacks must be proactively prevented or at least mitigated to meet the high availability requirements of the critical devices.

Moreover, any lossy or outdated data needs to be detected by the brokers and subscribers. The origin of data should be identifiable within the group communication paradi gm. Also data requiring confidentiality can be decrypted by the corresponding subscriber but neither by the broker nor by intruders.

### 5.2.2 Reference Pub-Sub Model

We consider that the utility employs a combined network (i.e., public and private), taking into account the applications' availability requirements and cost-effectiveness. To deal with the complexity of this heterogeneous network, we take advantage of the SeReCP middleware, which provides a scalable QoS-aware pub-sub system. This P2P-based pub-sub system selects the "strongest" nodes as brokers. These brokers are clustered depending on their autonomous system (AS) and geographical proximity in order to obtain the network state information in a scalable probing overhead (Fig. 5.1). SeReCP is devised for a messaging paradigm where, upon reception of a publication from a publisher, the broker transmits the data, taking into account the QoS requirement of the application

[1] Maximum acceptable latency in the message delivery

and the network state information, to the brokers responsible for delivery the data to the corresponding subscribers.

SeReCP combats DDoS attacks by distinguish between authorized and unauthorized traffic and then to either allowing the traffic to reach its destination or to drop/filter it, respectively. Thus, SeReCP acts as a firewall scattered over the wide area network to prevent any congested links to the target(s). SeReCP leverages some existing approaches to provide this. For example, to provide a QoS-aware robust overlay network, pub-sub platform for smart grid, and an overlay-based DDoS attack defense mechanism, we take advantage of [DGS15], [Bak+11], and[SK05], respectively. However, considering the SG security threats associated with the network model, we introduce a new advanced attack model and a mechanism, SeReCP, which counters these attacks in addition to covering scalability and QoS issues.

### 5.2.3 Perturbation/Attack Model

An adversary, whose aim is to render critical devices (publishers) inaccessible, can mount a DDoS attack against either subscribers through broker(s) or directly broker(s) that maintain the communication between those publishers and the corresponding subscribers. The attacks can be mounted for a short time, which force the peers to reset their communication as well as authentication. This introduces an unacceptable loss of availability for the critical applications.

On the other hand, for applications requiring high availability and low latency, the accidental failure of broker(s) providing connections between a publisher and its subscribers might pose safety-risks as, until a new connection is established over new broker(s), some of the critical node might be inaccessible.

For critical applications, replay and repudiation attacks can pose a high risk e.g., receiving an outdated measurement can result in a wrong decision for field devices. An adversary can obtain an elevated privilege by compromising some secrets to resend/delay some of the data.

We consider a strong threat model where: (1) An adversary can have access rights to some underlay routers to eavesdrop, capture, drop, resend, and alter the data traffic to mount a replay or DDoS attack against brokers. Exploiting the obtained elevated privilege, a large amount of zombies can launch an attack brokers to deny the service. (2) The secrets of some publishers, subscribers and brokers can be compromised by the intruders to attack brokers and subscribers (if attacker gains a right to pass authentication of the brokers).

### 5.2.4 Assumptions

We assume that all publishers/subscribers/brokers cannot be fully compromised, and only the secrets held by the minority of them can be compromised.

We consider that publishers know only the access brokers IP addresses and not each others.

We assume that all nodes have valid certificates issued by a Certification Authority and that each node's certification is delivered to Authorization Servers in a secure way.

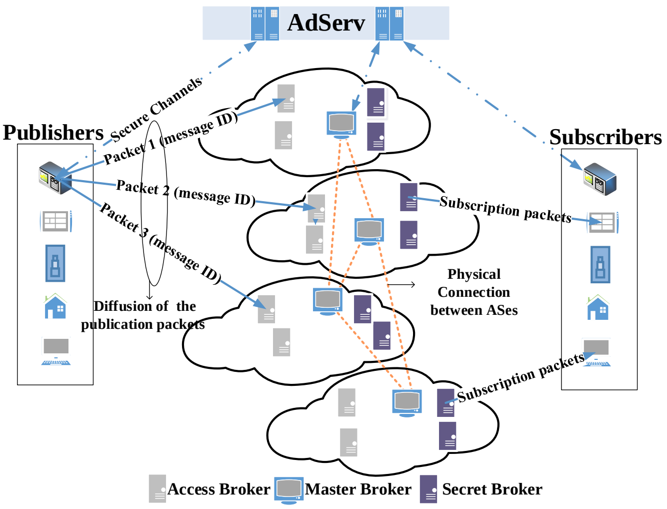


FIGURE 5.1: After obtaining the ticket from AdServ by using the secure channel, the publisher diffuses the data packets over N access brokers. Access brokers check the authenticator and hand them over secret broker(s) to check the validity of the ticket and distribute to the subscribers.

For devices that do not possess enough resources for asymmetric-key cryptography, we employ physical unclonable functions (PUF), combining symmetric-key and ID-based key cryptosystems [Sef+14]. This assumption is reasonable [Bak+11] considering the SG's relatively static communication node structure.

## 5.3 Developing the SeReCP Mechanisms

SeReCP's main goal is to provide: (a) a proactive DDoS attack protection by differentiating authorized/unauthorized traffic at the resource-rich broker nodes than being handled at the constrained subscriber nodes, and (b) a secure and reliable communication. SeReCP is designed for a messaging paradigm, providing E2E timely delivery guarantee using a lightweight pub-sub paradigm extending on [Bak+11] instead of data storage/querying [Kim+12] or complex event processing.

The main components of SeReCP are as follows (Fig.5.1):

**Publishers:** The devices that produce the data, which are required for the subscribers. This data can be a measurement for some applications or a command for some actuators. This data is signed and (if necessary) encrypted by the publisher.

**Subscriber:** The entity that needs the data to decide or to actuate depending on the application context.

**Administration servers (AdServ):** We consider these servers as trusted and robust for the DDoS attacks. They have three roles in the system: (1) Bootstrap node (2) Authorization server (3) Pub-sub system administrator. AdServ maintains all certifications of the nodes. When the nodes apply to AdServ to access the network, they obtain the security keys and other information regarding their respective role in the network by using a secure channel (public key or PUF).

**Brokers:** There are three type of brokers: (1) Access Brokers (AB), receiving the publications from publishers and validating their authenticators. (2) Secret brokers, transmitting the publications to the corresponding subscribers after verifying their tickets' authenticity, and (3) Master brokers, responsible for probing to the other clusters/master brokers and maintaining the network state.

After clustering all nodes according to their respective Autonomous System (AS) and geographical proximity (Fig. 5.1), AdServ dynamically chooses the "strongest" nodes as brokers for each cluster (versus employing dedicated brokers) in order to provide scalability. AdServ assigns the roles as publisher/subscriber/broker to each node. Every node in the network takes part as a publisher or/and subscriber or/and broker at the same time. AdServ informs the corresponding brokers about the publishers' advertisements[2] and their corresponding subscribers. The access broker IP addresses/IDs are also delivered to publishers in the initialization process by AdServ (the changes in the broker list are also delivered by AdServ, and we consider this to happen only infrequently).

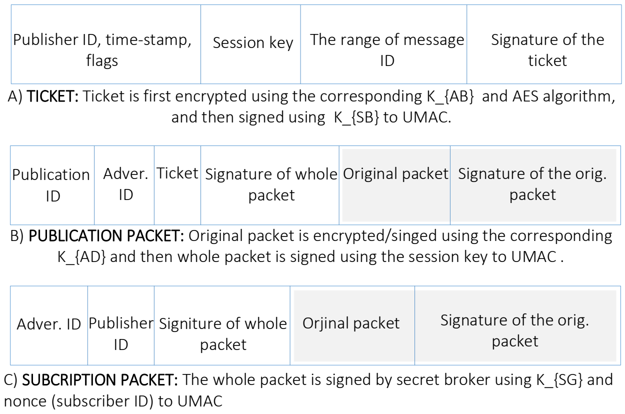
SeReCP provides security in two steps: (1) From publishers to the brokers, and (2) E2E (publishers to subscribers). We focus on the authentication between publishers and access brokers in order to avoid the DDoS attacks.

### 5.3.1 Authentication/Communication Protocol between Publishers and Access Brokers

When using the connection maintaining application or network layer state, the connection can be forced to reset by even short-time DDoS attack against access brokers. The loss of availability can cause some disturbance for the critical SG applications. In the construction of our authentication protocol between the publisher and brokers, we leverage Angelos, et al. [SK05]'s approach, which alleviates the necessity of application state at the brokers and provides high resilience against the DDoS attack. Taking advantage of this "stateless" authentication, publishers diffuse the data packets over N access brokers. In case of d percentage of N access brokers deny service due to the DDoS attacks (e.g., N = 100, d = 10), the dropped data can be checked/corrected in the subscribers by using forward error correction (FEC) or transmitting redundant replicated packets from the publisher (an acknowledgement mechanism can also be used for applications requiring relatively lower availability). However, [SK05] is based on a strong and potentially unrealistic assumption of fully trustworthy brokers. In [SK05], the authors also assume that the shared key encrypting the ticket cannot be obtained by an adversary in any way. An adversary who compromises the shared key can then mount a severe attack on the target (e.g., subscribers). Our approach removes these significant constraints.

**Key and Ticket Establishment:** When a publisher applies to AdServ using the secure channel in order to join the network, AdServ delivers three types of secrets to the publisher: (1) a secret key and an initial sequence number for each of their advertisements

[2] Advertisements are a type of publication and each of them has a group of subscribers. Each publisher can publish one or multiple advertisements. We assume that these assignments are managed by AdServ.


Publisher ID, time-stamp, flags
Session key
The range of message ID
Signature of the ticket
Publication ID
Adver. ID
Ticket
Signature of whole packet
Original packet
Signature of the orig. packet
Adver. ID
Publisher ID
Signiture of whole packet
Orjinal packet
Signature of the orig. packet

A) **TICKET:** Ticket is first encrypted using the corresponding K_{AB} and AES algorithm, and then signed using K_{SB} to UMAC.

B) **PUBLICATION PACKET:** Original packet is encrypted/singed using the corresponding K_{AD} and then whole packet is signed using the session key to UMAC .

C) **SUBCRIPTION PACKET:** The whole packet is signed by secret broker using K_{SG} and nonce (subscriber ID) to UMAC

FIGURE 5.2: Ticket and packet structures

(the subscriber groups also obtain the corresponding secret key and sequence number), (2) a session key, a 128 bit symmetric key, and (3) S (= number of the clusters) tickets valid for the corresponding access brokers. These secrets can be regularly updated depending on the criticality of the applications running on publishers[3]. Table 5.1 summarizes the keys used in SeReCP.

Fig. 5.2(A) illustrates the ticket consisting of a session key, a publisher ID, a range of publication ID numbers, a time-stamp, flags indicating ticket features, and a signature of the ticket. While the tickets are signed using a shared key $K_S$ (distributed to all secret brokers by AdServ) to UMAC [Bla+99], they are encrypted using the corresponding access broker's key $K_A$.

A packet sent to an access broker contains five parts, as shown in Fig. 5.2(B): (1) the publication ID, which is a monotonically increasing number and encrypted using the session key, (2) advertisement ID, (3) the corresponding ticket,(4) signature of the whole packet, which is generated using the session key, and (5) an original packet, which is encrypted and signed using a key ($K_O$) that is generated using a secret key and the sequence number of advertisement as inputs to a pseudo random function (PRF), more details in the Section 5.3.2.

**Communication Protocol between Publisher and Access Brokers:** When a publisher obtains the three secrets, it is ready to publish its data over access brokers. To transmit each packet, an access broker is selected in a pseudo-random manner. The selection is performed for each packet by using the last 4 digits of a random number as an index to the list of access brokers. The random number is derived using the session key and the publication ID as inputs to pseudo random generator (PRG). For each subsequent packet, the publication ID regularly increases, thus diffusing the packets over access brokers in a pseudo-random manner.

Once the access broker receives the packet from the publisher, it obtains the session key by decrypting the ticket of receiving packet using its respective $K_A$, and validates the packet's signature using the session key to UMAC. This provides a packet validation

[3] The tickets can be frequently updated by secret brokers with the same session key. However, the session key can be updated by AdServ using the public key.

with low computation and also protects from computational DDoS attacks. After the validation, the packet's publication ID (as decrypted using the session key) is checked to be within the acceptable range defined in the ticket and that it is larger than the last seen publication ID. The publisher ID and the last seen publication ID are only things stored by access brokers. In addition, the access broker validates whether the packet is routed to the correct access broker. To do this, it matches the respective access broker ID and the last four digits of the random number (derived by using the session key and publication ID along with PRG). These checks provide strong replay and repudiation attack protection with minimum memory occupation in access brokers. Finally, the access broker hands the packet containing the ticket in a decrypted form over to the corresponding secret brokers[4].

The ticket verification is delegated to the secret brokers to perform the authentication in separate nodes. The validity of the ticket is checked by fulfilling a UMAC validation using a shared key $K_S$ in secret broker(s). Moreover, the publication ID in the packet and the last seen publication IDs[5] are compared to determine whether there is an abnormal difference, thus dropping the packets fabricated with a random publication ID by a intruder compromising some of the access broker keys, $K_A$ and the shared key, $K_S$.

**Re-keying Procedure:** Ticket usage is restricted by the range of publication ID numbers and the time-stamp, i.e., 500 packets and 1-2 hours, thus avoiding reuse of the ticket by multiple zombies. Once a secret broker notices the ticket in the receiving packet is about to expire in terms of either the time or the range of publication ID numbers, it produces a new ticket by enlarging the range of publication ID number and signing using $K_S$. Then, the secret broker sends a new ticket to S access broker (randomly selecting one from each cluster). The access brokers issue new tickets to publishers after encrypting it using its respective $K_A$. However, in the re-keying process, the session keys of publishers are not changed. This can only be fulfilled by AdServ and we consider this done using the secure channel depending on the application criticality, e.g., hourly, daily.

**Multihoming-based Fast "Recovery":** SG applications have stringent latency requirements as delayed/lost messages could result in improper control operations. Taking these requirements into account, we propose multihoming based fast "recovery" mechanism, enhancing the approach of [SK05] by detecting/re-transmitting the dropped packets in a timely manner. To achieve this, we redesign the data diffusion mechanism by enabling publishers to forward every two consecutive data packets to two different network interfaces of a (pseudo) randomly selected pub-sub broker while spreading the data over the all access brokers. The access broker sets a timer on receiving one of the messages to check whether the other message arrives within the stipulated time period. If not, it requests the dropped message using the remaining functional network interface. We consider that the knowledge about which IP addresses belongs to the same access brokers is maintained by only publishers but not by public. Our experiments show that this mechanism provides high mitigation for delivery of the dropped packets without violating the defined maximum latency of the applications with high availability and latency requirement. This is

[4] AdServ constructs a hash table, mapping advertisement IDs with their corresponding secret broker ID, and then issues this list to all access brokers

[5] While secret brokers maintain publisher ID and the last seen S publication IDs of each publisher, access brokers save publisher ID and only the last seen publisher ID.

TABLE 5.1: The keys used by SeReCP

| **Keys** | **Usage** |
|---|---|
| Session Key | Signing/verifying the publication packets in publishers and access brokers, respectively |
| $K_O$ | Signing/verifying and encrypting/decrypting the original packet in publisher/subscribers, respectively |
| $K_A$ | Encryption/decryption of the ticket in the corresponding access brokers |
| $K_S$ | Verification of the ticket in the secret brokers |
| $SecretKey_i$ | Input to PRF to derivate $K_O$ |
| $K_G$ | Signing/verifying the subscription packet in secret broker/subscriber, respectively |

the case even when 30% of the access broker are under the DDoS attack. Furthermore, this mechanism does not introduce any additional overhead when there is no attack unlike replicated data delivery or FEC, and it provides much lower latency compared to classical acknowledge mechanisms e.g., TCP's cumulative ACK mechanism.

**How does SeReCP Overcome the Shortcomings of [SK05]:** To address the limitation of [SK05] of full trust in brokers, we develop a novel solution as: (1) SeReCP employs S tickets encrypted by S different $K_A$ rather than a ticket encrypted by a shared key. (2) The authentication in SeReCP is fulfilled in two separate brokers having different keys to check the ticket's diverse parts.

In SeReCP, an adversary whose aim is to launch a DDoS attack against the subscribers needs to compromise either all of S tickets maintained by publishers (this attack takes until ticket expire i.e., 500 packets and its result is relatively limited) or both $all of K_A$ and $K_S$ that are kept by the corresponding access brokers and secret brokers, respectively. Where an attacker compromises only some of S, $K_A$ and $K_S$ can suspect the packets to be fabricated by using compromised keys since their publication IDs are quite different from others. Thus, SeReCP renders a DDoS attack against the subscriber to be much harder to launch.

We employ a multihoming based fast "recovery" mechanism to meet high availability requirements without requiring replicated data delivery, as in [SK05]. Our experiments demonstrate that transmitting a replicated packet only to applications with a high availability requirement among the others, SeReCP meets the applications' availability requirements, even when 50% of access broker are under attack.

To provide secure communication between the brokers we consider a symmetric key pad similar to [Bak+11]. However, new pads can be regularly issued by AdServ using the secure channel, unlike [Bak+11]. This produces limited overhead, since we employ this method only between brokers but not across all nodes as in [Bak+11].

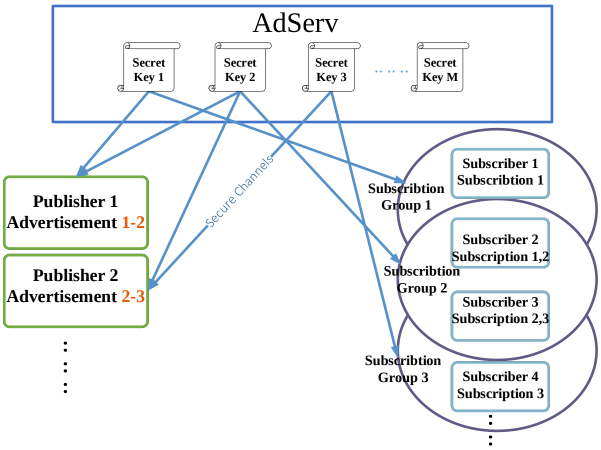

AdServ
Secret Key 1
Secret Key 2
Secret Key 3
Secret Key M
Secure Channels
Publisher 1 Advertisement 1-2
Publisher 2 Advertisement 2-3
Subscribtion Group 1
Subscribtion Group 2
Subscribtion Group 3
Subscriber 1 Subscribtion 1
Subscriber 2 Subscription 1,2
Subscriber 3 Subscription 2,3
Subscriber 4 Subscription 3

FIGURE 5.3: Secret Key Distribution

### 5.3.2 E2E Security and Group Key Management

**Group Key Management System:** SeReCP provides efficient DDoS protection by employing the above methods. However, to protect the E2E integrity and confidentiality between publisher and their subscribers, SeReCP requires a key management system. Hence, we introduce a group key management system (done by AdServ), which first identifies the advertisements of each publisher e.g., advertisements 1 and 2 for publisher 1, according to the application running on the device and the utility policy, and then issues the corresponding secret keys for the publishers, e.g., secret keys 1 and 2 for publisher 1. Additionally, AdServ clusters subscribers according to the advertisements they subscribe, e.g., subscribers 1 and 2 subscribe for advertisement 1 (a subscriber can also subscribe to multiple advertisements), and then issues the secret keys to the corresponding subscription group members e.g., secret key 1 for subscription group 1 (subscribers 1 and 2). This process is illustrated in Fig. 5.3.

**E2E Security:** Malicious intruders (during the transmission over the underlay network) can alter the data to launch a replay or repudiation attack against the subscribers. To cope with this, the publishers encrypt and sign each packet with $K_O$:

$K_O = f(SecretKey_i, SequenceNumber)$

where $f$ indicates PRF, $SecretKey_i$ is a symmetric key issued by AdServ for advertisement$_i$, and $SequenceNumber$ is the current sequence number of the packet of the advertisement$_i$. The subscribers use the pair of publisher ID and advertisement ID as an index to maintain the last seen sequence number in a list. Upon reception of a packet, using the publisher and advertisement IDs as index in the list, the last seen sequence number is obtained and then $SequenceNumber$ is calculated by adding 1 to it. The subscriber makes use of $SequenceNumber$ and $SecretKey_i$ to PRF in the process of derivation of the $K_O$.

