Resilient and Reliable Communication for First Responders with Ad Hoc Network and MPTCP

by

Yue Cui

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
Dec, 9, 2023

Keywords: OLSR, MPTCP, MANET

Approved by

Dr. Alvin Lim, Chair, Professor of Computer Science and Software Engineering
Dr. Shiwen Mao, Professor of Electrical and Computer Engineering
Dr. Cheryl Seals, Professor of Computer Science and Software Engineering
Dr. Xiao Qin, Professor of Computer Science and Software Engineering
Dr. Xiaowen Gong, Assistant Professor of Electrical and Computer Engineering

Abstract

In severe hazardous situations, reliable network communication is an important infrastructure for conveying vital real-time information from sensors embedded in first responders. Important communications and emergency information must be received quickly to make critical decisions, improve situation awareness, and save lives. It is always challenging to ensure that the network is continuously available. To ensure seamless, fault-tolerant, and secure communication, we propose a new communication system: Next-Generation First Responder Communication Hubs (NGFR Communication Hubs). In this novel architecture, we improve the reliability of our communication networks using Multi-Path TCP (MPTCP), Optimized Linked State Routing Protocol (OLSR), and Message Queuing Telemetry Transport (MQTT) for portable devices and cloud services. MPTCP enables rapid recovery when a TCP connection fails while maintaining an end-to-end connection feature. OLSR, as a proactive routing protocol in Ad Hoc networks, provides an infrastructure-less architecture in MANET. MQTT enables efficient transmission of sensor data. We also give solutions on solving the dynamic gateway switch problem in the OLSR network by utilizing a TCP Fast Open packet. We have implemented the above reliable communication hub and conducted extensive experiments in different configurations. The results show that our proposed architecture can be quickly deployed and provides the reliability of the communication network despite the failure of some network infrastructures.

Acknowledgments

I would like to express my deepest gratitude to Dr. Alvin Lim, my graduate advisor, for his encouragement and guidance throughout my journey at Auburn University. His support and guidance were instrumental in the success of my research projects and thesis completion.

I extend my heartfelt thanks to Dr. Shiwen Mao for sharing his valuable experience and insights into research. I am also grateful to Dr. Cheryl Seals for her invaluable guidance during my Ph.D. study. I would also like to thank Dr. Qin Xiao for his tremendous help and efforts. Additionally, I would like to express my gratitude to Dr. Xiaowen Gong for serving as my university reader and providing me with valuable feedback and unwavering support. I am deeply appreciative of the committee members for their time, support, and valuable advice throughout the preparation and implementation of my research and thesis.

My sincere appreciation goes to all my lab mates and colleagues for the invaluable knowledge and experiences. Lastly, I want to express my heartfelt thanks to my parents and friends for their support during my time at Auburn University. Their encouragement has been a driving force behind my achievements.

Table of Contents

Abstract ii

Acknowledgments iii

List of Abbreviations xi

1 Introduction 1

1.1 Background 3

1.1.1 MultiPath-TCP 3

1.1.2 MANET 8

1.1.3 MQTT 14

1.1.4 TCP Fast Open 15

2 Related work 17

2.1 MPTCP Based Transmission Scheme 17

2.2 Real-time Data Transmission in Ad Hoc Networks with Internet 19

2.3 Routing in Ad Hoc Networks 20

3 Problem Statement and Motivation 22

3.1 Problem Statement 22

3.2 Motivation 24

4 Conceptual Overview 25

4.1 Overview 25

4.2 NGFR Communication Hub in Network Stack 25

5 Resilient and Reliable Communication Hub 30

5.1 Motivation 30

5.2 Experiment Set Up 31

5.3 Experiment Configurations 31

5.4 Experiment Design 35

5.5 Experiment Data Selection 36

5.6 Stress Testing 36

5.7 Performance Metrics 36

5.8 Implementation and Result Analysis 37

5.8.1 One Publisher with MPTCP 40

5.8.2 Multiple publishers in MPTCP 41

5.8.3 Link Switch in TCP 50

5.9 Summary 60

6 Multiple Gateways in NGFR Communication Hub 61

6.1 Motivation 61

6.2 Proposed Solution 62

6.3 Experiment Configuration 63

6.4 Experiment Design 63

6.5 Result Analysis 66

6.6 Summary 70

7 Dynamic Gateway Switch in OLSR network 71

7.1 Motivation 71

7.2 Proposed Solution 71

7.3 Implementation 75
7.4 Experiment Set Up 76
7.5 Result Analysis 77
7.6 Summary 78

8 Conclusion and Future Work 82
8.1 Conclusion 82
8.2 Future Work 84
8.2.1 Security Challenges 84
8.2.2 Other Multiple-path Based Scheme 84

References 86

List of Figures

1.1 Comparison on TCP and MPTCP Protocol Stack 4

1.2 Initiation Process of a main flow and a subflow in MPTCP 5

1.3 Default Scheduler in MPTCP 7

1.4 Redundant Scheduler in MPTCP 7

1.5 Round-Robin Scheduler in MPTCP 7

1.6 Conceptual representation of a MANET 9

1.7 MANET extending to the Internet 9

1.8 Broadcast packets forward by MPR 13

1.9 An example of MQTT publish/subscribe architecture 15

3.1 TCP interface switch controller 23

4.1 Network Architecture in NGFR Communication Hubs 26

4.2 Prototype in NGFR Communication Hubs 28

4.3 NGFR Communication Hubs in TCP/IP Stack 29

5.1 Network Partition in a Private Network Setup 32

5.2 3 publishers with a maximum of 3 hops 32

5.3 2 publishers with a maximum of 4 hops 33

5.4 2 publishers with a maximum of 5 hops 34

5.5 Implementation of NGFR Communication Hubs 38

5.6 Screenshot of the running interface 39

5.7 Comparison of round trip time in different number of hops with one publisher . 42

5.8 Comparison of packet delivery ratios in different number of hops with one publisher 43

5.9 Comparison of throughput in different numbers of hops with one publisher . . . 44

5.10 Throughput split in publisher with 3 hops . . . 45

5.11 Throughput split in publisher with 6 hops . . . 46

5.12 Comparison of packet delivery ratio with 3 hops . . . 47

5.13 Comparison of throughput with 3 hops . . . 48

5.14 Comparison of round trip time with 3 hops . . . 49

5.15 Comparison of throughput in 3 hops . . . 51

5.16 Comparison of round trip time in 3 hops . . . 52

5.17 Comparison of packet delivery ratio in 3 hops . . . 53

5.18 Stress test for comparison of round trip time in 3 hops . . . 54

5.19 Stress test for comparison of packet delivery ratio in 3 hops . . . 55

5.20 Stress test for comparison of throughput in 3 hops . . . 56

5.21 Regular TCP splits in the publisher . . . 58

5.22 TCP with link detection splits in publisher . . . 59

6.1 2 publishers and 2 gateways with a maximum of 3 hops . . . 64

6.2 2 publishers and 2 gateways with a maximum of 5 hops . . . 65

6.3 Comparison of round trip time in different number of hops with two publishers and two gateways . . . 67

6.4 Comparison of packet delivery ratios in different number of hops with two publishers and two gateways . . . 68

6.5 Comparison of throughput in different number of hops with two publishers and two gateways . . . 69

7.1 Multiple gateways MANET . . . 72

7.2 Gateway failure . . . 72

7.3 Dynamic gateway selection with TFO . . . 75

7.4 HNA message format . . . 76

7.5 Throughput Comparison on Gateways with 3 Hops . . . 79

7.6 Throughput Comparison on Gateways with 4 hops 80

7.7 Throughput Comparison on Gateways with 5 hops 81

List of Tables

5.1 Experiment design 35

6.1 Experiment Design with Multiple Gateways 64

7.1 Experiment Design with Switch in Multiple Gateways 77

List of Abbreviations

ACK Acknowledgement

AODV Ad Hoc On-demand Distance Vector

cwnd Congestion Window

DSDV Destination Sequenced Distance Vector

DSR Dynamic Source Routing

ETX Expected Transmission Count

HNA Host and Network Association

IoT Internet of Things

LQ Link Quality

MANET Mobile Ad Hoc Networks

MPR Multipoint Relay

MPTCP Multipath TCP

MQTT Message Queuing Telemetry Transport

NLQ Neighbor Link Quality

OLSR Optimized Linked State Routing Protocol

PDR Packet Delivery Ratio

RTT	Round Trip Time
TC	Topology Control
TCP	Transmission Control Protocol
TFO	TCP Fast Open
TLS	Transport Layer Security
UDP	User Datagram Protocol

Chapter 1

Introduction

Reliable communication in networks is of importance, particularly in extreme hazardous environments such as conflagrations, earthquakes, hurricanes, and other emergencies[26]. The ability to transmit vital real-time information becomes crucial in these situations as important messages and emergency information are time-sensitive and essential for making critical decisions, enhancing real-time situation awareness, and ultimately, saving lives. However, ensuring that the network is always available in such challenging circumstances can be a significant obstacle [37].

With the evolution of technologies like Internet of Things (IoT) and smart cities, modern critical infrastructure networks are progressively relying more on communication networks. The advancement of these technologies has resulted in heightened interdependence among networks, as indicated in [22], where the failure of one network can lead to the failure of other interconnected networks. Noteworthy examples of such interdependence include the 2003 North American blackout[20], [67], the 2003 Italian blackout[46], and the 2012 Hurricane Sandy[28]. During the 2003 U.S. Northeastern power outage, abnormal connectivity outages affected 3,175 communication networks[20]. Undoubtedly, the role of reliable communication systems in today's critical infrastructure networks has gained increasing significance. One example that highlights the importance of reliable communication is the unprecedented winter storm that struck Texas, USA, on 16 February 2021[5]. The storm caused widespread disruptions, leading to cellular outages for all major carriers in the affected area. These outages not only resulted in a loss of communication but also inflicted irreparable economic damages.

To address the challenges of ensuring continuous and dependable communication during emergencies, we propose a novel network system architecture named Next Generation First Responders Communication Hubs (NGFR Communication Hubs). The primary objective of the communication hubs is to provide self-deployment, fault-tolerant, and secure communications in extreme and hazardous environments.

To achieve this goal, we employ several technologies and protocols within this novel network architecture. Multi-path TCP (MPTCP) is utilized to enhance network reliability by distributing data across multiple paths. This approach allows for improved fault tolerance and robustness, ensuring that even if one path fails, communication can continue through alternate paths.

Additionally, Optimized Linked State Routing Protocol (OLSR) is implemented to optimize the routing process within the network. OLSR is a proactive routing protocol that establishes and maintains routing tables, enabling efficient and reliable data transmission between network nodes. By leveraging OLSR, the NGFR Communication Hubs can dynamically adapt to changes in the network topology, ensuring that data reaches its intended destination promptly.

Furthermore, Message Queuing Telemetry Transport (MQTT) protocol is utilized to enhance the efficiency and reliability of message delivery. MQTT is a lightweight, publish-subscribe messaging protocol that enables reliable communication between devices, even in constrained network environments. By employing MQTT, the NGFR Communication Hubs can efficiently transmit critical messages and emergency information, ensuring that they reach the intended recipients in a timely manner.

A key aspect of our proposed network system architecture is its self-configuring nature, requiring minimal user involvement. The NGFR Communication Hubs are designed to automatically adapt and reconfigure themselves in response to changes in the network environment. This self-reconfiguration capability ensures that the network remains functional and reliable, even as conditions evolve during hazardous situations.

In summary, the Next Generation First Responders Communication Hubs (NGFR Communication Hubs) provide a novel network system architecture that aims to deliver seamless,

fault-tolerant, and secure communication in extreme and hazardous environments. By incorporating technologies such as Multi-path TCP (MPTCP), Optimized Linked State Routing Protocol (OLSR), and Message Queuing Telemetry Transport (MQTT), we enhance the reliability of networks using portable devices. Our ultimate objective is to create multi-path networks that require minimal user involvement while ensuring continuous and dependable communication, ultimately aiding first responders in their life-saving efforts.

1.1 Background

1.1.1 MultiPath-TCP

Nowadays, the majority of communication devices are equipped with multiple communication interfaces, such as 802.3 and WiFi[53]. Typically, these devices utilize a single communication interface for data exchange. However, the simultaneous utilization of multiple communication interfaces is anticipated to offer enhanced performance in terms of both throughput and delay[87]. This becomes particularly crucial in emergency situations. Multi-path TCP(MPTCP) is an extension of regular TCP which provides multiple TCP flows simultaneously. An Multipath-TCP capable device can partition a single TCP datastream among multiple subflows while remaining the regular TCP socket API to the application[9]. MPTCP begins its connection similarly to a regular TCP. The first established link will be the main flow of an MPTCP connection. If extra paths are available, it creates subflows for these paths.Figure 1.1 shows the comparison between a regular TCP and an MPTCP protocol stack.

Initiation of MPTCP Connection

When initiating a new connection, an option `MP_CAPABLE` will be carried in SYN, SYN/ACK, and ACK packets. This verifies whether the remote host supports MPTCP, as well as exchanges authentication information for establishing other subflows. After the main flow has started, further sub-flows can be added to the connection. The new subflow is also started as a regular TCP connection with the SYN/ACK exchange. `MP_JOIN` is used to identify the connection to be joined by the new subflow. Figure 1.2 shows the initiation of a main flow and a sublow

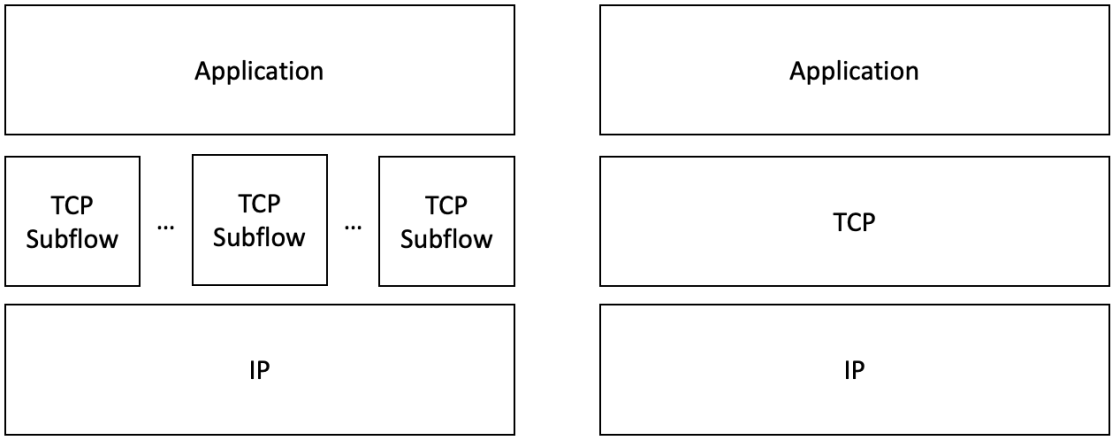


Figure 1.1: Comparison on TCP and MPTCP Protocol Stack

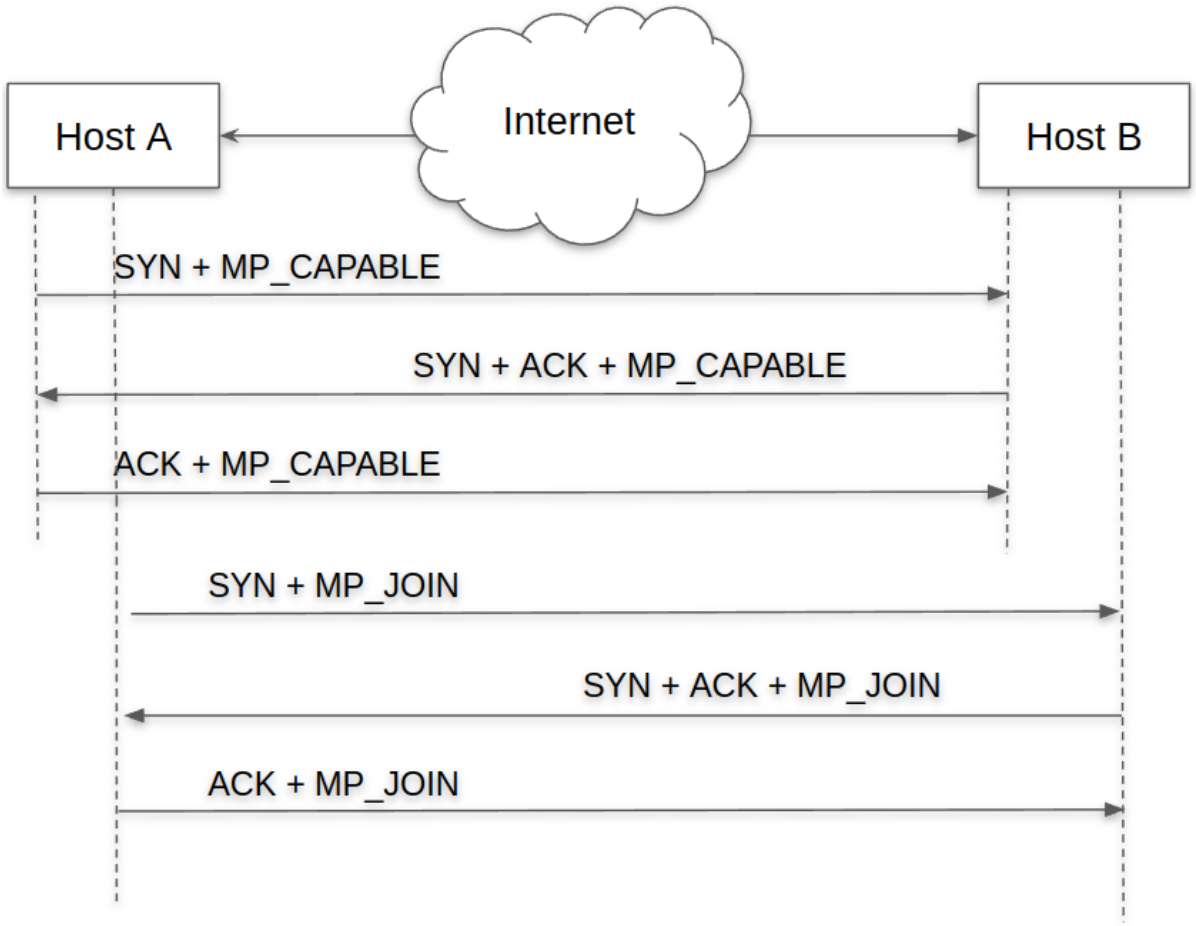


Figure 1.2: Initiation Process of a main flow and a subflow in MPTCP

in MPTCP. From the perspective of middle-boxes, each subflow is a normal TCP connection. MPTCP is responsible for splitting the data flows over these sub-flows at the source and re-assembling them at the destination. MPTCP is transparent to applications that use a regular TCP socket for communication. If the destination does not support MPTCP, the sender will use regular TCP for data transmission over the established connection.

MPTCP Scheduler in Heterogeneous Network

A heterogeneous network is a network that connects different types of devices with multiple protocols or service providers[75]. With MPTCP in heterogeneous networks, users can transport the connections between several subflows, benefiting from the best available services. Therefore, the core element of the MPTCP design is the scheduler, which is used for two tasks:first, choose a subflow among all the available TCP subflows and; second, decide how to distribute packets in subflows. There are three main scheduling policy implementations for MPTCP in Linux: Round-Robin, minRTT, and Redundant[79].

Round-Robin scheduler allocates data packets on a circular way among the available subflows[8]. As shown in Figure 1.5, there are two subflows available. The first one has an RTT of 1ms with a bandwidth of 1 packet/second. The second subflow has an RTT of 10ms with a bandwidth of 1 packet/second. With the Round-Robin scheduler, packets are assigned to one subflow after another. This scheduler ensures that each subflow's capacity is fully utilized as an equal distribution across all subflows. However, it does not consider the heterogeneous networks and not take into consideration of the bandwidth dissimilarity among subflows[25].

minRTT is the default scheduler for MPTCP in Linux to date[70],[71]. It sends packets first through the available path with the smallest estimated Round Trip Time (RTT) until the congestion window is full. Then it starts transmitting on the subflow with the next lowest RTT. Figure 1.3 shows the sending and receiving process with minRTT scheduler. There are two subflows available. The first one has an RTT of 1ms with a bandwidth of 1 packet/second. The second subflow has an RTT of 10ms with a bandwidth of 1 packet/second. With minRTT scheduler, subflow_1 will be chosen first before its congestion window is filled up. Then, data is sent on the subflow_2 since it has the next lowest RTT.

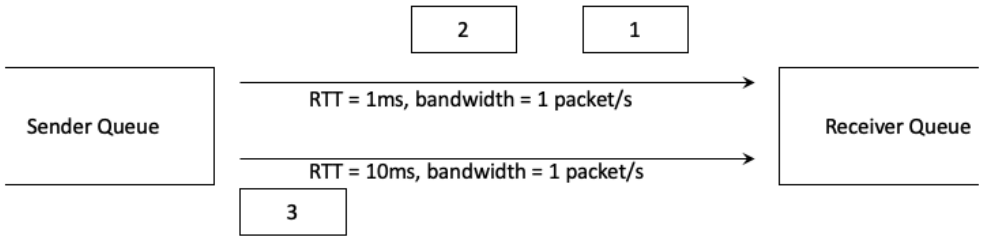


Figure 1.3: Default Scheduler in MPTCP

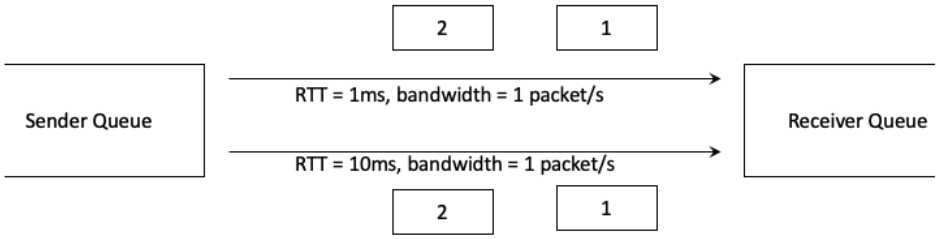


Figure 1.4: Redundant Scheduler in MPTCP

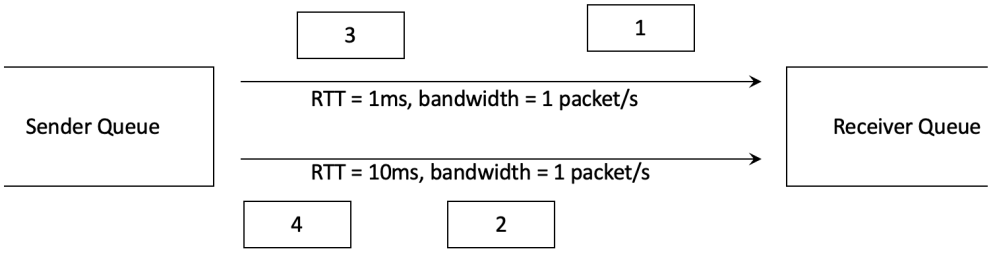


Figure 1.5: Round-Robin Scheduler in MPTCP

Redundant scheduler is another MPTCP scheduler. As shown in Figure 1.4, in this scheduler, traffics are sent on all available subflows in a redundant way[60]. There are two subflows available. The first one has an RTT of 1ms with a bandwidth of 1 packet/second. The second subflow has an RTT of 10ms with a bandwidth of 1 packet/second. With redundant scheduler, all packets will be duplicated and sent through both subflows regardless of the link quality. Redundant schedulers ensure reliable communication by delivering all segments redundantly[31]. It is anticipated that redundant schedulers can equalize the differences among subflows[27].In this redundant scheduler, the lowest possible latency is achieved by sacrificing bandwidth. However, this can be beneficial in some scenarios such ensuring the reliability of data delivery or dealing with unstable subflows.

1.1.2 MANET

Introduction

A Mobile Ad Hoc Network (MANET) [92] is a dynamic assembly of wireless mobile nodes and routers that autonomously establish and evolve a network, eliminating the necessity for pre-existing infrastructure or centralized management. This decentralized network architecture facilitates connectivity in environments devoid of traditional network infrastructure, making MANETs particularly advantageous in remote or emergency scenarios where rapid deployment is imperative [56]. As shown in Figure 1.6, MANETs allow for the seamless interconnection of devices, forming an Ad Hoc network that adapts to the dynamic movements and configurations of its constituent nodes.

Notably, MANETs serve as an ideal solution in situations such as catastrophes or emergencies where conventional infrastructures are absent or impractical due to geographical or temporal constraints [69]. The inherent ability of MANETs to spontaneously create a functional network without reliance on existing infrastructure makes them invaluable for first responders and emergency services, enabling fast and efficient communication in challenging environments.

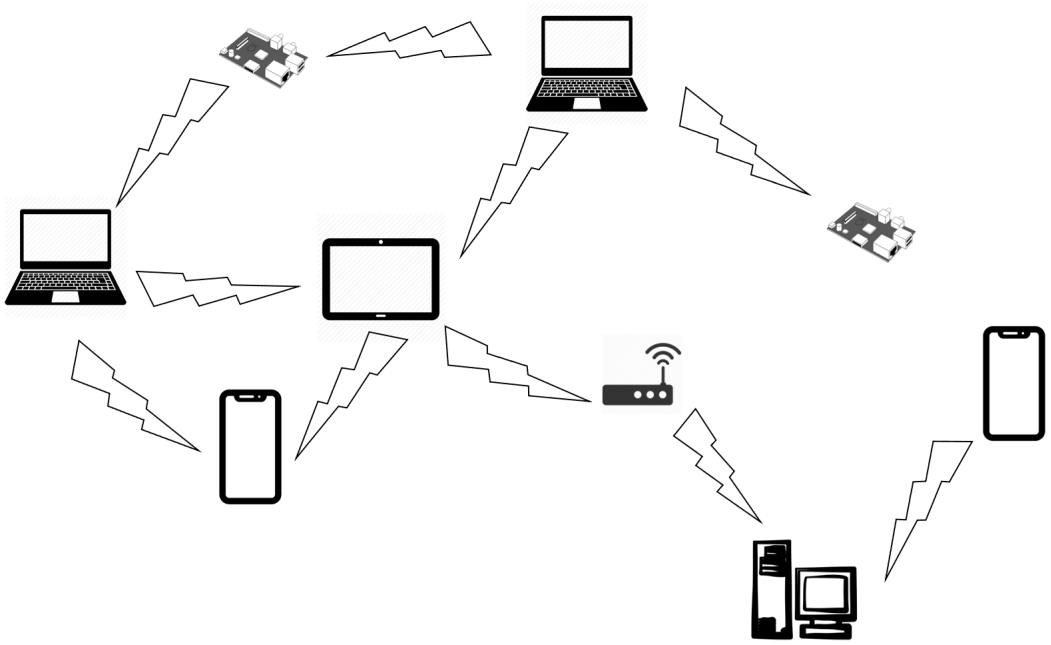


Figure 1.6: Conceptual representation of a MANET

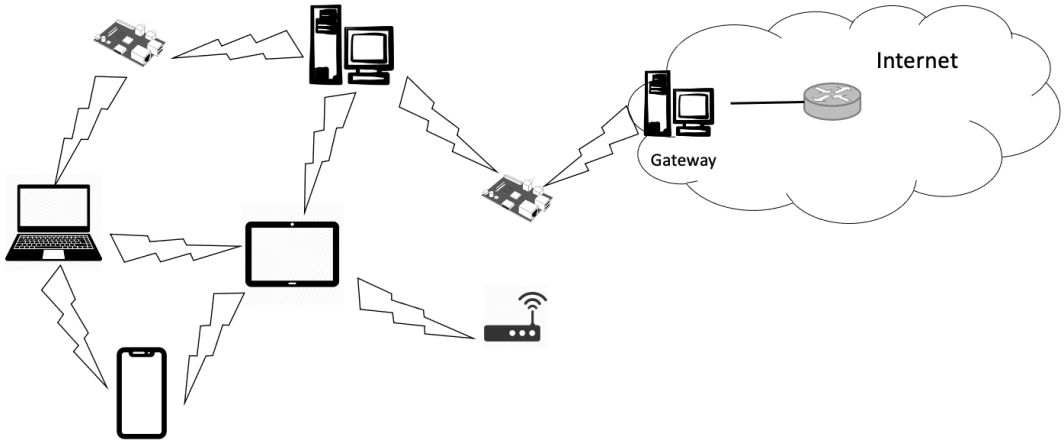


Figure 1.7: MANET extending to the Internet

Moreover, the versatility of MANETs extends beyond standalone scenarios, as illustrated in Figure 1.7. MANETs can seamlessly integrate with larger network environments, bridging the gap between isolated Ad Hoc networks and the broader Internet. This enhances the scalability and applicability of MANETs, making them adaptable to a variety of use cases and network architectures.