The subscribers can authenticate the original packet performing a UMAC validation using $K_O$. In case of the packet is forwarded by an intruder to a subscriber not in the correct group, the packet cannot be decrypted by the subscriber without $K_0$. In addition, since $K_O$ is derived using the current $SequenceNumber$, launching a replay attack is impossible for intruders if they have no the $SecretKey_i$ and no $SequenceNumber$.

Upon receiving a packet from an access broker, the secret broker extracts the ticket and publication ID, but adds the publisher ID into the subscription packet. To protect from repudiation attack, secret brokers sign the subscription packet using $K_G$ and a nonce (subscribers' respective ID) to UMAC as illustrated in Fig 2(C). $K_G$ is issued by AdServ to secret brokers and the corresponding subscription group (it does not need to happen frequently e.g., daily). Thus, even if intruders compromise a subscriber's secrets ($SecretKey_i$, $SequenceNumber$, and $K_G$), they cannot fabricate a packet for the subscribers by spoofing the IP addresses (as if it is coming from the publisher over the secret brokers), since the intruder must know subscriber IDs (a random 32 bit value) of the target subscribers in order to embody the UMAC signature.

To alter a packet, an intruder would need to compromise (a) all secrets of the secret brokers, (b) members of that subscription group, and (c) obtain detailed knowledge of the network/nodes structure and data flow relationships. This is often unrealistic.

## 5.4 Security Analysis

We now present the security analysis for SeReCP.

### 5.4.1 DDoS Attack

Rendering a publisher inaccessible is possible by launching a DDoS against either access brokers maintaining the communication between publisher and subscriber or the corresponding subscribers by gaining elevated privilege on the intermediate brokers[6]. To cope with direct attacks on the brokers, SeReCP employs packet diffusion in addition to redundant packet transmission or FEC, thus providing a significant mitigation in case d% of brokers are under DDoS attack. Moreover, taking into account the stringent latency and availability requirements, we propose a multihoming based fast "recovery" mechanism, proving rapid re-transmission of dropped packets. In the second case of attacks, SeReCP employs S different tickets and two-step authentication to make a DDoS attack against the subscribers much harder, keeping in mind this is the type of attacks launched by attackers who are able to compromise some secrets of the nodes. Thus, to mount a attack on the subscriber, an adversary must compromise both $K_A$ and $K_S$ belonging to access brokers and secret brokers, respectively. This is a difficult task for an attacker.

### 5.4.2 Replay Attack

Replay attacks pose risks for both brokers and subscribers. If the system is vulnerable to replay attack, an attacker can re-send the same packet in order to mislead brokers and subscriber. However, SeReCP uses two dedicated countermeasures to protect the system from replay attacks. (1) Access brokers keep the last seen publication ID along with the publisher ID. Upon reception of a packet, they check if the publication ID in the

[6] The publishers and subscribers IP addresses are not public and they permit only some predefined IP addresses to communicate. Hence we assume a direct DDoS attack on the publishers and subscribers is not possible.

receiving packet is larger than the last seen one. (2) Publishers sign the original packet $K_O$ derived by using $SecretKey_i$ and $SequenceNumber$ to PRF. Upon receipt of a packet, the subscribers derives $K_O$ by using $SecretKey_i$ and $SequenceNumber$ to PRF. Then, the signature is checked using $K_O$ whether the packet is altered/fresh, or not.

### 5.4.3 Repudiation Attack

Repudiation attacks are a severe concern over group communication since an adversary who compromised the secrets of a member of a subscription group can fabricate packets by spoofing a publisher's and secret broker's IP addresses. SeReCP employs a $K_G$ for each subscription group. AdServ issues these keys to the secret brokers and the corresponding subscribers. They also sign the packets using the subscriber ID (as nonce) and $K_G$ to UMAC. This provides a capability for subscribers to check whether the packet originates from the secret brokers and, implicitly, the publisher.

### 5.4.4 Drop and Delay Attack

An attacker can drop some of the packet to lead to failure of the authentication mechanism. Particularly, the dependence on the monotonically increasing message ID in the system might cause the failures if such an attack occurs. To simply mitigate this issue, the subscribers keep a reasonable number of missing sequence numbers.

## 5.5 Evaluation

The main goal of SeReCP is to provide high communication availability between publishers and subscribers, even during the high-volume DDoS attacks in order to avoid inaccessibility to some critical devices, which can pose severe safety risks on power grids. In this section, we present an evaluation as to how well SeReCP meets these goals and its additional overhead levels in terms of latency and traffic.

We consider that the wide area network used by the utility to manage the SG covers a territory or a country. To obtain realistic results we employed NorNet Testbed [Dre15] that contains multihomed[7] nodes spread over all of Norway. We deployed daemons at 30 nodes with 2-3 network connections.

To measure the round-trip time latency between publishers and subscribers (when the brokers interpose), we deploy publisher and subscriber daemons at two different nodes and broker daemon at the all other nodes. In each experiment we change the tasks of all nodes till the results stabilize. By doing so, we obtain the results across deployments. Correspondingly, for availability measurements, we deploy the publisher and subscriber daemons on the same node in order to enable the implementation of diverse attack scenarios.

[7] A multihomed node has connections to multiple Internet service providers (ISP) via different network interfaces.

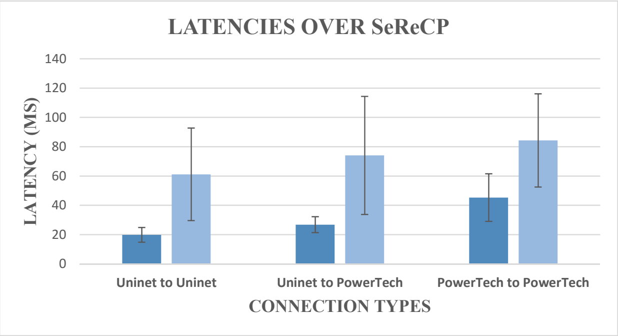


FIGURE 5.4: Latency results between publisher and subscriber as ISP connections. SeReCP (light blue/black) introduces only a 40ms additional latency compare to direct communication (dark blue/black)

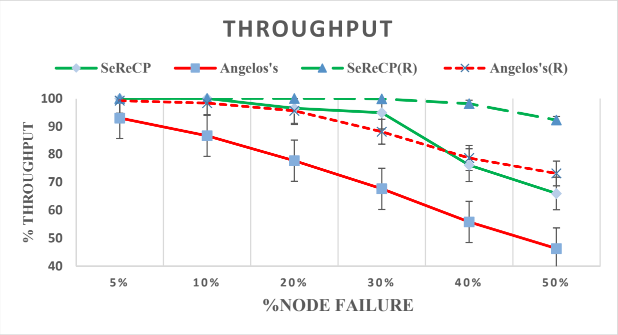


FIGURE 5.5: The normalized throughput results of SeReCP and Angelos [SK05]. In the case of transmitting replicated (R) packets, both SeReCP(R) and Angelos'(R) achieve higher network resilience.

Each SeReCP's publication packet (see Fig. 5.2(B)) contains 48 bytes of additional data (36 bytes ticket, 4 bytes publication ID, 4 bytes Advertisement ID, and the 4 bytes signature) over the original packet; the subscription packets include only 12 bytes additional data (Advertisement ID, publisher ID, and the signature are 4 bytes). Taking into account SG applications high availability requirements, this additional traffic constitutes a reasonable overhead.

Fig.5.4 shows the round-trip time latency between publishers and subscribers by comparing direct communication using UDP with the communication over the brokers using SeReCP. As we investigate whether our approach can be implemented for SG applications using the public networks, we present the actual latency results for both direct communication and SeReCP. In addition, although we obtain the results for each deployment case, since we see that the results mainly differ according to the ISPs (e.g., Uninet, PowerTech) between two ends, we illustrate the three representative combinations. The latency results show that SeReCP adds around 40ms latency in comparison to the direct connection. The ISPs' underlay infrastructures have significant effect on the latency. Although

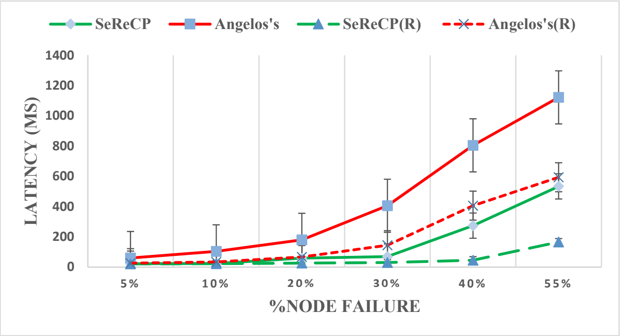


FIGURE 5.6: Latency measurements during the attacks. The transmitting replicated packet helps obtain lower latencies in both approaches

TABLE 5.2: Performance evaluation parameters

| Application | Availability | Latency | Priority |
|---|---|---|---|
| Wide Area Situational Awareness (WASA) | 100% | 200 ms | high |
| Real Time Pricing | 99.33% | 1150 ms | middle |
| Customer Information | 98.50% | 2000 ms | low |

SeReCP introduces an additional 40ms latency, the obtained latency values, between 60-80 ms, are reasonable for most of the SG applications which range at 200-2000 ms, as in Table 5.2 [US 10].

We first compare our approach with Angelos et al. [SK05] regarding throughput, in the normalized form, in the presence of DDoS attack. Throughput refers to the rate of successful message delivery. Fig. 5.5 denotes the successful delivery rates of SeReCP and [SK05] without/with duplicate packets in the case of different rates of failures in access brokers. Without duplicate packets, we see that while the connection performs well up to 30% failure of pub-sub brokers using SeReCP (recall that real time applications can perform well up to 10% packet drop by using UDP or TCP [Nah+01]), the performance drops after 5% failures for [SK05]. Utilizing our multihoming-based fast "recovery" mechanism, SeReCP delivers the dropped packets in time (i.e., the re-transmission time of the acknowledge mechanism of subscribers). On the other hand, in the case of sending duplicate packets, whereas SeReCP can maintain the connection without stalling up to 50% failure, the connection can perform well up to 20% failure for [SK05]. These results demonstrate that sending redundant replicated packets significantly enhances the throughput in the presence of DDoS attack. However, even by sending duplicate packets, [SK05] cannot introduce effectiveness as simple as SeReCP.

To assess the effectiveness of the approach, another important factor is latency in the presence of a DoS/DDoS attack. Fig. 5.6 shows the corresponding latency results for the experiments. By sending duplicate packets SeReCP provides reasonable latencies for

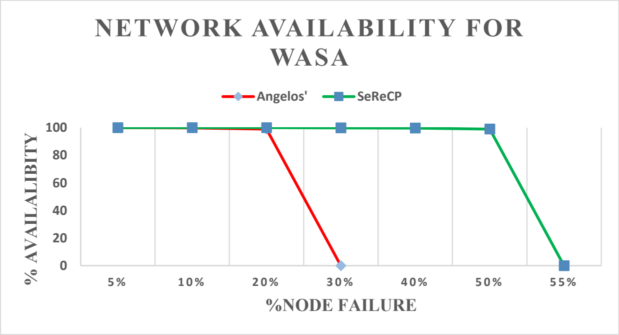


FIGURE 5.7: Network availability for WASA (high priority/critical)

the real-time applications[8] for up to 50% failures. [SK05] with duplicate packets, and similarly simple SeReCP, introduce similar curves and latencies up to 30% failures for the real-time applications.

We next evaluate the communication system to provide the required availability for SG applications. Availability refers to the rate of delivered packets that do not violate the application latency requirements. Therefore, a packet should arrive to the destination not only before the re-transmission time of the acknowledge mechanism, as is common practice, but also within the acceptable latency for SG applications. In light of the above results, we employ the duplicate packet method depending on the availability requirements of the applications. To do so, we categorize SG applications into three priorities with respect to availability and latency requirements (Table 5.2). We select three real SG applications [US 10] which represent general SG applications. The duplicate packet method is not used for Customer Information application. Note that while all packets are duplicated for WASA application, only 50% of packets (randomly chosen) are transmitted as duplicated for the Real Time Pricing application.

Using these three duplication methods (100%, 50%, and 0% duplication) for the corresponding applications, we obtain results regarding the network availability for each application in diverse failure rates of access brokers by employing SeReCP and [SK05]. Fig. 5.7 denotes that by employing 100% duplication, ([SK05] provides the required availability for WASA application for only up to 20% failure) SeReCP introduces the same availability up to 50% failure for WASA despite its strict latency requirement i.e., 200 ms. Fig. 5.8 depicts that although Real Time Pricing relatively has lower latency requirement (1150 ms), by duplicating only 50% of packets, the required availability can be provided up to around 30% and 20% failure by SeReCP and [SK05], respectively. Finally, without duplication of packets, SeReCP can still provide the required availability up to 30% failure due to the much lower latency requirement i.e., 2000 ms, [SK05] represents a sharp decline after 5% failure, as illustrated in Fig. 5.9. Overall, by employing the duplication of packets depending on the application requirements rather than duplication of all packets as in [SK05], SeReCP can introduce high resilience against DDoS attack and provide the required availability for each application at least up to 30% failure of access brokers.

[8] Real-time apps typically have 150-200ms latency [RGZ06].

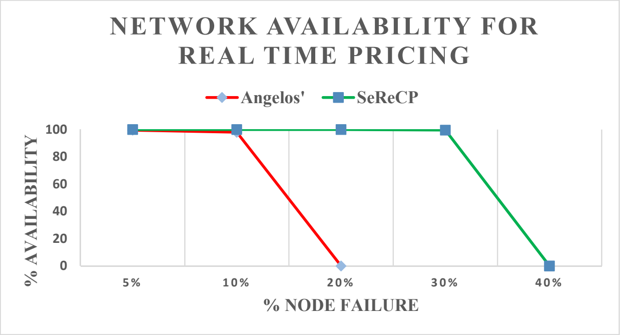


FIGURE 5.8: Network availability for RT Pricing (middle priority)

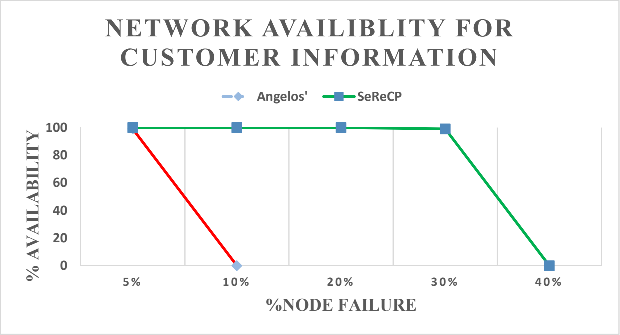


FIGURE 5.9: Network availability for Customer Information (Low priority)

In addition, if the first priority applications are safety-critical applications, then SeReCP sustains the required availability for up to 50% failures for these applications despite the second and the third priority cases shutting down after 30% failures.

**Discussion:** In our evaluation, we assess our approach for its ability to provide the required availability for real-time SG applications during the DDoS attack on the pub-sub brokers. We compare our approach with [SK05], which is also proposed to protect the E2E connection of real-time application from DDoS attacks. Firstly, we evaluate our approach in terms of its additional latency and overhead. The results denote that SeReCP's additional overhead and latency is reasonable for most SG applications. It is worth highlighting that, although the latency results, obtained the actual Internet infrastructure, are reasonable for most SG application, the ISP connections of the end hosts significantly affect the latency. Secondly, we evaluate the normalized throughput of SeReCP and [SK05] for real time applications. Without duplicate packets, whereas SeReCP can protect the E2E communication up to 30% failure, the connection stalls after 5% failure in [SK05]. This shows that SeReCP preserves the E2E communication even without duplicate packets up to 30% failures. With duplicate packets, SeReCP can maintain the same performance up to 50% failures. In the third evaluation, by employing different duplicate packet rates depending on SG applications' "priority", while SeReCP can provide the required availability for middle and low "priority" SG applications up to 30% failures, it provides the required availability for high "priority" appications up to 50% failure. If we consider pub-sub brokers size comparable to a typical Akamai setting (ca. 2500 nodes) [SK05], an attacker coordinating around 1,300,000 zombies can bring down 30% of access brokers. However, this volume of attacks are a small percentage of the DDoS attacks experienced in the past. Even in this situation, SeReCP can provide the required availability for high priority (safety critical) applications despite the failure of the low and middle priority ones. This substantiates SeReCP to be a promising approach to make the public network usable for SG applications in a secure and reliable manner.

## 5.6 Conclusion

The proposed SeReCP approach provides a proactive DDoS attack defense by using a pub-sub infrastructure in addition to providing secure E2E data delivery in a light-weight manner, considering SG applications requirements. To maintain the availability in case of a targeted or sweeping attack on an access broker maintaining the communication between a given publisher and its subscribers, we employ a packet diffusion mechanism, spreading the packets over access brokers in a pseudo-random manner due to its token-based authentication mechanism. Moreover, we propose a multihoming-based fast "recovery" mechanism, enhancing the packet diffusion mechanism by detecting and requesting the dropped packets in still access brokers rather than in the subscribers, thus enabling the system to meet the stringent latency requirements of SG applications. Finally, to preserve E2E confidentiality and integrity of the data, we propose a group key management system, which provides role-based access rights for both publisher and subscriber in addition to guard from replay attacks.

To assess the effectiveness of our approach against DDoS attacks, an actual SG test-bed was used. The experiments show that SeReCP introduces a small 40ms overhead acceptable for most SG applications. Furthermore, we conducted an DDoS attack by randomly bringing down access brokers, and compared the availability across SeReCP and state of the art [SK05]. The results showed that by assigning the rate of duplicate packets depending on the applications availability and latency requirements, SeReCP provides the required availability, up to 30% failure and 50% failure, of pub-sub brokers for the application with relatively lower requirements and for the application with stringent requirements, respectively. This showed that SeReCP, with its lightweight mechanism, can resist attacks much larger than we have seen to date. Overall, these results validate SeReCP to provide the required security for SG applications in instances where the SG uses public networks.

# Chapter 6

# Securing the Cloud-Assisted Smart Grid

To manage millions of SG devices and to handle large amounts of data in a reliable, scalable, and cost-effective way, the SG utilities increasingly extend their communication-based management system to the advocated cloud computing platform. These cloud platforms enable reliable and on-demand access to varied computing resources [BI14; BMR15]. Despite the advantages of the cloud, its usage of the public network and shared resources can expose the SG to security risks considering both the cyber and physical systems, e.g., power grid/appliances. In particular, DDoS attacks represent a major threat to the SG applications running in the cloud, considering SG applications' stringent latency requirements (in the range of 100 ms to 5 s) and reliability requirements (99.00 %–99.99%) [BI14].

As availability constitutes a safety property for SG applications (especially for control functions), deploying proactive defense mechanisms becomes indispensable for SG communication. Proactive defense mechanisms, e.g., moving/hiding the target [Sta+05; FPT12; Jia+14; DS17b], are introduced as countermeasures increasing the cost placed on the attacker to overwhelm the victim's resources. However, since these proactive defense mechanisms are mainly designed to mitigate DDoS attacks in typical web applications, they are not suitable for the SG applications' context due to the SG specific requirements of high availability and responsiveness [SK05].

To fill this gap, we propose a hybrid hierarchical cloud-extension concept (HHCEC), which is a SG-relevant cloud-assisted architecture. HHCEC provides ultra-high responsiveness and security with its (a) hybrid and geographically dispersed structure, and (b) specialized broker-based publish-subscribe communication system. Second, we propose a novel approach termed Port Hopping Spread Spectrum (PHSS), which acts as a strong defense against transport and application layers DDoS attacks, as well as the volume-based DoS/DDoS attacks, against the broker servers. PHSS is equipped with two distinctive features: (1) *port hopping*, changing the open port of the broker server as a function of the time and a secret shared between the broker server and the publishers[1], and (2) *packet spreading*, diffusing consecutive data packets over a number of broker servers versus a single broker server. This approach enables PHSS to instantiate replica broker servers to take over the attacked broker servers without blocking all traffic by taking advantage of the rapid-elasticity characteristic of the cloud.

[1] The terms client/publisher and server/broker are interchangeably used in the rest of the chapter. In addition, while every SG device/application server can be a publisher and/or subscriber, the brokers are dedicated servers for their respective roles.

The existing *port hopping* approaches assume that the secret (a cryptographic seed), if compromised, can be renewed by an Authorization Server using a public key-based re-keying approach, which requires high unaffordable computation complexities for different SG entities (cf. [LT04; FPT12]). Moreover, when using public key-based approaches, if the secret is compromised, the adversary can mount an attack on the broker to render inaccessible them. In such cases, the broker servers become unavailable for all publishers during the re-keying process, which in turn severely impacts the SG applications' service provision. Accordingly, an efficient reactive mechanism is highly necessary in order to minimize the impact of DDoS attacks against the open ports of broker servers, as a result of compromising the secret.

To address this issue, we introduce (1) *a token-based authentication mechanism* that allows for a light-weight periodic transmission of the secret to each client (publisher), and (2) *a shuffling-based containment mechanism* that quarantines *malicious clients*, without rendering the attacked broker server inaccessible. To do this, the containment mechanism repositions/shuffles the clients over the ports of the broker server with a negligible overhead.

To assess the efficiency of the proposed approach, we construct a proof-of-concept prototype using EC2-micro instance [Ama16] and the PlanetLab (http://planet-lab.org) test-bed. We evaluate PHSS's effectiveness in providing network availability by using the *shuffling-based containment mechanism* against DDoS attacks using the compromised secret. We also compare our approach with the public key-based re-keying method used by the existing *port hopping* mechanisms. Our results show that by containing the impact of the DDoS attack using the compromised secret in a notably shorter time period, PHSS provides high network availability of over 98% during the attack versus the typical 6̃0% availability achieved by using the public key-based re-keying method. Furthermore, after assessing the overhead (in terms of broker server throughput and response latency), the experimental results show that our proposed mechanism causes neither significant throughput degradation (i.e. <0.01% throughput degradation) nor additional latency. To summarize, our contributions are:

- A SG-relevant cloud extension, termed HHCEC, which utilizes a hybrid and geographically dispersed structure to meet the responsiveness and reliability requirements of SG applications.
- A strong proactive DDoS attack defense mechanism, called PHSS, which dynamically changes the open ports of the broker servers to efficiently drop the invalid packets in the firewall. Furthermore, PHSS diffuses consecutive data packets over a number of servers versus a single server in order to rapidly recover the attacked system in the cloud.
- A token-based authentication mechanism to prevent secrets from being compromised, as well as *a shuffling-based containment mechanism* to contain the damage of the DDoS attack utilizing the compromised secret in a shorter time.
- A proof-of-concept platform using Amazon's EC2 [Ama16] and PlanetLab nodes to evaluate our approach in terms of the availability of service provision for the SG applications over DDoS attacks and the overhead imposed by our approach.

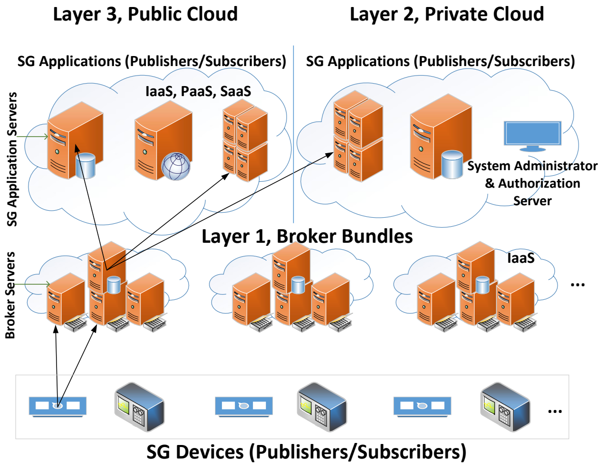

Layer 3, Public Cloud
Layer 2, Private Cloud
SG Applications (Publishers/Subscribers)
SG Applications (Publishers/Subscribers)
IaaS, PaaS, SaaS
SG Application Servers
System Administrator & Authorization Server
Layer 1, Broker Bundles
IaaS
...
Broker Servers
...
SG Devices (Publishers/Subscribers)

FIGURE 6.1: Hybrid Hierarchical Cloud Concept (HHCEC)

The remainder of this chapter is organized as follows: Section 6.1 details the security and attack models. Section 6.2 introduces the HHCEC, followed by the PHSS approach in Section 6.3 and their evaluation in Section 6.4.

## 6.1 Models, Goals and Assumptions

We now describe the system and attack models in addition to the security goals and assumptions driving our approach.

### 6.1.1 System Model

We consider the established SG model where the utility relies on a heterogeneous network (i.e., public and private) and a hybrid cloud infrastructure, as illustrated in Fig. 6.1. We posit that the utility takes into account the availability requirements of SG applications and the cost-effectiveness while buiding the network and the cloud infrastructure. As pub-sub systems inherently provide scalability and proactive DDoS attack defense for the constrained SG devices, we employ a broker-based pub-sub system on HHCEC.

Our system is designed for a message passing paradigm where publishers transmit their data to brokers, and the brokers subsequently deliver the data to the subscribers, e.g., the SG applications and devices.

A system administrator that considers the geographical distance and the latency between the brokers and publishers, assigns each publisher to a broker bundle. Furthermore, the system administrator monitors/maintains the latency between the broker bundles and the publishers to re-assign the publishers to a new broker bundles in case of detecting intolerable latency.

Our approach distinguishes between authorized and unauthorized traffic before it reaches the resource-constrained SG devices by countering the DDoS attacks in the well-provisioned broker servers in terms of computation capacity and bandwidth. Thus, the proposed approach adopts the functionality of a distributed firewall to avoid any overwhelmed link to the SG nodes.

### 6.1.2 Security Goals

Our security objective is to guarantee delivery of the publisher data to the corresponding subscribers within the time window specified in the application requirements. To meet the high availability requirements of the critical devices, any DDoS attacks using privilege of the compromised secret must be eliminated or at least mitigated by containing the impact of the DDoS attack in a reasonable time period.

### 6.1.3 Attack Model

The broker servers, which maintain the communication between the publishers and the subscribers, can be targeted by an attacker with the intent to render critical devices inaccessible.

We consider a robust threat model where the attacker:

- Controls a minority of publishers/clients that behave maliciously, referred to as *malicious clients*.
- Can eavesdrop, capture, drop, resend, and alter some of the traffic between the publisher and the brokers to launch DDoS attacks against brokers.
- Can disclose the secret of the *malicious clients*. Accordingly, the attacker can launch a DDoS attack against the open port of the broker server.

### 6.1.4 Assumptions

As in contemporary attack models, we assume that (a) publishers obtain only the IP addresses of the broker servers and (b) valid certificates are issued by a Certification Authority to all brokers/publishers/subscribers[2] and to Authorization Servers in a secure way. Since we focus on the broker defense against DDoS attacks, the protection of the Authorization Server is beyond the scope of the paper.

### 6.1.5 Limitations

It is worth mentioning that the pathological case of attackers that can fully saturate the Internet backbone links for HHCEC is beyond the scope of this approach.

[2] We suppose that our approach is deployed on the SG devices that possess enough resources for asymmetric-key cryptography.

## 6.2 Cloud Computing for Smart Grid

In this section, we review the utility of the cloud for SG applications. Afterwards, we highlight the existing limitations behind the direct usage of the cloud structure in the SG context. Finally, we describe the technical details behind our proposed cloud-assisted architecture that addresses such limitations. We also present existing approaches related to the adoption of cloud computing for the SG in Section 2.3.

Typically, the realization of smart grids causes a very large increase in data volume due to the implementation of real time metering, monitoring and pricing applications. This massive data also needs to be collected and processed in real time. As control decisions are solely based on such data, it significantly affects the stability and reliability of the SG. Thus, data parallelism and high computational capabilities play key roles in analyzing and processing this large amount of data [BI14].

However, the variable resource needs of the SG applications, as matching the varying SG operational behavior, is a challenge for the SG utilities. These applications operates in idle mode on dedicated hardware until a particular situation occurs, e.g., detected abnormality in the grid voltage. This results in inefficient resource usage. Consequently, using a cloud computing platform becomes a viable solution to address these issues due to its featured rapid elasticity [BI14]. In fact, as SG applications have strict availability, response time and security requirements, the direct usage of the cloud for the SG encounters the following limitations [BI14].

1. *Guaranteed Service Availability*: while availability, real-time responsiveness, guaranteed consistency, and fault tolerance are the properties indirectly affecting the safety of the SG, they are typically liveness properties for cloud service providers. Avoiding single point of failure scenarios and potential communication bottlenecks is a must to achieve high availability in the use of the typical cloud for the SG.

2. *High Responsiveness*: for data efficiency in the Cloud, an outer layer of the Cloud can be built to provide data aggregation, multiplexing towards the main applications. This would eliminate the potential data transfer bottleneck and contribute to the responsiveness of the applications.

3. *Data Confidentiality*: some SG applications require high confidentiality to prevent data sharing or information leakage, which the cloud service providers typically do not provide. On the other hand, some SG applications need relatively less security protection. This security diversity forces the SG utility to employ diverse resources with different security assurances in the cloud adoption.

In the next section, we introduce an SG related cloud-extension concept that overcomes the above-mentioned limitations resulting from the direct usage of the cloud in the SG context.

### 6.2.1 Hybrid Hierarchical Cloud Concept (HHCEC) for the SG

Providing the specific SG requirements is the driver behind proposing a 3-layer HHCEC cloud-assisted architecture, as depicted in Fig. 6.1. *The first layer* is composed of *Broker*

*Bundles*, which are dispersed based on the grid topology throughout the utility territory. Each *Broker Bundle* can consists of several broker servers. The goal of the *Broker Bundles* is to handle the time-sensitive data in a location surrounding the source rather than in a remote center. This layer provides an interface to support data concentration, data preprocessing, short-term redundant data storage (using replica shards), proactive defense against DoS/DDoS attacks and multiplexing for applications running in the other layers. Since this layer is composed of public cloud infrastructures, data requiring high confidentiality is saved/forwarded in an encrypted form so that it can be decrypted solely by the destination [DS17b].

*The second layer* is an in-house private cloud infrastructure comprised of application servers that process data requiring high availability and/or confidentiality. This layer controls and monitors the *Broker Bundles* of the first layer and assigns the SG devices to the corresponding *Broker Bundles*. Furthermore, the second layer accommodates applications performing analysis, batch processing, permanent archiving, and visualization functions.

Applications/data requiring less security are delegated to *the third layer*, which consists of public cloud infrastructure(s). This layer communicates and shares corresponding data with third parties.

While the public clouds in the first layer are built using the infrastructure as a service (IaaS) model, the public clouds in the third layer can be constructed using IaaS, platform as a service (PaaS), and/or software as a service (SaaS) models depending on the applications' requirements. On the flip side, the private cloud in the second layer is located in-house to strictly ensure no physical data access by third-party.

We utilize a pub-sub system as a communication platform on HHCEC for SG applications. The brokers of this pub-sub system reside in the *Broker Bundles*. The communication between the SG devices and the layers 2 and 3 is not direct, but goes through the *Broker Bundles*, as shown in Fig. 6.1. The application servers and the SG devices can be either publishers or subscribers. We assume that their roles are assigned by a system administrator residing in the in-house cloud architecture provided by the second layer.

As a summary, the proposed cloud-assisted architecture HHCEC accommodates the pub-sub based SG communication platform while taking into account the SG security requirements. Next, we describe the proposed DDoS attack defense mechanism, PHSS, that guards the broker servers residing in the *Broker Bundles*.

## 6.3 Port Hopping Spread Spectrum (PHSS)

In this section we detail the technical concepts behind our proposed defensive mechanism, required for securing the aforementioned cloud-assisted SG structure. The proposed PHSS consists of two main mechanisms: (1) *port hopping* and (2) *packet spreading*, which provide for a robust DDoS protection for the pub-sub broker servers.

### 6.3.1 Port Hopping

The *port hopping* system of PHSS periodically changes the open port of the broker server over time, as illustrated in Fig. 6.2, according to a pseudo-random sequence known by

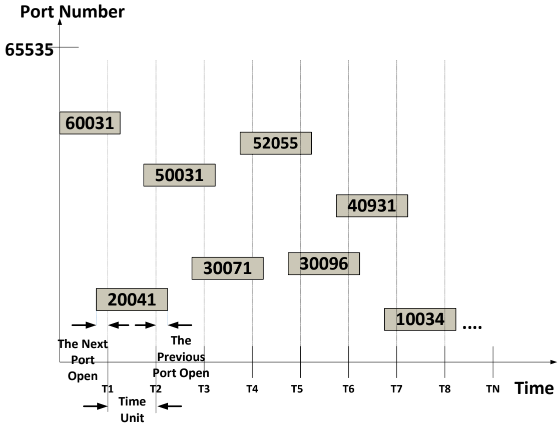


FIGURE 6.2: Port Hopping Approach

both the clients and broker server. This sequence is produced by the broker and the clients using a shared secret, the time and a pseudo random function (PRF). In addition, to avoid clients sending packets to the previous or the next port due to time sync error or communication latency, the broker server leaves those ports open for a certain time period, corresponding to the maximum latency between the broker and the clients [FPT12]. (see Fig. 6.2). In this context, two challenges must be considered: (1) time synchronization attacks or clock drift [FPT12] and (2) compromising of the shared secret by the attacker.

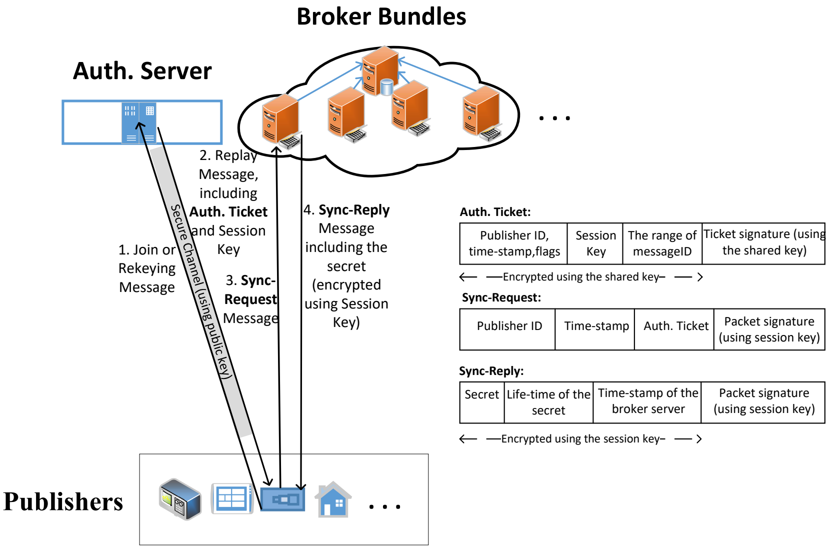
### Time synchronization attacks/clock drift

To address the first challenge, PHSS takes advantage of a secure synchronization approach between the brokers and clients. To perform the secure synchronization, each client first obtains a respective session key (128 bits symmetric key) and an authentication ticket (which also includes the session key) from an Authorization Server via a secure channel during the process of joining the network (see messages # 1 and # 2 in Fig. 6.3). The authentication tickets (akin to Kerberos ticket [NT94]) are encrypted and signed using a shared key[3] known by the broker servers. The session key of a given client is derived by decrypting the authentication ticket (inside the sync-request message of the client) by using the shared key in the broker servers. Thus, the sync-request message's integrity is checked using the session key by the broker servers.

To synchronize the secret and time, each client sends a sync-request message to the broker including the respective authentication ticket and time-stamp. As a response, a sync-reply message, including the current secret, the life-time of the secret and a time-stamp, is issued by the broker server. The sync-reply messages are issued to each client by encrypting and signing with the respective session key, derived by decrypting the authentication ticket inside the sync-request message. This synchronization process is illustrated in Fig. 6.3 (3. and 4. messages).

A client receiving the sync-reply message can synchronize the time with the broker server, as reported in [FPT12]. The life-time of the secret is randomly generated to avoid

[3] A symmetric key.


Broker Bundles
Auth. Server
. . .
2. Replay Message, including Auth. Ticket and Session Key
Secure Channel (using public key)
1. Join or Rekeying Message
3. Sync-Request Message
4. Sync-Reply Message including the secret (encrypted using Session Key)
Publishers
. . .

Auth. Ticket:

| Publisher ID, time-stamp,flags | Session Key | The range of messageID | Ticket signature (using the shared key) |
|---|---|---|---|

⟵ —Encrypted using the shared key– ⟶

Sync-Request:

| Publisher ID | Time-stamp | Auth. Ticket | Packet signature (using session key) |
|---|---|---|---|

Sync-Reply:

| Secret | Life-time of the secret | Time-stamp of the broker server | Packet signature (using session key) |
|---|---|---|---|

⟵ —Encrypted using the session key– ⟶

FIGURE 6.3: Authentication and synchronization protocol

synchronization attacks. Before the end of the lifetime of the current secret, each client issues a new sync-request message to the broker server to derive a new secret and time-sync info[4]. The regular re-synchronization employed by our approach provides protection against clock drift and time synchronization attacks, which are main concerns in the existing *port hopping* approaches [LT04; FPT12].

### Shared Secret Compromise by the Attacker

Another concern associated with the second challenge is the compromise of the secret shared among all clients, which poses a high security threat for the system. The existing *port hopping* approaches use a PRF and a long-term clients secret, which increases the risk of attacks [LT04; FPT12]. As a consequence of compromising the secret, SG applications would experience an unacceptable degradation of availability until new secrets are issued to all clients via the secure channel (using a public key). To address this issue, in PHSS, each client regularly requests the current secret from the broker server, as mentioned above.

The regular renewal of the secret by using the token-based authentication provides a limited mitigation since the attacker can continuously compromise the clients' secrets and thus, launch a direct DDoS attack against the open port. In this case, the containment of the damage of the attack on the broker server is only possible by retaining *malicious clients* in quarantine.

[4] The synchronization is fulfilled a few times in a day by each client. The overhead of this process is negligible in comparison to the daily traffic of client/broker server.

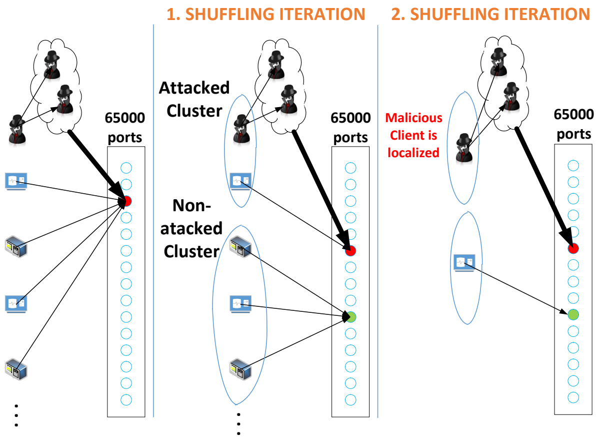


FIGURE 6.4: Port *Shuffling*

***Shuffling-based Containment Mechanism.*** We develop a *shuffling-based (repositioning) containment mechanism*, which contains the impact of *malicious clients* by localizing /quarantining them and then renewing their keys via Authorization Server, as illustrated in Fig. 6.4. The shuffling idea is roughly inspired by [Jia+14], but our mechanism does not require moving target servers and additional servers. In the *shuffling-based containment mechanism*, when the broker server detects the DDoS attack on the open port[5], it randomly shuffles and splits all clients $N$ into $p$ clusters by considering that all clients are suspicious clients $N_s$, ($N_s$ = $N$). New secrets[6] are then transmitted to each of the $p$ clusters. This process is simply called a *shuffling* iteration. An overview of the variables and constants used in the shuffling-based process is given in Table 6.1. After the clients start using their new secrets, the port(s) under attack indicate that the corresponding secret(s) are compromised. The clients who do not use these compromised secrets are removed from $N_s$[7]. Then, the clients of $N_s$ are shuffled and re-clustered by issuing new secrets for each new cluster. This technique progressively quarantines the *malicious clients*, which provides a quick localization of the *malicious clients* $c$ without disturbing all traffic. The number of *shuffling* iterations is denoted as $x$. To investigate the effects of $p$ and $c$ on the number of *shuffling* iteration $x$ (indicating also the containment duration), we perform a mathematical analysis as follows:

$$|N|/(p/c)^x \leq 1 \tag{6.1}$$

**Lemma.** *For a fixed $N$, if $|N|/(p/c)^x \leq 1$, then the compromised clients $c$ are localized in $x$ shuffling iterations by splitting the $N_s$ into $p$ clusters in each shuffling iteration.*

[5] To detect the attack we simply probe the port periodically, but more complicated methods can be used for the detection like [GP01].

[6] For each secret, the broker server concurrently opens the corresponding ports. A client using a given secret communicates over the port opened for that secret

[7] The benign clients can continue the transmission over the last issued secrets/ports without disturbing their traffic.