The significance of MANETs is further underscored by their role in shaping emerging applications in the realm of Smart Cities and the Internet of Things (IoT) [88, 44, 13]. As cities and IoT ecosystems embrace innovative connectivity solutions, MANETs provide a foundation for dynamic and responsive networks that can intelligently adapt to the evolving needs of these complex environments.

In summary, MANETs are increasingly becoming an integral part of the Internet due to their remarkable attributes. Their flexibility, self-configuration capabilities, independence from traditional infrastructure, ease of maintenance, self-administration capabilities, and cost-effectiveness collectively position MANETs as a transformative force in networking. As technological landscapes evolve, MANETs are poised to play an important role in enabling resilient, adaptive, and fast deployable wireless networks across a spectrum of applications and scenarios.

Routing Protocols in MANET

In MANET, the frequent and unpredictable joining and leaving of nodes introduces a considerable challenge when it comes to devising efficient routing strategies within MANETs[16]. The establishment of efficient communication pathways relies on the deployment of specialized routing protocols. These protocols can be broadly categorized into two main types: table-driven, commonly referred to as proactive protocols [3], and demand-driven, known as reactive protocols. Each type offers a distinct approach to handling the dynamic nature of MANETs.

Proactive protocols, exemplified by Destination Sequenced Distance Vector (DSDV) [73] and Optimized Link State Routing (OLSR) [45], operate on a constant quest for up-to-date routing information. In these protocols, each node diligently maintains comprehensive routing

tables, ensuring a continuous flow of information regarding network topology and node connectivity. However, this dedication to real-time information comes at a cost – the perpetual exchange of control packets throughout the network introduces a consistent overhead. While proactive protocols provide the advantage of readily available routing information, their constant information dissemination can lead to increased network traffic and potential resource utilization challenges[66].

On the contrary, reactive protocols, such as Ad Hoc On-demand Distance Vector (AODV) [72] and Dynamic Source Routing (DSR) [47], adopt a more selective and responsive strategy. In reactive protocols, routing information is only propagated when triggered by a specific request from the source to the destination. While this approach minimizes control packet overhead, it introduces latency when establishing new routes or adapting to changes in network topology. Reactive protocols excel in scenarios where resource conservation and reduced network traffic are critical considerations, but they may experience delays in route setup due to their on-demand nature[38].

The choice between proactive and reactive protocols in MANETs depends on the specific requirements of the network and the trade-offs that align with its operational goals[50]. The decision hinges on factors such as the frequency of topology changes, the need for real-time information, and the tolerance for latency in route establishment. Consequently, the exploration and refinement of both proactive and reactive routing protocols continue to be at the forefront of MANET research, seeking to strike an optimal balance between timely information availability and resource-efficient network operations.

OLSR in MANET

Optimized Link State Routing Protocol (OLSR) [17] is a proactive IP routing protocol optimized for MANET. In the OLSR network, nodes engage in periodic exchanges of `HELLO` messages and topology control (TC) messages to calculate local link and neighborhood information. This continuous information exchange forms the basis for maintaining an updated understanding of the network's topology[33]. Additionally, each node maintains a copy of routes to all destinations within the network. To minimize the overhead of flooding messages

and reduce redundant re-transmissions in the OLSR network, each node(n) is required to select a subset of its symmetric one-hop neighbors known as Multipoint Relay(MPR) nodes as shown in Figure 1.8. This selection is pivotal, as MPRs are strategically chosen to collectively cover all strict symmetric two-hop neighbors of the node(n)[68]. Notably, only MPRs can re-transmit packets received from node n, thereby minimizing the occurrence of redundant transmissions and reducing the likelihood of duplicate messages circulating within the network. This strategic selection of MPRs adds an efficiency layer to the OLSR protocol, optimizing the utilization of control message dissemination while ensuring a well-managed and responsive network.

There are two phases in the message propagation process in OLSR network. Initial Phase (Link Sensing): Each node in the network initiates the process by transmitting `HELLO` messages to its immediate neighbors. This broadcast occurs every 2 seconds and serves the purpose of assessing the status of links. Importantly, these HELLO messages only occurred between immediate neighbors and cannot be propagated to nodes further down the network hierarchy. A `HELLO` message encompasses various details, including message types, originator address, valid time, and the one-hop neighbors of the originator. Utilizing the neighbor list provided in received `HELLO` messages, nodes can effectively identify their two-hop neighbors. The information empowers each node to construct a neighbor table, often referred to as the MPR set. Within this set, each node makes selections to designate a subset of neighbors as multi-point relays (MPR(n)), enabling them to forward control packets originating from the node N[74]. The selection of MPR(n) involves satisfying two conditions: (1) possessing the transmission range to reach all two-hop neighbors, and (2) minimizing the number of MPRs to prevent unnecessary overhead in the network[64]. Additionally, the MPR set undergoes recalculation whenever a change in one-hop or two-hop neighbors is detected.

Second Phase (Topology Sensing): Topology sensing involves the dissemination of topology control (TC) messages every 5 seconds. These messages are broadcasted to construct an intra-forwarding table and are exclusively forwarded by the MPR nodes. The information embedded in TC messages allows each node to independently create a topology table, also enabling the selection of a set of MPR selectors. From this set, one node is chosen as an MPR, and details of the MPR selectors are added to the TC packet. Using the information derived

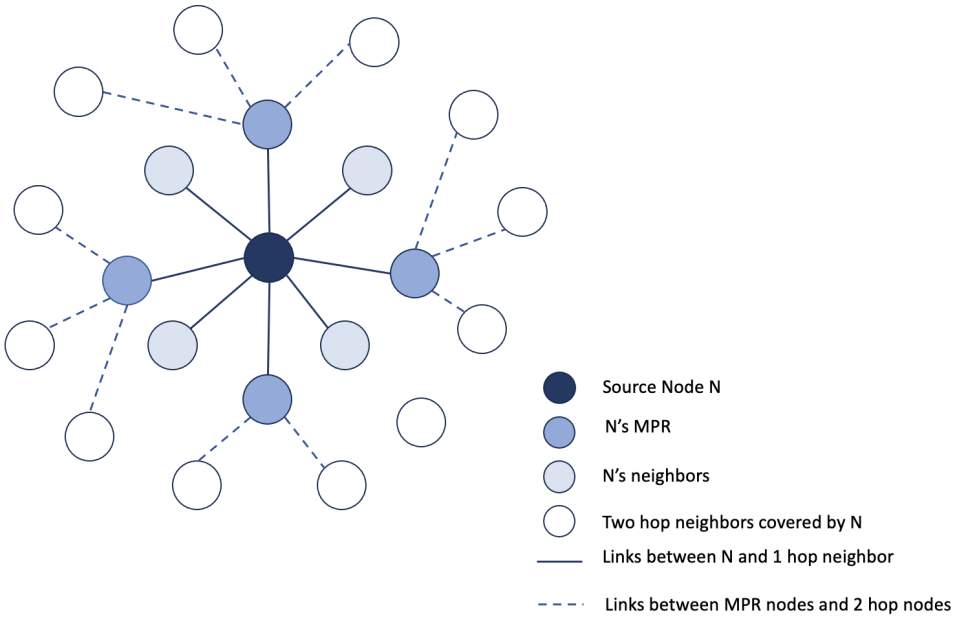


Figure 1.8: Broadcast packets forward by MPR

from TC packets, each node constructs a topology table that includes details such as possible destinations, the last-hop node to the destination, and MPR Selector Set sequence numbers. This mechanism facilitates the creation of paths to destination nodes by the originator of the TC message.

Multiple Interfaces in an OLSR Node

Multiple interfaces could exist in one OLSR node, only some of which participate in the OLSR network. These nonparticipating interfaces may connect to different networks. If a node in the OLSR network announces itself as a gateway to specific networks with Host and Network Association (HNA) messages, other nodes in the OLSR network will have the ability to access this network by setting up an internet route based on the HNA information. The gateway node generates an HNA message containing pairs of (network address, network) corresponding to the connected hosts and network. Upon receiving an HNA message, the node should update its current `A_network_address`. Using this feature, the nodes in the OLSR network have the ability to connect to a desired network through this gateway node.

1.1.3 MQTT

Message Queuing Telemetry Transport (MQTT) [85]is a lightweight publish-subscribe network protocol for the IoT. There are three components, subscriber, publisher and broker. The publish/subscribe communication model operates on the principle that components expressing interest in specific information register their interest, a process known as subscription, thereby becoming subscribers. On the other hand, components wishing to disseminate particular information become publishers. There could be multiple publishers and subscribers on one network. For each publisher/subscriber, it could publish/subscribe to multiple topics. The intermediary responsible for facilitating the transmission of data from publishers to subscribers is known as the broker. The broker plays a crucial role in coordinating subscriptions, with subscribers typically required to explicitly contact the broker to subscribe. In this model, the seamless flow of information between publishers and subscribers is coordinated by the broker, ensuring efficient communication within the system. Messages exchanged between the publisher and the

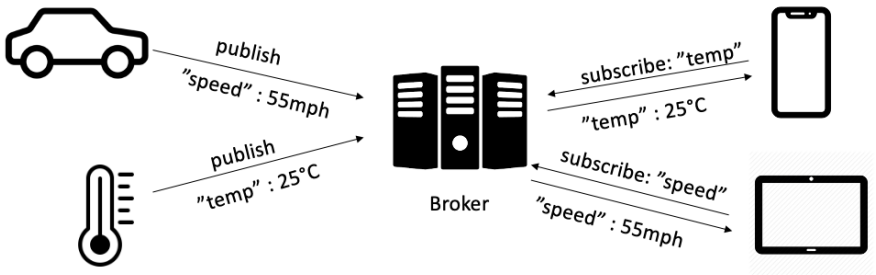


Figure 1.9: An example of MQTT publish/subscribe architecture

subscriber are often topic-based. Subscription and publications are restricted to a predefined set of topics. 1.9 is an example architecture with multiple publishers and multiple subscribers with topic publication and subscription.

The MQTT protocol, renowned for its lightweight nature and publish/subscribe architecture, is well-suited for deployment in Wireless Sensor Networks. This suitability arises from its compatibility with low-end, battery-operated sensor devices and its ability to function seamlessly over networks with limited bandwidth.[40]

The initiation of an MQTT connection always starts by an MQTT client, which could be either a publisher or a subscriber, sending a `CONNECT` message to the broker and the broker responds with a `CONNECT` message. The connection remains open until the client sends a `DISCONNECT` message. Once the connection is established, an MQTT client is able to publish/subscribe topics. Transport Layer Security(TLS) is used to encrypt the whole MQTT communication. Port 8883 is standardized for a secured MQTT connection whereas port 1883 is used for a non-secured MQTT connection.

1.1.4 TCP Fast Open

TCP Fast Open (TFO) [76] stands as an extension of the traditional TCP protocol, introducing a mechanism aimed at expediting the establishment of TCP connections between clients and

servers. The primary objective of TFO is to mitigate the latency associated with the standard TCP three-way handshake, the process integral to establishing a TCP connection.

In the conventional TCP connection setup, the client and server engage in the exchange of three packets (SYN, SYN-ACK, ACK) before actual data transmission can commence. This sequence introduces a degree of delay, particularly noticeable in scenarios involving short-lived connections or connections to servers characterized by high latency.

TCP Fast Open revolutionizes this process by enabling the server to embed data directly within the SYN-ACK packet. This innovative approach essentially merges the initial connection request with the transmission of data. Consequently, the client gains the ability to initiate data transmission immediately after the handshake is completed, bypassing the need to await a separate ACK (acknowledgment) packet from the server.

By circumventing the traditional sequential exchange of packets, TCP Fast Open significantly accelerates the initiation of data transfer[90], proving especially beneficial for scenarios where reducing latency is paramount. This extension optimizes the efficiency of TCP connections, particularly for short-lived connections or situations involving servers with inherently high latency, contributing to an enhanced and more responsive networking experience.

Initiation of TCP Fast Open Connection

Client sends a TCP SYN packet to the server, including a TFO cookie option indicating support for TFO. Server receives the SYN packet and checks if it supports TFO. If TFO is supported, the server generates a TFO cookie and includes it in the SYN-ACK packet. Server sends the SYN-ACK packet back to the client, containing the TFO cookie and the initial data payload requested by the client (if any). Client receives the SYN-ACK packet, verifies the TFO cookie, and sends an ACK packet back to the server. Server receives the ACK packet and completes the connection establishment. At this point, the client can start sending data immediately. By eliminating the need for a separate round-trip for data transmission, TCP Fast Open can significantly reduce latency for certain types of connections. This is particularly beneficial for protocols or applications that establish short-lived connections or rely on frequent connections to the same server.

Chapter 2

Related work

2.1 MPTCP Based Transmission Scheme

Regular Transmission Control Protocol (TCP) was initially designed for wired networks but has been modified for use in wireless networks, leading to challenges like increased packet loss and delays[91].This stems from the attributes of wireless channels, including a high error rate, interference, fading, obstructions, and more issues[86]. Although many algorithms have been proposed to improve TCP performance. When path failure occurs, data loss is inevitable, causing TCP to re-establish a new connection. Additionally, TCP lacks support for multi-homed terminals, such as smartphones, tablets, and laptops equipped with multiple heterogeneous interfaces for transmission. This limitation motivated the Internet Engineering Task Force (IETF) to introduce the Multi-Path Transmission Control Protocol (MPTCP), which facilitates concurrent traffic forwarding on various paths through multiple network interfaces like Wi-Fi, 5G/LTE, and Ethernet.

The MPTCP architecture is introduced in RFC 6182 [43] and has been published as an experimental standard in RFC 6824 [29] in 2013. In 2020, MPTCP has been pushed to the standard track in RFC 8684 [30]. Since its publication, there has been a wealth of research related to the design, implementation, and performance of MPTCP. [62] evaluated the performance of MPTCP with 4G and 5G network situations using a simulation test-bed. [14] In particular, [97] proposes a solution for streaming high quality mobile video with MPTCP in heterogeneous wireless network. [19] proposed a cross-layer scheduler for video streaming

over MPTCP. This scheduler utilizes information from both the application and transport layers to rearrange data transmission, prioritizing the critical segments of the video.[101] proposes an improved algorithms for multi-hop routing and establishing subpaths in MANETs. MPTCP has gained widespread popularity due to its ability to enhance performance, including seamless handover between network interfaces to support user mobility, resilience to link failures, and the aggregation of bandwidth from multiple paths[77], [54], [99].[7] proposed an multi-pathing community WiFi networks that are self-configuring with Wireless Mesh Networks(WMNs). In the proposed design, they used OLSR as an IP routing protocol for mobile Ad Hoc networks for WMNs. However, the solution still requires a private DHCP for WMNs and does not fully utilize the feature of Ad Hoc networks.[24] analyzes LTE and WiFi for up-link and down-link traffics and concludes that MPTCP can be beneficial with longer flows in the transmission.

There are works aiming to improve the packet transmission reliability of control packets in MPTCP. [84] provided numerical evaluation of different schedulers of MPTCP in comparing their throughput and reliability. Authors in [51] proposed a packet scheduling mechanism to reduce the out-of-ordered packet in the receiver buffer by adding a reordering considering mechanisms to each packet. [6] proposed an Opportunistic Routing technique to reduce MPTCP delay by reducing the number of transmissions. The Opportunistic Routing is a routing model employed to enhance the delivery rate and reliability of data transmission in wireless networks through the broadcasting method, allowing multiple relays to deliver data for each subflow. Authors in [58] proposed a novel framework for gathering scheduling information from diverse network entities and performing optimization from a global perspective. The framework leverages existing or easily accessible MPTCP parameters. Within this framework, we present a centralized optimization algorithm designed to achieve general proportional fairness in user throughput. [41] developed a packet scheduling protocol that confers benefits not only to long-lived flows but also to shorter ones. More precisely, the protocol initiates the classification of MPTCP paths into fast and slow paths. Subsequently, it temporarily halts the slow path when a substantial divergence in path performance exists between the fast and slow paths. This strategic freezing of the slow path facilitates the rapid transmission of small amount of data via the

expeditious fast path. [98] proposed a low latency MPTCP Scheduler for live video streaming in mobile networks.

2.2 Real-time Data Transmission in Ad Hoc Networks with Internet

Due to the expanding number of connected devices brought about by the widespread use of the Internet of Things(IoT), the amount of data being generated are growing exponentially[36], [15], [102], [89]. Scholars in [32], [61], [55] have proposed solutions for delivering massive real time data generated from these devices. However, these solutions are heavily rely on the current Internet structure and assume that the Internet is always reliable and capable for transmitting massive real time data. The capacity of wireless links is usually limited due to the interference and high transmission overhead [21]. Additionally, the ability to quickly adjust to the new topology of the network as well as maintaining the reliable transmission brings another challenge [11].

The characteristics of Ad Hoc Network provides the possibility to extend the Ad Hoc Network in the wireless sensor transmission. An Ad Hoc sensor network not only faces challenges in hardware design, but also in communication protocols and application design of the sensor network[94]. The dynamically changed network topology brings the challenge in designing such architecture. CodeBlue[63], designed for emergency medical care, presents a novel network architecture employing Ad Hoc networks for the deployment and transmission of multiple sensors. Nevertheless, the inherent limitations of Ad Hoc networks pose challenges to maintaining reliable transmission and adapting to rapidly changing network topology. Another design hurdle involves the transmission overhead within the Ad Hoc network. In CodeBlue, a substantial volume of information is exchanged among the Ad Hoc network, potentially leading to a reduction in the transmission rate within this network.

Integration of the Ad Hoc Network with other types network is another important research topics that many scholars have addressed before. [39] proposed a wireless networking architecture by connecting the MANETs to a Cellular network via a Terrestrial gateway. The proposed architecture reduces the network deployment expenses while delivering voice, messaging and

low-rate data services to mobile users. [10] also proposed an integration heterogeneous network called CAMA, a cellular aided mobile Ad Hoc network. CAMA utilizes a CAMA agent to manage the control information in the cellular network. They also proposed a position-based routing protocol for exploring the reachable neighbors. [96] proposed an iCAR system for integrating the cellular and Ad Hoc relaying system.

Improving Quality of Experience (QoE) for real-time data transmission in Mobile Ad Hoc Networks (MANETs) poses a significant and complex challenge.[35] proposed an optimal bandwidth allocation strategy in MANETs, minimizing the loss-induced distortion associated with a video source. [34] identified major factors that affect the QoE of voice communication in MANETs.

2.3 Routing in Ad Hoc Networks

Due to mobility of the MANET, it is inevitable to have nodes join and leave in the network frequently[83]. To have the gateways adapt to the change and maintain the connectivity of the MANET is of importance. Several techniques have been proposed for interconnecting MANET with the Internet. While most papers concentrate on integrating MANETS with the internet without being tied to any specific protocol, some also describe the fixed gateway discovery mechanism. [48] introduced the MIPMANET approach, designed to furnish MANET nodes with Internet access and Mobile IP mobility services.This involves the utilization of Mobile IP with a foreign agent care-of address and reverse tunneling. MIPMANET employs Ad Hoc on-demand distance vector (AODV) for the transmission of packets between a Mobile IP foreign agent and a visiting node seeking to establish an Internet connection. Notably, MIPMANET allows a visiting node to transition from its current foreign agent to a new one, a process referred to as handoff, only if the new agent is at least two hops closer.In[93], scholars proposed an implementation for integrating MANET and Internet architectures. Their approach considers diverse MANETs interacting with the fixed Internet, each characterized by its own Time-To-Live (TTL). The proposed architecture involves two network-layer programs, DSDVd and MIPd, which interact with the system kernel through socket interfaces. [12] proposed a solution for integrating MANET with Mobile IP, introducing the concept of a border router with

two interfaces. The interface connected to the Internet is configured to use normal IP routing mechanisms for incoming and outgoing packets in MANET. Meanwhile, the interface connected to MANET utilizes the dynamic source routing (DSR) protocol to route packets within the MANET.[80] proposed an efficient and load balancing gateway switch method using IP tunnelling. [100] provides a solution of Ad Hoc dynamic gateway based on mobile IP.[65] proposed a method to reach the network gateway by selecting paths based on trusted nodes and uncongested routes. In this instance, a hybrid gateway selection scheme is introduced that relies on trust, incorporating parameters such as node trust, route trust, and residual route load capacity in MANET.

Chapter 3

Problem Statement and Motivation

3.1 Problem Statement

In hazardous situations, the ability to maintain a reliable network transmission is crucial for first responders to make time-sensitive decisions and effectively respond to emergencies. Unfortunately, when an emergency occurs, it is not uncommon for power outages and breakdowns in cable services to cause network infrastructures to fail, rendering WiFi access points and cellular networks unavailable.

Modern wireless and mobile devices are often equipped with multiple interfaces, allowing them to connect to different networks simultaneously[81]. The network manager of an operating system can switch between these interfaces to establish a connection with the best available link when the current connection goes down. However, accomplishing this is not a easy task. As an end-to-end protocol, each TCP connection is uniquely defined by [source-IP-address, source port, destination-IP-address, destination port][82]. If the current interface is disabled, TCP does not have the ability to maintain the connection with a different interface. It is inevitable to break the current end-to-end connection in order to switch to a different interface.

To perform a seamless switch with TCP, we create a controller to oversee all available interfaces in TCP. The controller uses a round-robin scheduler with a priority queue for link switching as shown in Algorithm 1. Initially, all the available interfaces are stored in a queue with a priority value. The priority is calculated based on the round trip time of each link. The link with the lowest round trip time has the highest priority. If the current interface is disabled, switch to the interface with the next highest priority and update the current queue.

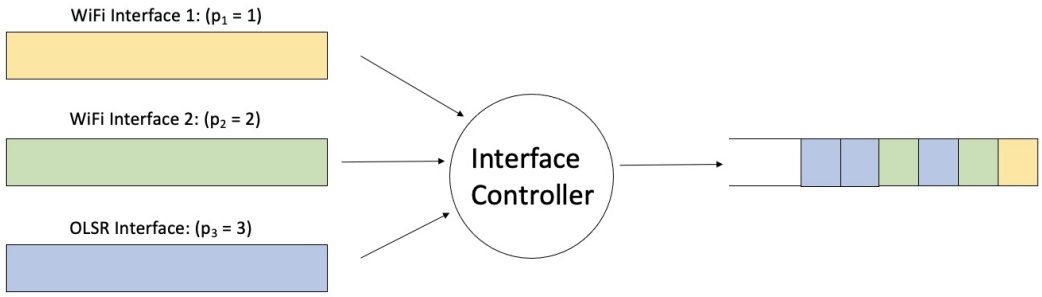


Figure 3.1: TCP interface switch controller

Figure 3.1 shows an example of the interface controller with three interfaces, two WiFi interfaces and one OLSR interface respectively. Initially, three interfaces are all available with WiFi 1 having the highest priority value. When WiFi 1 is not available, the interface controller switches to WiFi 2 and updates the priority value for both WiFi 1 and WiFi 2. If WiFi 2 is disabled, the interface controller switches the current interface to OLSR.

Algorithm 1 TCP Weighted Round Robin Switch

```
Q: list contains all the UP interfaces
n: length of Q
i: from 0 to n -1
while Q is not empty do
    if i is not UP then
        Q[i].dequeue()
        i++
        switch(i)
        updateQueue()                    ▷ re-calculate RTT for all interfaces
    end if
end while
```

Such design breaks the end-to-end connection and introduces a controller in the network stack which introduces more overhead in the network. Additionally, such frequent switching between links can lead to longer disconnections and disruptions in communication. Therefore, it is essential to prioritize maintaining a stable and uninterrupted connection during critical situations.

In emergency scenarios, data integrity is of importance. Any loss of data can have severe consequences, potentially resulting in life-threatening situations. To ensure data integrity, it is

recommended to use TCP over UDP. TCP provides reliable and ordered data delivery, making it more suitable for transmitting critical information. This is especially crucial when transmitting various types of data, such as real-time video, audio, and text messages, to multiple receivers simultaneously. Accomplishing this task requires powerful hardware resources and a network infrastructure with high-quality connectivity. Integrating different types of data transmission among a large number of devices poses significant challenges.

3.2 Motivation

The aforementioned problems make it difficult for first responders to react quickly and efficiently in emergency situations. Moreover, civilians in the affected areas also suffer from the loss of Internet connectivity, amplifies the difficulties encountered during emergencies. It is imperative to develop a resilient and rapidly deployable communication hub that utilizes all available wireless and mobile devices in the area. Such a hub would enhance the reliability of real-time data transmission, enabling first responders to receive critical information without disruption and allowing civilians to maintain essential connectivity in emergency situations.

By leveraging the collective resources of wireless and mobile devices, this communication hub would establish a decentralized network that can adapt to changing conditions and provide seamless connectivity. The hub would intelligently allocate network resources, prioritize critical data, and optimize the utilization of available bandwidth. Through this approach, the communication hub would improve the reliability and efficiency of real-time data transmission, enabling first responders to make informed decisions promptly and ensuring that civilians have access to vital information and assistance.

In summary, the creation of a resilient and fast-deploying communication hub that harnesses the capabilities of wireless and mobile devices is crucial in hazardous situations. By addressing challenges such as maintaining a stable connection, preserving data integrity, and integrating various types of data transmission, this communication hub would significantly enhance the reliability of real-time data transmission. This, in turn, empowers first responders to effectively respond to emergencies and enables affected civilians to stay connected and informed during critical situations.

Chapter 4

Conceptual Overview

4.1 Overview

The proposed Next Generation First Responder (NGFR) Communication Hub aims to enhance the reliability of the network by enabling rapid deployment, self-organization, and self-connection capabilities. This communication hub possess the ability to automatically select the most optimal and secure communication links without interrupting the current connection. Furthermore, the NGFR Communication Hub is equipped with the capability to efficiently pull and push real-time sensor data from sensors that can be dynamically attached to first responders. This adaptability in communication and sensing functionalities is made possible by the innovative self-organizing and secure communication hub network architecture, illustrated in Figure 4.1. This architecture not only ensures robust connectivity but also enables the dynamic integration of diverse sensor inputs, enhancing the overall responsiveness and effectiveness of first responders in critical situations.

4.2 NGFR Communication Hub in Network Stack

The NGFR Communication Hub is intricately designed following the principles of the current TCP/IP stacks, as depicted in Figure 4.3. In this structure, sensor data is generated and gathered at the application layer. The Message Queuing Telemetry Transport (MQTT) protocol protocol serves as the interface between mobile devices and Internet-connected devices. Multipath TCP (MPTCP), operating at the transport layer, enhances the stability and reliability of communication hubs. Meanwhile, the Optimized Link State Routing (OLSR) protocol acts

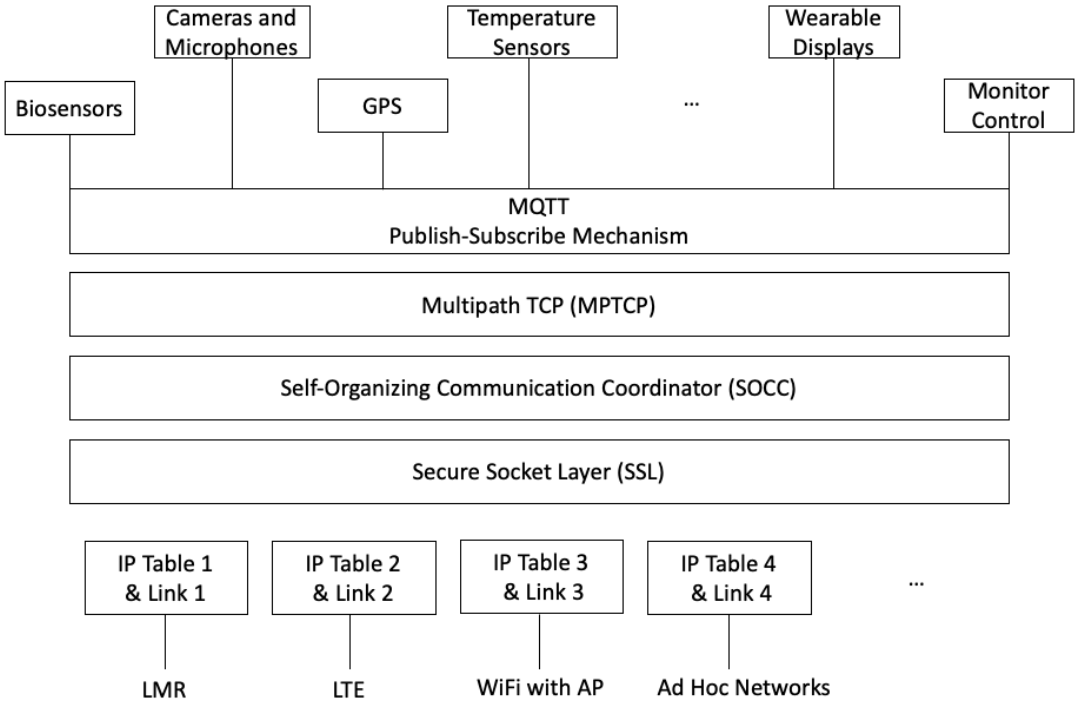


Figure 4.1: Network Architecture in NGFR Communication Hubs

as the routing protocol for mobile devices within the network. The proposed design for the NGFR Communication Hubs is illustrated in Figure 4.2, providing the potential deployment and interaction within a Mobile Ad Hoc Network (MANET).

Within the MANET, a dynamic network is formed by wireless and mobile devices such as laptops, Raspberry Pis, and mobile phones. The OLSR protocol takes charge of establishing connections between these devices and identifying gateways for efficient communication. Notably, some devices exhibit multiple interfaces with potentially unstable wireless connections, while others lack direct Internet access. Nevertheless, these devices can still connect to the Internet through the gateways selected by the OLSR protocol.

All devices operating within the MANET demonstrate the capability to function as both publishers and subscribers, facilitating a bidirectional flow of information. The central broker, hosted on a cloud server, plays a pivotal role in maintaining a stable Internet connection. Multiple publishers and subscribers connected to the Internet can seamlessly interact with the broker, establishing a robust communication ecosystem within the NGFR network. This comprehensive design ensures adaptability, reliability, and effective communication across a diverse array of devices and network conditions.

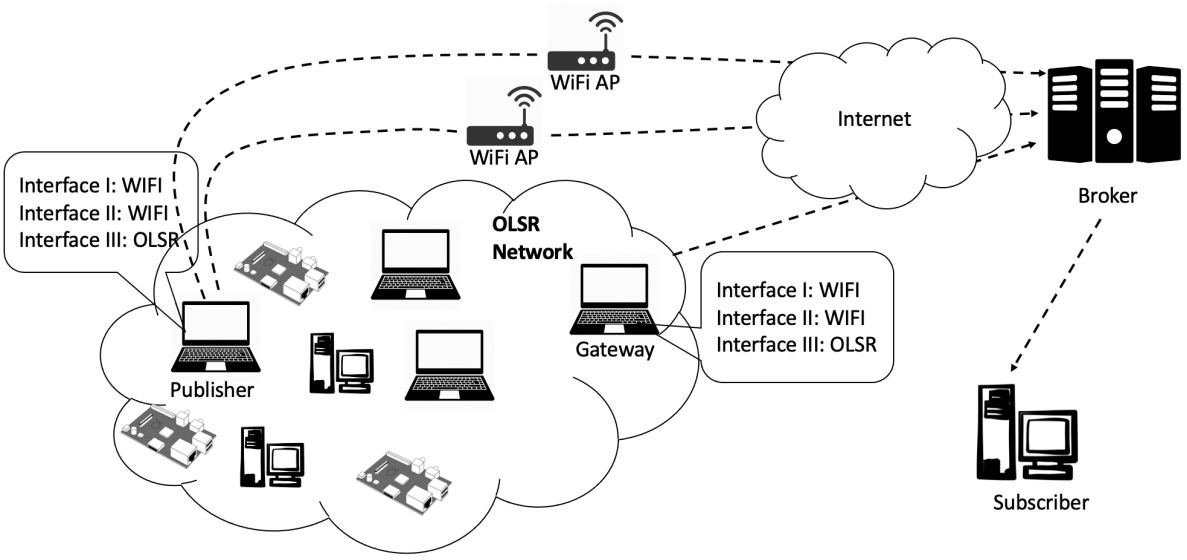


Figure 4.2: Prototype in NGFR Communication Hubs

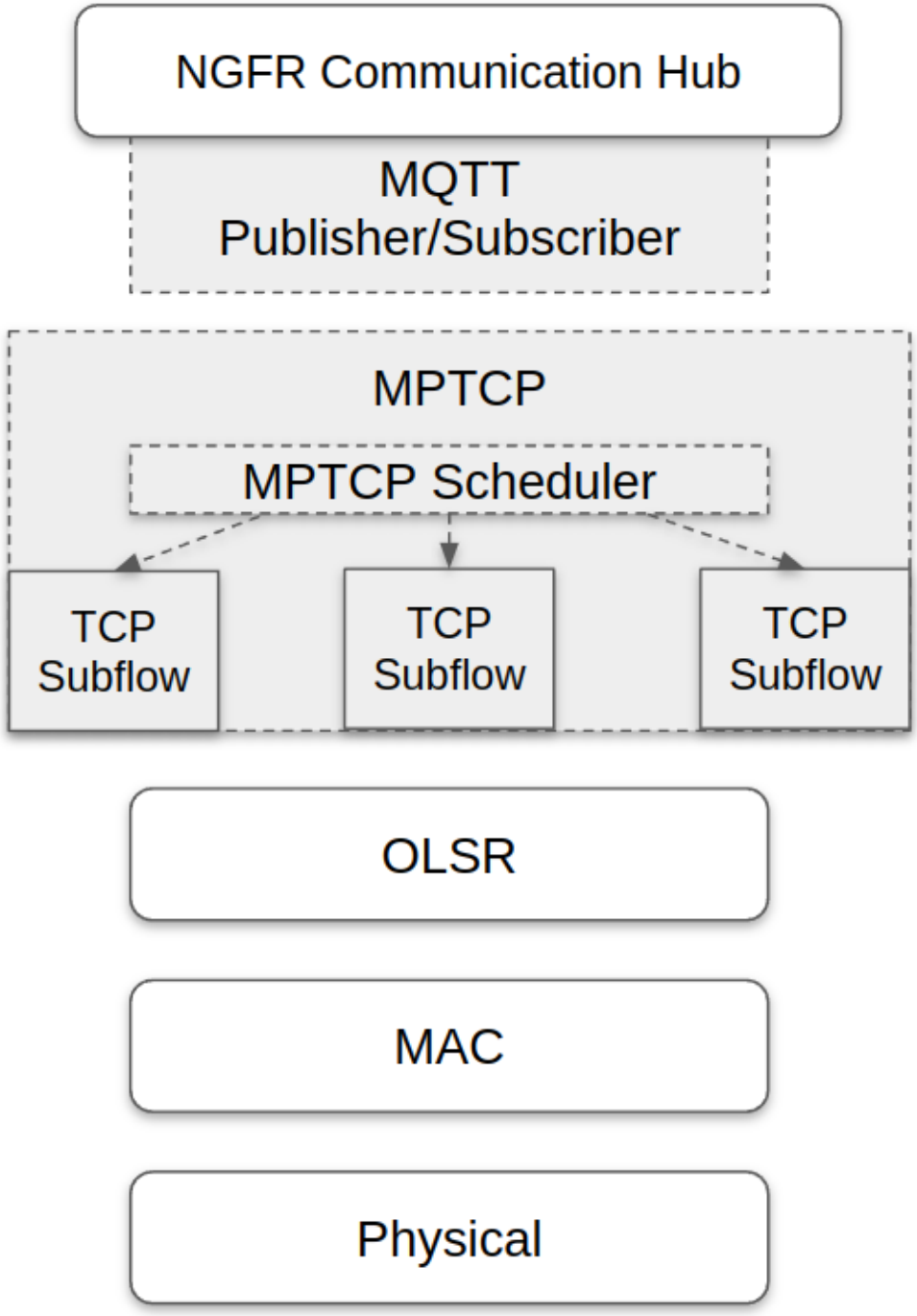


Figure 4.3: NGFR Communication Hubs in TCP/IP Stack

Chapter 5

Resilient and Reliable Communication Hub

5.1 Motivation

With the aforementioned conceptual design, we developed and implemented the NGFR Communication Hubs. The Communication Hub enhances network reliability through switching to the most efficient and secure links without disrupting the existing connection. It employs rapid deployment, self-organization, and self-connection functionalities, ensuring a resilient connection. Furthermore, it is equipped to seamlessly retrieve and transmit real-time sensor data from a variety of sensors that can be dynamically connected to first responders. The self-organizing and secure communication hub network architecture facilitates these adaptive communication and sensing capabilities.

To obtain the desired authenticity of the results, we designed extensive experiments in various network configurations to verify the design and implementation of our NGFR Communication Hubs. The design of the experiment should aim to accomplish the following objectives:

Various types of data transmission - The system should aim to support the transmission of various types of data with minimal bandwidth requirements. This capability is crucial in emergency situations where real-time communication and information exchange play a vital role.

Smooth Switch with MPTCP - For devices with multiple interfaces, disabling any interface in order would force it to switch to a different one. With MPTCP enabled, the switch should not cause any disconnect in data transmission. Additionally, the switch should not require user interaction.

Impact on number of hops in MANET - A gateway is considered as the last hop of an OLSR network. Thus, the number of hops is defined by the hops between the node itself and the gateway. In above-mentioned different configurations in MANET, the less number of hops, the better the performance should be.

Impact on number of publishers in MANET - More publishers in the MANET requires more bandwidth and also could increase overheads in the network. Thus, the more publishers/subscribers in the network, the worse the performance should be.

The rest of the chapters are arranged as follows: we introduce the set up of the experiment, various configurations for different scenarios, experimental data selection, stress testing and performance metrics for evaluation.

5.2 Experiment Set Up

The experimental procedures take place within the confines of an indoor setting, where a carefully configured private network setup has been established. This network is consist of three distinct subnets, each controlled by a corresponding router, as illustrated in Figure 5.1. Within this framework, Wireless 1 and Wireless 2 are independent subnets, each assigned unique IP addresses to ensure segregation and facilitate efficient data management. The interconnection of these subnets is achieved through the collaboration of three routers—Router 1, Router 2, and Router 3—which are seamlessly linked via a network switch. This structured network architecture provides a conducive environment for conducting systematic experiments, allowing for the exploration and evaluation of various aspects related to connectivity, data transmission, and overall network performance.

5.3 Experiment Configurations

Figures 5.3, 5.4 and 5.2 are the three selected configurations for the OLSR network. We use an open-source implementation of OLSR `OLSRd` for the routing algorithm in MANET. The publisher (P) is the node that designed to publish sensor data. It has three IEEE 802.11 interfaces,

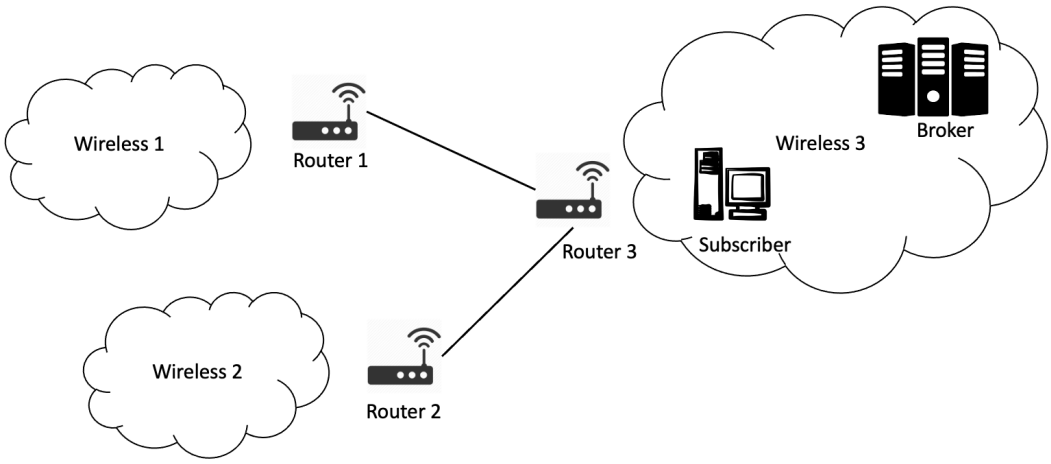


Figure 5.1: Network Partition in a Private Network Setup

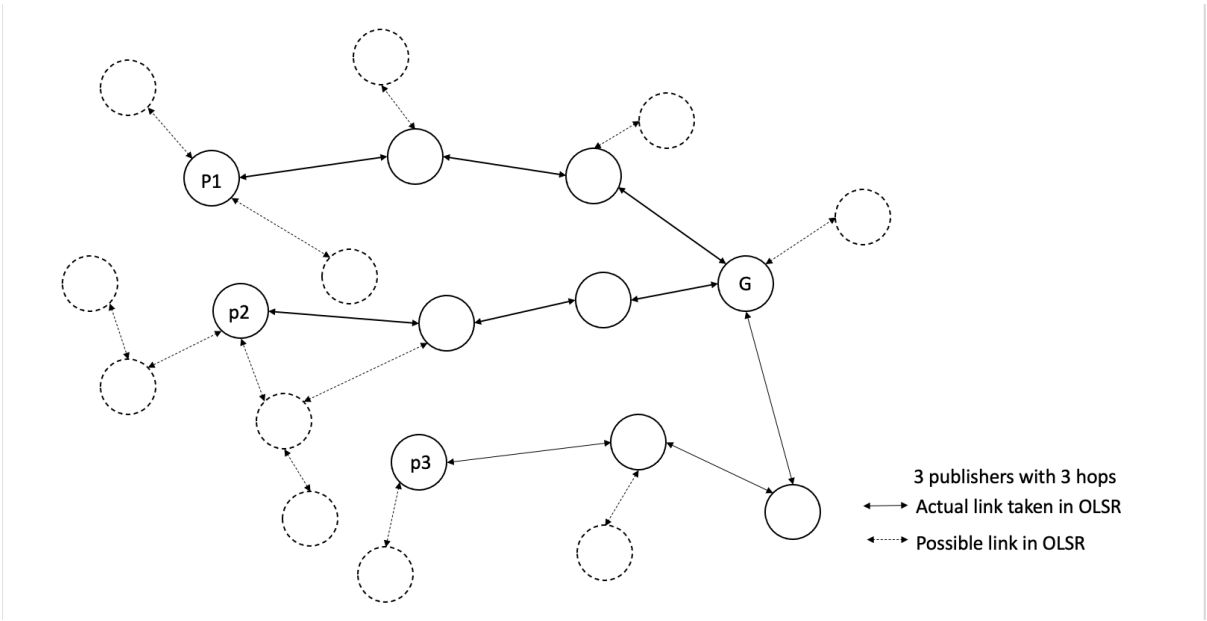


Figure 5.2: 3 publishers with a maximum of 3 hops

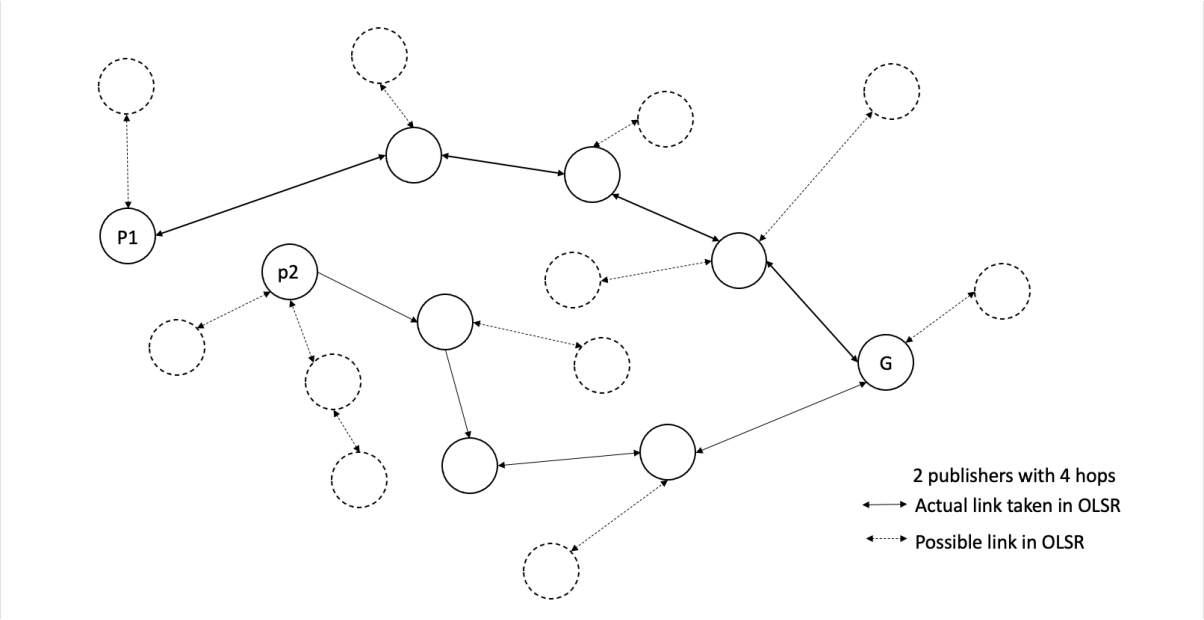


Figure 5.3: 2 publishers with a maximum of 4 hops

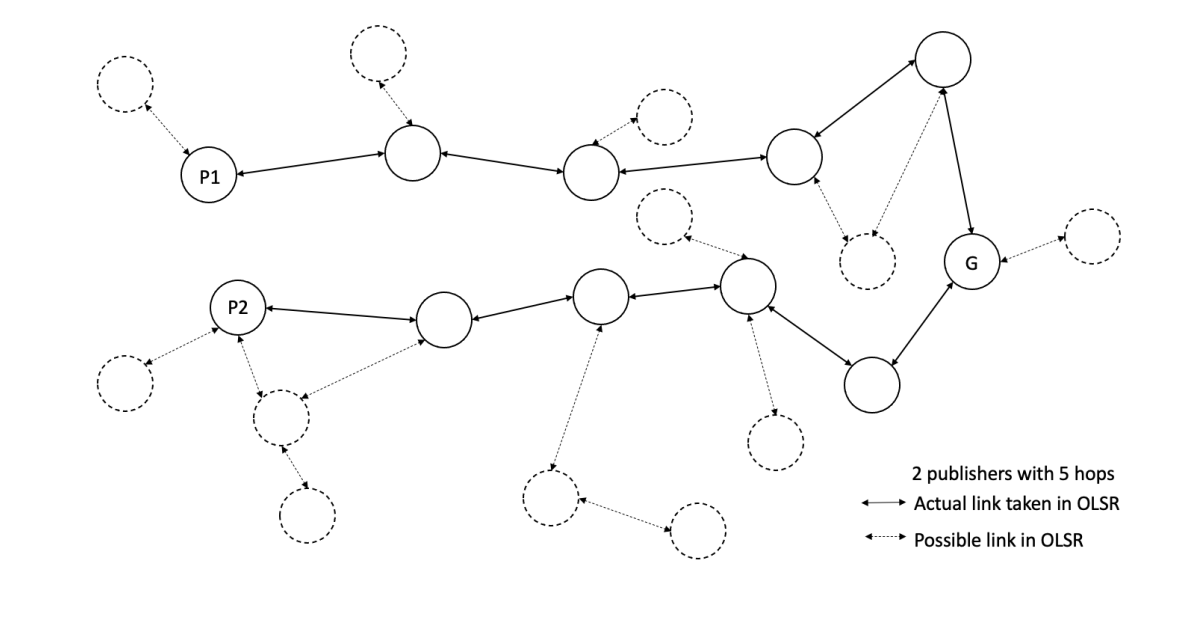


Figure 5.4: 2 publishers with a maximum of 5 hops

all USB connected. The gateway (G) has two interfaces, IEEE 802.11 and IEEE 802.3, respectively. Both the publisher and the gateway use an x86_64 Linux kernel 4.19.126 with `mptcp` extension v0.95.1. Paho MQTT is used in the application layer to publish and subscribe to sensor data. The solid line in the figures indicates the path taken in the OLSR network. The dotted nodes in the figures represent the existing nodes in the OLSR network yet not be selected.

5.4 Experiment Design

There are 25 executions in each configuration, separated into 5 groups. The average values are calculated every 0.1 seconds. At the beginning of each experiment, all interfaces are up, and each is assigned with an IP address to a different network. MPTCP will choose one or more paths to transmit data.

There are 3 phases in each execution as shown in Table 5.1. The first 60 seconds is the first phrase. Devices send data for 30 seconds from the beginning without any disruption. At 30^{th} second, one WiFi AP mode interface is brought down. The devices continue to send data for another 30 seconds. The second phrase comes at 60^{th} seconds, when the second WiFi AP mode interface is removed, leaving only the Ad Hoc network. At 90^{th} seconds, we come to the third phrase, when one of the WiFi AP mode interfaces is brought up. After 30 seconds of communication, the second WiFi AP mode interface is brought up. At this point, all three interfaces are available. The execution continues for the last 30 seconds and ends at 150^{th} seconds.

	Status		
Time	Interface I(WiFi)	Interface II (WiFi)	Interface III(OLSR)
0-30	up	up	up
30-60	down	up	up
60-90	down	down	up
90-120	up	down	up
120-150	up	up	up

Table 5.1: Experiment design

5.5 Experiment Data Selection

To achieve the goal of sending various types of sensor data and to better simulate the real-world situation, we selected three types of data for testing: live audio, live video, and text messages. The audio was captured via an USB sound card and processed by an open source python library `PyAudio` with a sampling rate of 44100, and chunk size of 1024 bytes. The video frames were captured via an USB camera and encoded with H.264. The text message was generated with 1024 bytes per second.

5.6 Stress Testing

Stress testing serves as a critical method for evaluating the robustness and reliability of a system. In the context of NGFR Communication Hubs, stress tests are systematically conducted for each configuration to evaluate their performance under challenging conditions. To execute these stress tests, the widely adopted open-source tool, `iperf3`, is employed. `iperf3` is renowned for its capability to actively measure the maximum achievable bandwidth on IP networks, making it a valuable asset in assessing the NGFR Communication Hubs' capabilities.

The stress tests align with the experiment design detailed in Table 5.1, ensuring a standardized and consistent approach to the evaluation process. By subjecting the NGFR Communication Hubs to stress tests under various configurations, we not only verify their ability to withstand adverse conditions but also gain a comprehensive understanding of their performance limits and capabilities across different scenarios. This methodical approach contributes to a thorough assessment of the NGFR Communication Hubs' functionality and reliability in diverse operational contexts.

5.7 Performance Metrics

Round Trip Time (RTT)is the duration from a packet being sent to when it receives a acknowledgement (ACK) from the broker. It is used to measure the latency of the network. The RTT can be calculated by:

$$Latency = avgRTT\_Server + avgRTT\_Client$$

`avgRTT_Server` is the average time between the broker receives the packet and the receiver sends the ACK of that packet back to the client. `avgRTT_Client` is the average time between the client sends the packet and the client receives the ACK from the broker.

Packet Delivery Ratio (PDR) defined as the ratio of successfully delivered packets compared to the total transmitted packets in the network.

$$PDR = \frac{\sum Number\ of\ packet\ receive}{\sum Number\ of\ packet\ send}$$

Throughput is defined as the rate at which the data is delivered successfully in a given period of time. It is calculated to show the reliability of the network during the experiment time. It is calculated as the total size of the payload in data packets divided by time, which is a period of 10 seconds.