TABLE 6.1: Variables and Constants Definition.

| Symbol | Definition |
|---|---|
| N | The set of clients |
| $N_s$ | The set of suspicious clients |
| p | The number of clusters/secrets/open ports |
| x | The number of *shuffling* iterations |
| c | The number of *malicious clients* |
| $S_a$ | The set of secrets used by attacked ports |

**Proof.** To localize a *malicious client* in $x$ *shuffling* iterations, first, $N_s$ is set equal to $N$ ($N_s = N$) and then it is split into $p$ clusters ($p$ is equal to $|N|^{\frac{1}{x}}$). The broker server issues a different secret for each cluster. After the first *shuffling* iteration, the clients of the cluster(s) whose secret(s) are not used to launch an attack on the corresponding port(s) are removed from $N_s$. This iteration continues until $|N_s| \leq p$, and a different port is assigned to each suspicious client, which enables to localize the *malicious client*. In addition, if $c \geq 1$, $N_s$ is further split into p clusters in each clustering/*shuffling* iteration, and $p$ is assigned to ($p = |N|^{\frac{1}{x}} * c$).

A speedy localization of the *malicious client(s)* minimizes the loss of network availability. To this end, in the extreme case, we can assign each client to a different cluster, namely issuing a different secret per client ($p = |N|$), and thus finding the malicious one after a *shuffling* iteration ($x = 1$) based on the above lemma. However, opening a large number of ports poses a high risk of vulnerability to attacks that target the entire port range. In addition, building larger clusters in each *shuffling* iteration, e.g., splitting into two clusters ($p = 2$) in each *shuffling* iteration, increases the duration of the containment, thus affecting the network availability. Thus, we need to localize the *malicious clients* $c$ in a minimum number of *shuffling* iterations $x$, and open a minimum number of ports $p$ (equal to the number of the clusters and the issued secrets) in each *shuffling* iteration. To minimize the two parameters ($p$ and $x$) for $N$ clients, we create a corresponding optimization problem:

$$\text{minimize } A(p,x) = p * x \tag{6.2}$$

$$\text{subject to } |N|/(p/c)^x \leq 1 \tag{6.3}$$

To find the minimum values of $x$ and $p$, inequality (6.3) is expressed as

$$|N|/(p/c)^x \leq 1 \Longrightarrow |N| \leq (p/c)^x \Longrightarrow p \geq c * |N|^{1/x} \tag{6.4}$$

and the result is substituted into equation (6.2) in order to express $A(p,x)$ as a function of one variable:

$$A(x) = (c * |N|^{1/x}) * x, \;\; x \neq 0 \tag{6.5}$$

To compute the minimum value of (6.5), the Closed Interval Method [HW03] is used. We have to solve $A'(x) = 0$. Thus,

$$c * (|N|^{\frac{1}{x}} - \frac{|N|^{\frac{1}{x}} \ln(|N|)}{x}) = 0,\ x \neq 0 \tag{6.6}$$

Solving the above equation gives

$$x = \ln(|N|) \tag{6.7}$$

Substituting the solution (6.7) into (6.3) results in $p = \sqrt[\ln(|N|)]{|N|} * c$.

**Algorithm 1** Containment Algorithm

```
**Input:** A set $N = \{n_1, n_2, \ldots, n_i\}$ of clients, $c = 1$ as the first estimation
**Output:** Suspicious clients $N_s = \{n_{s1}, n_{s2}, \ldots, n_{sj}\}$ equal to compromised clients
$N_s \leftarrow N$
$(p, x) \leftarrow$ OPTIMUM$(Ns, c)$
CLUSTER$(Ns, p)$
while $|N_s| \geq p$ do                ▷ if $|N_s| \leq p$, the compromised ones are contained
    Check the ports to find the attacked ones.
    Remove the clients not using the attacked ports/the secrets $S_a = \{s_{a1}, s_{a2}, \ldots, s_{ak}\}$ from $N_s$
    if $c \geq |S_a|$ then CLUSTER$(Ns, p)$
    else
        $c \leftarrow |S_a|$
        OPTIMUM$(Ns, c)$
procedure OPTIMUM$(Ns, c)$                ▷ finds min p and x
    $x = \ln(|N_s|)$
    $1 > |N_s|/(p/c)^x \implies p = \sqrt[\ln(|N_s|)]{|N_s|} * c$
    return $p, x$
procedure CLUSTER$(N_s, p)$
    Randomly split $N_s$ into $p$-clusters and then issue $p$-secrets to the corresponding clients
```

### Adaptive Algorithm

We embody an adaptive optimization algorithm, which sets $c = 1$ and then computes the optimum $p$ and $x$ by solving the optimization problem above. After the execution of each *shuffling* iteration, if the number of compromised secrets is higher than $c$, the algorithm increases the number of issued secrets (clusters) $p$ based on the number of compromised secrets $c$[8]. The pseudo-code of the optimization-based containment algorithm is shown in Algorithm 1.

To summarize, in order to address the clock drift and compromising the secret key in the *port hopping* mechanism of PHSS, we develop *a token-based authentication mechanism*

[8] An intelligent attacker who can pause his/her attack over time and/or cooperate with the others cannot evade this containment algorithm but might delay it.

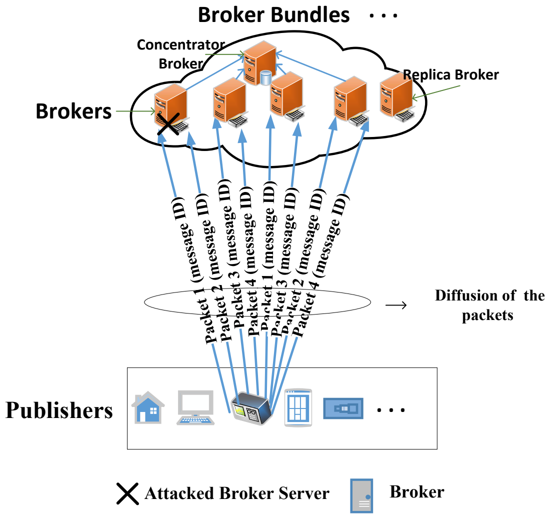


FIGURE 6.5: Packet Spreading

and *a shuffling-based containment mechanism*. The idea behind the token-based authentication is to complicate the compromise of secrets. The *shuffling-based containment mechanism* is further introduced to localize the compromised secrets without rendering the broker server inaccessible for all the clients, unlike typical *port hopping* [LT04; FPT12] or moving target mechanisms [Jia+14].

### 6.3.2 Packet Spreading

An attacker who controls a larger Botnet can bring down targeted brokers by flooding their entire ports and thus overcoming the *port hopping* mechanism. In such a case, the time period for re-establishing the connection could violate the availability requirements. To address this issue, we employ the data spreading mechanism [SK05; DS17b], which transmits by spreading consecutive data packets to broker servers within a *Broker Bundle* in a pseudo-random manner, as illustrated in Fig. 6.5. As shown in the figure, a *Broker Bundle* might consist of normal brokers, concentrator brokers and replica brokers. The role of the concentrator broker is to reassemble the packets received from the normal brokers. When some of the broker servers are brought down by the DDoS attack, we employ transmitting duplicate packets methods to "recover" the dropped data and meet the availability requirements. In that way, the dropped packets do not affect the reassembling process, as the concentrator broker uses the duplicate packets for reassembling.

Moreover, utilizing the rapid-elasticity characteristic of the cloud computing, new/ready replica broker server(s) are instantiated to take over the attacked broker server(s). This provides an efficient attack mitigation also in cases of persistent threat. The IP addresses of the new replicas can be delivered to the publishers in an encrypted form by performing a process similar to the sync-process.

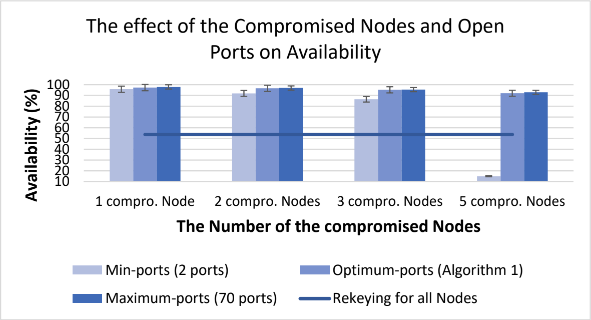


FIGURE 6.6: Implementation of PHSS on EC2 instance/server for 21 PlanetLab nodes

## 6.4 Evaluation

In order to validate and provide realistic results on the efficiency of the proposed approach, we build a proof-of-concept prototype which consists of two EC2 micro instances (EC2) [Ama16] and 21 PlanetLab nodes. To represent the SG applications with their strict requirements, we deploy a pseudo-state estimation application, which requires a latency of less than one second ($< 1s$) and a minimum of 30 samples per second [WYB15] for a power grid that spans continental Europe.

Hence, we employ all the properly functioning PlanetLab nodes (21 nodes) in Europe as publisher clients of the SG and two EC2 instances in EU-Central-1 (Frankfurt). The first EC2 instance represents a broker server in a *Broker Bundle*, while the second EC2 instance is a subscriber running the SG application in the third layer of HHCEC.

### 6.4.1 Evaluation Metrics

The evaluation metrics used to assess our approach are availability, and throughput and latency overheads.

1. Availability: As responsiveness is a dominant concern for SG applications, we focus on network availability that refers to the success rate of timely delivery of the messages from SG publishers to subscribers over the broker server. This metric is used to measure the level of network availability between the beginning of the attack exploiting the compromised secret and the containment of the impact of the attack. For the containment of the impact of the attacks we use PHSS's *shuffling-based containment mechanism* and the traditional approach, which launches a re-keying process for all the clients using public key. Then, we compare their efficiency in providing availability during the same attack period.

2. Throughput and latency overhead: Throughput is defined as the successful response rate of the broker server for the pseudo-state estimation application. The throughput overhead refers to the throughput decrease caused by PHSS on the broker server by comparing it with the simple transmission overhead. Furthermore, the additional latency imposed by PHSS is used as metric in the evaluation of our approach.

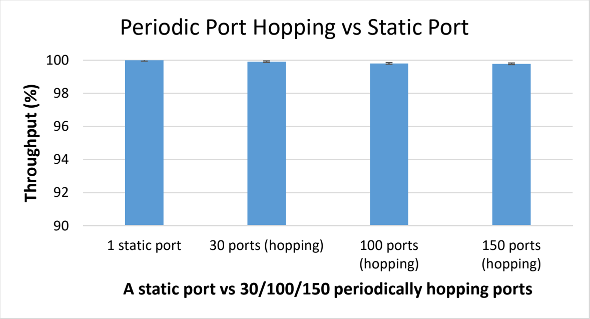


FIGURE 6.7: The effect of PHHS on the Throughput

### 6.4.2 Proof-of-concept Prototype-based Evaluation

Our proposed software architecture is a middleware between the network stack and the pub-sub layer in the broker servers and publishers. The middleware in the broker server (i.e., the server stack) conducts the following tasks: (1) switching the open port depending on PRF and the current secret, (2) answering the clients' synchronisation messages and (3) executing Algorithm 1 to contain the impact of the DDoS attack utilizing the compromised secret.

The client side middleware (i.e., client stack) is responsible for: (1) producing the corresponding open port number of the broker servers using current secret to PRF, and (2) synchronising/updating the time/secret by sending a sync-request message to the broker servers. Moreover, to obtain a new secret while an attack is ongoing, each client stack sends a sync message each time when the server stack transmits a message requesting for a sync-request message.

The number of open ports $p$ for the clusters in each *shuffling* iteration and the number of the *malicious clients* $c$ are the two key factors for the efficiency of PHSS, as pointed out in Section 6.3.1. Therefore, we evaluate the efficiency of PHSS for these factors by comparing with the public key-based re-keying process.

In the public key-based approach the Authorization Server issues different secrets to each client to localize the *malicious clients* and countermeasure the impact caused by the DDoS attack. However, this also increases the risk of attacks targeting the entire port range, since the broker server opens a different port each client.

Benchmark attack duration for our experiments is the period for containing the DDoS attack's impact through a public key-based approach. During this period, the successful message delivery rate of the pseudo-state estimation application refers to the network availability provided by the containment mechanisms. As the state estimation is one of the critical SG applications, we employ 4096 bits public key in our evaluation when comparing PHSS with the public key-based containment mechanism.

To the best of our knowledge, our proof-of-concept implementation-based experiment is the first real-world experiment of the *port hopping* approach in the literature. The related existing approaches focus only on the local network performance in case of a DoS attack or clock accuracy of *port hopping* mechanism [LT04; BHK07a; FPT12].

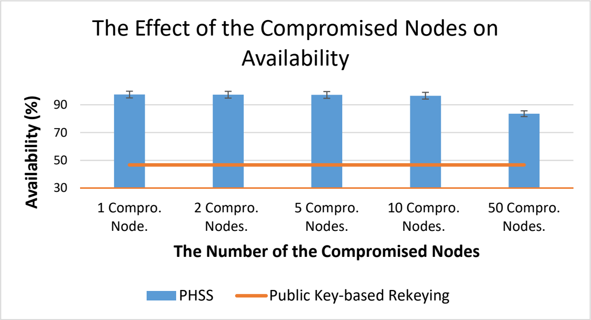


FIGURE 6.8: The effectiveness of PHSS while increasing the *malicious clients*

### Results Discussion

Fig. 6.6 demonstrates that the number of open ports $p$ significantly affects the availability, especially with the increase in the number of *malicious clients*. However, instead of opening the maximum number of ports (21 ports in this experiment, i.e., a port for each client), opening an optimum number of ports computed using Algorithm 1 provides availability close to the maximum availability provided by 21 clients even when the number of *malicious clients* increases.

The straight line in Fig. 6.6 shows the successful delivery rate in the time period between the beginning of a DDoS attack that exploits the compromised secret and the containment of the impact of the DDoS attack by using the public key-based approach. PHSS, using Algorithm 1, provides an **availability** over 98% in each case, whereas the public key-based approach caters an availability under 60% . The only case where PHSS provides lower availability than the public key-based containment approach is when using the minimum ports (2 ports) in each *shuffling* iteration despite the existence of more than a single *malicious client*.

Another aspect of the evaluation of our approach is the overhead in terms of service degradation of the broker server and the additional latency induced when PHSS is operating. To do this, we run the pseudo-state estimation application on the proof-of-concept prototype using both static port and *port hopping* mechanisms with variant numbers of the open ports.

Fig. 6.7 shows that with up to 150 hopping ports, neither the switching ports nor the opening ports results in a significant impact. The **throughput** degradation of the broker server is <0.01% for 30 ports, which implies a successful response rate of the broker server for the pseudo-state estimation application. Opening more than 150 ports causes abnormal behavior of the broker server, but thanks to our optimization used by Algorithm 1, PHSS does not need such a high number of open ports, $p$. Moreover, we did not observe any significant additional **latency** when using our approach.

## 6.4.3 Emulation-based Evaluation

To assess the effectiveness of our approach in large networks, we emulate the proof-of-concept in EC2's local network by creating 100 clients[9]. We employ Algorithm 1 to find

[9] More than 100 clients are not supported by the EC2-micro instance.

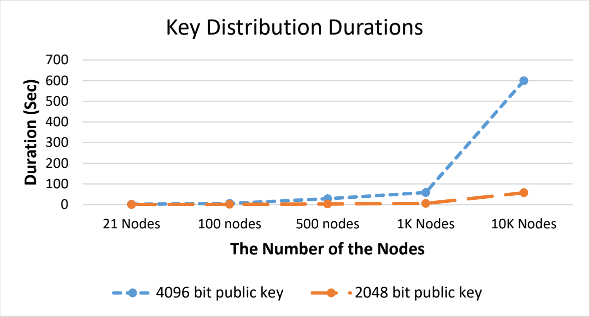


FIGURE 6.9: The rekeying process duration of different size of keys and nodes

the optimum number of open ports $p$ in each run. In addition, the network includes different number of *malicious clients* in each run.

Fig. 6.8 shows that with the increase in the number of clients from 21 to 100 (see Fig. 6.6), the public key-based containment method is able to contain, in a relatively longer time period, the impact of the DDoS attack that uses compromised secrets to discover server's open port. Accordingly, a notably higher loss of availability occurs.

Considering the case where PHSS is deployed, PHSS maintains an **availability** performance of up to 98% even where the *malicious clients* are up to 10%. Beyond 10%, the performance linearly degrades, as depicted in Fig. 6.8. The reason for the degradation is the increase in the number of the quarantined *malicious clients* that need to obtain new keys using the public key. Hence, if all clients are malicious, our approach loses its efficiency. However, PHSS takes advantage of the different session key for each client, which eliminates a high fraction of potential key breaches.

Finally, we demonstrate the duration of the key distribution ranging from 21 to 10K nodes for different sizes of the public key (i.e., 2048 bits and 4096 bits) in the case of usage of public key-based re-keying employed by the existing approaches. Fig. 6.9 shows that the increase in the number of clients strongly impacts the duration of containment of the damage of the DDoS attack as well as the network availability indirectly. As PHSS does not need the public key to sanitize all clients except the *malicious clients*, it significantly outperforms the public-key based re-keying approach when the number of clients increases. In addition, the key size is also an important factor: as shown in Fig 6.9, the re-keying process using 4096 bits key takes ten times longer than in the case of 2048 bits.

### 6.4.4 Synopsis

The evaluation of our approach focuses on the availability of the network and the induced overhead (i.e., throughput and latency). The experimental results denote that during DDoS attacks using the compromised secret, PHSS can provide network availability which is higher than 98% compared to the public key-based re-keying mechanism that provides availability below 60%. An increase in the number of the clients does not have a significant effect on the performance of PHSS, whereas it considerably affects the public key-based re-keying mechanism.

Unless all clients are malicious, PHSS significantly outperforms the public-key based re-keying approach. In addition, PHSS introduces negligible throughput and latency overheads, as depicted in the results.

## 6.5 Conclusion

We have proposed a cloud-assisted DDoS attack resilient communication platform. Our first contribution was a hierarchical hybrid cloud-assisted architecture (HHCEC), aimed at meeting scalability and security requirements of the SG applications in the cloud. We employed a publish-subscribe system on the HHCEC, as the pub-sub message-passing paradigm that matches with the SG's data acquisition paradigm. However, DDoS attacks against the brokers pose availability risk for time-sensitive critical SG applications. To cope with this, we proposed the port hopping spread spectrum (PHSS). The *port hopping* mechanism of PHSS basically prevents the brokers from transport and application layer DDoS attacks by switching the open port over time in a pseudo-random manner.

This enables the broker to drop the invalid packets in the firewall to avoid the applicat ion-based filtering. In addition, to overcome the relatively high-volume flooding attack, PHSS spreads the consecutive packets over the brokers in a *Broker Bundle*.

Furthermore, the existing *port hopping* mechanisms use a secret shared between all parties to produce the same open port number in the same time, which pose a high security risk in the case of the compromise of the secret. The containment of the impact of the DDoS attack utilizing the compromised secret is fulfilled by an Authorization Server using a secure channel (a public key-based re-keying process) in the existing works. However, the brokers become unavailable until the public key-based re-keying process is done, which leads to the loss of availability that violates the requirements of the SG applications. To address this issue of the existing approaches, we employ a *token-based authentication mechanism*, enabling the brokers to regularly issue the secret in encrypted form by using the session key of each publisher. Moreover, to contain the damage of the DDoS attack employing the compromised secret, we introduce a *shuffling-based containment mechanism*, which delivers new secrets to each cluster after shuffling and clustering the publishers. By repeating this process on the clients in the cluster(s) whose secrets are still used for the attack, the port *shuffling* mechanism progressively isolates the *malicious clients*.

Using a proof-of-concept platform consisting of Amazon EC2 micro instances and PlanetLab nodes, we evaluated the effectiveness of our approach in providing availability in the case of DDoS attacks using the compromised secret. We did so by comparing PHSS with the public key-based rekeying mechanism. The results show that our approach significantly increases availability in comparison to the public key-based re-keying mechanism, since it contains the impact of the DDoS attack utilizing the compromised secret in a notably shorter time period.

# Chapter 7

# Towards DDoS Attack Resilient Wide Area Monitoring Systems

The effectiveness of cyber-control systems is determined by achieving real-time and accurate state information as obtained from an efficient and reliable communication schema. Thus, runtime state estimation constitutes a critical element to maintain the SG performance and resilience over any network failures transpiring as either operational failures or as deliberate attacks. In practice, this state assessment is achieved using Wide Area Monitoring Systems (WAMS) that use Phasor Measurement Units (PMUs, and also known as Synchrophasors) for data acquisition to monitor real-time power transmission and to detect grid instabilities [Mar+14]. The PMUs periodically sample the voltage and current parameters of the power system, and subsequently forward the sampled data to the Phasor Data Concentrator (PDC) for processing [KAR13].