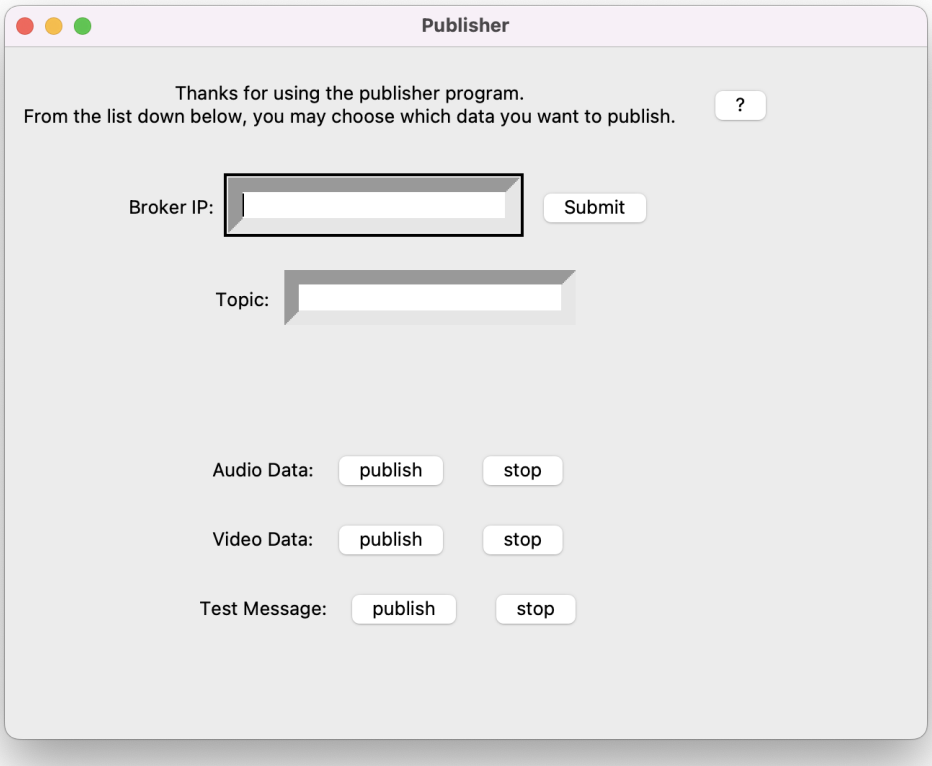
$$Throughput = \frac{\sum Total\ size\ of\ data}{time}$$

The switch time for each link is defined by the average period of time between the initiation of one link break and the switch to another link. It indicates how fast the data transmission can be switched between different links.

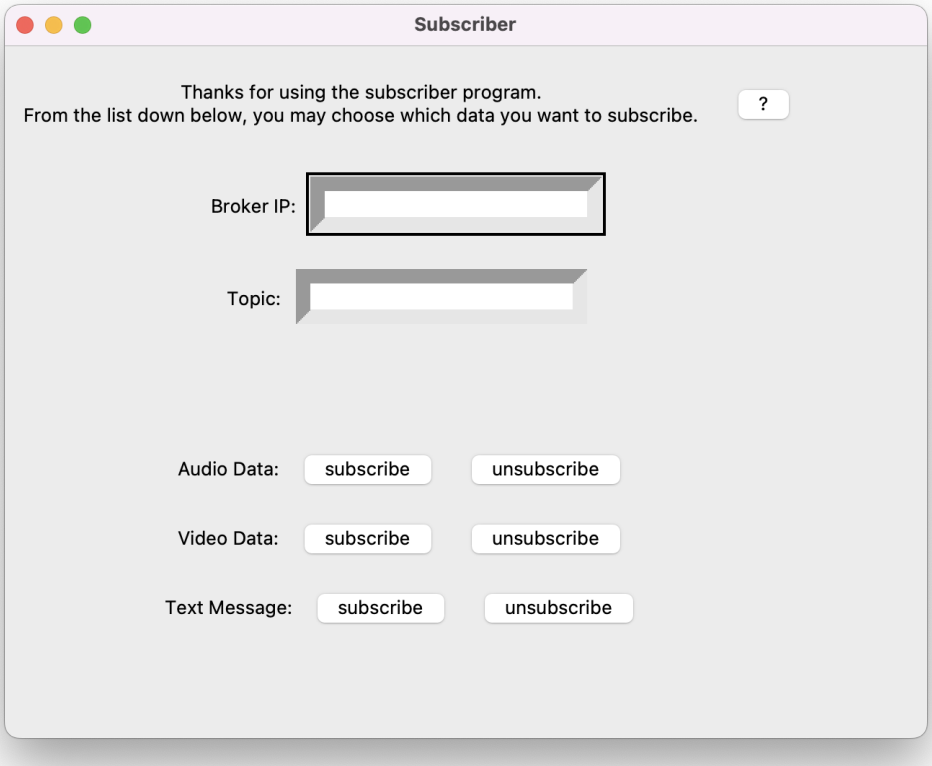
$$Time_{switch} = \frac{\sum Switch\ time}{number\ of\ runs}$$

5.8 Implementation and Result Analysis

NGFR Communication Hub Publisher is implemented in *Python* with the ability to collect real-time audio, video and text data, and publish to the broker. NGFR Communication Hub Subscriber has the similar design of the software interface, and is able to subscribe to real-time audio, video and text data with the topic. Figures 5.5 and 5.6 show the GUI of the implemented program and the running screenshot of the NGFR Communication Hub, respectively.



(a) Publisher



(b) Subscriber

Figure 5.5: Implementation of NGFR Communication Hubs

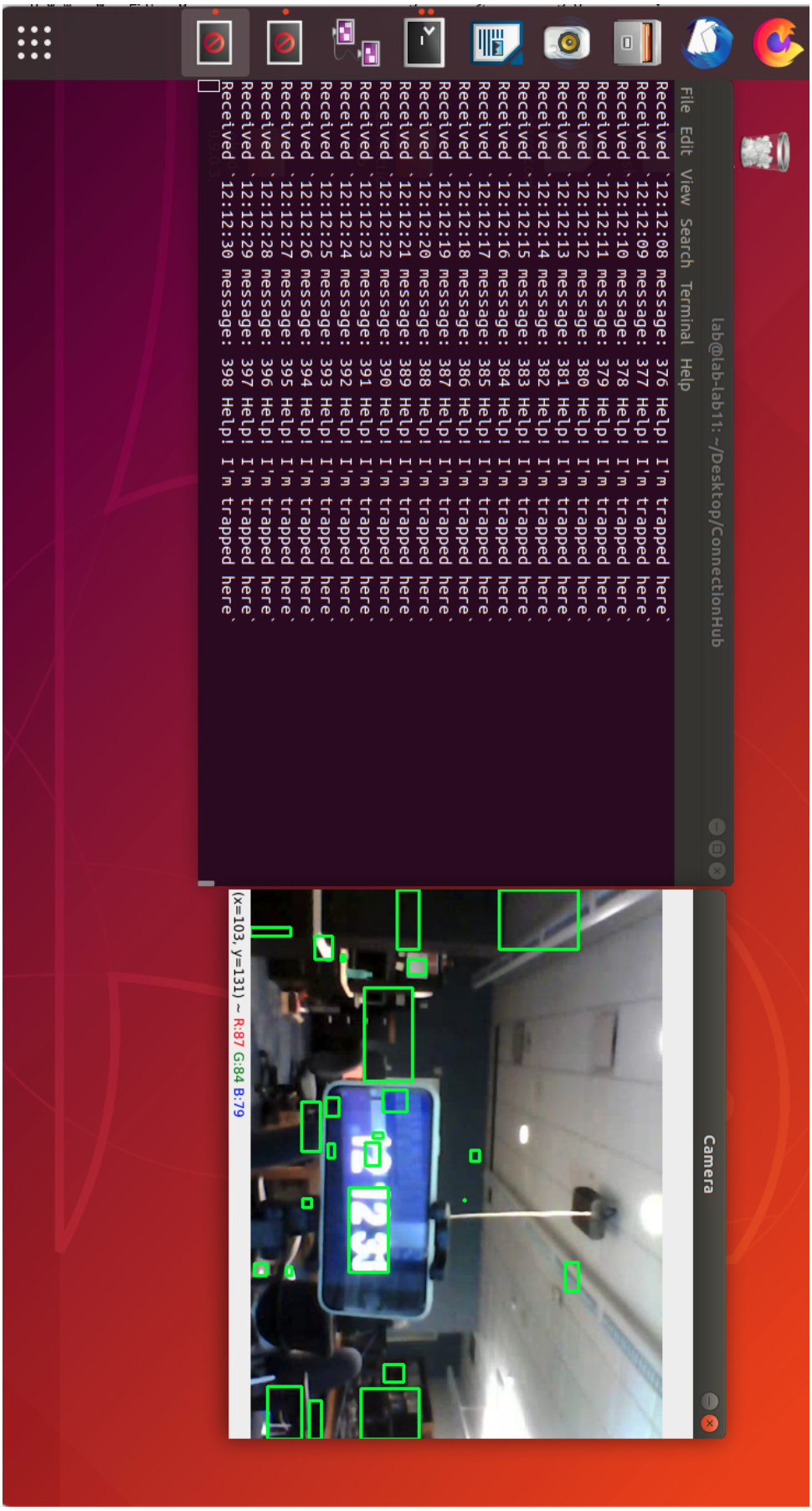


Figure 5.6: Screenshot of the running interface

5.8.1 One Publisher with MPTCP

We test with only one publisher in the OLSR network to see different performance when increasing the number of hops. The publisher is equipped with three USB 802.11 interfaces and is implemented with the MPTCP default scheduler. We use the default MPTCP congestion control algorithm, LIA[78] for all the experiments.

Result from the Broker

At 0^{th} second, three interfaces are available. There is no latency in transmission. The throughput is stable with a high packet delivery ratio.

At 30^{th} second, the first WiFi AP mode interface is brought down. The latency and packet deliver ratio of both setups do not change a lot as shown in Figure 5.7 and in Figure 5.8. Packet delivery ratios are close to 100% before 60^{th} second. Figure 5.9 also shows that the throughput is similar.

At 60^{th} second, the second WiFi AP mode interface is brought down, leaving only the Ad Hoc network interface. The publisher will have to use the Ad Hoc mode with OLSR routing algorithm to route all data packets to the gateway. Unlike WiFi AP mode where packets are directly delivered to the gateway, nodes selected by OLSR algorithm in the Ad Hoc network need to relay the packets. Latency begins to dramatically increase, and packet delivery ratio drops a lot with an increasing number of hops. As a result, throughput drops as well. However, the connection has still remained. Some glitches are observed during the experiment for 8 and a higher number of hops in the MANET setup, especially for video transmission. The performance of text transmission has no problem when delivering data.

At 90^{th} second, a WiFi AP mode interface is brought up. It takes a few seconds for MPTCP to adjust its flow to a less congested path. After 100^{th} second, the performance begins to improve. There is a large increase in throughput in 8 and a higher number of hops, indicating that with only the OLSR interface, a larger number of hops may introduce more congestion in the MANET. When a less congested link is available again, MPTCP balances the load. Thus,

latency becomes lower and fewer retransmissions occur. The quality of the video is improved by observation during the experiment.

At 120^{th} second, the second WiFi AP mode interface is brought up. The latency and packet delivery ratio are stable as before.

Result from the Publisher

Regarding the publisher's perspective, we categorized throughput based on interfaces. An illustration of this is depicted in Figure 5.10, showcasing the results with a 3-hop distance to the gateway. Throughout the 0^{th} to 30^{th} second period, all interfaces effectively transmit data, with WiFi1 handling the majority of the traffic. At the 30^{th} second mark, WiFi1 is deactivated, leading to an immediate surge in throughput through WiFi2. By the 60^{th} second, WiFi2 is also disabled, leaving only the Ad Hoc network interface operational. Subsequently, from the 90^{th} second to the 120^{th} second, WiFi1 and WiFi2 are reactivated, respectively. Notably, there are no transmission disconnections during the experiment, and the transitions between different interfaces are seamless, maintaining an uninterrupted end-to-end connection. The accumulated throughput is calculated as the sum of all possible interfaces. Figure 5.11 provides another instance of the results in a 6-hop scenario.

5.8.2 Multiple publishers in MPTCP

Subsequently, we conducted tests using multiple publishers to assess the performance of the NGFR Communication Hubs. Each publisher is equipped with three network interfaces, comprising two WiFi interfaces and one OLSR interface, respectively. All publishers adhere to the identical experimental schedule outlined in 5.1.

From 0^{th} second to 30^{th} second, three interfaces are actively available. The throughput, as shown in Figure 5.13, indicates an increase along with an increasing number of publishers. At 30^{th} second, there is a slight drop in throughput due to interface switching. However, the throughput recovers immediately. The latency and the packet delivery ratio exhibit minimal changes in both configurations as evident in Figure 5.14 and in Figure 5.12. Packet delivery ratios are close to 100% before 60^{th} second.

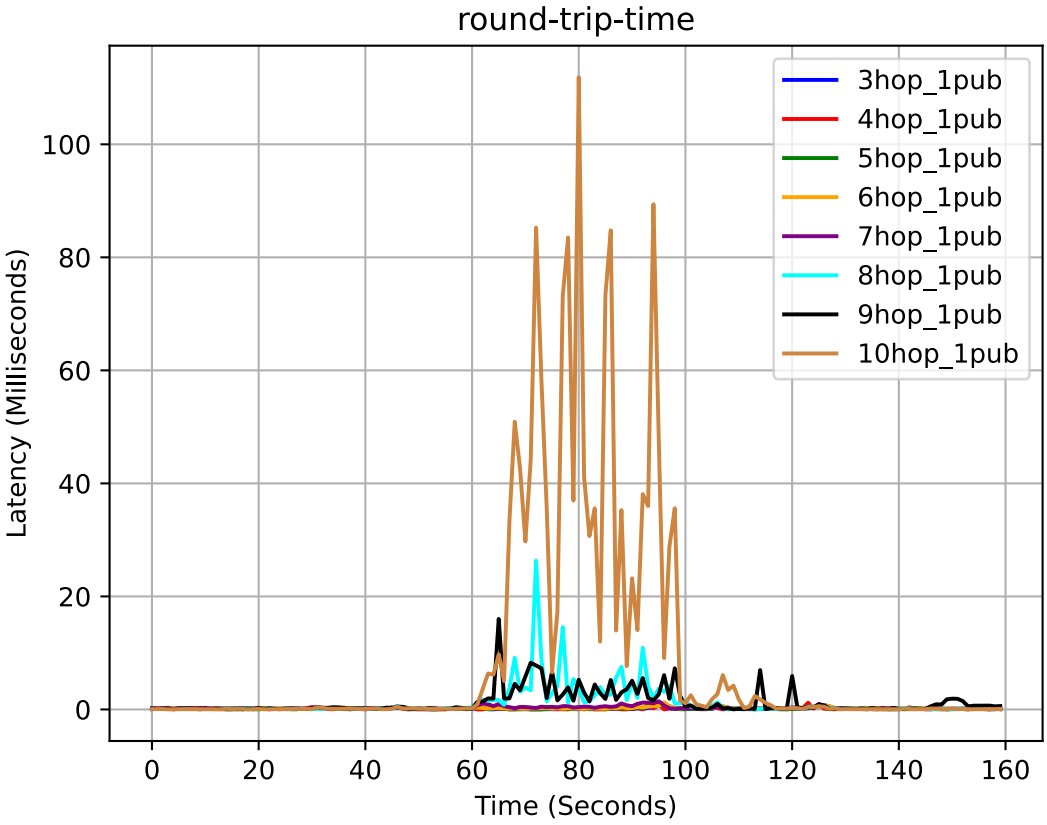


Figure 5.7: Comparison of round trip time in different number of hops with one publisher

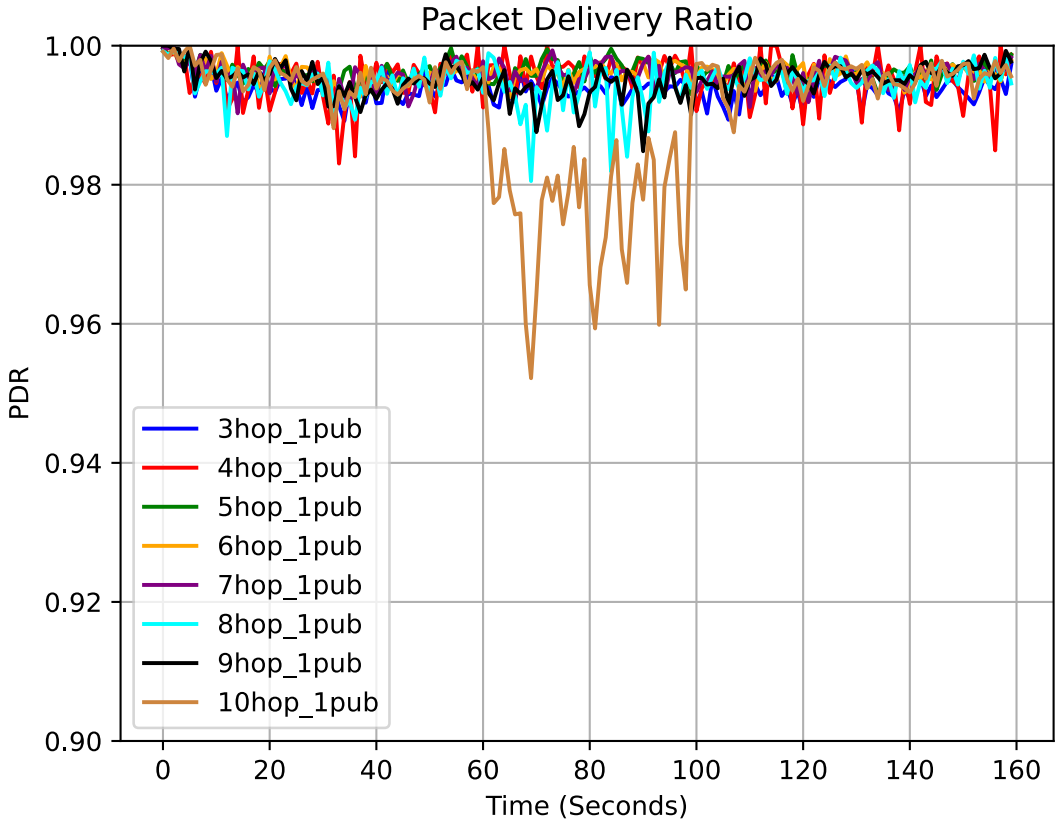


Figure 5.8: Comparison of packet delivery ratios in different number of hops with one publisher

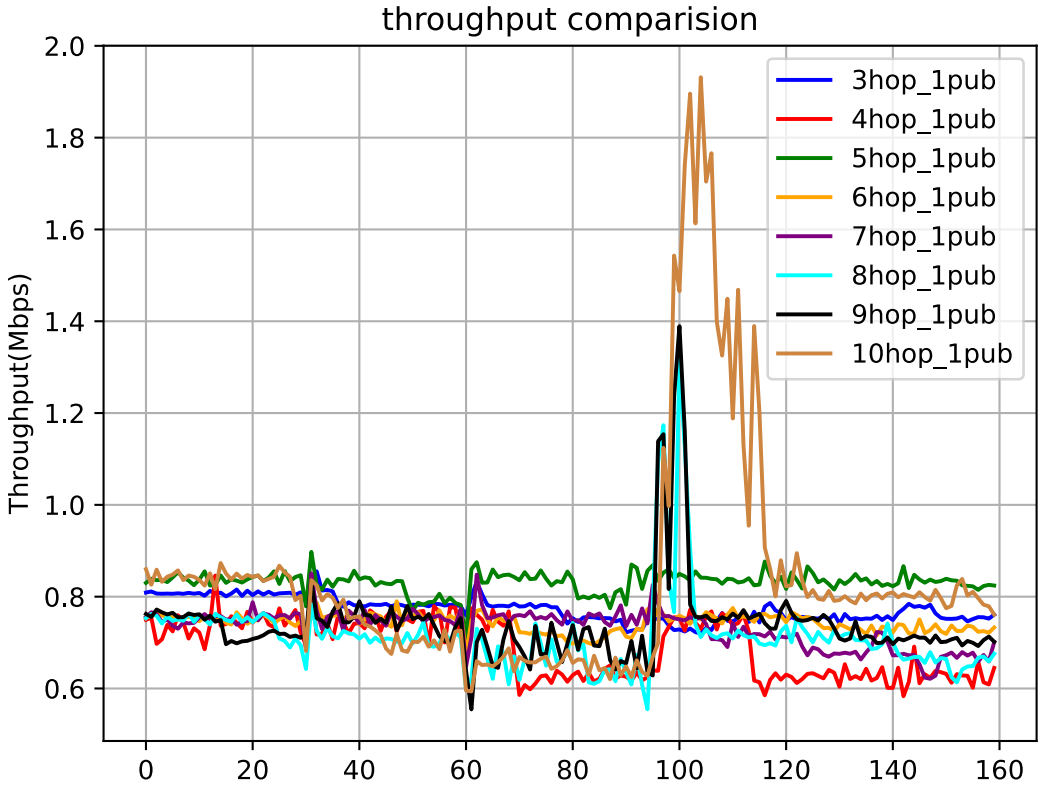


Figure 5.9: Comparison of throughput in different numbers of hops with one publisher