As WAMS form the core of SG operations, this criticality also makes the WAMS susceptible to attacks that can exploit communication level vulnerabilities to compromise the critical WAMS requirements on low-latency and high-availability. The transport layer is particularly vulnerable to cyber-security attacks - Denial-of-Service (DoS) and Distributed DoS (DDoS) attacks can be conducted towards the PMUs or PDCs to cause transmission delays or loss of measurements. Consequently, this can result in a severe degradation in SG performance in terms of inaccurate predictions of transmission status, network metering failures or delays in the mitigation of power network failures [Kar14; WL13].

In this chapter, we extend upon the advocated Multipath-TCP (MPTCP) approach to provide a resilient and efficient communication scheme for the WAMS phasor measurement processes. The basic MPTCP provides long-duration communication connections [Rai+11] and provides reactive mitigation against attacks with its diverse multi-path functionality. However, in order to achieve proactive and robust protection of the transport and application layer from DoS/DDoS attacks, we introduce a novel stream hopping mechanism termed as MPTCP-H that is directly integrated into MPTCP. The proposed hopping mechanism hides open port numbers by timely refreshing of the sub-flows, with new port numbers, without causing data traffic interruptions. This approach of hiding port numbers is shown to provide high coverage against transport and application layer DoS/DDoS attacks. The results from MPTCP-H demonstrate that the proposed approach indeed secures the system with minimal additional latency and message overhead.

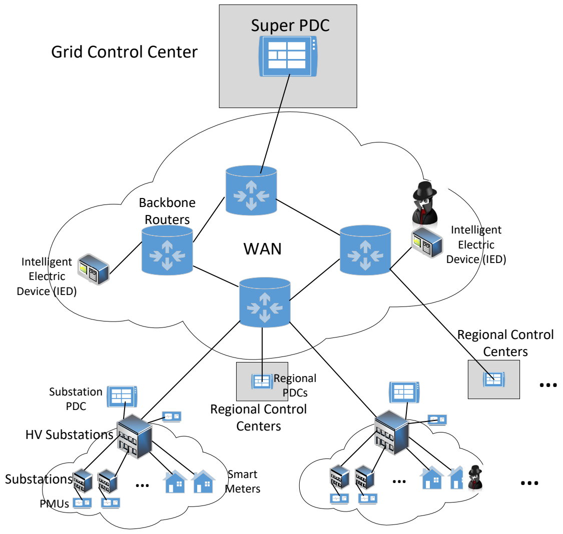


FIGURE 7.1: A SG network

*Contributions in this chapter:*

- A practical threat model where the DoS/DDoS attacks can occur in the WAN via compromised devices, and accordingly saturate the WAMS devices.
- A novel defense mechanism that mitigates DoS/DDoS attacks by periodically switching the MPTCP connection subflows.
- Empirical validation of the MPTCP-H's overhead which shows that MPTCP-H performs equal to or better than the standard transportation protocols in terms of latency and congestion attributes.

## 7.1 Background

This section outlines the technical characteristics of WAMS in a SG. We also provide a background on MPTCP operations that are used in our proposed MPTCP-H extension.

### 7.1.1 Wide Area Measurement Systems (WAMS)

Accurate estimation and monitoring of the state of the power network is critical for SG operations. The traditional Supervisory Control and Data Acquisition (SCADA) systems are employed for periodically monitoring the sampling measurements at predefined time intervals, e.g., per second [AJZ05]. In order to manage the SG in a reliable and efficient manner, WAMS offer low-latency, high-precision and time-synchronized measurements by taking advantage of phasor measurements (both magnitude and phase angle) obtained from the deployed Phasor Measurement Units (PMUs) [AJZ05]. Whereas SCADA systems are unable to handle the dynamic snapshots of a power system, the advanced WAMS support real-time behavior of the power system to mitigate unexpected power blackouts. While the WAMS technology supports the SG control functions with real-time state monitoring, any inaccuracies in the state information arising from communication perturbations or assessment errors, can also detrimentally affect the SG stability.

In this chapter, we focus on a multi-tier WAMS architecture that interfaces, in turn, with the high voltage (HV) substation PMUs followed by substations PDCs, regions PDCs and control center PDC (cf. Fig. 7.1 [KAR13]), where the HV substation PDCs also connected with PMUs in the neighboring substations (ca. 20-40 PMUs) [KAR13].

In the hierarchical architecture, the measurements of PMUs are forwarded to the substation PDCs that sort the received data by timestamps and examine any missing data for requisite analysis. The substation PDCs then transmit the prepared measurements to the regional PDCs for subsequent forwarding to the national monitoring centers, as shown in Figure 7.1. The characteristics of WAMS are as follows [KAR13].

A HV substation of the Power Grid (Substation PDC):

- $\sim$ 20-40 PMUs connected to the PDC.
- PMU data rates (60-120 fps for 60Hz systems).
- Tolerable internal latency ($\sim$3-10 ms).
- Applications requiring fast response as well as local visualization and archiving.

Regional centers of WAMS (Regional PDC):

- Responsible for a large number of PMUs ($\sim$50-500).
- Data rates between 30-60 fps.
- Tolerable internal latency ($\sim$10-100 ms).
- Applications for regional operation, e.g. state estimation.

Main control center (Super PDC):

- Accommodation of a very large number of PMUs (a few thousand PMUs).
- Low data rates ($\sim$1-30 fps).
- Tolerable internal latency ($\sim$100 ms-1s).
- Applications that perform visualization combining SCADA and Synchrophasor data.

### 7.1.2 Multipath TCP (MPTCP)

Multipath TCP is a recent TCP extension [For+13] and an Internet Engineering Task Force (IETF) standard, which is still in its experimental phase. MPTCP allows a single TCP connection to make simultaneous use of multiple paths by opening several subflows, each using a different interface and routed through a different path in the network. In practice, MPTCP is a TCP connection that uses TCP options to enable multipath functionality

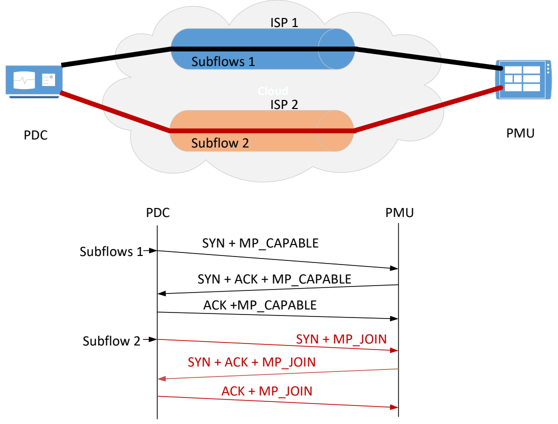


FIGURE 7.2: MPTCP connection

without requiring any changes at the application level. Hence, for a given application, an MPTCP connection behaves exactly like a regular TCP connection.

In MPTCP, the initial 3-way handshake consists of a SYN, a SYN/ACK and an ACK, as in the regular TCP. The difference with MPTCP is that each party asks the other party through an `MP_CAPABLE` TCP option whether it supports MPTCP. At this stage, they also share their keys in cleartext in order to identify and authenticate future subflows for the connection. This handshake and the subflows are depicted in Fig. 7.2.

Each subflow is identified with a 4-tuple of <source address/port, destination address/ port>, which is created after the initial MPTCP handshake and exchange of keys. To add new subflows into an existing connection, a token derived from the initial key and `MP_JOIN` in the TCP options are used in the handshake process of the new subflow, as illustrated in Fig. 7.2 [For+13].

Note that each subflow has its respective sequence numbers similar to a regular TCP connection. In addition, the specification of MPTCP identifies a different sequence number that interrelate packets delivered over multiple subflows within a single MPTCP connection [For+13].

### The Advantages of Utilizing MPTCP in the Phasor Measurement Communication of WAMS

High communication latency, resultant from a connection re-establishment of TCP due to a broken or stalled connection, can violate the latency requirements of phasor measurements [Paa+14]. In contrast for MPTCP, when the first subflow is initialized to transmit phasor measurements, the other subflows are created concurrently. Since one of the MPTCP subflows used to transmit the measurements is likely functioning normally (with high likelihood), thus the overall phasor measurement traffic is not disturbed or delayed.

Moreover, using MPTCP, a higher network utilization and a fairer allocation of resources to subflows is provided by efficiently addressing the congestion response of the corresponding subflows. The detailed advantages of MPTCP-based networks appear in [Rai+11].

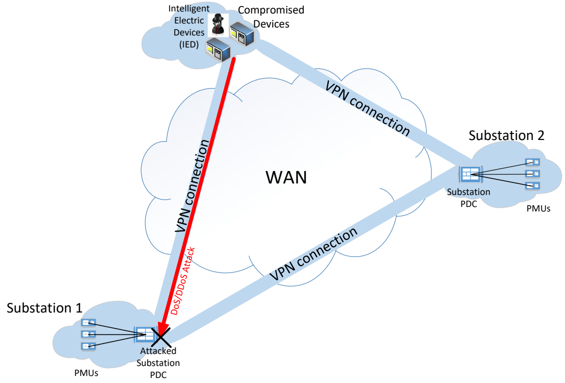


FIGURE 7.3: Illustration of Network and Attack Models

## 7.2 System and Security Models

In this section, we first present the SG system and threat models. Subsequently, we outline the potential compromise of SG devices and the resultant security vulnerabilities in the SG networks. We also discuss the deficiencies of current intrusion detection systems to detect these threats.

### 7.2.1 System Model

Similar to contemporary SG models, we consider that the power utility employs a heterogeneous network (i.e., composite public and private networks) to construct a SG wide area network (WAN) for cost-effectiveness and applications availability. For WAN, the utilities could use the links leased from the carriers and also dedicated network links. WAN is typically used not only for various kinds of WAMS devices but also for different types of SG devices such as Smart Meters.

In addition, we assume that gateway-to-gateway (versus host-to-host) virtual private networks (VPNs) exist in the WAN to provide secure channels. Thus, every node at a given local area network can access the other local area networks only through the proscribed gateway-to-gateway tunnels [Lav+10].

In the geographically demarcated SG operational area where the WAMS acquire the state measurement data, some SG devices might be compromised by exploiting the vulnerabilities. The compromised devices could grant the attackers elevated privileges for overwhelming the devices resources, as illustrated in Fig. 7.3.

Although the IEEE standards C37.118.1/2 [Mar+14] specify Synchrophasor measurements and data transmissions, respectively, IEC 61850 [AAH16] is the de-facto standard for specifying the substation and utility automation [AAH16]. In addition, we also assume that both the PMUs and the PDCs support the IEC 61850-90-5 standard to provide MPTCP connection authentication through the standardized key distribution center (KDC). IEC 61850-90-5 recommends UDP for data transmission of WAMS as a transport layer protocol due to its a lightweight mechanism [AAH16]. However, in this work, we employ MPTCP for the data transmission and show that it introduces near similar performance characteristics to UDP for phasor measurement traffic.

### 7.2.2 Attack and Threat Model

Our threat model covers two types of Denial-of-Service attacks, namely: (1) transport layer attacks, where the adversary consumes the device's processing and networking resources by exploiting the information obtained from the port scanning or sweeps, and (2) application layer attacks, which exploit the vulnerability of the application to saturate the device resources.

In our threat model, the attackers are malicious entities which are compromised devices able to access the WAN (Intranet) where the phasor measurement devices are located. Furthermore, the malicious devices are assumed to have the ability to launch DoS/DDoS attacks to saturate the resources of PDCs and PMUs. However, to mount more sophisticated attacks inside the WAN, the attackers have to guess the open port numbers or obtain elevated privileges (e.g., scanning the open ports). In addition, we do not trust the devices inside the WAN. We preclude the case of an insider attacker physically accessing the phasor measurement devices.

The level of effort needed to mount DoS/DDoS attacks naturally also depends on the type of the topology. As an example, a mesh topology connecting the the SG devices within a WAN is highly vulnerable to DoS attacks. This occurs as any compromised device in the mesh can be exploited to carry out an internal DoS attack which overwhelms the service of multiple nodes at the same time.

### 7.2.3 Compromise of SG Devices

The deployment of devices in a wide geographical area makes it difficult to protect them from being physically compromised. This is often observed in devices used for monitoring the grid where an attacker can accesses the physical devices and compromise them. On the other hand, a house owner can have full physical access to many deployed devices e.g., smart meter [AP17].

The device can be compromised either by using login credentials or by exploiting a vulnerability. 1) Login credentials can be obtained using: social engineering, side channel attacks, eavesdropping (unprotected communication), and passwords guessing, and 2) Identifying a vulnerability is possible for an attacker either by buying zero-day exploits or by scanning the device. The attacker also needs to develop a exploit code using the vulnerability to plant malware on the device to exploit it. In addition, to compromise the devices, the attacker can also directly connect to the local network behind the firewall that the devices are connected to [PSZ17].

As the compromised nodes act similar to the normal nodes, such "internal" attacks pose a higher threat potentially leading to significant damages to the SG communication network and even to the control system of power network. As a result, the compromised nodes can be exploited by different malware or viruses attacking critical SG devices [AP17].

In particular, the SG can be significantly affected by DoS/DDoS attacks since it heavily depends on the availability of the communication network. In this paper, we mainly focus on internal DoS/DDoS attacks in WAN networks where the attackers use the malicious devices inside a WAN to launch DoS/DDoS attacks on critical SG devices (i.e., PDCs or PMUs) to induce data transmission delays or block data delivery [AP17].

### 7.2.4 Security Vulnerabilities of WAMS

We outline a study conducted on a testbed using real WAMS devices [Mor+11] to highlight their security vulnerabilities.

Morris et al. [Mor+11] conducted tests to evaluate the vulnerability of PMUs and PDCs in terms of the attacks originating from inside a WAN by building a testbed consisting of PMUs, PDCs, a router and a Network Analyzer [Mor+11]. They launched TCP flooding (SYN and FIN) and UDP garbage flooding attacks on the devices for both specific and also random ports. The test results showed that all devices under flooding attacks are eventually overwhelmed and start to deny service when the traffic volume increases beyond the data processing ability of the device [Mor+11].

Based on the collected results, the authors suggestion to mitigate these issues is that utilities should be enabled to monitor the volume of the network traffic in order to detect and/or limit transmission of the traffic to the devices. Moreover, the fuzzing tests conducted in [Mor+11] show that even individual packets can result in device failures i.e., resetting the devices.

These test results indicated that DoS/DDoS attacks can be a serious threat to the safety and reliability of the power network. Such DoS/DDoS attacks can lead to partial loss of availability, and thus leading to the incorrect state estimation of the power network, or, impediments to the mitigation on power system failures [Mor+11]. For this reason, a proactive defense mechanism needs to be employed to mitigate the DoS/DDoS attacks for WAMS.

### 7.2.5 Deficiencies in Intrusion Detection Systems

For providing security protection to IT infrastructures, the traditional security solutions, e.g, firewalls, intrusion detection systems (IDS), or Virtual Private Networks (VPN), are both common and efficient. However, as SG devices are typically resource constrained (computational, bandwidth, memory), the direct adoption of these IT-level security solutions is largely not possible [BHK07a].

Typical IT servers need stronger security protection than the edges/clients. However, in SG communication networks, the control center servers and edge nodes (e.g., relays, circuit breakers) require the same level of security, since the edge nodes can also pose safety similar to that of the servers. Moreover, given that SG devices have constrained resources, directly utilizing the IT-based DDoS defense/authentication mechanisms might not provide the expected security protection for the SG applications. Therefore, lightweight and proactive DDoS protection mechanisms are desired for securing SG communication networks [BHK07a].

Moreover, the classical Intrusion Detection Systems (IDS) or Intrusion Prevention Systems (IPS) are not adequate for guaranteeing SG communication security. SG communication systems have also inherited many new challenges and security threats from its own machine to machine communication structures and other issues of computer networking technologies. For this reason, the IDS systems utilized for SG communication have to consider the issue of handling resource-frugal devices over both traditional computer networks and M2M networks [AP17].

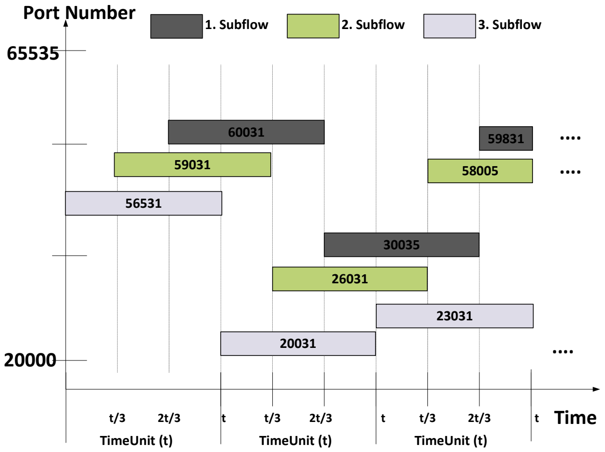


FIGURE 7.4: Stream hopping of MPTCP-H

Based on this background, we now present our proposed approach that provides efficient defense against transport and application layer DoS attacks on WAMS.

## 7.3 MPTCP-H Architecture

The MPTCP-extension (MPTCP-H) aims to provide proactive protection against transport and application layer DoS attacks. The main idea behind MPTCP-H is to employ a stream hopping mechanism alongside the MPTCP multipath functionality. In order to achieve that, MPTCP-H develops two innovations:

1. Stream Hopping, where subflows are switched over random ports which increases the attack cost, unlike in the typical fixed MPTCP flow.
2. Authentication, which handles the authentication between the PMU and the PDC whenever a new connection is created.

### 7.3.1 The Stream Hopping Technique

The traditional security systems such as firewalls, IDS and IPS are unsatisfactory to defend phasor measurement traffic against DoS/DDoS attacks due to their passive and unaccomplished structure. In existing IT systems, the continuous change of network attack type gives an advantage to the attackers over the protection systems. The malicious attacker is in the dark side, while the protector is in the bright side. Therefore, the adversary solely requires discovery of a few vulnerabilities whereas the protector must guarantee that the system does not have any exploitable vulnerabilities [LWC14].

To address the advantage the attackers have over the protectors, moving target defense (MTD) methods have been proposed [Afz17]. This mechanism is a new proactive defense method in which the protector constantly changes the attack surface of the system to boost the cost of an successful attack for the attacker. Port hopping [LT04; FPT12] is a specific MTD method that periodically switches a port of a service in a pseudorandom manner, confusing potential intruders. The port hopping mechanism facilitates both the detection and filtering of unauthenticated packets and does not need require changes in the existing systems and protocols [LWC14].

However, this port hopping scheme requires all clients to know a secret key used by the server to calculate the port number for the current time slot. In the case of disclosure of the secret key, the open port of the devices can be direct target of DoS/DDoS attacks, which exposes high security risks for WAMS. Furthermore, the implementation of the port hopping technique is not practically feasible for TCP connections.

On the other hand, MPTCP allows simultaneous use of $s$ subflows over different paths[1] to distribute data across these subflows, while maintaining a standard TCP interface for the applications. This characteristic of the MPTCP connection enables the implementation of a port hopping-like technique, called stream hopping, by periodically switching the subflows over different IP-addresses/interfaces[2].

### Subflow Switch

To realize the stream hopping technique of MPTCP-H, we extend the functionality of MPTCP by periodically opening new subflows that use different paths, each of which is used for an allocated time period $t$. In MPTCP-H, only PDC is allowed to initiate $s$ new subflows using TCP-like handshake, as illustrated in Fig. 7.2. After establishing an MPTCP connection with a PMU, the PDC opens new subflows by selecting new random port numbers on its side. To establish a new subflow, the PDC periodically sends a SYN packet containing an `MP_JOIN` option to the PMU. After checking the authentication of the new subflow, the PMU acknowledges the SYN with the same type of message (`MP_JOIN`) and binds the new subflow to the MPTCP connection. The three-way handshake ends with the acknowledgement message of the PDC.

MPTCP-H also allows a given PMU to randomly select new port numbers on their side to announce to PDC the new subflows in the MPTCP connection. In order to do this, each PMU periodically hands over the selected port numbers to the PDC, such that the PDC can use them to connect to the PMU. To this end, PMUs transmit a `ADD_ADDR`/ `REMOVE_ADDR` message through an existing subflow, which informs the receiver of the PMU's alternative existing addresses/no longer existing addresses, respectively. The PDC initiates new subflows over the delivered port numbers of each PMU by performing a three-way handshake carrying `MP_JOIN` command as illustrated in Fig. 7.2. Subsequently, the expired subflows are closed by sending `FIN`. This periodic switching of

[1] In this work, a path between a sender and a receiver is defined by a 4-tuple of source and destination address/port pairs. Changing one of the tuples creates a new path. We interchangeably use the subflow and path terms.

[2] Moreover, in MPTCP, the single-homed nodes can create subflows over different port numbers of IP-addresses pair using MPTCP "ndiffport" path manager [Paa17]. Since PMUs are single-homed devices, for each new subflow we use different port numbers but the same pair of IP-addresses. Furthermore, MPTCP provides higher performance and robustness than normal TCP when the number of subflows per pair of IP addresses gets increased [ZSP16]. The reason for the performance improvement is the utilization of available network paths in an efficient manner.

subflows is the basis for the subflow hopping technique which makes reconnaissance of the victim's address difficult for attacker.

### Phase Shift

To keep the renewing period of subflows shorter than the attacker's subflow port number reconnaissance time, MPTCP-H creates multiple subflows with $t/s$ phase shifts versus multiple subflows activated for the same time period $t$. In Fig. 7.4, stream hopping in conjunction with the phase shift is depicted where each shaded bar represents a subflow of the active connection ($s = 3$) at a particular point in time. Each subflow is active for the allocated time $t$ and is substituted with a new subflow with a new port number when the allocated time expires. The renovation of the subflows do not overlap each other, but take place with a $t/s$ phase shift. By doing so, we assure the MPTCP-H connection of having a subflow initialized within a period of time not exceeding $t/s$ at each instance. In addition, by finely calibrating the number of the subflows $s$ on $t/s$, depending on the attacker's the reconnaissance time, we can assure that the MPTCP-H connections have a functional subflow throughout the attack duration. The reason is that, throughout the attack, there is a subflow whose lifetime is shorter than the $t/s$ which is the time for the reconnaissance of a subflow by attacker.

### Attack-resistance

The shuffling of the active port numbers (and the subflows) increases the difficulty for an attacker who discovers the port number of subflows through port scanning to launch connection-flooding attacks. As the subflows expire after the allocated time, the possible maximum damage caused by an attacker who discovers the port numbers of the subflows is limited to that specific time duration. As new subflows get activated, the attacker must guess or once again scan the port numbers for the new subflows to maintain the attack. This limits the attacker to mount persistent attacks on the active ports or forces them to blindly guess or aggressively scan the active ports. The consequence of either is the limited damage potential from an attack.

Furthermore, as PMU and PDC randomly and separately choose their next ports, MPTCP-H does not need a shared secret key to determine the port number while opening a new subflow, unlike existing pseudorandom port hopping mechanisms. This protects the system from the effects of a probable shared key disclosure.

In MPTCP, allowing only one side to initiate new subflows is possible and we delegate this responsibility to PDCs. Since the PDCs typically have higher importance than a single PMU, MPTCP-H configures PDCs to initiate new subflows. Therefore, even if an attacker uncovers the varying open port numbers to some extent, he is unable to saturate the PDC resources by sending forged messages to initiate new subflows.

The proposed stream hopping mechanism of MPTCP-H is akin to Frequency Hopping Spread Spectrum (FHSS) [Dix94] technique which enables secure radio communication. If an attacker plans to jam or decipher the radio signal in FHSS, he needs to discover the hopping sequence or monitor the entire wide frequency band to capture the signal. Likewise, the subflow hopping of MPTCP-H has the same impact - increasing the difficulty for the attacker by changing port number over time. In addition, when DoS/DDoS attacks take place, the data traffic can be distributed or duplicated over several subflows for redundancy/resiliency.

**Performance Consideration**

Since MPTCP-H requires frequent opening and closing of subflows (TCP connections), a probable degradation in the performance and throughput of the system should be considered. Firstly, we introduce an equation that calculates the additional data traffic overhead (per second) of MPTCP-H:

$$\begin{aligned} s * 1/t * (4 * handshake\, message\ (MP\_JOIN) + 4 * FIN\ message) + \\ ADD\_ADDR\, message + REMOVE\_ADDR\, message * 1/t = \ Message\, Overhead \end{aligned} \tag{7.1}$$

As seen in the equation (7.1), $t$ and $s$ are key factors in the calculation of the overhead. We present two scenarios to show their effects on the overhead. Then, we assess the scenarios' results to see how to properly calibrate the factors' values.

*The first scenario:* If $t$ is equal to 1 second, $s$ is 10 subflows, and the packet length is equal or greater than 40 bytes in the equation (7.1), then the overhead is 3280 bps (between a PDC and a PMU).

*The second scenario:* If t is equal to 5 seconds, $s$ is 5 subflows, and the packet length is equal or greater than 40 bytes in the equation (7.1), then the overhead is 326 bps (between a PDC and a PMU).

Considering the second scenario, if there are 40 PMUs that connect to a PDC, $40 * 326/2kbps = 6.48kbps$ inbound traffic and $40 * 326/2\ kbps = 6.48\ kbps$ outbound traffic (overhead) are created by MPTCP-H.

To compare the overhead with the measurement traffic, we need to calculate the max PMUs traffic for a PDC: $40\ PMUs * 70 < bytes\ (packet\ size) * 120fps = 336\ kbps$

When we compare the overhead of the second scenario (4.32 kbps) with the inbound traffic of PDC (336 kbps), we see that MPTCP-H introduces a DDoS mitigation mechanism at the expense of a reasonable overhead (14.5%). In addition, it is worth noting that the difference between the first scenario and the second scenario indicates that decreasing $t$ and rising $s$ sharply boost the total overhead, meaning the calibration of those values has high importance for obtaining the minimal overhead with the required security.

Furthermore, as detailed in the evaluation section, we did not observe any perturbation in the system performance while frequently switching the subflows (TCP connections).

### 7.3.2 Authentication for Initiating New Subflows

A MPTCP connection between a PMU and PDC is initiated by exchanging initial keys that are used to authenticate new subflows for the connection. However, no secure mechanism is declared by the MPTCP specification for the exchange of the initial keys. IEC 61850-90-5 specifies the key distribution center (KDC) [AAH16] which introduces a symmetric key coordination between the publishers and subscribers (i.e., PMU-PDC).

To provide secure authentication, we use keys provided by KDC instead of the initial keys. The idea is akin to the one reported in [PB12], where an application-layer key (SSL/TSL) is proposed to be used for the authentication.

By using an application-layer key, i.e., the KDC keys, instead of the initial keys exchanging in the clear-text, we address the existing MPTCP's security issue related to key disclosure. By doing so, PMUs and PDCs become more robust against JOIN-flooding attacks.

Moreover, MPTCP-H secures the handshake process of establishing new subflows as follows: When initiating a new subflow, the PDC transmits the initial synchronization message including a 32-bit token which is a cryptographic hash of the receiver's initial (KDC) key, produced by the SHA-1 algorithm, and truncated to the most significant 32 bits. This token is used to associate the subflows to the MPTCP connection and also provide the security mechanism to block unauthenticated new subflows initiated by attackers [For+13].

Upon receiving a SYN that contains an `MP_JOIN` option, a valid token, and a random number, the PDC responds by sending a SYN/ACK including an `MP_JOIN` option, a random number, and a truncated (leftmost 64 bits) Hash-based Message Authentication Code (HMAC). Finally, following the PDC's transmission of an ACK with a HMAC, the PMU sends an ACK to the PDC, which makes the connection ready for data transfer. The random numbers (nonces) averts replay attacks on the authentication method. The HMAC exchange along with the random number secures the establishment process, since if the HMAC is incorrect, the connection is refused [For+13].

## 7.4 Security Analysis of MPTCP-H

We now present the threat scenarios and the related security analysis for the proposed MPTCP-H technique.

Computer network attacks can be categorized as: (i) active attacks, and (ii) passive attacks. An active attack involves the exploitation of compromised data or devices to mount attacks on the network, such as data injection, data modification or packet drop attacks. In a passive attack, the attacker needs to collect critical information on the network and to learn network properties or transmitted data by using attack types such as sniffing or eavesdropping. Passive attacks are widely employed to collect information paving the way for an active attack [PSZ17].

The SG can be targeted to induce a power outage which can be performed by portioning the power grid. The power grid portioning can be carried out in cyber means, by intentionally transmitting a trip command to a circuit breaker (CB). Triggering trip commands can be accomplished by launching the following active attacks: 1) directly compromising the CB; 2) prompting a wrong control decision at the central controller which sends a trip message to the CB; or 3) changing the controller commands while they are on the path between CB and the central controller [PSZ17].

As we focus on the security of phasor measurement traffic between PMU and PDC in this work, the second method is the most probable to be used by an attacker to conduct an attack on the grid after gathering critical information using passive attack methods. The attacker can cause an incorrect control decision by the controller by perturbing the phasor measurement traffic, providing information about the state of the grid. To accomplish this, the attacker mounts a DoS/DDoS attack against the open ports of either the PMU generating the measurements or the PDC processing those measurements. However, since MPTCP-H reshuffles the open ports, the attacker must guess or discover the open ports to launch a DoS/DDoS attack. A blind attacker is very unlikely to realize a successful attack by randomly selecting a port number and flooding garbage data to affect the phasor's operation. Even if the attacker successfully guesses or discovers the open port, the available time for attacking is limited, and after some allocated time $t$, the ports get shuffled. In the case of a DOS/DDoS attack against all the discovered open ports, the attacker has to continually scan the ports of the devices and continually adapt its attack according to the periodically varying port numbers of the subflows. This makes conducting an efficient attack a difficult task for the attacker, making MPTCP-H a successful mitigation technique against such attacks.

Another threat includes the compromising of the KDC keys for both PMUs and PDCs for the exchange of an initial key. These devices have scarce resources and can be saturated using a relatively small number of malicious authenticated connection requests by the attacker. For this attack, exposing the open port numbers for a short time would be enough to overwhelm the service of the devices. To protect the PDCs, carrying more importance than PMUs from the above mentioned attack, MPTCP-H grants the right of opening new subflows to the PDCs. Thus, the PDCs are able to refuse any requests to open new subflows and protect their resources from being depleted by the attacker. A complementary scheme that adds protection on the keys can also be included in MPTCP-H.

Configuring PMUs to accept traffic only from the PDCs (and vice versa) also represents a suitable defense approach for the PMUs. This can be achieved through a whitelisting approach that provides the PMUs (or PDCs) with a list of the authorized PDCs (or PMUs). For an attacker spoofing the IP addresses, MPTCP-H renders such a DoS attack to become unlikely by periodically varying the MPTCP subflows using new port numbers. To do so, each subflow is continuously recreated after some lifetime $t$ using new port numbers. This introduces a defense against threats related to attacker spoofing the IP address to consume the target's resources by transmitting forged packets to its open ports.

MPTCP inherently introduces new challenges for the traditional security approaches, making them no longer sufficient for MPTCP. For instance, since an IDS monitors and categorizes the traffic of a connection based on the 5-tuple, it sees the subflows of an MPTCP connection as an independent TCP connection, and thus cannot discover the correlation to reassemble MPTCP traffic correctly. Moreover, MPTCP enabling a sender to employ all available routes at the same time causes the fragmentation of data among the routes. For this reason, an IDS cannot have adequate knowledge on any of the streams to detect the malicious data, which leads to an exploitable vulnerability for cross-path data fragmentation attacks. Z. Afzal [Afz17] investigates possible attacks using these vulnerabilities and introduces solutions to address them.

Overall, MPTCP-H constitutes a proactive defense mechanism for time-critical communications. We opted for a proactive mechanism vs. a reactive mechanism, as deploying a reactive approach e.g., Intrusion Detection Systems (IDSs) would consume more time

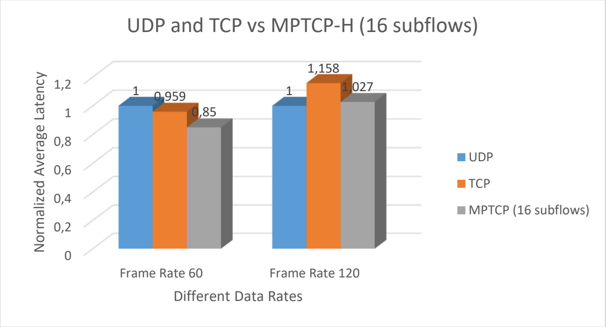


FIGURE 7.5: Normalized latency of MPTCP-H (4 network interfaces with 16 subflows), UDP and TCP

for mitigation. Reactive mechanisms have to detect the attacker at first and then report the attack to the systems or administrators for prevention. Moreover, a careful placement of IDSs in the network is required to detect internal attacks. Still, MPTCP-H can be used complementary to an IDS.

## 7.5 MPTCP-H Implementation

For the implementation of our mechanism we use the Linux Kernel implementation of Multipath-TCP (mptcp_v0.91) [Paa17] which is the reference and most common implementation of IETF [For+13]. We implement our MPTCP-H mechanism using the Enhanced Socket API of B. Hesmans et al.[Hes17], which enables us to have control over individual subflows. This API allows us to open new subflows with custom IP addresses and port numbers and closing them whenever needed. G. Demaude and P. Ortegat [DO17] develop a Java Native Interface (JNI) tool that enables us to use Java language to manage the Native C socket API. In the implementation of MPTCP-H on a Virtual Machine (the Linux Kernel with mptcp_v0.91), we manage the native C socket API with the above mentioned tool.

In our implementation, while PMU runs on the host (physical computer), PDC runs on a Virtual Machine. In WAMS, the phasor measurement traffic between a PMU and PDC is similar to a server-client model. The PMU acts as a server by sending measurement messages each time the PDC (the client) transmits a request message for the measurement. To implement this scenario, we develop a middleware between the application layer and MPTCP in the client side for MPTCP-H and provide two applications acting as PMU-PDC in client and server sides. After establishing a MPTCP connection, the PDC (client) additionally opens a fixed number of Multipath-TCP subflows $s$ for the connection. The subflows are periodically switched by the PDC (client). In other words, the subflows are closed over time and replaced with new subflows. Each of the new subflows is created with a random port number as explained in Section 7.3.1. The implementation of the idea and threat model are fulfilled by Ferdaus Nayyer during his master thesis as a joint work.

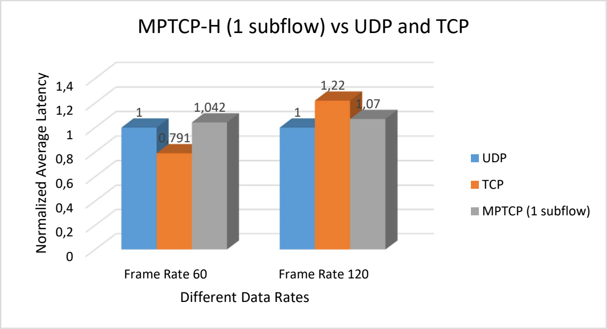


FIGURE 7.6: Normalized latency of MPTCP-H (1 network interface with 1 subflow), UDP and TCP

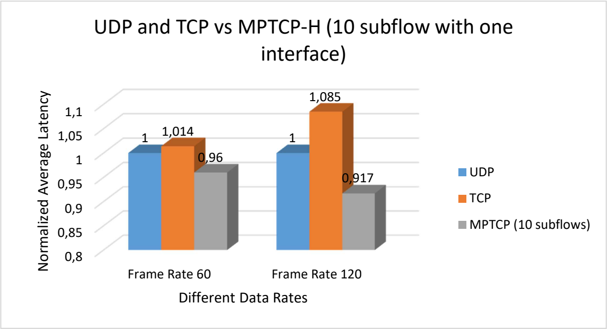


FIGURE 7.7: Normalized latency of MPTCP-H (10 subflows in one network interface), UDP and TCP

## 7.6 MPTCP-H Evaluation

As the proposed defense mechanism is targeted at time-sensitive critical WAMS applications, we need to particularly assess the system availability and the overhead, in terms of additional latency and message. Thus, we employ three metrics in the evaluation of the approach: (1) the system availability, (2) the latency, and (3) the overhead messages caused by MPTCP-H. Firstly, in Section 7.6.1, we evaluate our approach regarding the second and the third metrics, i.e., latency and message overhead, in attack-free conditions in order to test our approach in the case of different network topologies and data rates. Secondly, in Section 7.6.2, we test MPTCP-H under both DoS attack and attack-free conditions in terms of the system availability (the first metric) and additional latency (the second metric) by comparing TCP.

### 7.6.1 Attack-free conditions

In the following sections we evaluate our approaches under DoS attack conditions.

### Latency Assessment for MPTCP-H

To assess the impact on latency, the NorNet testbed is used for evaluating latency, which provides realistic results [Dre15]. The testbed consists of a collection of multihomed nodes distributed throughout Norway. Two nodes with 2-3 network connections are driven by daemons (i.e., a PMU and PDC). In our experiments evaluating impact on latency, three representative types of PMU-PDC topologies are implemented: 1) 4 network interfaces and 16 subflows (full-mesh), 2) a single network interface and a single subflow, and 3) a single network interface but multiple subflows.

We utilize two different data rates (60 fps and 120 fps) in each experiment to simulate realistic phasor measurement traffic of WAMS. As the measurement traffic of WAMS typically has proscribed data rates, we evaluate the proposed approach regarding induced latency or congestion rather than throughput of the system.

According to IEEE C37.118.2-2011, Synchrophasor measurement traffic can be transmitted over TCP/IP or UDP/IP. UDP provides faster data delivery given its lightweight characteristics [Mar+14]. We compare the proposed approach with TCP and UDP in the transmission of Synchrophasor measurements to assess its performance.

Fig.7.5 presents the normalized average latency versus data rates for varied protocols. The latency values are normalized utilizing the latency of UDP as a base - as suggested for Synchrophasor data transfer by the IEEE Standard for Power Systems C37.118.2-2011 [Mar+14]. Fig 7.5. shows that MPTCP-H introduces less latency than TCP (and even UDP) in transmitting 60 frames per second (fps). On the other hand, for the 120 fps data rate, while TCP provides the worst latency, UDP outperforms MPTCP-H in terms of latency. We see from Fig.7.5 that TCP's latency is relatively low for the data rates of 60 fps due to its congestion handling mechanisms. However, when the data rates are high (120fps), UDP's connectionless approach provides better latency than TCP. That being said, MPTCP-H with multiple subflows provides latency results close to UDP even in the case of high data rates (120 fps).

We also conducted experiments on single-homed PMU and PDC to analyse if MPTCP-H has any shortcomings in these scenarios. Fig.7.6 shows that while the latency results for TCP are similar to the results of the previous experiment, MPTCP-H's latency degrades slightly. However, the overall latency of MPTCP-H is still relatively close to the latency of the UDP for both data rates of 60 fps and 120 fps.

Finally, to demonstrate the effect of the port-based multiple subflows structure of the MPTCP-H on the latency, we conducted experiments that compare UDP and TCP with MPTCP-H that uses 10 subflows over single-homed nodes (with 1 network interface). Fig. 7.7 highlights that MPTCP-H does not introduce any additional latency, and, instead, decreases latency even when the data rate increases to 120 fps. TCP's latency increases with the data rate.

### Message Overhead of MPTCP-H

To measure the additional overhead, we deploy a PMU and a PDC on a host and on Virtual Machines, respectively. In this work, while the message overhead refers to the protocol-specific message transmission, all traffic implies the message overhead plus the application layer message transmission. To calculate the message overhead, we run each

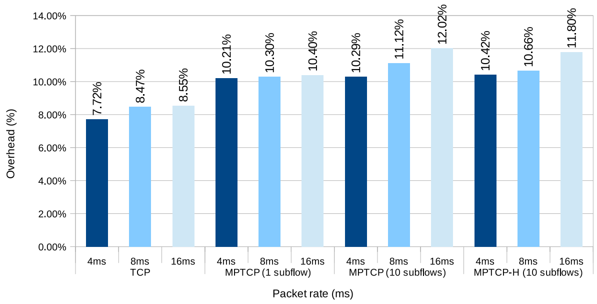


FIGURE 7.8: TCP, MPTCP vs MPTCP-H for message overhead

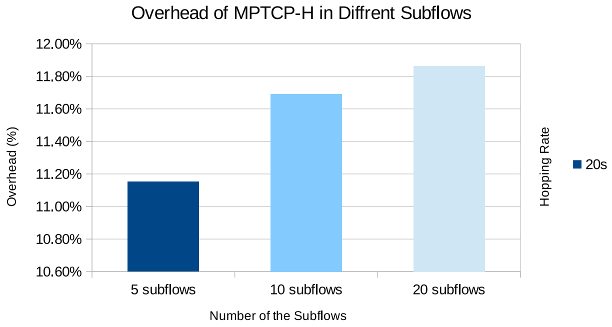


FIGURE 7.9: The effect of the number of subflows on the overhead

experiment for 5 minutes with different hopping rates, number of the subflows, and application layer message rates, i.e, 4ms (250 fps), 8ms (120 fps), 16s (60 fps). Subsequently, we find the ratio of the overhead messages to the whole traffic for each run. We conduct our experiments in the fix time period (5 min) since phasor measurement traffic acts as a continuous data stream unlike typical web applications. By doing so, we find the additional message overhead in the case of phasor measurement traffic.