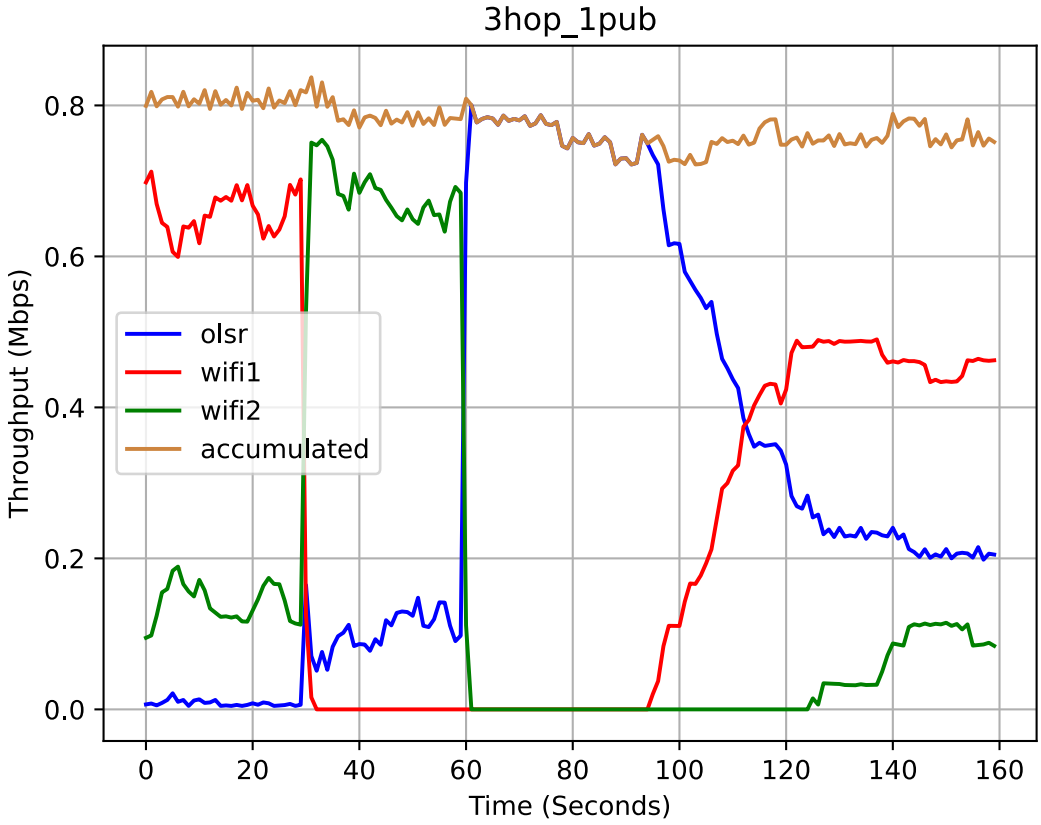


Figure 5.10: Throughput split in publisher with 3 hops

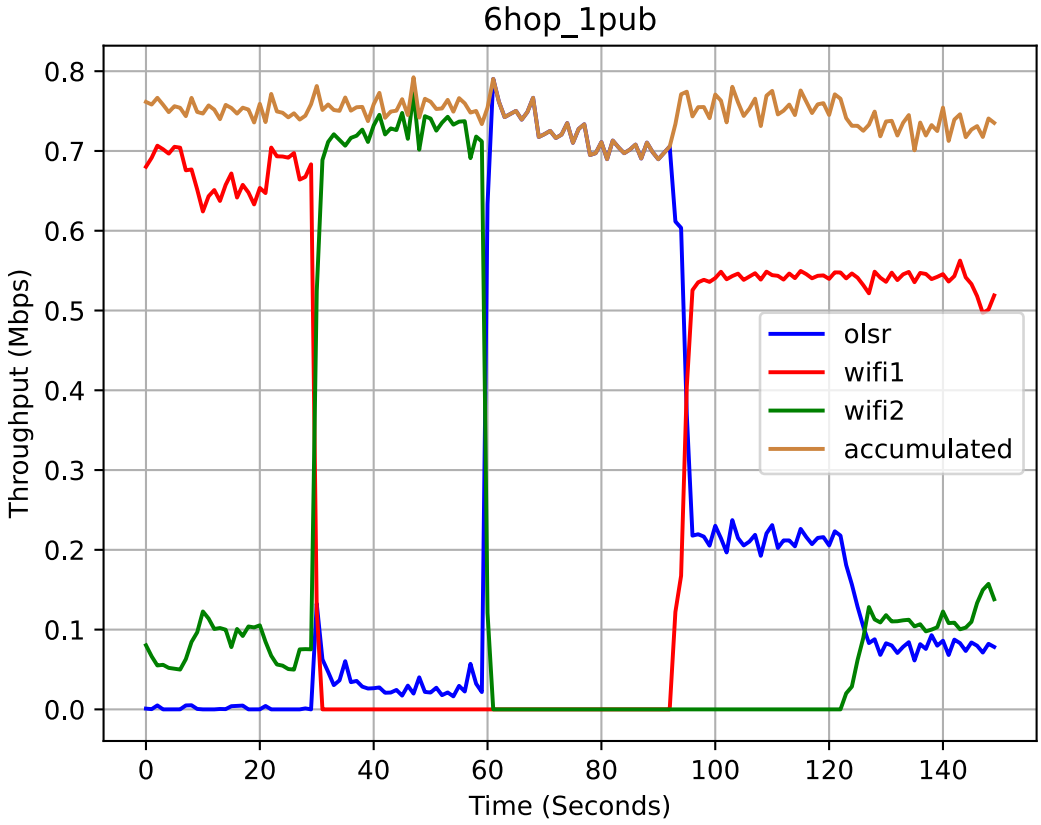


Figure 5.11: Throughput split in publisher with 6 hops

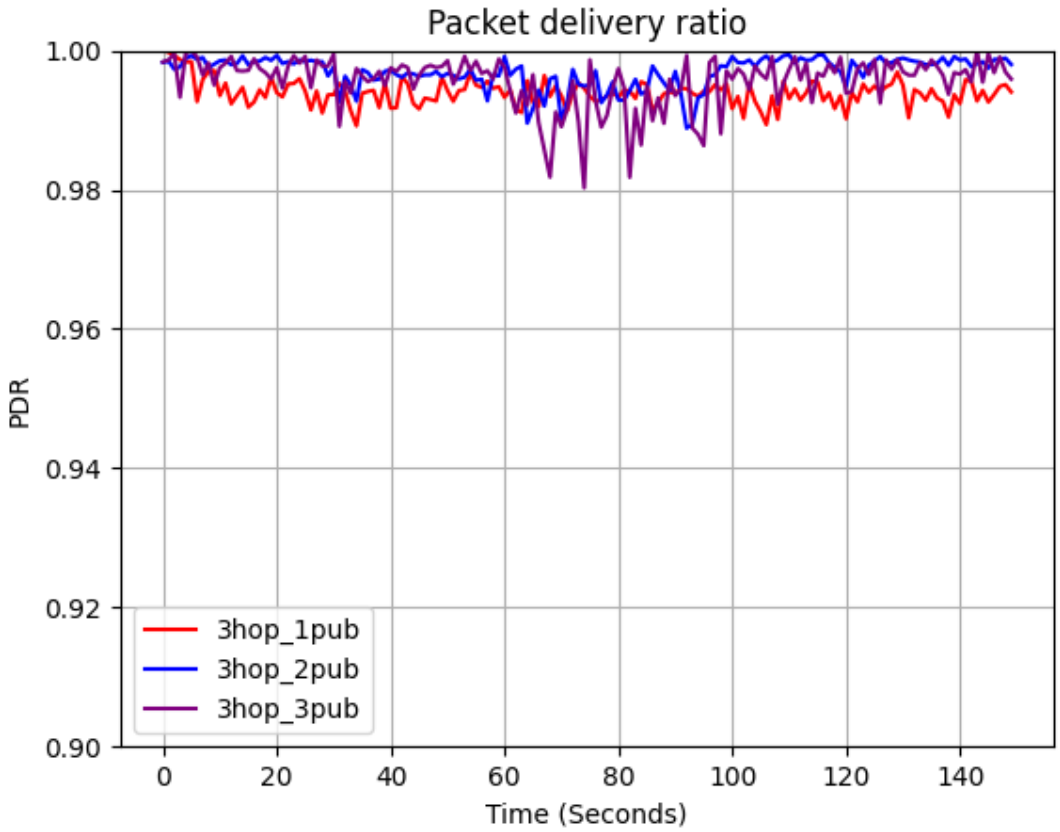


Figure 5.12: Comparison of packet delivery ratio with 3 hops

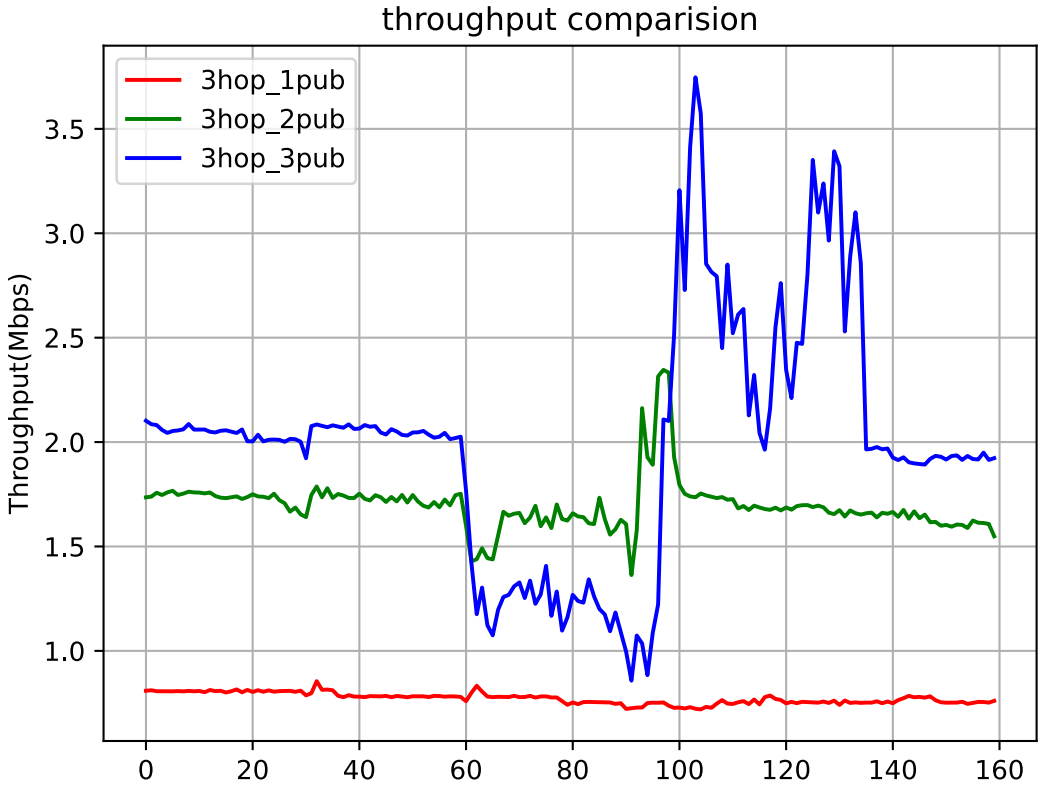


Figure 5.13: Comparison of throughput with 3 hops

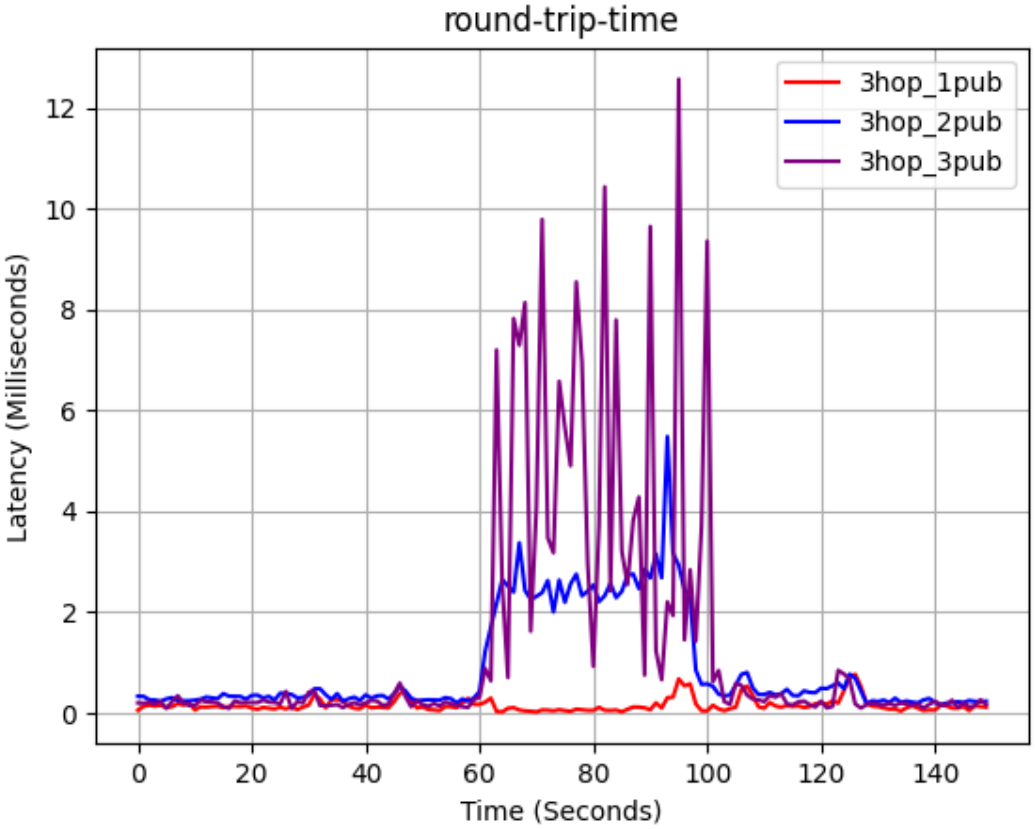


Figure 5.14: Comparison of round trip time with 3 hops

Upon reaching the 60^{th} second, the second WiFi AP mode interface is deactivated, leaving only the Ad Hoc network interface. The publishers are then compelled to use the Ad Hoc mode with OLSR routing algorithm to route all data packets to the gateway. With a solitary publisher in the network, latency remains low, and the packet delivery ratio remains close 100%. Although the throughput experiences minimal change, introducing more publishers leads to a substantial increase in latency and a subsequent drop in throughput, indicative of heightened congestion within the network. Nevertheless, the connection remains intact.

By the 90^{th} second, a WiFi AP mode interface is reactivated, resulting in increased throughput in scenarios with multiple publishers. The increase is due to congestion with only the OLSR interface in the publisher. After 100^{th} second, the performance recovers. Thus, latency becomes lower and fewer retransmissions occur. Observations during the experiment indicate an enhancement in video quality.

At 120^{th} second, the second WiFi AP mode interface is brought back. Latency and the packet delivery ratio maintain stability, and the throughput achieves a steady state as well.

MPTCP Scheduler

The scheduling decision is crucial in MPTCP because it can impact performance and quality of experience[57]. In the current implementation of Multipath TCP in the Linux kernel, the scheduler always prefers the subflow with the smallest round-trip time to send data. There is another implementation with a redundant scheduler where it sends duplicate data over all the subflows. Figures 5.15, 5.16, and 5.17 show the comparison of the throughput, round-trip time and packet delivery ratio between a default scheduler and a redundant scheduler. It is obvious that throughput increases with the redundant scheduler, since all subflows are sending duplicated data. However, in Figure 5.17 we see that with a redundant scheduler, the packet delivery ratio is always higher than the default scheduler.

5.8.3 Link Switch in TCP

Furthermore, we executed a comparative analysis by conducting a test employing the regular TCP protocol, and the results are depicted in Figure 5.21. This experiment followed the same

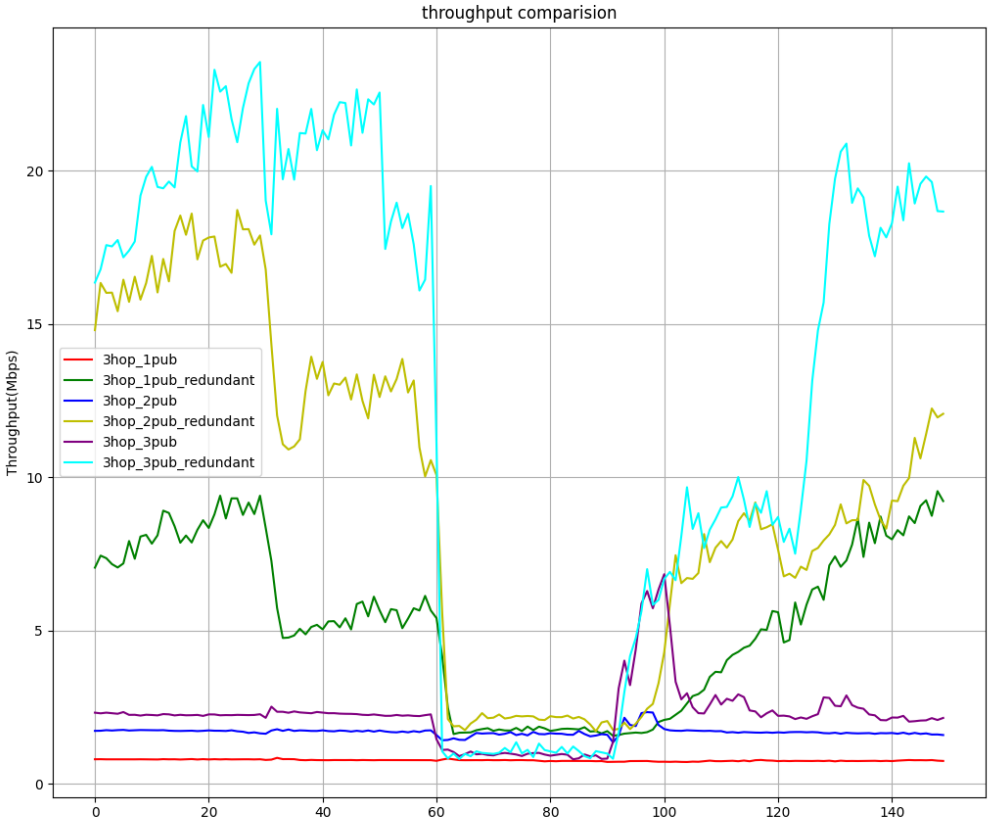


Figure 5.15: Comparison of throughput in 3 hops

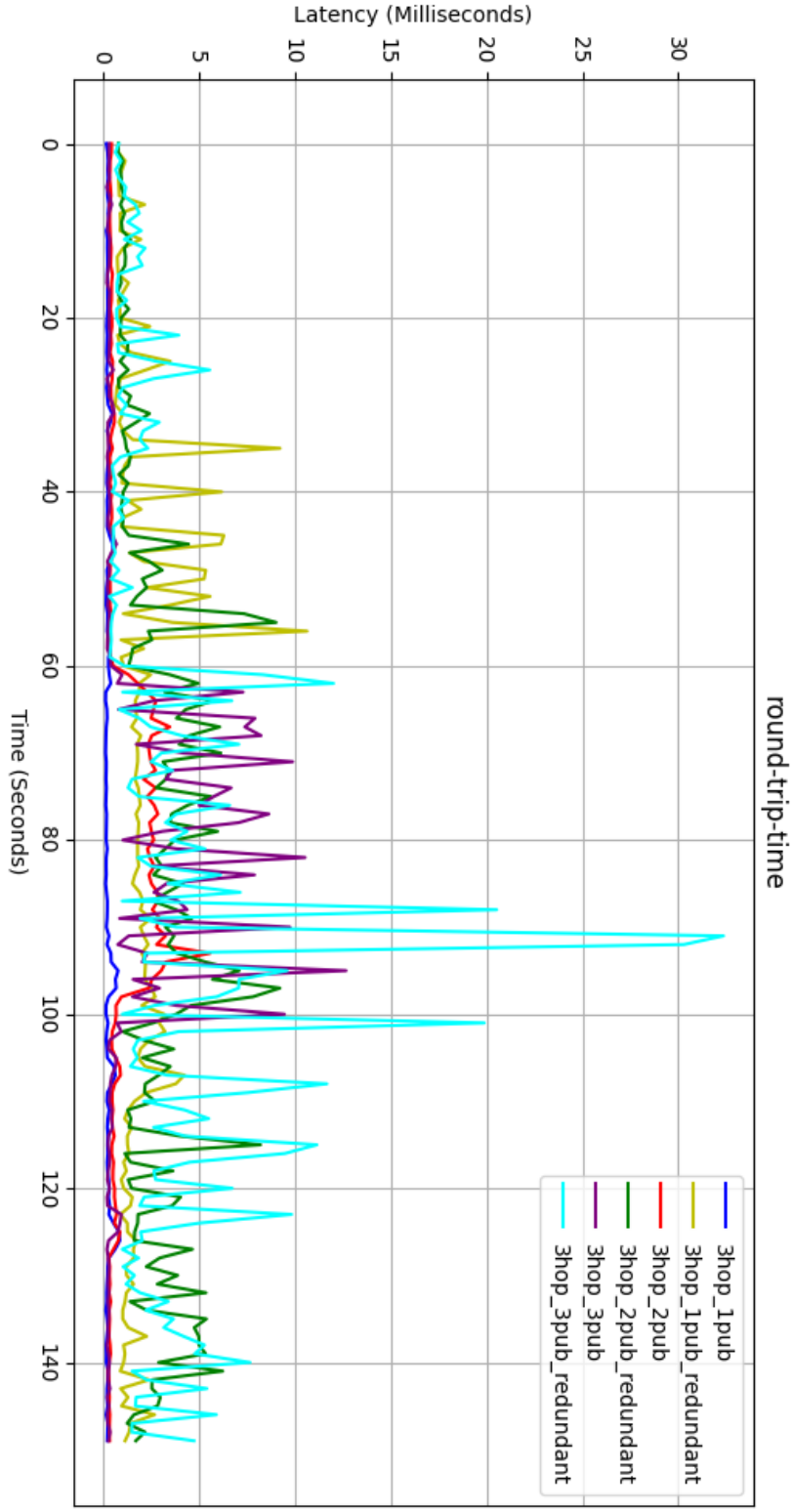


Figure 5.16: Comparison of round trip time in 3 hops

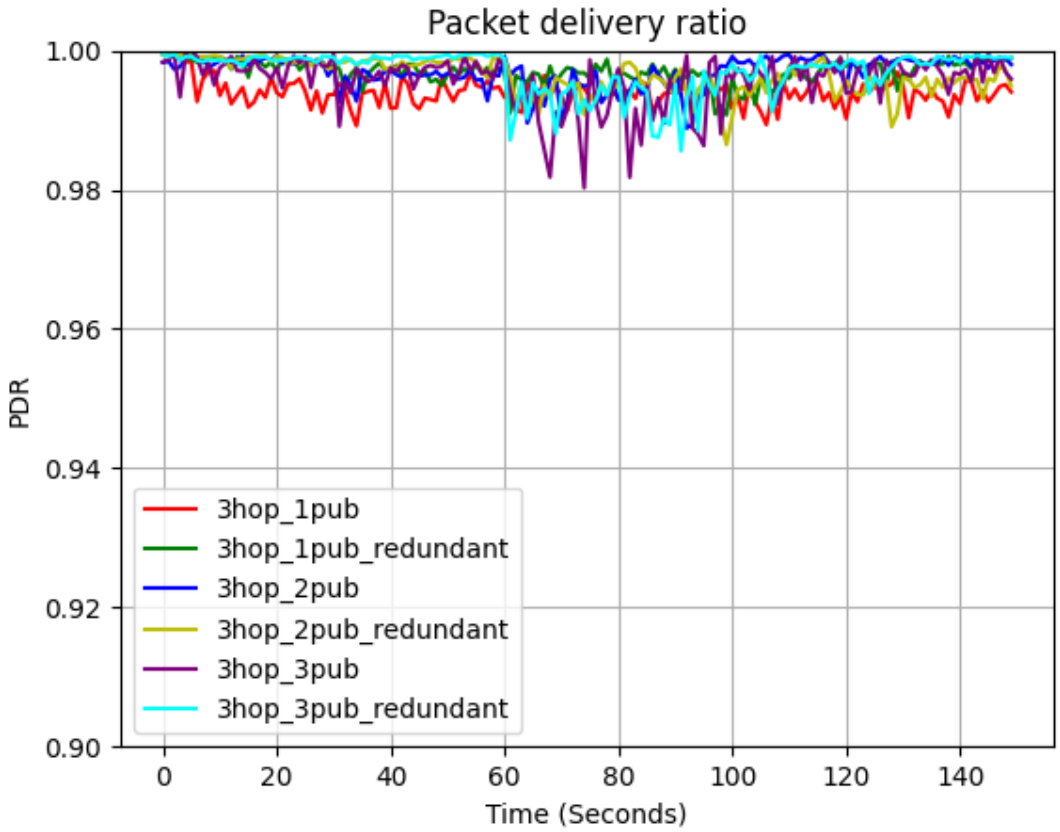


Figure 5.17: Comparison of packet delivery ratio in 3 hops

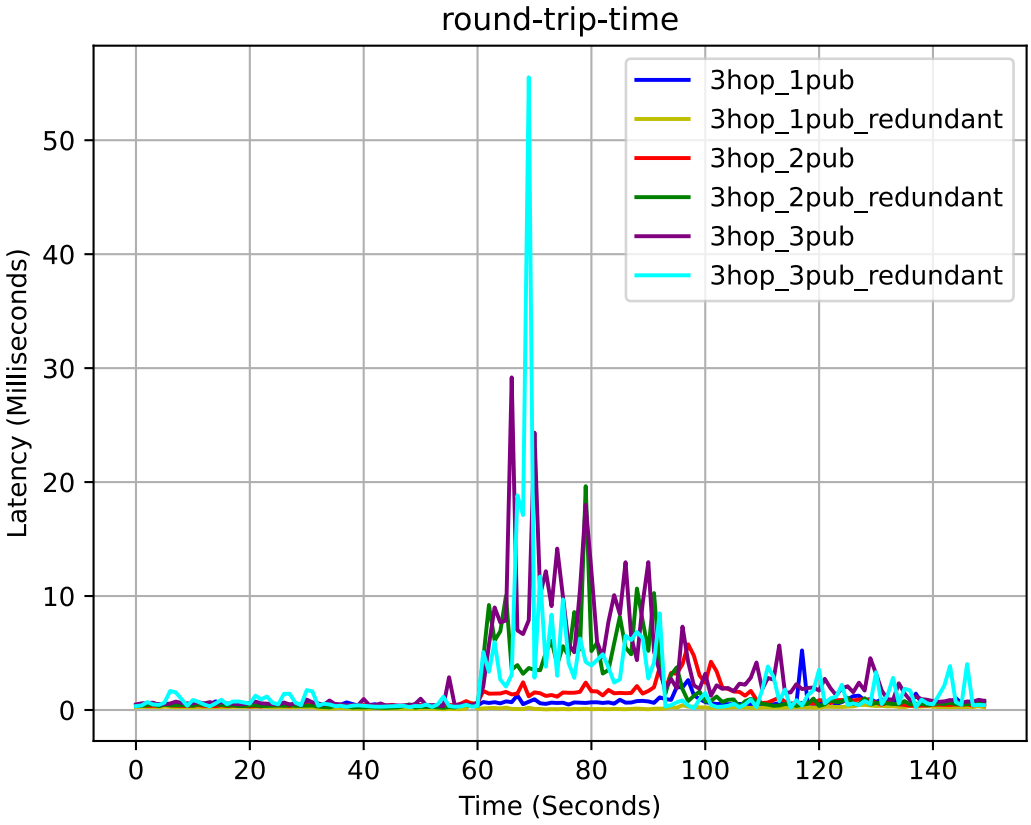


Figure 5.18: Stress test for comparison of round trip time in 3 hops

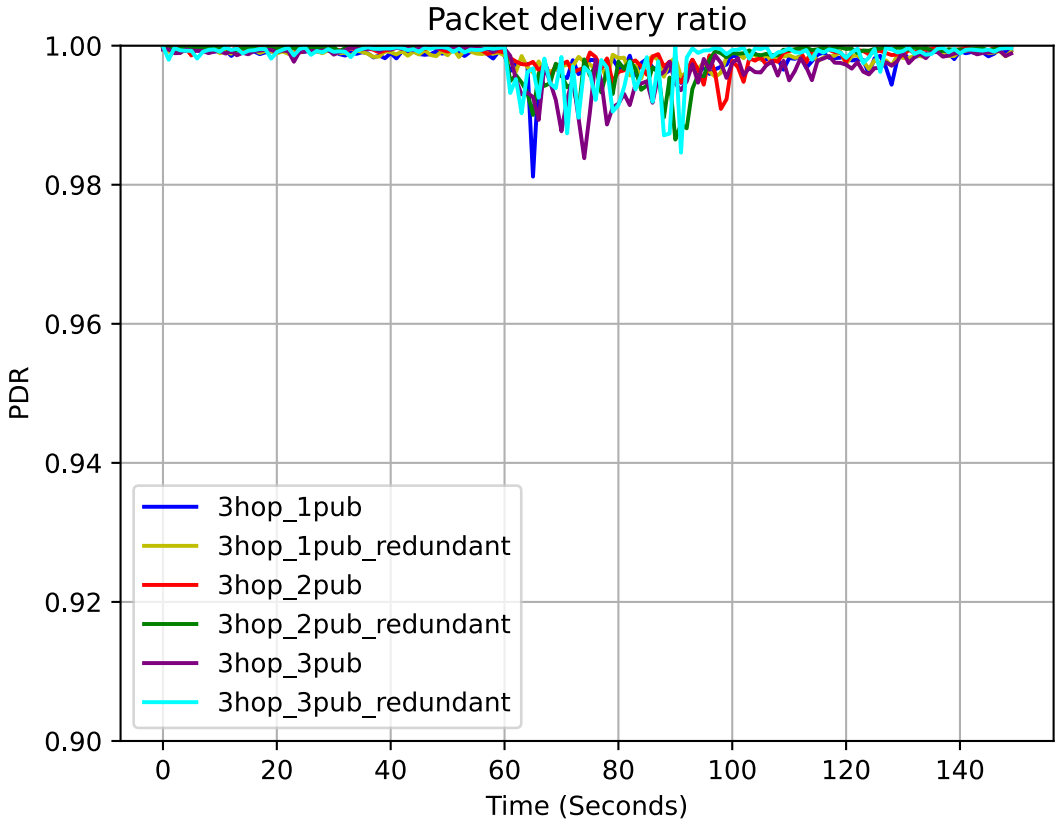


Figure 5.19: Stress test for comparison of packet delivery ratio in 3 hops

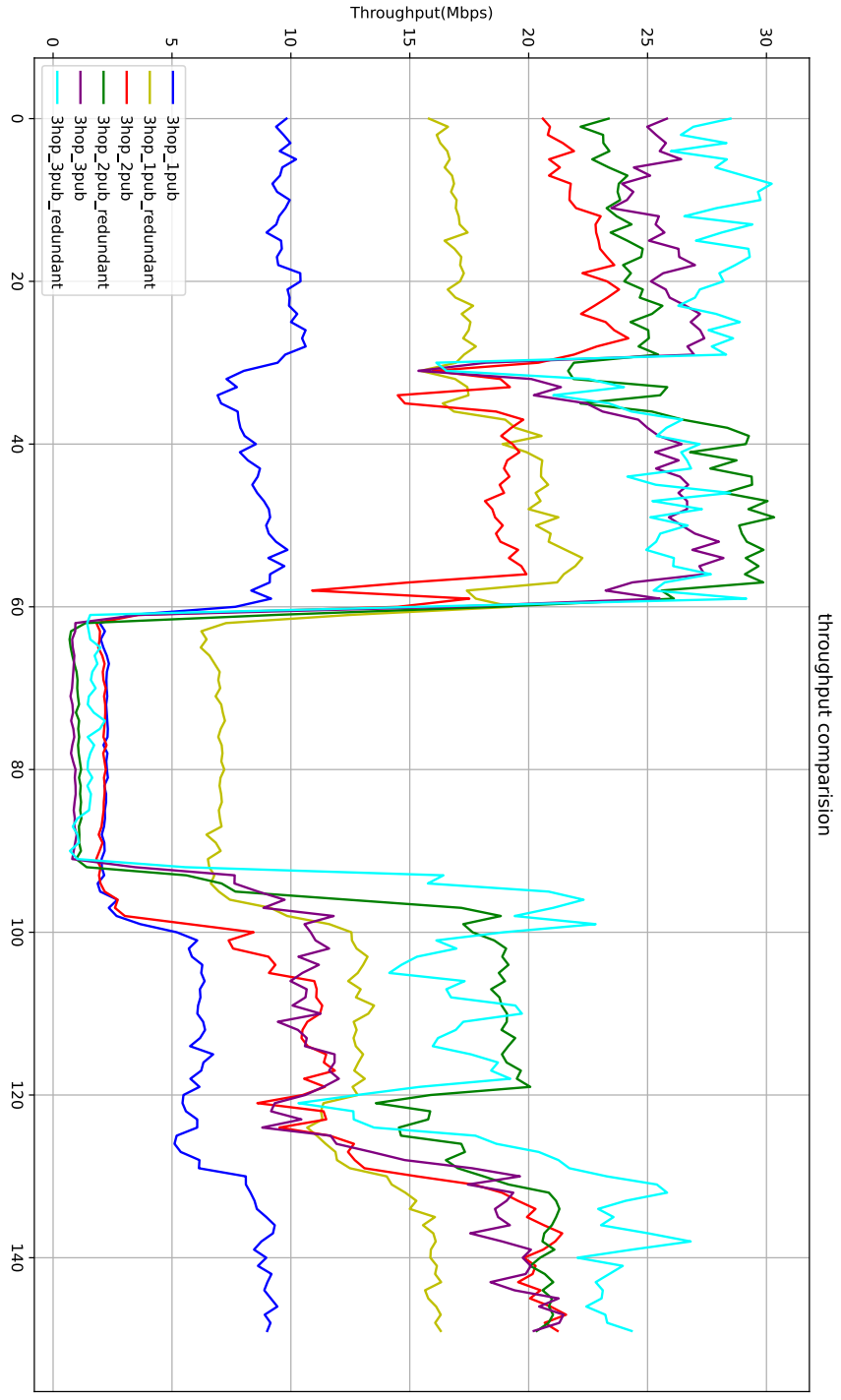


Figure 5.20: Stress test for comparison of throughput in 3 hops

scheme and setup as the previous one. Initially, all interfaces were accessible from the 0^{th} second to the 30^{th} second, with the transmission primarily utilizing a single WiFi AP mode interface.

At the 30^{th} second, we disabled this specific interface, leading to a complete breakdown of the connection and resulting in the loss of all transmitted data. Despite the presence of two additional interfaces, the conventional TCP protocol encountered difficulties in re-establishing the connection through alternative interfaces, mirroring the challenges identified in the previous test.

This scenario underscores the significance of the MPTCP as the transport layer protocol, particularly in emergency situations where infrastructure vulnerabilities may disrupt connectivity. The extended disconnection observed in the TCP protocol test, lasting until the same WiFi AP mode interface became operational again at the 90^{th} second, emphasizes the potential risks associated with relying on a single interface. In contrast, the proposed connection hub aims to mitigate such risks by facilitating seamless interface transitions, minimizing disconnection durations, and ensuring the continuous flow of critical data even in adverse conditions.

By employing the TCP interface controller proposed in 1, we have the capability to emulate a seamless transition among all interfaces, and the outcomes of this simulation are depicted in Figure 5.22.

We apply the identical experimental framework and configurations, where all interfaces remain accessible from the 0^{th} second to the 30^{th} second.

At 30^{th} second, when the initial WiFi AP mode interface is disabled, the link switch controller detects the disconnection and efficiently re-establishes the connection utilizing the second WiFi AP mode interface. Similarly, at the 60^{th} second, when the second WiFi AP mode interface undergoes disconnection, the link switch controller breaks the connection from the server and adeptly re-establishes the connection, utilizing the OLSR interface. The average duration for the switch between interfaces is measured at 350 ms.

The incorporation of the link switch controller empowers TCP to seamlessly navigate between interfaces, providing a valuable adaptive mechanism. However, it is essential to note a trade-off in this process. While TCP gains the ability to switch between links, this dynamic

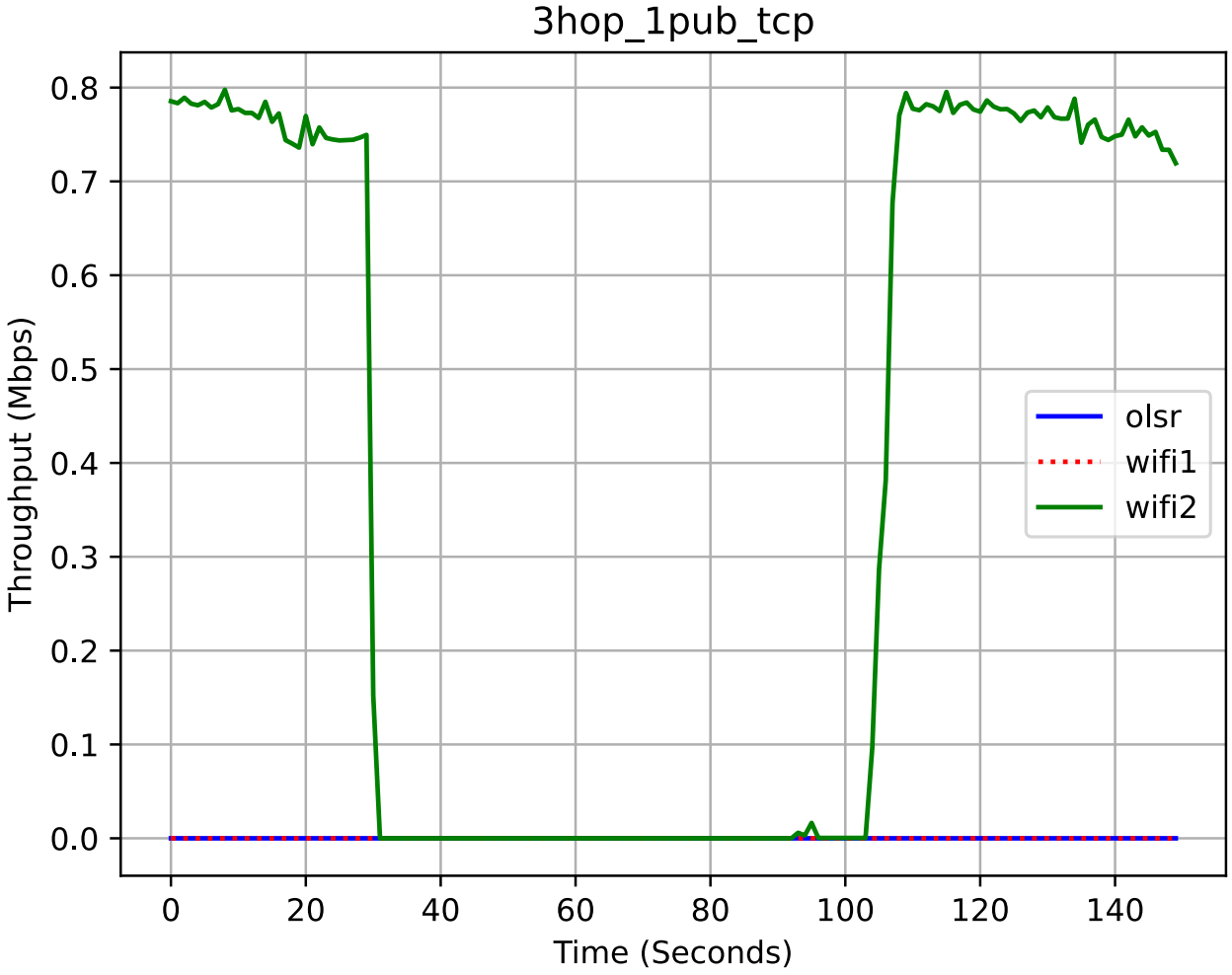


Figure 5.21: Regular TCP splits in the publisher

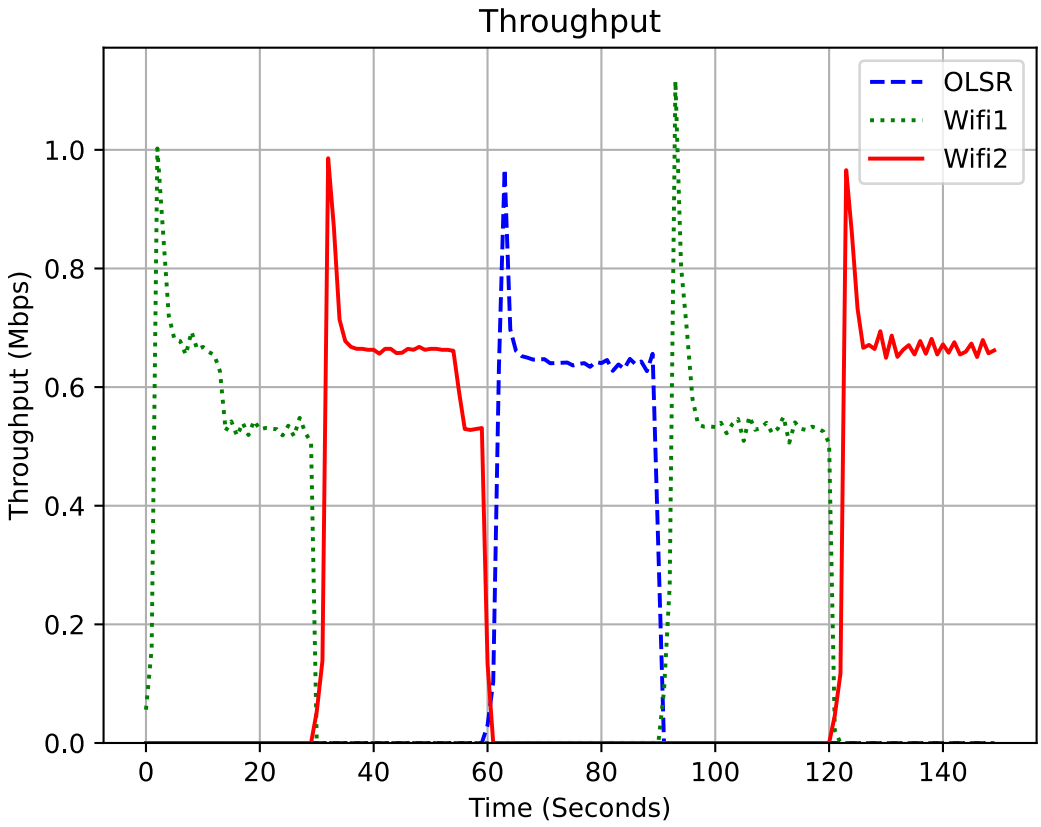


Figure 5.22: TCP with link detection splits in publisher

operation results in the disruption of the end-to-end connection. Furthermore, the protocol is unable to leverage the simultaneous availability of multiple interfaces, showcasing a limitation in its ability to harness the full potential of diverse network pathways.

5.9 Summary

The proposed NGFR Communication Hub design is validated by implementing a prototype and testing it in an indoor environment. Introducing MPTCP provides smooth switch between different interfaces while maintaining an end-to-end connection. Compared to the traditional MANET application used with regular TCP, the NGFR Communication Hub provides a reliable and resilient transmission. We complete extensive tests with different network configurations. The test result shows that the NGFR Communication Hub is capable of self-connecting, self-organizing, and rapidly deployed.

Although the proposed method combining MPTCP into MANET helped to improve the reliability of network communication, this design still suffer from several limitations, such as bandwidth-constrained, variable capacity links and energy-constrained operations. Due to the natural of MANET, the performance is always limited by the wireless channel, the interference of nearby nodes and the computation capability of the device. Furthermore, the dynamic topology causes frequent link failures and high error rates, which makes it difficult to maintain a desired degree of Quality of Service(QoS). In addition, live streaming video always requires high bandwidth and low transmission latency. Therefore, providing a robust and low-latency video streaming solution is a challenge.

Chapter 6

Multiple Gateways in NGFR Communication Hub

Based on the results obtained from the previous experiment, it is evident that as the number of hops in the network increases, it leads to higher latency, decreased throughput, and a lower packet delivery ratio. Thus, a potential solution to mitigate these issues is to introduce additional gateways to the OLSR network. By strategically placing more gateways, it is possible to alleviate the problems associated with excessive hops, thereby improving overall network efficiency and enhancing the transmission of data packets.

6.1 Motivation

Drawing insights from the outcomes of the preceding experiment, a discernible trend emerges: an increase in the number of hops within the network correlates with heightened latency, diminished throughput, and a decreased packet delivery ratio. The observed relationship underscores the impact of network topology on performance metrics, where an augmented number of hops introduces additional traversal points, contributing to the observed adverse effects on latency, throughput, and packet delivery.

As the data reveals, each additional hop introduces a potential point of delay and a subsequent increase in the time taken for data to traverse the network. This cumulative effect manifests in elevated latency, as the signals encounter multiple intermediary nodes before reaching their intended destination. Consequently, the latency metric exhibits an upward trend as the network's hop count escalates.

Simultaneously, the experiment highlights a decrease in throughput as the number of hops increases. The throughput reduction is attributable to the increased complexity and extended

route length, resulting in a slower overall data transfer rate. The inefficiencies introduced by additional hops contribute to a diminished capacity for the network to transmit data efficiently.

Moreover, the packet delivery ratio experiences a decline with an expanding number of hops. This decline is indicative of a heightened likelihood of packet loss or delivery failures. The intricate nature of routing through multiple hops introduces complexities that can lead to packet drops, resulting in an overall reduction in the packet delivery ratio.

6.2 Proposed Solution

One viable strategy to address the challenges posed by the observed issues is the introduction of additional gateways to the Optimized Link State Routing (OLSR) network. This proactive approach involves strategically deploying extra gateways within the network topology, with the overarching goal of mitigating the problems associated with an excessive number of hops. This strategic placement aims to enhance overall network efficiency and optimize the transmission of data packets.

By introducing supplementary gateways, the network architecture undergoes a beneficial transformation. The strategic placement of these gateways serves to reduce the average number of hops that data packets need to traverse to reach their destination. This reduction is pivotal in curtailing the detrimental impact on latency, throughput, and packet delivery ratio, as previously identified. The additional gateways create alternative and potentially shorter paths for data, circumventing the need for extensive and convoluted routes.

Furthermore, the strategic deployment of gateways facilitates load balancing within the network. With multiple gateways, the network can distribute traffic more evenly, preventing congestion on specific routes and ensuring a more equitable utilization of resources. This load-balancing effect contributes to enhanced network resilience and responsiveness, particularly in scenarios where nodes frequently connect and disconnect.

In summary, the introduction of additional gateways to the OLSR network represents a proactive and effective solution to mitigate the challenges associated with an elevated number of hops. This approach not only optimizes network efficiency but also bolsters overall performance, offering a robust foundation for the seamless transmission of data packets within the

network. As network designs evolve, the strategic placement of gateways emerges as a crucial element in enhancing the agility and responsiveness of OLSR networks.

6.3 Experiment Configuration

Figures 6.1 and 6.2 are the selected configurations for the experiment design. To ensure a stable connection for more nodes in the OLSR network, an extra gateway is introduced into the OLSR network. The network comprises two publishers (P) and two gateways (G), with each publisher specifically configured to connect to a designated gateway.

6.4 Experiment Design

We extend the same experiment scheme from the previous experiment design. In each configuration, there are a total of 25 executions divided into 5 groups. The average values are computed at intervals of 0.1 seconds. At the beginning of each experiment, all interfaces are activated and assigned unique IP addresses corresponding to different networks.

Each execution consists of three phases, as indicated in Table 6.1. The initial phase lasts for 60 seconds, during which devices transmit data continuously without any interruptions. At the 30^{th} second, one of the WiFi AP mode interfaces is deactivated, and data transmission continues for another 30 seconds. The second phase begins at the 60^{th} mark, where the second WiFi AP mode interface is deactivated, leaving only the Ad Hoc network active. The third phase starts at the 90^{th} second mark, where one of the WiFi AP mode interfaces is reactivated. Following 30 seconds of communication, the second WiFi AP mode interface is also reactivated, resulting in all three interfaces becoming available simultaneously. Within the network, there exists multiple gateways to choose from. Each publisher establishes a connection with a specific gateway, and this assigned gateway remains stable throughout the duration of the connection.

Time	Status Interface I(WiFi)	Interface II (WiFi)	Interface III(OLSR)	Gateway Switch
0-30	up	up	up	-
30-60	down	up	up	-
60-90	down	down	up	-
90-120	up	down	up	-
120-150	up	up	up	-

Table 6.1: Experiment Design with Multiple Gateways

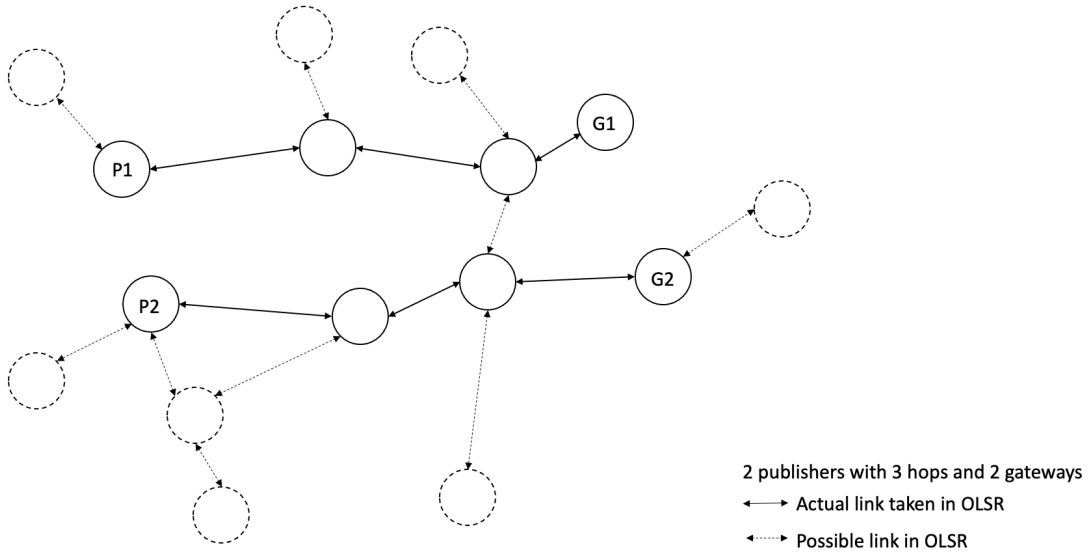


Figure 6.1: 2 publishers and 2 gateways with a maximum of 3 hops

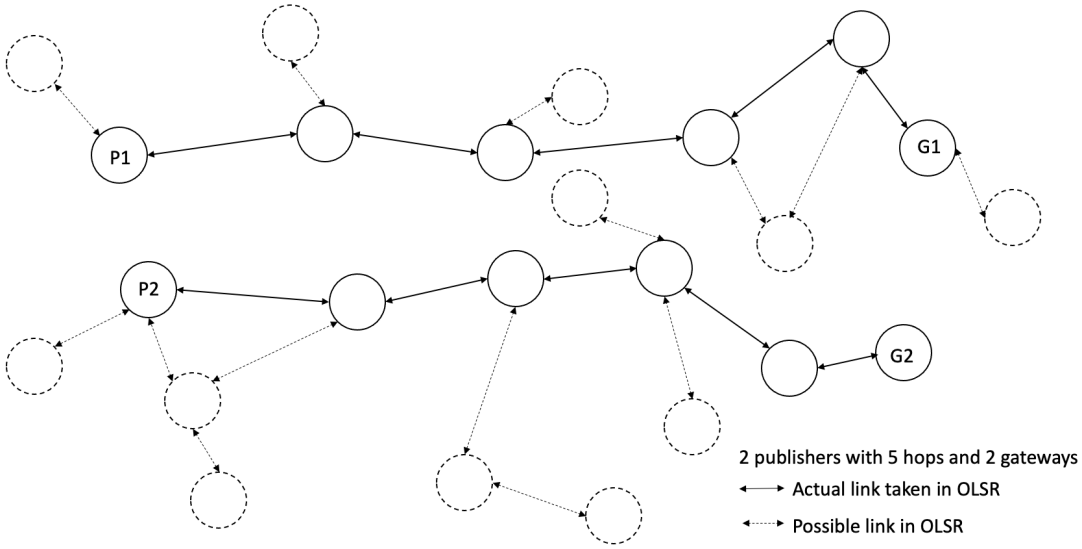


Figure 6.2: 2 publishers and 2 gateways with a maximum of 5 hops

6.5 Result Analysis

All the publishers are equipped with three network interfaces, two WiFi interfaces and one OLSR interface respectively. All the publishers follow the same experiment schedule as showed in Table 6.1.

At 0^{th} second, three interfaces are available. There is no latency in transmission. The throughput is stable with a high packet delivery ratio.

At 30^{th} second, the first WiFi AP mode interface is deactivated. However, the latency and packet delivery ratio remain relatively stable, as shown in Figure 6.3 and Figure 6.4. The packet delivery ratios remain close to 100% until the 60^{th} second, and Figure 6.5 illustrates that the throughput in different configurations are similar.

At the 60^{th} second, the second WiFi AP mode interface is taken offline, leaving only the Ad Hoc network interface. To transmit data packets to the gateway, the publisher must utilize the Ad Hoc mode with the OLSR routing algorithm. While there is a noticeable increase in latency, the decrease in packet delivery ratio and throughput is relatively minimal compared to single gateway experiment.Additionally, during the experiment, fewer glitches are observed in video transmission compared with the single gateway configuration.

At the 90^{th} second, a WiFi AP mode interface is brought up, requiring MPTCP to adapt its flow to a less congested path, which takes a few seconds. Subsequently, from the 100^{th} second onward, there is a discernible improvement in performance, specially in terms of latency. This observation provides evidence that introducing an additional gateway in the OLSR network effectively mitigates congestion levels, especially when dealing with a higher number of hops and multiple publishers in the MANET.

At the 120^{th} second, the second WiFi AP mode interface is activated, without affecting the stable latency and packet delivery ratio observed previously.

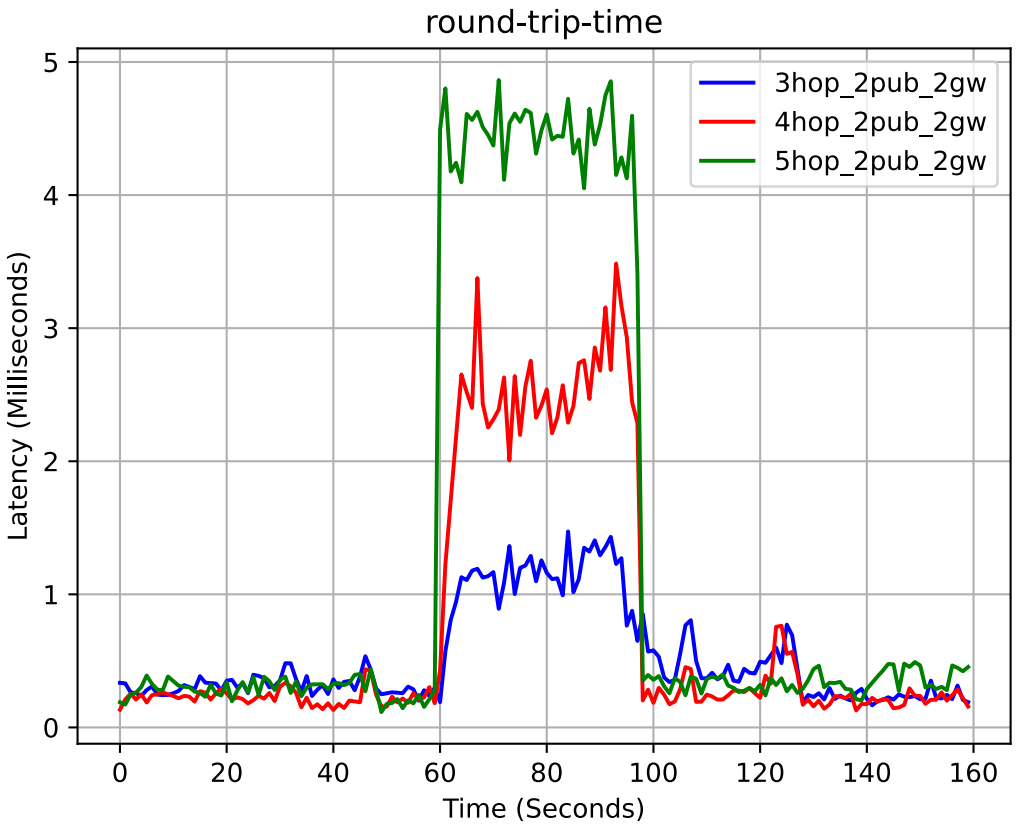


Figure 6.3: Comparison of round trip time in different number of hops with two publishers and two gateways

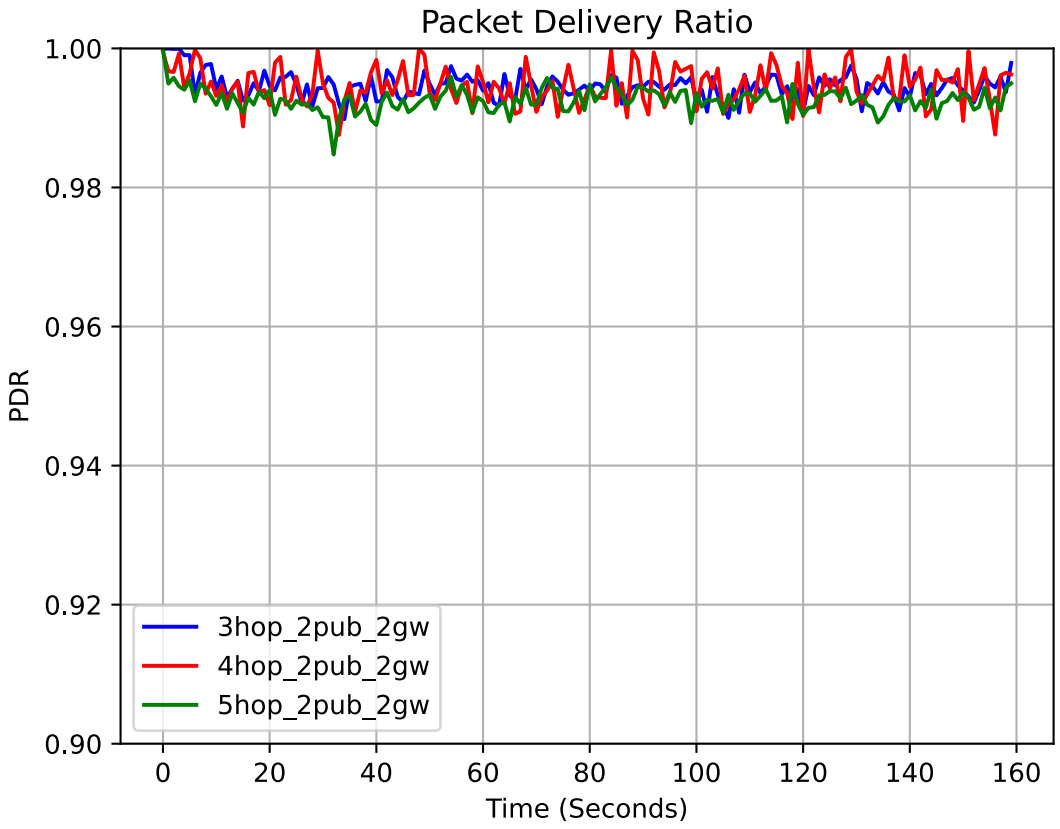


Figure 6.4: Comparison of packet delivery ratios in different number of hops with two publishers and two gateways

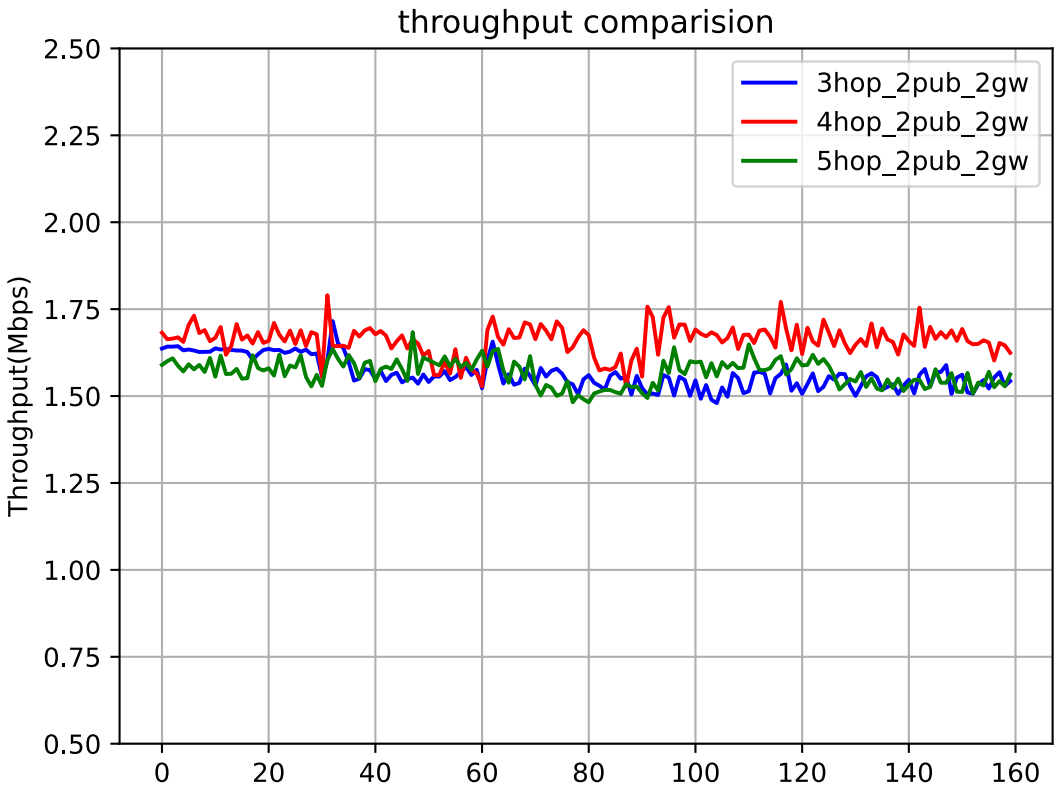


Figure 6.5: Comparison of throughput in different number of hops with two publishers and two gateways

6.6 Summary

By introducing more gateways in the system, we can create a more robust and efficient network infrastructure. Multiple gateways act as entry points for the OLSR network helps creating a more stable network environment.