We first compare TCP, plain MPTCP, and MPTCP-H in terms of the additional message overhead, since TCP is recommended by IEEE standard C37.118.2 for phasor measurement traffic and is a reliable transportation protocol like MPTCP and MPTCP-H. Fig. 7.8 demonstrates that increasing the message rate causes a slight decrease in the message overhead ratio. The reason is that since the increase of application layer message rate does not lead to a linear raise in the message overhead of any protocols, the ratio of the overhead messages to all traffic decreases. In addition, we see that utilizing MPTCP (1 subflow) instead of TCP introduces around 2% of additional message overhead due to MPTCP's additional protocol messages. When we consider high capacity of contemporary network devices, this additional message overhead is reasonable for WAMS. Furthermore, we compare MPTCP (10 subflows) with MPTCP-H (10 subflows) to assess the message overhead caused by our mechanism. As seen in the Fig. 7.8, MPTCP-H does not introduce significant message overhead in comparison to the plain MPTCP. Moreover, it causes an additional 2% of message overhead compared to TCP, similar to plain MPTCP.

Fig. 7.9 shows that when the number of the subflows $s$ increases from 5 to 20, the message overhead also goes up to near 1%. The reason is that the increasing of the number

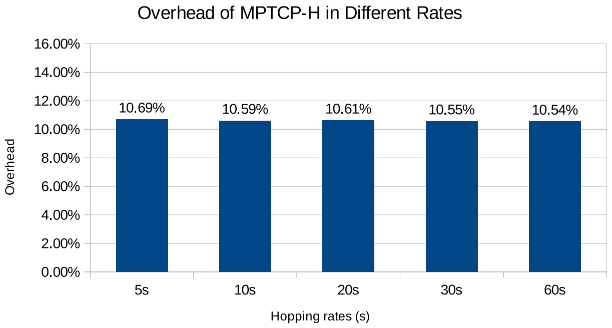


FIGURE 7.10: The effect of hopping rates on the overhead

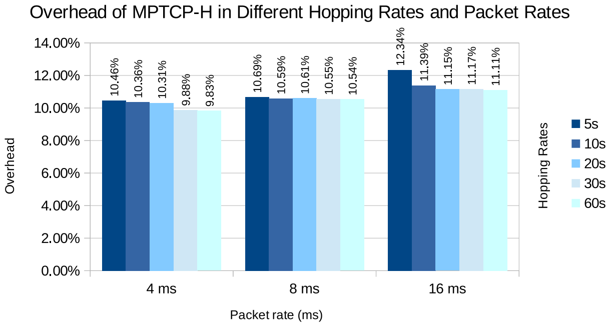


FIGURE 7.11: The effect of hopping rates and packet rates on the overhead

of the subflows (sub-TCP connections) causes additional protocol-based message overhead. The results denote that the number of the subflows $s$ should be minimized due to the high overhead in a network with numerous PMUs. On the other hand, involving a smaller number of subflows eases the discovery of the open ports as explained in Section 7.3.1. Therefore, $s$ should be adapted for different network topologies considering a probable adversary's attack coordination speediness and the trade-off between the $s$-related message overhead and the security consideration.

To show the effect of various hopping rates $t$ on the message overhead, we conduct experiments using 5 subflows in different hopping rates (time periods) $t$. The results indicate that reducing the time period of switching subflows slightly increases the message overhead, as illustrated in Fig. 7.10. However, the increase in message overhead is not as high as $s$. Therefore, we can select the shortest time period/hopping rate $t$ without considering the message overhead.

Lastly, we assess the effect of both different hopping rates and packet rates on the message overhead. The results demonstrate that when the message rate is high (4ms), the ratio of the overhead messages is much lower than the one in the low message rate (16ms), as shown in the Fig. 7.11. This implies that a higher message rate does not lead to a significant message overhead in MPTCP-H. Moreover, the effect of different hopping rates is clearly seen at the low message rate due to existence of less application layer messages in the whole traffic at a low message rate. However, even in the worst case ($t$ = 5s), the increase of the ratio of the message overhead is less than 1%.

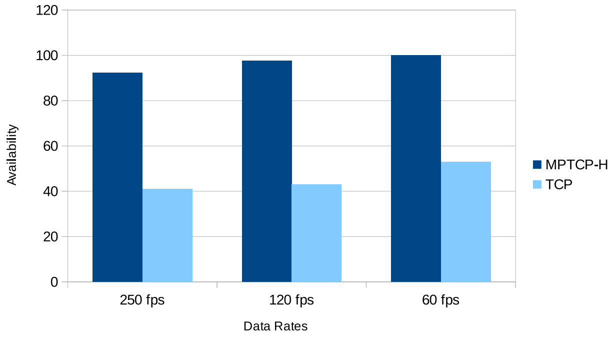


FIGURE 7.12: The system availability provided by MPTCP-H and TCP under DoS attack

### 7.6.2 Under DoS attack conditions

In the following sections we assess our approach under DoS attack conditions.

#### Assessment of the System Availability

We test the availability provided by MPTCP-H and TCP under the DoS attack. The availability refers to the successful delivery rate of the phasor measurements. The attack scenario in our evaluation is setup as follows: The attacker scans the all ports of the target (i.e., PDC or PMU) and then launches a SYN flooding attack against the ports for 5 minutes. We employ different phasor measurement rates, i.e., 250, 120 and 60 fps, while testing the availability of MPTCP-H and TCP under DoS attack.

Fig. 7.12 shows that under the DoS attack, MPTCP-H at a low data rate (60 fps) provides 100% availability. However, the provided availability degree decreases down to 92% with the increase of the data rates from 60 to 250 fps. The reason for this is that until the MPTCP-H switches subflows/ports under attack, mass amounts of data are transmitted in the high data rate scenarios, which can not be handled by the acknowledge mechanism of MPTCP-H. Alternatively, in Fig. 7.12 we see that TCP cannot provide more than 53% availability for any data rate when the PMU/PDC is under attack. Furthermore, with data rate increase, the provided availability degree sharply decreases around 10% like in the case of MPTCP-H.

#### Evaluation the Additional Latency

As we target time-sensitive WAMS applications, we also assess our approach in terms of latency in both DoS attack and attack-free conditions. We run each experiment for the three phasor measurement rates.

Fig. 7.13 demonstrates that the DoS attack causes around 2 ms of additional latency for each data rate when the system uses MPTCP-H. However, as seen in Fig. 7.14 the DoS attack leads to more than 20 ms additional latency for TCP, which is not acceptable by most WAMS applications. Moreover, when we look at Fig. 7.14 and Fig. 7.13, it is clear that MPTCP-H does not cause any additional latency in attack-free cases in comparison to TCP.

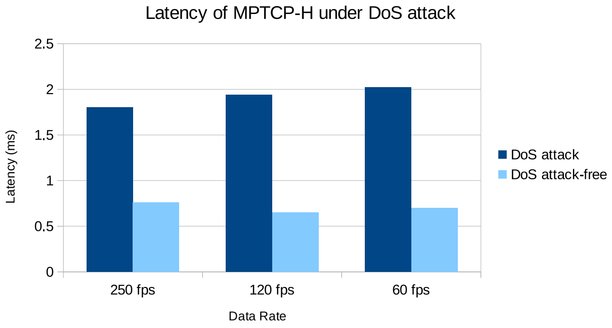


FIGURE 7.13: Latency of MPTCP-H under DoS attack

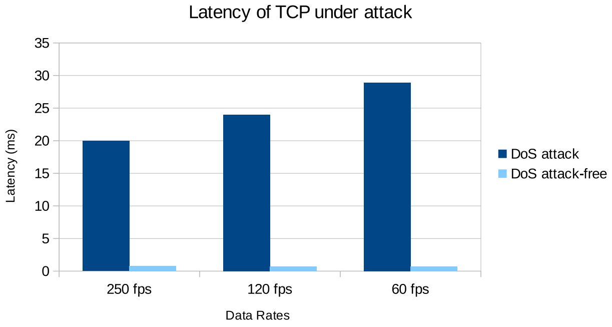


FIGURE 7.14: Latency of MPTCP-H under DoS attack

## Summary

The experiments for latency showed that MPTCP-H, with different network topologies, does not induce any additional latency for the phasor measurement traffic in WAMS in comparison to UDP, as recommended by the IEEE standard C37.118.2 and IEC 61850. Furthermore, using MPTCP instead of TCP introduces reasonable additional message overhead for the contemporary network devices. On the other hand, we test our approach under DoS attack conditions in terms of the system availability and latency. The results show that when the PMU/PDC is under DoS attack, whereas MPTCP-H provides over 92% availability for each data rate, the availability provided by TCP is under 53%. In addition, while the DoS attack causes around 2 ms of additional latency for MPTCP-H, it leads to more than 20 ms of additional latency for TCP. Overall, we can see from the experiments that MPTCP-H provides a significant mitigation of the DoS/DDoS attack with a reasonable overhead.

## 7.7 Conclusion

In this paper, we first surveyed the possible DoS attack threats against the WAMS (i.e., PMUs and PDCs). As a countermeasure against possible DoS attacks, we have proposed an MPTCP-extension, termed MPTCP-H, which basically switches the subflows, removing each subflow after a proscribed $t$ period and adding a new one with a new port, thus hiding the session information from an attacker who is capable of scanning the ports.

As real-time delivery is a crucial requirement for the phasor measurement traffic, we evaluated our approach regarding the additional latency and message with the standard UDP and TCP. The results show that our approach demonstrates a latency performance competitive with even the most lightweight transport protocol of UDP. In addition, MPTCP-H does not introduce any significant additional message overhead in comparison to plain MPTCP and TCP. Further, the experiment results obtained under DoS attack condition indicate that while MPTCP-H provides over 92% availability, TCP cannot provide an availability above 53%.

In this paper, we showed that MPTCP-H, with its lightweight mechanism, can mitigate the attacks originating from inside the WAN. Overall, these results validate that MPTCP does not introduce significant additional overhead that can disturb the phasor measurement traffic while at the same time providing protection against DoS attacks.

Moreover, we plan to test our approach under more sophisticated attacks where the attacker can continuously scan using powerful computers.

# Chapter 8

# Summary and Conclusion

The Smart Grid (SG), differing from the classical power grid with fixed generation sources, dynamically coordinates multiple heterogeneous power sources, distribution and load balancing activities to provide reliable and cost efficient energy services. This is achieved by tightly interlinking the power producers and consumers (the physical resources) using advanced computing/communication technologies (the cyber resources) to form an adaptive cyber-control system, i.e., a state machine. The effectiveness of such cyber-control systems is determined by achieving real-time and accurate state information as obtained from an efficient and reliable communication schema [KK13].

To support the communication requirements of the SG, utilities typically prefer dedicated private E2E communication networks. However, this may not always be achievable due to cost and technical restrictions. Therefore, the SG communication network could become a heterogeneous network consisting of multiple private networks and public network i.e., the Internet [Bud+10; KK13].

However, where the Internet infrastructure is employed for SG communication, the reliability and security issues of the Internet can pose risk for SG operations. Two main problems regarding these issues have been explored in this thesis. The first problem is that the current Internet infrastructure does not inherently provide the necessary QoS guarantees for the safety-critical applications, requiring both low latency and high reliability, due to the Internet's best-effort versus guaranteed delivery service performance. To address this problem, the contribution **(C1)** proposes an overlay network based approach, which provides a disjoint multipath in the Internet and smart resource allocation for critical applications. Thus SG applications obtain high QoS-assurance in the Internet infrastructure. The second problem is due to the Internet's security vulnerabilities that can be exploited by hackers, causing security and safety risks for not only the cyber-system but also for physical-systems, e.g., electrical grid/appliances. In particular, DDoS attacks can be considered as a major threat to the SG applications using over public network. In this regard, we consider three type of DoS/DDoS attacks: 1) volume-based DDoS attacks, 2) transport and application layer DoS/DDoS attacks, and 3) internal DoS/DDoS attacks launched by compromised SG devices. Contributions **(C2)**, **(C3)**, and **(C4)** address these three type of attacks respectively. Contribution **(C2)** uses P2P broker-based pub-sub system with a data diffusion mechanism, which provides protection from volume-based DDoS attacks. Contribution **(C3)** uses a port hopping mechanism whose key disclosure risk is addressed. This mechanism provides a strong mitigation for transport and application layer DoS/DDoS attacks. Finally, contribution **(C4)** focuses on internal attackers that mount application and transport layer attacks exploiting common secrets. To do this, we propose an extension to multipath-TCP which hides the open ports from attacker without using common secrets with other SG devices.

## 8.1 Thesis Contributions

In this section, we briefly summarize the contributions made by this thesis.

- **Contribution 1: Reliable Communication for the SG over Public Networks**

  We propose an overlay network, HetGrid, that addresses the following requirements of SG applications running on public networks: (1) reliable real-time performance, (2) fault-tolerant communication, and (3) E2E QoS-managed delivery. HetGrid selects the overlay nodes with the most adequate resource provisioning to manage inter-Autonomous System (AS) communication rather than place a dedicated server into each domain. In addition, HetGrid needs only local underlay knowledge to enable reliable communication across the network.

  HetGrid's main contributions for SG communication can be summarized as follows: 1) **High reliability over the public networks:** HetGrid strives to build a physically-disjoint multipath, and meets the strict QoS requirements of SG applications via a light-weight, low-overhead communication architecture. To achieve high reliability, it employs Source Routing-based QoS Routing (SRQR) and Compensative Multi-Routing (CMR) mechanisms, and 2) **Application-adaptive and criticality aware resource allocation:** SG applications not only need flow-based (periodic) data acquisition, but also aperiodic data accusation (e.g., alert messages) with diverse QoS requirements. This necessitates a smart resource allocation on the overlay network. Thus, HetGrid employs Altruistic Flow Allocation (AFA) in order to reserve/allocate the "best" paths (in terms of QoS metrics) for high priority (critical) applications in a distributed manner.

- **Contribution 2: A Secure and Reliable Communication Platform for the SG**

  Taking into consideration the security requirements and threats for the SG, we propose a novel pub-sub approach, SeReCP, which provides secure/reliable communication in the case of a volume-based DDoS attack and for link/node failures. In addition, considering the high availability requirements of the SG traffic, we propose a multihoming-based fast "recovery" mechanism in addition to the data diffusion approach, which provide minimum drop/ack/re-transmission over attacks on the intermediate pub-sub brokers. Moreover, given the constraints of SG devices and for their group communication requirements, we introduce a novel group key management mechanism, which provides replay and repudiation attack protection in addition to confidentiality and integrity assurance. Lastly, the evaluation of SeReCP is performed on a real test-bed NorNet, which validates the effectiveness of SeReCP in terms of availability under the attack and, also, its low overhead. SeReCP shows stable communication performance for up to 30% of pub-sub brokers being attacked.

- **Contribution 3: Securing the Cloud-Assisted SG**

  Application and transport layer DDoS attacks represent a serious threat to SG applications, like volume-based DDoS attacks addressed by SeReCP **(C2)**. To mitigate the risk related to DDoS threats, we propose an SG-relevant Hierarchical Hybrid Cloud-Extension Concept (HHCEC) along with a DDoS attack defense mechanism, termed as Port Hopping Spread Spectrum (PHSS). HHCEC is a cloud-assisted architecture designed to meet scalability and security requirements of the SG applications in the cloud. To prevent transport or application-layer DDoS attacks on HHCEC, PHSS switches the open port of server as a function of time and a secret shared between authorized clients and server, thus efficiently dropping packets with invalid port numbers. In addition, PHSS spreads the data packets over all the servers versus a single server to provide a robust protection against DDoS attacks that would affect some of the servers. This approach enables PHSS to instantiate replica servers to take over the attacked servers without blocking all traffic by utilizing the rapid-elasticity characteristic of the cloud. Moreover, PHSS leverages a port shuffling mechanism in order to quarantine malicious clients in a notably short time. Accordingly, the effect of launching a DDoS attack based on the compromised secret is minimized. We evaluate our approach by building a proof-of-concept prototype using Amazon's EC2 and the PlanetLab test-bed. In a DDoS attack scenario, the proposed approach obtains a significant availability improvement of >38%, highlighting its efficiency in comparison to existing approaches. The results also indicate a negligible overhead of less than 0.01% throughput degradation for the proposed approach.

- **Contribution 4: Towards DDoS Attack Resilient Wide Area Monitoring Systems**

  We first surveyed the possible DoS attack threats against the WAMS devices (i.e., PMUs and PDCs) in both the substation network and WAN of the SG. Based on this survey, we introduce a practical threat model where the DoS/DDoS attacks can occur in the substation network or WAN via compromised SG devices, and accordingly saturate the WAMS devices. To counter these threats, we propose a novel defense mechanism, MPTCP-H, which mitigates DoS/DDoS attacks by periodically switching subflows of the MPTCP connection over new port numbers. Furthermore, MHPTCP-H does not need a shared secret between communicating parties, thus avoiding the DoS/DDoS attack from the compromised SG devices. Empirical validation of the MPTCP-H's overhead shows that MPTCP-H performs equal to or better than the standard transportation protocols in terms of latency and congestion attributes.

## 8.2 Limitations and Future Works

When utilities employ the Internet infrastructure for the critical application, they face many security and reliability problems. In this thesis we mainly focus on availability-related problems, as loss of availability posses safety risks for the grid and, perhaps more importantly, for human beings. However data integrity and confidentiality violations are also serious issues in the SG. In particular, data integrity violations can cause critical safety problems in the power grid, when considering the dispersed SG devices in a large geographical area where the SG devices are vulnerable to being physically compromised. Moreover, data integrity violations can also affect communication availability. For example, by sending mass amount of fabricated messages triggering a heavy-duty computation, the compromised SG devices can cause saturation of some control center servers, which can pose significant safety risks for the SG. This problem will be addressed in future research.

# Appendix A

# Appendix A

## A.1 Path Selection and Cost Function Definition

In SRQR, we employ the shortest path (least-cost) routing algorithm for path selection between the ingress and the egress SN. We aim to find the least cost (weight) path which meets the QoS requirements in addition to balancing the link load. Hence, we need to define the weight of the links and the function which computes the weight of paths for the shortest path algorithm.

Let the overlay path pass through $n$ SNs (from SN$_s$ to SN$_d$). Proportional Bandwidth Shortest Path (PBSP) [LM04] defines the path weight function by including the influence of all the concave metrics (e.g., bandwidth, etc.) as: $\sum_i^{n-1}(\frac{B_{i,i+1}}{B_{i,i+1}-R_B} * \frac{C_{i,i+1}}{C_{i,i+1}-R_C})$, where $C_{i,i+1}$ and $R_C$ are residual and required any other concave metric, respectively. The aim of the definition is to maximize the residual bandwidth and other metrics at any link for any path (cf. [LM04]). However this path weight function does not include the influence of additive metrics (e.g., latency). We include the additive metrics' influence over the weight of the path as:

$$PathWeight = \partial_{conc}/n * \partial_{add}, \tag{A.1}$$

where the influence of the concave ($\partial_{conc}$) and the additive ($\partial_{add}$) metrics. We define the influence of the additive metrics ($\partial_{add}$) (latency and reliability) over the weight of the path as;

$$\partial_{add} = (\ell * \Re), \tag{A.2}$$

where $\ell$ and $\Re$ are the influence of latency and reliability over the weight of the path respectively.

Firstly, the latency's influence ($\ell$) is defined based on following criteria. Let $P_n$ and $P_m$ be the probability of choosing the paths which pass through $n$ and $m$ intermediate nodes from $SN_s$ to $SN_d$ respectively:

if $\sum_i^{n-1} L_{i,i+1} > \sum_j^{m-1} L_{j,j+1}$ than $P_n < P_m$.