Partitioning the OLSR network into smaller sub-networks offers several advantages. Firstly, it enables us to isolate different segments of the network, ensuring that issues in one area do not affect the entire system.

Additionally, this partitioning enhances throughput stability, meaning the network can sustain a steady and reliable data transfer rate, even with multiple publishers sending data.

Reducing latency is another crucial benefit. With multiple gateways, more publishers are capable of sending packets simultaneously, leading to a reduction in communication delays. This is especially crucial for real-time applications such as real-time video, audio, and crucial messages.

Overall, the experiment proves that introducing multiple gateways improves the packet deliver ratio, alleviating the high latency time in long number of hops scenario.

Chapter 7

Dynamic Gateway Switch in OLSR network

7.1 Motivation

From the previous test result, we notice that with the increasing of the publishers in the network or the increasing of the number of hops, the throughput drops dramatically and also brings the latency to the network. Introducing new gateways to the system and partition the network into different sub networks will help to solve the problem. However, it is inevitable for nodes in the MANET to be mobile and causing the connection and disconnection with the gateway frequently. It is easy to switch to the best gateway for nodes in MANET. However, for routing packets outside the MANET via gateway, disconnecting the current gateway will bring problems to the routing table. Figure 7.1 shows an example of multiple gateways in one OLSR network. Nodes are partitioned into two small sub-nets to get the best performance by utilizing the nearest gateway. If Gateway_1 is not available anymore, nodes in OLSR_1 will automatically route to find Gateway2 as their gateways. However, the cloud server is not aware of this switch and keeps sending acknowledgements back to Gateway_1 as shown in Figure 7.2.

7.2 Proposed Solution

In addressing the challenge of managing multiple gateways within the Optimized Link State Routing (OLSR) protocol, we have proposed an innovative solution that leverages TCP Fast Open (TFO) to efficiently transmit the current selected gateway information to the cloud server.

Traditionally, the TCP handshake incurs a substantial delay, equivalent to one full Round-Trip Time (RTT). The RTT, representing the time taken for a packet to travel from the sender to

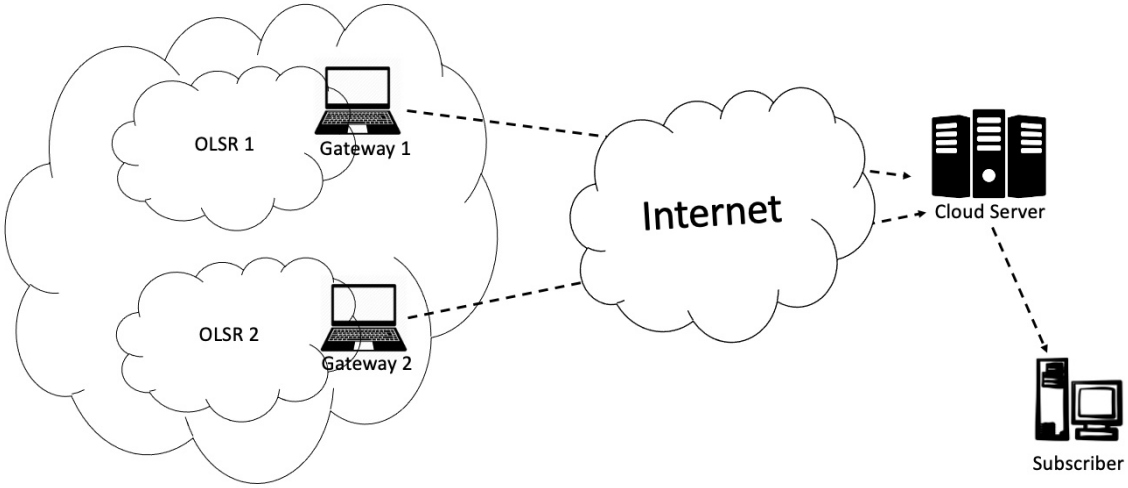


Figure 7.1: Multiple gateways MANET

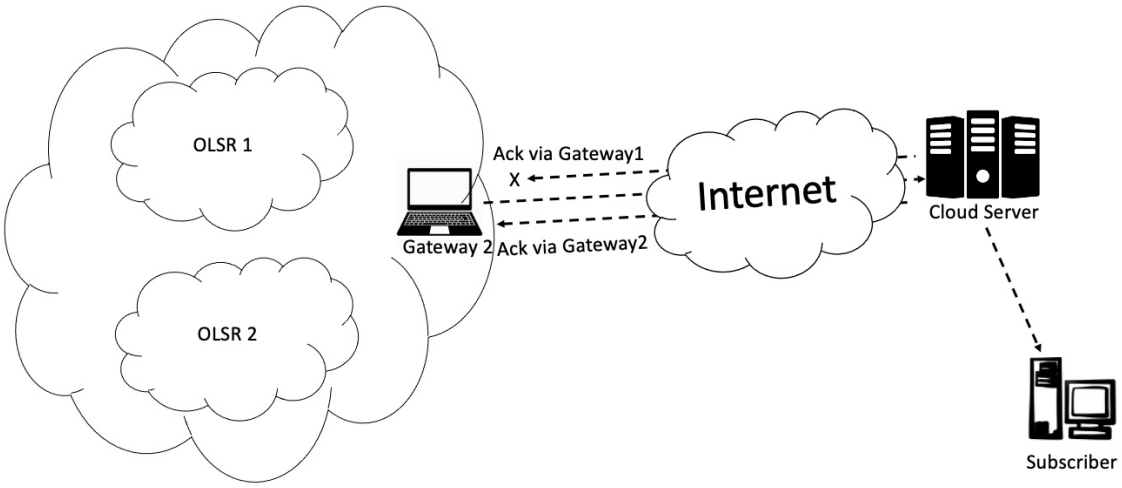


Figure 7.2: Gateway failure

the receiver and back, becomes a critical factor, particularly for traffic characterized as 'short-lived' and 'time-sensitive.' Examples of such traffic include web browsing, where users visit websites. In scenarios where the propagation delay is high or the mobile network exhibits sluggish performance, the impact of one RTT can significantly degrade the overall user experience.

For 'short-lived' and 'time-sensitive' traffic, one RTT becomes a substantial duration, and its overhead can be particularly notable. Although in conventional scenarios, one RTT might not drastically impact transmission performance, the situation is different in a Mobile Ad Hoc Network (MANET). In a dynamic MANET environment characterized by rapid node movements, frequent connection, and disconnection events, minimizing RTT assumes paramount importance. This reduction in RTT is crucial for enhancing the throughput of the MANET, ensuring a more responsive and efficient communication network.

By introducing TFO into the solution for managing multiple gateways in OLSR, we aim to expedite the communication process and reduce the latency associated with the TCP handshake. This optimization becomes especially pertinent in the context of MANETs, where the agility and responsiveness of the network are paramount. Through the incorporation of TFO, our proposed solution seeks to minimize the impact of RTT, fostering improved throughput and performance in dynamic MANET scenarios marked by frequent node mobility and connectivity changes.

TCP Fast Open (TFO) emerges as an innovative transport layer solution designed to circumvent the overhead associated with one full Round-Trip Time (RTT) between a client and a server. Specifically, it targets the elimination of the TCP three-way handshake for repetitive connections, a process that traditionally consumes one complete RTT before the actual data exchange begins.

In a standard TCP connection, the initial round-trip involves the establishment of the connection, with the true communication initiating only from the third packet onward. TFO comes into play when dealing with repeated connections, leveraging a cryptographic cookie requirement to streamline the process. During the initial interaction between the client and the server, a traditional three-way handshake occurs, during which the server shares a cryptographic cookie with the client.

For subsequent connections, the client optimizes the process by encapsulating both the request and the cryptographic cookie within the SYN packet itself. By doing so, the client effectively piggybacks this information onto the initial packet, negating the need for a full three-way handshake. Upon receiving this SYN packet, the server authenticates the client using the embedded cookie, promptly accepts the connection request, and even initiates data transmission in the subsequent ACK packet.

In essence, TCP Fast Open introduces a clever mechanism to expedite connection establishment for repeated connections, leveraging the sharing and reuse of cryptographic cookies. By doing away with the necessity for a full three-way handshake for each connection, TFO significantly enhances the efficiency of communication between clients and servers, especially in scenarios involving repetitive interactions. This optimization contributes to a reduction in latency and an overall improvement in the responsiveness of the network.

In situations where nodes within the Optimized Link State Routing (OLSR) protocol need to transition to a new gateway, the implementation of TCP Fast Open (TFO) presents a valuable solution for efficiently communicating this change to the server. During the initiation of the connection between the publisher and the server, the server distributes a connection cookie to the publisher. This cookie serves as a crucial element in the subsequent communication process.

Given the table-driven nature of the OLSR routing protocol, each node in the network maintains a gateway table. In the event of a gateway change, the publisher utilizes TFO to transmit the updated gateway information to the cloud server. This mechanism ensures a streamlined and rapid exchange of information without relying on traditional connection establishment procedures.

Upon receiving the new gateway information, the server initiates a verification process by scrutinizing the cryptographic cookie, as depicted in Figure 7.3. If the cookie is validated successfully, the cloud server proceeds to decode the received message and promptly updates the existing routing table with the new gateway information. This seamless process enables the maintenance of end-to-end connections without necessitating substantial alterations to the current network architecture.

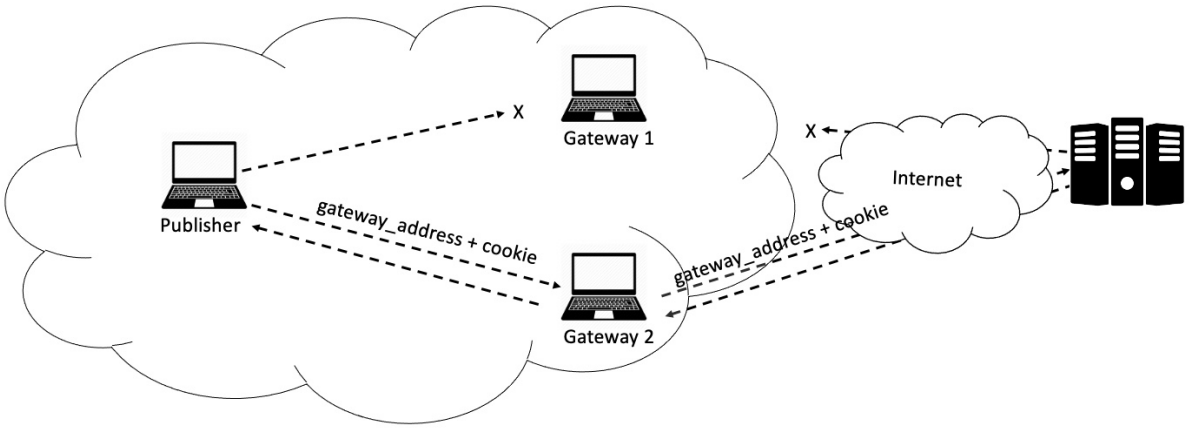


Figure 7.3: Dynamic gateway selection with TFO

The proposed approach, leveraging TFO in conjunction with the distribution and validation of connection cookies, provides an efficient means of handling gateway transitions in OLSR networks. By facilitating swift communication between nodes and the cloud server, this method ensures the continuity of connections while minimizing disruption and maintaining the integrity of the overall network structure.

7.3 Implementation

The HNA (Host and Network Association) message serves as a periodic broadcast within an OLSR network. It is generated when a node's non-OLSR interface establishes a connection with another network, indicating that the node can act as a gateway to that network. This message plays a crucial role in informing other nodes within the network about the availability of external networks and their associated network addresses and netmasks.

When nodes within the OLSR network intend to connect to an external network, they rely on the HNA messages to determine the appropriate gateway. The node sending the relevant HNA message for the desired destination network is selected as the gateway. In cases where multiple gateways are available, the node with the highest link quality is chosen.

The format of an HNA message is shown in Figure 7.4. For each publisher, with such information, we can retrieve the gateway node information and save for the next step.

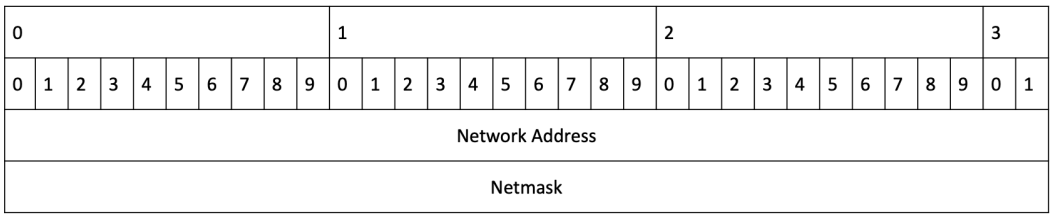


Figure 7.4: HNA message format

The implementation of HNA message is accomplished by sending a ICMP packet to the desired destination. We define an interval of 1 second for each ICMP check. By default, the interval for an HNA message is the same as a TC message, which is 2 seconds. The validation time for an HNA message is 3 times of the interval. To adapt the mobility of the OLSR network, we define an interval of 2 seconds for each HNA message and 2 seconds for validation time. We also implement an ETX protection method to avoid frequent change in the gateway switch. ETX [23] is calculated to by:

$$ETX = \frac{1}{NLQ * LQ}$$

LQ (Link Quality) is the probability for a successful packet transmission from this neighbor to current node[59]. Whereas the NLQ (Neighbor Link Quality) is the bidirectional Link Quality. A good link is considered as have an ETX of 1.0. An ETX protection method will keep checking the ETX of the gateway. If a new gateway joins the network and has a higher ETX value, even it has less number of hops to the publisher, it will not be selected as the gateway.

7.4 Experiment Set Up

We adopt the same experimental framework as in the previous experiment design. Each configuration comprises a total of 25 executions distributed across 5 groups. Average values are computed at intervals of 0.1 seconds. At the initiation of each experiment, all interfaces are activated, and unique IP addresses corresponding to different networks are assigned. Each publisher is linked to a distinct gateway.

Every execution unfolds in three phases, delineated in Table 7.1. There are 4 switches in the experiment. The initial phase spans 30 seconds, during which devices transmit data continuously without any interruptions. At the 30^{th} second, gateway_1 disconnects from the desired network, resulting an initiation of a gateway switch. The traffic that used to go through from gateway_1 has shifted to gateway_2. The second switch begins at the 60^{th} mark, where gateway_2 is disconnected from the network and gateway_1 is brought back. The third switch starts at the 90^{th} second mark, where the interface that connects to the desired network on gateway_1 is brought down and the interface on gateway_2 is brought back. Following 30 seconds of communication, the last switch begins at 120^{th} second, where the interface that connects tot he desired network on gateway_1 is brought back.

Time	Status			Gateway Switch
	Interface I(WiFi)	Interface II (WiFi)	Interface III(OLSR)	
0-30	up	up	up	-
30-60	down	up	up	on
60-90	down	down	up	on
90-120	up	down	up	on
120-150	up	up	up	on

Table 7.1: Experiment Design with Switch in Multiple Gateways

7.5 Result Analysis

We test our solutions with 3 hops, 4 hops and 5 hops of OLSR, respectively. The results are shown as in Figure 7.5, Figure 7.6 and Figure 7.7.

Initially, at the 0^{th} second, each publisher establishes a connection to a gateway, and the traffic distribution through each gateway is approximately even. At 30^{th} second, a significant event unfolds: the first gateway is disconnected. This prompts all the nodes in OLSR_1 to swiftly transition from relying on gateway_1 to utilizing gateway_2. Remarkably, this switch does not result in any disruption to the ongoing traffic flow; instead, the throughput in gateway_2 experiences an immediate boost.

At 60^{th} second, another pivotal switch occurs – gateway_2 is disconnected, and gateway_1 is seamlessly reconnected. All nodes in OLSR_2 now establish connections with gateway_1.

The 90^{th} second marks another noteworthy development, where gateway_1 is deliberately deactivated, and the interface on gateway_2 is reactivated. This prompts the traffic to once again switch its route, this time reverting back to traversing via gateway_2.

Upon reaching the 120^{th} second, both gateways are concurrently activated. Nodes in OLSR_1 and OLSR_2 select the least ETX node as their gateways. However, in experiments involving 3 hops and 4 hops, illustrated in Figure 7.5 and Figure 7.6, the traffic exhibits no inclination to switch from gateway_2 to gateway_1. This lack of switching is attributed to the similar cost associated with both gateways, emphasizing the stability and balance in the network's routing decisions.

7.6 Summary

Due to the dynamic nature of Mobile Ad Hoc Networks (MANETs), where nodes are frequently mobile and experience frequent connection and disconnection events, presented challenges in routing with multiple gateways. To overcome the gateway disconnection problem, we proposed a solution centered around the integration of TCP Fast Open (TFO) within the Optimized Link State Routing (OLSR) protocol. The motivation behind this approach stemmed from the need to efficiently transmit selected gateway information to the server, particularly during dynamic gateway switches.

In the context of OLSR networks, where gateway transitions pose a challenge to routing tables and connection stability, the proposed approach seamlessly integrates TFO with the distribution and validation of connection cookies. This integration ensures fast communication between nodes and the server during gateway switches, maintaining the continuous connections without compromising the overall network structure.

We have done extension experiments to validate our proposed solution.The proposed approach demonstrates its effectiveness in managing multiple gateways in a dynamic MANET environments, offering a practical solution for real-world applications.

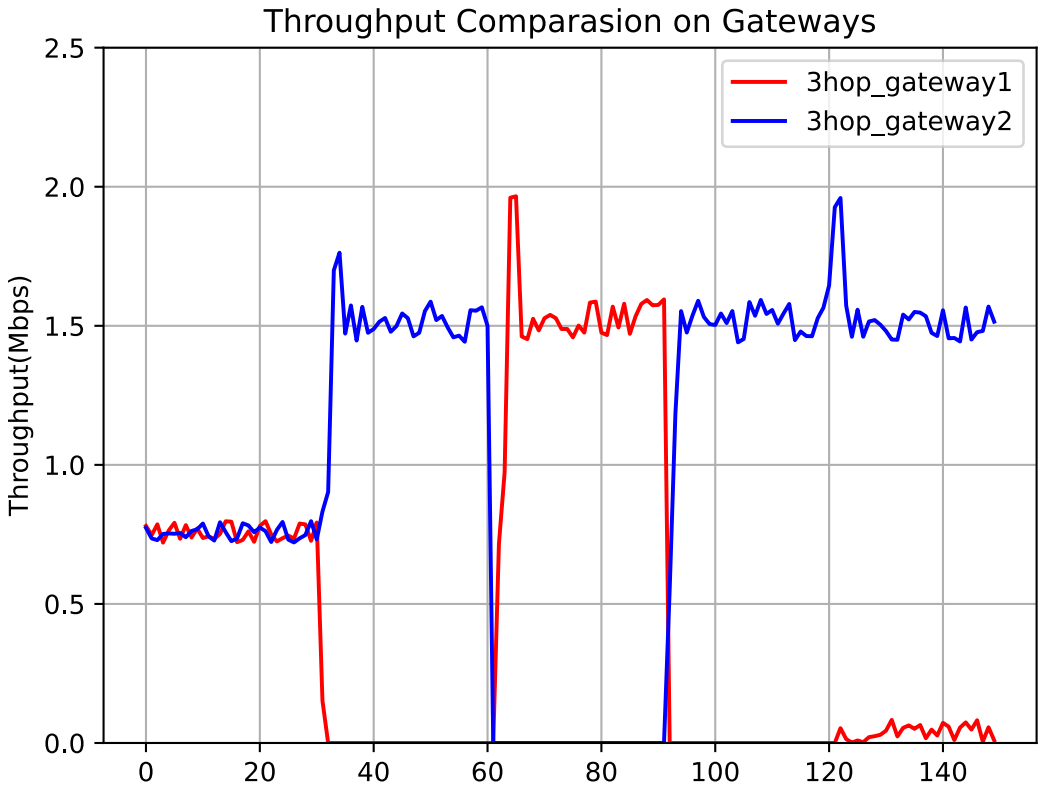


Figure 7.5: Throughput Comparison on Gateways with 3 Hops

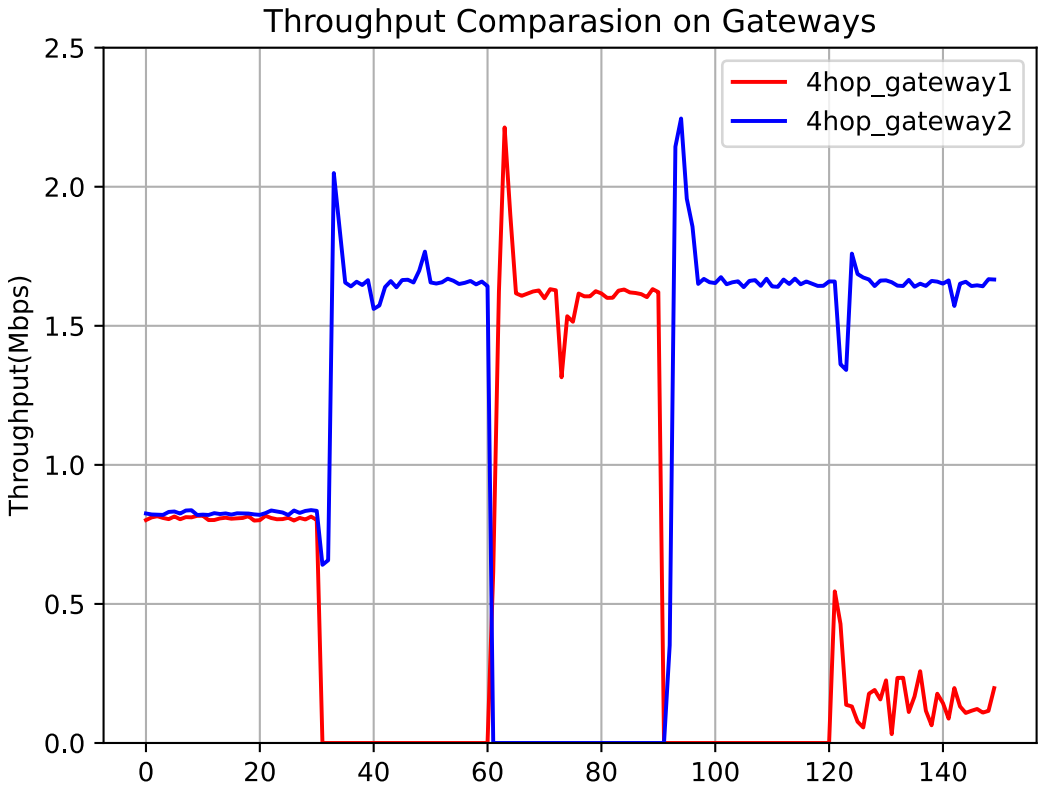


Figure 7.6: Throughput Comparison on Gateways with 4 hops

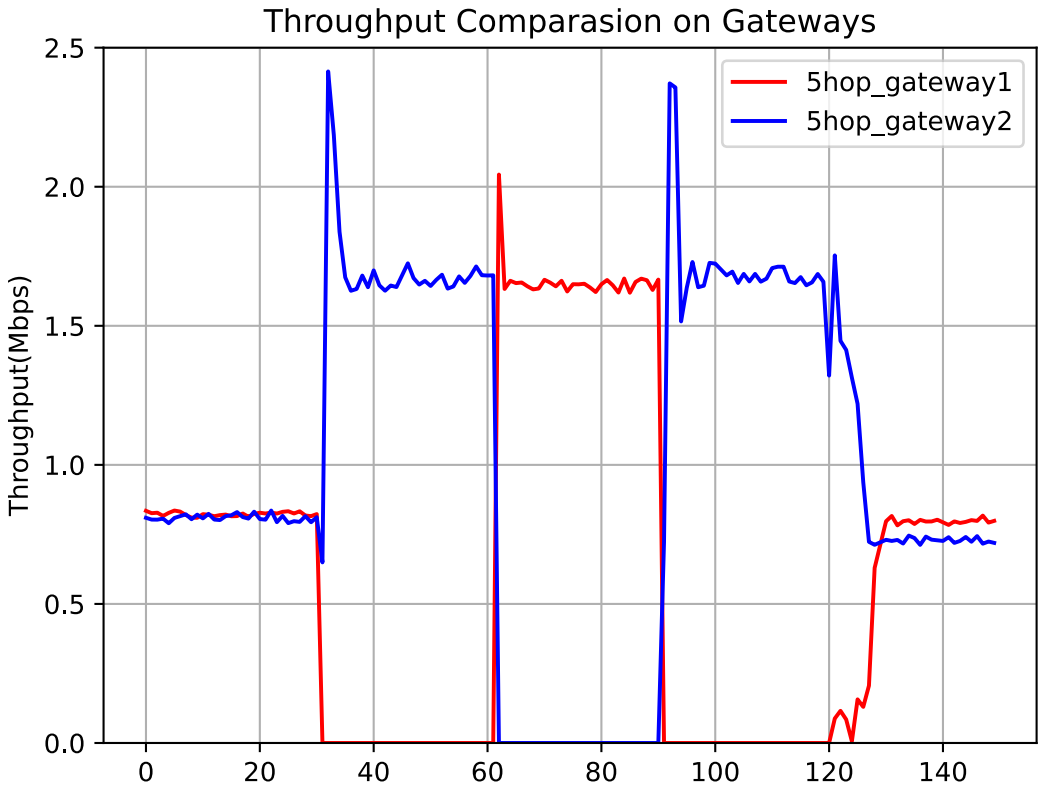


Figure 7.7: Throughput Comparison on Gateways with 5 hops

Chapter 8

Conclusion and Future Work

8.1 Conclusion

The Next Generation First Responders Communication Hubs (NGFR Communication Hubs) represents a cutting-edge and innovative network system architecture, designed to revolutionize communication in extreme and hazardous environments. Our primary goal is to establish a seamless, fault-tolerant, and secure communication infrastructure that can support first responders in their critical missions.