In the definition of the latency's influence ($\ell$) , our aim is to minimize the current latency at any link for any path, selecting the minimum latency path, as in PBSP, thus: if $\sum_i^{n-1} L_{i,i+1} < R_L$, $\sum_j^{m-1} L_{j,j+1} < R_L$ and $\frac{R_L}{R_L - \sum_i^{n-1} L_{i,i+1}} > \frac{R_L}{R_L - \sum_j^{m-1} L_{j,j+1}}$ then

($P_n = \frac{R_L - \sum_i^{n-1} L_{i,i+1}}{R_L} < P_m = \frac{R_L - \sum_j^{m-1} L_{j,j+1}}{RL}$). The weight of the paths can be specified as 1 /

$P$ . The latency ($\ell$) is defined as:

$$\ell = \frac{R_L}{R_L - \sum_i^{n-1} L_{i,i+1}}. \tag{A.3}$$

Although the reliability is probabilistic metric, it can be converted additive ones by taking logarithm their product [IP11]. Base on this concept, by using a similar approach of the latency, the influence of the reliability over the weight of the path can be defined as:

$$\Re = \frac{\sum_i^{n-1} logR_{i,i+1}}{\sum_i^{n-1} logR_{i,i+1} - logR_R}. \tag{A.4}$$

Let put $\ell$ and $\Re$, defined above, into the equation (A.2):

$$\partial_{add} = \frac{RL}{RL - \sum_i^{n-1} L_{i,i+1}} * \frac{\sum_i^{n-1} logR_{i,i+1}}{\sum_i^{n-1} logR_{i,i+1} - logR_R}. \tag{A.5}$$

Finally, $\partial_{add}$ and $\partial_{conc}$ can be put into the equation (A.1) to get the path weight equation as:

$$PathWeight = \sum_i^{n-1} \left( \frac{B_{i,i+1}}{B_{i,i+1} - R_B} \right) / n *$$

$$* \frac{\sum_i^{n-1} logR_{i,i+1}}{\sum_i^{n-1} logR_{i,i+1} - logR_R} * \frac{R_L}{R_L - \sum_i^{n-1} L_{i,i+1}}. \tag{A.6}$$

# Bibliography

[AAH16] I. Ali, M. A. Aftab, and S. M. S. Hussain. “Performance comparison of IEC 61850-90-5 and IEEE C37.118.2 based wide area PMU communication networks”. In: *Journal of Modern Power Systems and Clean Energy* 4.3 (2016), pp. 487–495.

[Afz17] Z. Afzal. “Towards Secure Multipath TCP Communication”. In: *Diss. Karlstads Universitet* (2017).

[AJZ05] M. Anjia, Y. Jiaxi, and G. Zhizhong. “PMU placement and data processing in WAMS that complements SCADA”. In: *IEEE Power Engineering Society General Meeting, 2005* (2005), pp. 1–4.

[Alb+15] M. Albano et al. “Message-oriented middleware for smart grids”. In: *Computer Standards and Interfaces* 38 (2015), pp. 133–143.

[Ali+13] S. Alishahi et al. “Quality of service guarantee in smart grid infrastructure communication using traffic classification”. In: *Proc. of 22nd International Conference and Exhibition on Electricity Distribution (CIRED)* (2013), pp. 0803–0803.

[Ama16] Amazon Web Services. “Amazon Web Services (AWS) - Cloud Computing Services”. In: *https://aws.amazon.com/ (Last visited on 08-08-2017)* (2016).

[And+01] D. Andersen et al. “Resilient overlay networks”. In: *Proc. of the 8th ACM symposium on Operating systems principles (SOSP)* 35.5 (2001), pp. 131–145.

[AP15] S. Asri and B. Pranggono. “Impact of Distributed Denial-of-Service Attack on Advanced Metering Infrastructure”. In: *Wireless Personal Communications* 83.3 (2015), pp. 2211–2223.

[AP17] R. Ahmad and A. Pathan. “A Study on M2M (Machine to Machine) System and Communication: Its Security, Threats, and Intrusion Detection System”. In: *Security Solutions and Applied Cryptography in Smart Grid Communications* (2017), pp. 179–214.

[Bae+15] J. Baek et al. “A secure cloud computing based framework for big data information management of smart grid”. In: *IEEE Transactions on Cloud Computing* 3.2 (2015), pp. 233–244.

[Bak+11] D. E. Bakken et al. “Smart generation and transmission with coherent, real-time data”. In: *Proc. of the IEEE* 99.6 (2011), pp. 928–951.

[BHK07a] G. Badishi, A. Herzberg, and I. Keidar. “Keeping denial-of-service attackers in the dark”. In: *IEEE Transactions on Dependable and Secure Computing* 4.3 (2007), pp. 191–204.

[BHK07b] I. Baumgart, B. Heep, and S. Krause. "OverSim: A Flexible Overlay Network Simulation Framework". In: *IEEE Global Internet Symposium* (2007), pp. 79–84.

[BI14] D. Bakken and K. Iniewski. "Smart Grids: Clouds, Communications, Open Source, and Automation". In: *CRC Press* (2014), pp. 435–446.

[Bla+98] S. Blake et al. "An architecture for differentiated services". In: *IETF draft RFC 2475* (1998).

[Bla+99] J. Black et al. "UMAC: Fast and Secure Message Authentication". In: *Proc. of 19th Annual International Cryptology Conference* (1999), pp. 216–233.

[BMR15] S. Bera, S. Misra, and J. P. C. Rodrigues. "Cloud Computing Applications for Smart Grid: A Survey". In: *IEEE Transactions on Parallel and Distributed Systems* 26.5 (2015), pp. 1477–1494.

[BOP94] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson. "TCP Vegas". In: *Proc. of the conference on Communications architectures, protocols and applications - SIGCOMM* 24.4 (1994), pp. 24–35.

[BS11] R. Berthier and W. H. Sanders. "Specification-Based Intrusion Detection for Advanced Metering Infrastructures". In: *Proc. of 17th IEEE Pacific Rim International Symposium on Dependable Computing* (2011), pp. 184–193.

[Bud+10] K. C. Budka et al. "Communication network architecture and design principles for smart grids". In: *Bell Labs Technical Journal* 15.2 (2010), pp. 205–227.

[Cio+15] C. Ciontea et al. "Smart grid control and communication: The SmartC2net Real-Time HIL approach". In: *Proc. of IEEE Eindhoven PowerTech* (2015), pp. 1–6.

[DBC09] W. Dantas, A. Bessani, and M. Correia. "Not quickly, just in time: Improving the timeliness and reliability of control traffic in utility networks". In: *Proc. of the 5th Workshop on Hot Topics in System Dependability* (2009).

[Dec+10] G. Deconinck et al. "Communication overlays and agents for dependable smart power grids". In: *Proc. of 5th International Conference on Critical Infrastructure (CRIS)* (2010), pp. 1–7.

[DGS14] K. Demir, D. Germanus, and N. Suri. "Robust and real-time communication on heterogeneous networks for smart distribution grid". In: *IEEE International Conference on Smart Grid Communications (SmartGridComm)* (2014), pp. 386–391.

[DGS15] K. Demir, D. Germanus, and N. Suri. "Robust QoS-aware communication in the smart distribution grid". In: *Peer-to-Peer Networking and Applications* 10.1 (2015), pp. 193–207.

[Dix94] R. C. Dixon. "Spread spectrum systems: with commercial applications". In: *Wiley New York* (1994).

[DO17] G. Demaude and P. Ortegat. "https://github.com/reirep/matcp-java.git". In: *Last accessed on 03-08-2017* (2017).

[Dre15] T. Dreibholz. "The NorNet Testbed A Large-Scale Experiment Platform for Real-World Experiments with Multi-Homed Systems," in: *https://www.simula. no /research/projects/nornet (Last visited on 08-08-2017)* (2015).

[DS17a] K. Demir and N. Suri. "Securing the Cloud-Assisted Smart Grid". In: *Submitted* (2017).

[DS17b] K. Demir and N. Suri. "SeReCP: A Secure and Reliable Communication Platform for the Smart Grid". In: *Proc. of the 22nd IEEE Pacific Rim International Symposium on Dependable Computing (PRDC)* (2017), pp. 175–184.

[DS17c] K. Demir and N. Suri. "Towards DDoS Attack Resilient Wide Area Monitoring Systems". In: *Proc. of the 12th International Conference on Availability, Reliability and Security (ARES )* (2017), pp. 1–7.

[EBB08] M. El Hachimi, M. A. Breton, and M. Bennani. "Efficient QoS implementation for MPLS VPN". In: *Proc. of International Conference on Advanced Information Networking and Applications, AINA* (2008), pp. 259–263.

[For+13] A. Ford et al. "TCP extensions for multipath operation with multiple addresses". In: *IETF RFC 6824* (2013).

[FP12] Z. Fu and M. Papatriantafilou. "Off the Wall: Lightweight Distributed Filtering to Mitigate Distributed Denial of Service Attacks". In: *Proc. of IEEE 31st Symposium on Reliable Distributed Systems* (2012), pp. 207–212.

[FPT12] Z. Fu, M. Papatriantafilou, and P. Tsigas. "Mitigating distributed Denial of Service attacks in multiparty applications in the presence of clock drifts". In: *IEEE Transactions on Dependable and Secure Computing* 9.3 (2012), pp. 401–413.

[GP01] T. M. Gil and M. Poletto. "MULTOPS : a data-structure for bandwidth attack detection". In: *Proc. of the 10 th USENIX Security Symposium* (2001), pp. 23–28.

[Gun+13] V. C. Gungor et al. "A Survey on smart grid potential applications and communication requirements". In: *IEEE Transactions on Industrial Informatics* 9.1 (2013), pp. 28–42.

[Hei+15] F. Heimgaertner et al. "A security architecture for the publish/subscribe C-DAX middleware". In: *Proc. of IEEE International Conference on Communication Workshop (ICCW )* (2015), pp. 2616–2621.

[Hes17] B. Hesmans. "A socket API to control Multipath TCP". In: *https://tools.ietf.org/html/draft-hesmans-mptcp-socket-00 Last accessed on 03-08-2017* (2017).

[HW03] E. Hansen and W. G. Walster. "Global optimization using interval analysis". In: *CRC Press* (2003).

[HWJ08] J. Han, D. Watson, and F. Jahanian. "Enhancing end-to-end availability and performance via topology-aware overlay networks". In: *Computer Networks* 52.16 (2008), pp. 3029–3046.

[IP11] D. Ilie and A. Popescu. "Unicast QoS Routing in Overlay Networks". In: *Network Performance Engineering* (2011). Ed. by Demetres D Kouvatsos, pp. 1017–1038.

[Jia+14] Q. Jia et al. "Catch me if you can: A cloud-enabled DDoS defense". In: *Proc. of 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)* (2014), pp. 264–275.

[Kan+09] P. M. Kanabar et al. "Evaluation of Communication Technologies for IEC 61850 based Distribution Automation System with Distributed Energy Resources". In: *IEEE Power & Energy Society General Meeting* (2009), pp. 1–8.

[KAR13] M. Kanabar, M. G. Adamiak, and J. Rodrigues. "Optimizing Wide Area Measurement System architectures with advancements in Phasor Data Concentrators (PDCs)". In: *Proc. of IEEE Power & Energy Society General Meeting* (2013), pp. 1–5.

[Kar14] B. Karthikeyan. "Detecting and Isolating Distributed Denial of Service Attack in Smart Grid Systems". In: *Diss. National Institute of Technology Rourkela* (2014).

[Kho+13] S. S. Khorasani et al. "QoS Assurance in Smart Grid for IP-based Applications of Mashhad Electric Energy Distribution Company". In: *Proc. of 22nd International Conference and Exhibition on Electricity Distribution (CIRED)* (2013), pp. 0906–0906.

[Kim+12] Y. Kim et al. "SeDAX: A Scalable, Resilient, and Secure Platform for Smart Grid Communications". In: *IEEE Journal on Selected Areas in Communications* 30.6 (2012), pp. 1119–1136.

[KK13] R. H. Khan and J. Y. Khan. "A comprehensive review of the application characteristics and traffic requirements of a smart grid communications network". In: *Computer Networks* 57.3 (2013), pp. 825–845.

[KMR02] A. D. Keromytis, V. Misra, and D. Rubenstein. "SOS: Secure Overlay Services". In: *Electrical Engineering* 32.4 (2002), pp. 61–72.

[Lav+10] D. M Laverty et al. "Telecommunications for Smart Grid: Backhaul solutions for the distribution network". In: *IEEE PES General Meeting* (2010), pp. 1–6.

[LK12] S. I. Lee and S. G. Kang. "NGSON: Features, state of the art, and realization". In: *IEEE Communications Magazine* 50.1 (2012), pp. 54–61.

[LM04] Z. Li and P. Mohapatra. "QRON: QoS-aware routing in overlay networks". In: *IEEE Journal on Selected Areas in Communications* 22.1 (2004), pp. 29–40.

[LT04] H.J.C. Lee and V.L.L. Thing. "Port hopping for resilient networks". In: *IEEE 60th Vehicular Technology Conference (VTC)* (2004), pp. 3291–3295.

[LWC14] Y. Luo, B. Wang, and G. Cai. "Effectiveness of Port Hopping as a Moving Target Defense". In: *Proc. of 7th International Conference on Security Technology* (2014), pp. 7–10.

[Mah+13] K. Maheshwari et al. "Toward a reliable, secure and fault tolerant smart grid state estimation in the cloud". In: *IEEE PES Innovative Smart Grid Technologies Conference, ISGT* (2013), pp. 1–6.

[Mar+14] K. E. Martin et al. "An Overview of the IEEE Standard C37.118.2 Synchrophasor Data Transfer for Power Systems". In: *IEEE Transactions on Smart Grid* 5.4 (2014), pp. 1980–1984.

[Med+01] A. Medina et al. "BRITE: an approach to universal topology generation". In: *Proc. of Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems* (2001), pp. 346–353.

[Mor+11] T. Morris et al. "Cybersecurity risk testing of substation phasor measurement units and phasor data concentrators". In: *Proc.of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW)* (2011), pp. 1–4.

[Nah+01] E. M. Nahum et al. "The effects of wide-area conditions on WWW server performance". In: *Proc. of the ACM SIGMETRICS international conference on Measurement and modeling of computer systems* 29.1 (2001), pp. 257–267.

[Nav+13] J. Navarro et al. "The Information System of INTEGRIS: INTelligent Electrical GRId Sensor Communications". In: *IEEE Transactions on Industrial Informatics* 9.3 (2013), pp. 1548–1560.

[NSS10] E. Nygren, R. K. Sitaraman, and J. Sun. "The Akamai network". In: *ACM SIGOPS Operating Systems Review* 44.3 (2010), p. 2.

[NT94] B.C. Neuman and T. Ts'o. "Kerberos: an authentication service for computer networks". In: *IEEE Communications Magazine* 32.9 (1994), pp. 33–38.

[OK10] M. Oldak and B. Kilbourne. "Communications requirements comments of utilities telecom council". In: *Department of Energy, Washington, DC, USA* (2010).

[Paa+14] C. Paasch et al. "Experimental evaluation of multipath TCP schedulers". In: *Proc. of the ACM SIGCOMM workshop on Capacity sharing workshop (CSWS )* (2014), pp. 27–32.

[Paa17] C. Paasch. "Multipath TCP in the Linux Kernel". In: *http://www.multipath-tcp.org, Last visited on 23-04-2017* (2017).

[PB12] C. Paasch and O. Bonaventure. "Securing the MultiPath TCP handshake with external keys". In: *Work in Progress, draft-paasch-mptcp-ssl-00* (2012).

[Pon93] G. Pongor. "OMNeT: Objective Modular Network Testbed". In: *Proc. of International workshop on Modelling ,Analysis & Simulation on computer and telecommunication system (MASCOT)* (1993), pp. 323–326.

[Pre+14] T. Predojev et al. "A real-time middleware platform for the smart grid". In: *Proc. of IEEE Online Conference on Green Communications, OnlineGreenComm* (2014), pp. 1–6.

[PSZ17] S. Paudel, P. Smith, and T. Zseby. "Attack Models for Advanced Persistent Threats in Smart Grid Wide Area Monitoring". In: *Proc. of the 2nd Workshop on Cyber-Physical Security and Resilience in Smart Grids - CPSR-SG* (2017), pp. 61–66.

[Rai+11] C. Raiciu et al. "Improving datacenter performance and robustness with multipath TCP". In: *Proc. of the ACM conference on SIGCOMM* 41.4 (2011), p. 266.

[RGZ06] S. Ren, L. Guo, and X. Zhang. "ASAP: An AS-aware peer-relay protocol for high quality VoIP". In: *Proc. of International Conference on Distributed Computing Systems* (2006), pp. 70–80.

[Riz+14] T. A. Rizzetti et al. "Methods of availability assurance for communication of PMU in a smart grid based on IP protocol". In: *Proc. of 49th International Universities Power Engineering Conference (UPEC)* (2014), pp. 1–6.

[RVC00] E. Rosen, A. Viswanathan, and R. Callon. "Multiprotocol label switching architecture". In: *RFC 3031* (2000).

[Sef+14] V. Seferian et al. "PUF and ID-based key distribution security framework for advanced metering infrastructures". In: *Proc. of IEEE International Conference on Smart Grid Communications (SmartGridComm)* (2014), pp. 933–938.

[SK05] A. Stavrou and A. D. Keromytis. "Countering DoS attacks with stateless multipath overlays". In: *Proc. of the 12th ACM Conference on Computer and Communications Security* (2005), pp. 249–259.

[Sta+05] A. Stavrou et al. "MOVE: An End-to-End Solution To Network Denial of Service". In: *Proc. of the ISOC Symposium on Network and Distributed System Security (SNDSS)* (2005), pp. 81–96.

[Ste+14] C. Stefanovic et al. "SUNSEED - An evolutionary path to smart grid comms over converged telco and energy provider networks". In: *Proc. of 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace and Electronic Systems (VITAE) - Co-located with Global Wireless Summit* (2014), pp. 1–5.

[Sub+04] L. Subramanian et al. "OverQoS: An Overlay Based Architecture for Enhancing Internet QoS." In: *NSDI* 4.6 (2004), pp. 71–84.

[US 10] U.S. Department of Energy. "Implementing the National Broadband Plan by Studying the Communications Requirements of Electric Utilities To Inform Federal Smart Grid Policy | Department of Energy". In: *Technical Report* (2010).

[Vul+12] A. Vulimiri et al. "More is Less : Reducing Latency via Redundancy". In: *Proc. of the 11th ACM Workshop on Hot Topics in Networks (HotNets-XI)* (2012), pp. 13–18.

[Wan+13] G. Wang et al. "An efficient relay node selection scheme to improve the performance of P2P-based VoIP applications in Chinese internet". In: *Multimedia Tools and Applications* 64.3 (2013), pp. 599–625.

[Wei+10] D. Wei et al. "An integrated security system of protecting smart grid against cyber attacks". In: *Innovative Smart Grid Technologies Conference (ISGT)* (2010), pp. 1–7.

[WL13] W. Wang and Z. Lu. "Cyber security in the Smart Grid: Survey and challenges". In: *Computer Networks* 57.5 (2013), pp. 1344–1371.

[Wro97] J. Wroclawski. "The use of RSVP with IETF integrated services". In: *RFC 2210* (1997).

[WYB15] Y. Wang, P. Yemula, and A. Bose. "Decentralized communication and control systems for power system operation". In: *IEEE Transactions on Smart Grid* 6.2 (2015), pp. 885–893.

[Yan+09] H. Yang et al. "Message-Oriented Middleware with QoS Awareness". In: *Service-Oriented Computing* (2009), pp. 331–345.

[YBG11] Q. Yang, J. A. Barria, and T. C. Green. "Communication Infrastructures for Distributed Control of Power Distribution Networks". In: *IEEE Transactions on Industrial Informatics* 7.2 (2011), pp. 316–327.

[Zha+02] B. Zhao et al. "Brocade: Landmark routing on overlay networks". In: *Peer-to-Peer Systems* (2002), pp. 34–44.

[Zha+10] Y. Zhang et al. "Wide-Area Frequency Monitoring Network (FNET) Architecture and Applications". In: *IEEE Transactions on Smart Grid* 1.2 (2010), pp. 159–167.

[Zha+11] Y. Zhang et al. "Distributed Intrusion Detection System in a Multi-Layer Network Architecture of Smart Grids". In: *IEEE Transactions on Smart Grid* 2.4 (2011), pp. 796–808.

[ZJT13] S. T. Zargar, J. Joshi, and D. Tipper. "A survey of defense mechanisms against distributed denial of service (DDOS) flooding attacks". In: *IEEE Communications Surveys and Tutorials* 15.4 (2013), pp. 2046–2069.

[ZSP16] S. Zannettou, M. Sirivianos, and F. Papadopoulos. "Exploiting path diversity in datacenters using MPTCP-aware SDN". In: *Proc. of IEEE Symposium on Computers and Communication (ISCC)* (2016), pp. 539–546.