At the core of the NGFR Communication Hubs lies three advanced technologies that work in harmony to enhance network reliability on portable devices. First, we utilize Multi-path TCP (MPTCP), a groundbreaking approach that enables simultaneous data transmission over multiple network paths which ensures that even in the face of network disruptions or failures, communication remains uninterrupted.

Additionally, the Optimized Linked State Routing Protocol (OLSR) plays a vital role in the NGFR Communication Hubs. OLSR is a proactive routing protocol that constantly updates and maintains the network topology, allowing for efficient and dynamic path selection. This proactive nature minimizes delays in data transmission, crucial in time-sensitive scenarios where every second counts.

Furthermore, to facilitate a seamless transition between gateways within the OLSR network, we have introduced a solution that actively retrieves the current gateway information from the routing table. This information is then disseminated to the server using a TCP Fast Open packet. This approach effectively addresses the issue of gateway switching failures by

preserving the existing connection while enabling a rapid and transparent switch between different gateways.

We employ the Message Queuing Telemetry Transport (MQTT) protocol for efficient and lightweight data transfer. This protocol optimizes communication between devices by utilizing a publish-subscribe model, reducing overhead and ensuring rapid data delivery to the intended recipients.

The NGFR Communication Hubs aims to be user-friendly and require minimal user involvement during operation. First responders can focus entirely on their life-saving efforts without worrying about complex network configurations. Through extensive experiments, it shows that out design handles the network's dynamic management, adaptability, and fault tolerance, allowing sensor data to be transmitted on continuous and dependable communication, regardless of the challenges posed by the environment.

In summary, the NGFR Communication Hubs represent an innovative network architecture designed to augment the communication capabilities of first responders in emergency situations. This infrastructure is specifically engineered to scale efficiently in densely populated networks with large amount of devices, even in the highly unstable network conditions. The NGFR Communication Hubs offer support for reliable sensor data delivery in Ad Hoc network, a versatile node discovery scheme, and a decentralized security model, thereby enhancing the resilience and reliability of communication for first responders operating in challenging environments.

We believe that such network architecture designed for emergency and disaster response poses various significant research challenges. Through NGFR Communication Hubs, our aim is to integrate these challenges into a cohesive system that offers routing, addressing, security, and data prioritization. Establishing such an infrastructure is imperative for unlocking the potential advantages associated with these next-generation wireless devices.

8.2 Future Work

8.2.1 Security Challenges

The challenge of securing MANET arises from its self-organized nature and the limitations of conventional security solutions, a topic not extensively addressed in this work. Specifically, any device within the transmission range can join the Communication Hub, participating in the exchange of published or subscribed data. This behavior poses a potential threat[49] to our design, capable of disrupting ongoing communication or compromising the routing tables of other devices. The latter issue presents a significant drawback in our design. While most research assumes that messages containing topological information are sufficiently secured and resistant to compromise[2], our design heavily relies on reading and modifying the routing table. Therefore, it is imperative to consider solutions that ensure the integrity of the routing table.

Many works have proposed solutions dealing with the compromised routing tables. Most of the works are based on cryptography to secure messages containing the topology information essential fro calculating the routing tables. [1] proposed a secured version of OLSR named SOLSR. Their approach relies on the signature and timestamp of each OLSR control message. A signature is generated for each control message and transmitted with the message to thwart malicious nodes from altering or falsifying topology information. Additionally, a timestamp is associated with each signature to estimate the freshness of the message.

8.2.2 Other Multiple-path Based Scheme

Recently, the networking community and the IETF have worked on the design and implementation of the QUIC[4] protocol aiming at providing the services of TCP, TLS and HTTP atop UDP. QUIC is being finalized within the IETF[42]. QUIC is a transport layer network protocol. In addition to HTTP, it may accommodate other types of traffic. An initial design for Multipath QUIC[18] has already been proposed. Similarly to Multipath TCP, Multipath QUIC allows the simultaneous usage of multiple network paths for a given connection. When an application requests a new connection to be established, the Path Manager module in MPQUIC is engaged.

This module is responsible for adding and removing paths during the connection lifetime unrelated to the actual data transfer. As a consequence, MPQUIC engages Stream Scheduler. By default, it runs under a round-robin policy, but the actual policy may be adjusted to specific needs[95].

QUIC is a stream based protocol. Using parallel streams in the same connection to carry different topics from MQTT makes publishing/subscribing process paralleled with different priorities and mitigate the HOL (Head Of Line) blocking issue.Researchers in [52] have studied the performance of MQTT integrating with wireless, wired, and long-distance test-beds constructed using Raspberry Pi 3B devices. The result indicates that MQTT with QUIC surpasses MQTT with TCP in processor usage, memory usage, and latency. It will be another interesting and challenge topic to apply the NGFR Communication Hub with QUIC for a more secured and fast transport layer protocol.

References

[1] Cedric Adjih, Thomas Clausen, Philippe Jacquet, Anis Laouiti, Paul Muhlethaler, and Daniele Raffo. Securing the olsr protocol. In *Proceedings of Med-Hoc-Net*, pages 25–27. Citeseer, 2003.

[2] Asma Adnane, Christophe Bidan, and Rafael Timóteo de Sousa Júnior. Trust-based security for the olsr routing protocol. *Computer Communications*, 36(10-11):1159–1171, 2013.

[3] Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad hoc networks*, 3(3):325–349, 2005.

[4] A. Langley al. The quic transport protocol: Design and internet-scale deployment. *In Proc. ACM SIGCOMM, pp*, pages 183–196, 2017.

[5] Peter Aldhous, Stephanie M Lee, and Zahra Hirji. The texas winter storm and power outages killed hundreds more people than the state says. *BuzzFeed News, May*, 26, 2021.

[6] Mohammed Aljubayri, Tong Peng, and Mohammad Shikh-Bahaei. Reduce delay of multipath tcp in iot networks. *Wireless Networks*, 27:4189–4198, 2021.

[7] Kelvert Ballantyne, Wahab Almuhtadi, and Jordan Melzer. Autoconfiguration for faster wifi community networks. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 938–941, 2015.

[8] Sébastien Barré et al. *Implementation and assessment of modern host-based multipath solutions.* PhD thesis, Catholic University of Louvain, Louvain-la-Neuve, Belgium, 2011.

[9] Sébastien Barré, Christoph Paasch, and Olivier Bonaventure. Multipath tcp: from theory to practice. In *NETWORKING 2011: 10th International IFIP TC 6 Networking Conference, Valencia, Spain, May 9-13, 2011, Proceedings, Part I 10*, pages 444–457. Springer, 2011.

[10] Bharat Bhargava, Xiaoxin Wu, Yi Lu, and Weichao Wang. Integrating heterogeneous wireless technologies: a cellular aided mobile ad hoc network (cama). *Mobile Networks and Applications*, 9:393–408, 2004.

[11] Nabil Bitar, Steven Gringeri, and Tiejun J Xia. Technologies and protocols for data center and cloud networking. *IEEE Communications Magazine*, 51(9):24–31, 2013.

[12] Josh Broch, David A Maltz, and David B Johnson. Supporting hierarchy and heterogeneous interfaces in multi-hop wireless ad hoc networks. In *Proceedings Fourth International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN'99)*, pages 370–375. IEEE, 1999.

[13] Rasa Bruzgiene, Lina Narbutaite, and Tomas Adomkus. Manet network in internet of things system. *Ad hoc networks*, 66:89–114, 2017.

[14] Luomeng Chao, Celimuge Wu, Tsutomu Yoshinaga, Wugedele Bao, and Yusheng Ji. A brief review of multipath tcp for vehicular networks. *Sensors*, 21(8):2793, 2021.

[15] Shanzhi Chen, Hui Xu, Dake Liu, Bo Hu, and Hucheng Wang. A vision of iot: Applications, challenges, and opportunities with china perspective. *IEEE Internet of Things journal*, 1(4):349–359, 2014.

[16] Mahima Chitkara and Mohd Waseem Ahmad. Review on manet: characteristics, challenges, imperatives and routing protocols. *International journal of computer science and mobile computing*, 3(2):432–437, 2014.

[17] Thomas Clausen and Philippe Jacquet. Optimized link state routing protocol (olsr). Technical report, 2003.

[18] Q. De Coninck and O. Bonaventure. Multipath quic: Design and evaluation. In *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*, pages 160–166, 2017.

[19] Xavier Corbillon, Ramon Aparicio-Pardo, Nicolas Kuhn, Géraldine Texier, and Gwendal Simon. Cross-layer scheduler for video streaming over mptcp. In *Proceedings of the 7th International Conference on Multimedia Systems*, pages 1–12, 2016.

[20] James H Cowie, Andy T Ogielski, B Premore, Eric A Smith, and Todd Underwood. Impact of the 2003 blackouts on internet communications. *Preliminary Report, Renesys Corporation (updated March 1, 2004)*, page 9, 2003.

[21] Yong Cui, Hongyi Wang, Xiuzhen Cheng, and Biao Chen. Wireless data center networking. *IEEE Wireless Communications*, 18(6):46–53, 2011.

[22] Michael M Danziger, Amir Bashan, Yehiel Berezin, Louis M Shekhtman, and Shlomo Havlin. An introduction to interdependent networks. In *Nonlinear Dynamics of Electronic Systems: 22nd International Conference, NDES 2014, Albena, Bulgaria, July 4-6, 2014. Proceedings 22*, pages 189–202. Springer, 2014.

[23] Douglas SJ De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 134–146, 2003.

[24] Shuo Deng, Ravi Netravali, Anirudh Sivaraman, and Hari Balakrishnan. Wifi, lte, or both? measuring multi-homed wireless internet performance. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, IMC '14, page 181–194, New York, NY, USA, 2014. Association for Computing Machinery.

[25] Thomas Dreibholz, Robin Seggelmann, Michael Tüxen, and Erwin Rathgeb. Transmission scheduling optimizations for concurrent multipath transfer. 11 2010.

[26] Sisi Duan, Sangkeun Lee, Supriya Chinthavali, and Mallikarjun Shankar. Reliable communication models in interdependent critical infrastructure networks. In *2016 Resilience Week (RWS)*, pages 152–157, 2016.

[27] Benevid Felix, Igor Steuck, Aldri Santos, Stefano Secci, and Michele Nogueira. Redundant packet scheduling by uncorrelated paths in heterogeneous wireless networks. In *2018 IEEE Symposium on Computers and Communications (ISCC)*, pages 00498–00503, 2018.

[28] Federal Emergency Management Agency (FEMA). Hurricane sandy in new jersey and new york: building performance observations, recommendations, and technical guidance. *Mitig. Assess. Team Rep.*, pages p–223, 2013.

[29] Alan Ford, Costin Raiciu, Mark J. Handley, and Olivier Bonaventure. TCP Extensions for Multipath Operation with Multiple Addresses. RFC 6824, January 2013.

[30] Alan Ford, Costin Raiciu, Mark J. Handley, Olivier Bonaventure, and Christoph Paasch. TCP Extensions for Multipath Operation with Multiple Addresses. RFC 8684, March 2020.

[31] Alexander Frommgen, Tobias Erbshäußer, Alejandro Buchmann, Torsten Zimmermann, and Klaus Wehrle. Remp tcp: Low latency multipath tcp. In *2016 IEEE international conference on communications (ICC)*, pages 1–7. IEEE, 2016.

[32] José Antonio Galache, Takuro Yonezawa, Levent Gurgen, Daniele Pavia, Marco Grella, and Hiroyuki Maeomichi. Clout: Leveraging cloud computing techniques for improving management of massive iot data. In *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications*, pages 324–327. IEEE, 2014.

[33] Carles Gomez, D Garcia, and Josep Paradells. Improving performance of a real ad-hoc network by tuning olsr parameters. In *10th IEEE Symposium on Computers and Communications (ISCC'05)*, pages 16–21. IEEE, 2005.

[34] Christian Gottron, André König, Matthias Hollick, Sonja Bergsträßer, Tomas Hildebrandt, and Ralf Steinmetz. Quality of experience of voice communication in large-scale mobile ad hoc networks. In *2009 2nd IFIP Wireless Days (WD)*, pages 1–6, 2009.

[35] Pejman Goudarzi and Mehdi Hosseinpour. Qoe enhancement for video transmission over manets using distortion minimization. *Scientia Iranica*, 19(3):696–706, 2012.

[36] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.

[37] Vehbi C. Gungor and Gerhard P. Hancke. Industrial wireless sensor networks: Challenges, design principles, and technical approaches. *IEEE Transactions on Industrial Electronics*, 56(10):4258–4265, 2009.

[38] Alex Hinds, Michael Ngulube, Shaoying Zhu, and Hussain Al-Aqrabi. A review of routing protocols for mobile ad-hoc networks (manet). *International journal of information and education technology*, 3(1):1, 2013.

[39] Cao Huang, Xiaojun Guo, and Zeguo Liu. A wireless networking architecture using manet for mobile communications of remote pastoral areas in tibet. In *Conference of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)*, pages 800–803. Atlantis Press, 2013.

[40] Urs Hunkeler, Hong Linh Truong, and Andy Stanford-Clark. Mqtt-s—a publish/subscribe protocol for wireless sensor networks. In *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*, pages 791–798. IEEE, 2008.

[41] Jaehyun Hwang and Joon Yoo. Packet scheduling for multipath tcp. In *2015 Seventh international conference on ubiquitous and future networks*, pages 177–179. IEEE, 2015.

[42] J. Iyengar and M. Thomson. Quic: A udp-based multiplexed and secure transport. *Internet Engineering Task Force Internet-Draft draft-ietf-quic-transport-*, 20, April 2019.

[43] Jana Iyengar, Costin Raiciu, Sebastien Barre, Mark J. Handley, and Alan Ford. Architectural Guidelines for Multipath TCP Development. RFC 6182.

[44] Elizabeth Jacob and P Sivraj. Performance analysis of manet routing protocols in smart city message passing. In *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1255–1260. IEEE, 2016.

[45] Philippe Jacquet, Paul Muhlethaler, Thomas Clausen, Anis Laouiti, Amir Qayyum, and Laurent Viennot. Optimized link state routing protocol for ad hoc networks. In *Proceedings. IEEE International Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century.*, pages 62–68. IEEE, 2001.

[46] Chris W Johnson. Analysing the causes of the italian and swiss blackout, 28th september 2003. In *Proceedings of the 12th Australian Workshop on Safety Critical Systems and Software-Related Programmable Systems, Adelaide, Australia*, pages 21–30. Citeseer, 2007.

[47] David Johnson, Yin-chun Hu, and David Maltz. The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4. Technical report, 2007.

[48] Ulf Jonsson, Fredrik Alriksson, Tony Larsson, Per Johansson, and Gerald Q Maguire. Mipmanet-mobile ip for mobile ad hoc networks. In *2000 First Annual Workshop on Mobile and Ad Hoc Networking and Computing. MobiHOC (Cat. No. 00EX444)*, pages 75–85. IEEE, 2000.

[49] Hamela Kanagasundaram and A Kathirvel. Eimo-esolsr: energy efficient and security-based model for olsr routing protocol in mobile ad-hoc network. *IET Communications*, 13(5):553–559, 2019.

[50] Gunseerat Kaur and Poonam Thakur. Routing protocols in manet: An overview. In *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, volume 1, pages 935–941. IEEE, 2019.

[51] Han Ah Kim, Bong hwan Oh, and Jaiyong Lee. Improvement of mptcp performance in heterogeneous network using packet scheduling mechanism. In *2012 18th Asia-Pacific Conference on Communications (APCC)*, pages 842–847, 2012.

[52] Puneet Kumar and Behnam Dezfouli. Implementation and analysis of quic for mqtt. *Computer Networks*, 150:28–45, 2019.

[53] M Kuzlu, M Pipattanasomporn, and S Rahman. Review of communication technologies for smart homes/building applications. In *2015 IEEE Innovative Smart Grid Technologies-Asia (ISGT ASIA)*, pages 1–6. IEEE, 2015.

[54] Li Li, Ke Xu, Tong Li, Kai Zheng, Chunyi Peng, Dan Wang, Xiangxiang Wang, Meng Shen, and Rashid Mijumbi. A measurement study on multi-path tcp with multiple cellular carriers on high speed rails. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 161–175, 2018.

[55] Tingli Li, Yang Liu, Ye Tian, Shuo Shen, and Wei Mao. A storage solution for massive iot data based on nosql. In *2012 IEEE International conference on green computing and communications*, pages 50–57. IEEE, 2012.

[56] Yao-Nan Lien, Hung-Chin Jang, and Tzu-Chieh Tsai. A manet based emergency communication and information system for catastrophic natural disasters. In *2009 29th IEEE international conference on distributed computing systems workshops*, pages 412–417. IEEE, 2009.

[57] Yeon-sup Lim, Erich M Nahum, Don Towsley, and Richard J Gibbens. Ecf: An mptcp path scheduler to manage heterogeneous paths. In *Proceedings of the 13th international conference on emerging networking experiments and technologies*, pages 147–159, 2017.

[58] Jianwei Liu, Anjan Rayamajhi, and James Martin. Using mptcp subflow association control for heterogeneous wireless network optimization. In *2016 14th International*

Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), pages 1–8, 2016.

[59] Tao Liu and Alberto E Cerpa. Data-driven link quality prediction using link features. *ACM Transactions on Sensor Networks (TOSN)*, 10(2):1–35, 2014.

[60] Igor Lopez, Marina Aguado, Christian Pinedo, and Eduardo Jacob. Scada systems in the railway domain: enhancing reliability through redundant multipathtcp. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 2305–2310. IEEE, 2015.

[61] Youzhong Ma, Jia Rao, Weisong Hu, Xiaofeng Meng, Xu Han, Yu Zhang, Yunpeng Chai, and Chunqiu Liu. An efficient index for massive iot data in cloud environment. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2129–2133, 2012.

[62] Imtiaz Mahmud, Tabassum Lubna, and You-Ze Cho. Performance evaluation of mptcp on simultaneous use of 5g and 4g networks. *Sensors*, 22(19):7509, 2022.

[63] David J Malan, Thaddeus Fulford-Jones, Matt Welsh, and Steve Moulton. Codeblue: An ad hoc sensor network infrastructure for emergency medical care. In *International workshop on wearable and implantable body sensor networks*, 2004.

[64] Devesh Malik, Krishna Mahajan, and MA Rizvi. Security for node isolation attack on olsr by modifying mpr selection process. In *2014 First International Conference on Networks & Soft Computing (ICNSC2014)*, pages 102–106. IEEE, 2014.

[65] R Manoharan and S Mohanalakshmie. A trust based gateway selection scheme for integration of manet with internet. In *2011 International Conference on Recent Trends in Information Technology (ICRTIT)*, pages 543–548. IEEE, 2011.

[66] Shima Mohseni, Rosilah Hassan, Ahmed Patel, and Rozilawati Razali. Comparative review study of reactive and proactive routing protocols in manets. In *4th IEEE International Conference on Digital Ecosystems and Technologies*, pages 304–309, 2010.

[67] A Muir and J Lopatto. Final report on the august 14, 2003 blackout in the united states and canada: causes and recommendations. 2004.

[68] Dang Nguyen and Pascale Minet. Analysis of mpr selection in the olsr protocol. In *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, volume 2, pages 887–892. IEEE, 2007.

[69] E Onwuka, A Folaponmile, and M Ahmed. Manet: A reliable network in disaster areas. *Jorind*, 9(2):105–113, 2011.

[70] Christoph Paasch, Simone Ferlin, Ozgu Alay, and Olivier Bonaventure. Experimental evaluation of multipath tcp schedulers. In *Proceedings of the 2014 ACM SIGCOMM workshop on Capacity sharing workshop*, pages 27–32, 2014.

[71] Christoph Paasch, Ramin Khalili, and Olivier Bonaventure. On the benefits of applying experimental design to improve multipath tcp. In *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, pages 393–398, 2013.

[72] Charles Perkins, Elizabeth Belding-Royer, and Samir Das. Ad hoc on-demand distance vector (aodv) routing. Technical report, 2003.

[73] Charles E Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *ACM SIGCOMM computer communication review*, 24(4):234–244, 1994.

[74] Sefali Prajapati, Nimisha Patel, and Rajan Patel. Optimizing performance of olsr protocol using energy based mpr selection in manet. In *2015 Fifth International Conference on Communication Systems and Network Technologies*, pages 268–272. IEEE, 2015.

[75] Tie Qiu, Ning Chen, Keqiu Li, Mohammed Atiquzzaman, and Wenbing Zhao. How can heterogeneous internet of things build our future: A survey. *IEEE Communications Surveys Tutorials*, 20(3):2011–2027, 2018.

[76] Sivasankar Radhakrishnan, Yuchung Cheng, Jerry Chu, Arvind Jain, and Barath Raghavan. Tcp fast open. In *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, pages 1–12, 2011.

[77] Costin Raiciu, Sebastien Barre, Christopher Pluntke, Adam Greenhalgh, Damon Wischik, and Mark Handley. Improving datacenter performance and robustness with multipath tcp. *ACM SIGCOMM Computer Communication Review*, 41(4):266–277, 2011.

[78] Costin Raiciu, Mark Handley, and Damon Wischik. Coupled congestion control for multipath transport protocols. Technical report, 2011.

[79] Costin Raiciu, Christoph Paasch, Sebastien Barre, Alan Ford, Michio Honda, Fabien Duchene, Olivier Bonaventure, and Mark Handley. How hard can it be? designing and implementing a deployable multipath TCP. In *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, pages 399–412, San Jose, CA, April 2012. USENIX Association.

[80] V. Rajeshkumar and P. Sivakumar. Comparative study of aodv dsdv and dsr routing protocols in manet using network simulator-2. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(12):2319–5940, 2013.

[81] Dipankar Raychaudhuri and Narayan B Mandayam. Frontiers of wireless and mobile communications. *Proceedings of the IEEE*, 100(4):824–840, 2012.

[82] Timothy Rooney. *Introduction to IP address management*, volume 17. John Wiley & Sons, 2010.

[83] Radhika Ranjan Roy. *Handbook of mobile ad hoc networks for mobility models*, volume 170. Springer, 2011.

[84] Muge Sayit, Erdem Karayer, Chi-Dung Phung, Stefano Secci, and Selma Boumerdassi. Numerical evaluation of mptcp schedulers in terms of throughput and reliability. In *2019 11th International Workshop on Resilient Networks Design and Modeling (RNDM)*, pages 1–6, 2019.

[85] Cigdem Sengul and Anthony Kirby. Message Queuing Telemetry Transport (MQTT) and Transport Layer Security (TLS) Profile of Authentication and Authorization for Constrained Environments (ACE) Framework. RFC 9431, July 2023.

[86] Varun Kumar Sharma, Lal Pratap Verma, and Mahesh Kumar. Cl-adsp: Cross-layer adaptive data scheduling policy in mobile ad-hoc networks. *Future Generation Computer Systems*, 97:530–563, 2019.

[87] Hang Shi, Yong Cui, Xin Wang, Yuming Hu, Minglong Dai, Fanzhao Wang, and Kai Zheng. {STMS}: Improving {MPTCP} throughput under heterogeneous networks. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, pages 719–730, 2018.

[88] Bhagya Nathali Silva, Murad Khan, and Kijun Han. Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. *Sustainable cities and society*, 38:697–713, 2018.

[89] Dhananjay Singh, Gaurav Tripathi, and Antonio J Jara. A survey of internet-of-things: Future vision, architecture, challenges and services. In *2014 IEEE world forum on Internet of Things (WF-IoT)*, pages 287–292. IEEE, 2014.

[90] Erik Sy, Tobias Mueller, Christian Burkert, Hannes Federrath, and Mathias Fischer. Enhanced performance and privacy for tls over tcp fast open. *arXiv preprint arXiv:1905.03518*, 2019.

[91] Ye Tian, Kai Xu, and Nirwan Ansari. Tcp in wireless environments: problems and solutions. *IEEE Communications Magazine*, 43(3):S27–S32, 2005.

[92] C.K. Toh. *Wireless ATM and Ad-Hoc Networks: Protocols and Architectures*. Springer US, 1997.

[93] Y Tseng. Mobile ip and ad hoc networks: An integration and implementation experience. *IEEE Computer*, 2003.

[94] M. Tubaishat and S. Madria. Sensor networks: an overview. *IEEE Potentials*, 22(2):20–23, 2003.

[95] Jing Wang, Yunfeng Gao, and Chenren Xu. A multipath quic scheduler for mobile http/2. In *Proceedings of the 3rd Asia-Pacific Workshop on Networking*, APNet '19, page 43–49, New York, NY, USA, 2019. Association for Computing Machinery.

[96] Hongyi Wu, Chunming Qiao, S. De, and O. Tonguz. Integrated cellular and ad hoc relaying systems: icar. *IEEE Journal on Selected Areas in Communications*, 19(10):2105–2115, 2001.

[97] J. Wu, C. Yuen, B. Cheng, M. Wang, and J. Chen. Streaming high-quality mobile video with multipath tcp in heterogeneous wireless networks. *in IEEE Transactions on Mobile Computing*, 15(9):1, September 2016.

[98] Yitao Xing, Kaiping Xue, Yuan Zhang, Jiangping Han, Jian Li, Jianqing Liu, and Ruidong Li. A low-latency mptcp scheduler for live video streaming in mobile networks. *IEEE Transactions on Wireless Communications*, 20(11):7230–7242, 2021.

[99] Kaiping Xue, Jiangping Han, Hong Zhang, Ke Chen, and Peilin Hong. Migrating unfairness among subflows in mptcp with network coding for wired–wireless networks. *IEEE Transactions on Vehicular Technology*, 66(1):798–809, 2016.

[100] B. Yu, W. Zhang, and L. Li. Design and implementation of ad-hoc dynamic gateway based on olsr and mobile ip. *Wireless Communications Networking and Mobile Computing*, 5:1–4, 2009.

[101] Tongguang Zhang, Shuai Zhao, Yulong Shi, Bingfei Ren, Bo Cheng, and Junliang Chen. The implementation of improved mptcp in manets. In *2017 IEEE 25th International Conference on Network Protocols (ICNP)*, pages 1–2, 2017.

[102] Jun Zheng, David Simplot-Ryl, Chatschik Bisdikian, and Hussein T Mouftah. The internet of things [guest editorial]. *IEEE Communications Magazine*, 49(11):30–31, 2011.