Enhancements to the Multipath Transmission Control Protocol for Internet of Things Wireless Networks

Aljubayri, Mohammed

Awarding institution:
King's College London

Enhancements to the Multipath Transmission Control Protocol for Internet of Things Wireless Networks

Mohammed Aljubayri

Engineering Department

King's College London

A thesis submitted to King's College London in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

January 2022

I would like to dedicate this thesis to my loving parents, my brother and my sisters who supported me during my PhD study. I also would like to dedicate this thesis to my super, kind, respectful and wonderful wife and children who helped me a lot. From my heart, I say to all of you THANK YOU VERY MUCH.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university.

Mohammed Aljubayri

January 2022

Acknowledgements

I wish to express my thanks to my first supervisor Prof. Mohammad Shikh-Bahaei for his guidance during my study. His knowledge and support were unlimited and highly appreciated. I also would like to thank my second supervisor Prof. Mischa Dohler for his assistance and valuable time. Finally, thanks to all of my friends and KCL colleagues for their cooperation.

Abstract

The main theme of this thesis is about enhancing the performance of the Multipath Transmission Control Protocol (MPTCP) in Internet of Things (IoT) wireless networks. MPTCP is a transport layer protocol that simultaneously transmits data through multiple paths using several network interfaces (e.g., Wi-Fi, 5G and Ethernet). This has attracted a lot of research in industry and academia. However, the next generation IoT applications are expected to have strict Quality of Service (QoS) requirements. Traditional MPTCP is not able to satisfy such requirements, therefore, there is an opportunity to improve the protocol.

This thesis first proposes a cross-layer design to enhance MPTCP. A network utility maximisation framework is formulated with joint congestion control, routing, and scheduling design for ad-hoc networks. The objective of the optimisation problem is to maximise user utility, subject to source rate, scheduling, and queuing delay constraints. This joint optimisation problem is solved using the Lagrangian and two scheduling algorithms, specifically perfect scheduling and distributed scheduling, are proposed. The stability and convergence of the above algorithms are proved in fixed channels and time-varying channels scenarios. The results reveal that the proposed multipath algorithms outperform the existing schemes in terms of improving source rate and moderating congestion price.

The thesis also compares several variants of MPTCP to determine the best protocol in wireless IoT networks environments. The candidates are conventional MPTCP, MPTCP-TSC (Traffic Split Control) and ReMP TCP (redundant

MPTCP). ReMP TCP provided the lowest number of transmissions and the lowest delay. To further diminish delay, the thesis introduces a routing technique, called Opportunistic Routing (OR), to the above protocols. OR is a networking protocol in which the traffic is broadcast to all wireless nodes that can hear the transmission. Hence, the reliability of correct data transmission in a network is increased. The simulation results show that all OR-based MPTCP schemes are superior to the schemes without OR. Also, the most efficient joint selection is OR with ReMP TCP.

To reduce the latency of MPTCP further and improve its reliability, the thesis investigates the feasibility of applying Full Duplex (FD) technology to MPTCP in Internet of Vehicles (IoV) networks. FD-capable devices can transmit and receive on the same radio frequency bandwidth concurrently. This property is exploited by the MPTCP protocol; and a new design, called the FD-based multi-path transmission control protocol (FDMP), is proposed for ultra-reliable low-latency communication (URLLC) applications. FDMP incorporates a modified scheduler and congestion control mechanism, as well as a proactive acknowledgment (ACK) mechanism and a novel re-transmission strategy. The simulation results demonstrate that FDMP outperforms the benchmark MPTCP and FD is feasible.

Abbreviations

1G	First Generation
ACK	Acknowledgment
AIMD	Additive Increase Multiplicative Decrease
ALM	Adaptive Loss Management
AR/VR	Augmented Reality/Virtual Reality
AT&T	American Telephone and Telegraph
B5G-IoV	Beyond 5G Internet of Vehicles
BALIA	Balanced Linked Adaptation
BBR	Bottleneck Bandwidth and Round-trip propagation time
BCCPS	BBR-Based Congestion Control and Packet Scheduling
BLEST	Blocking Estimation
CDMA	Code Division Multiple Access
CMT-SCTP	Concurrent Multipath Transfer Using Stream Control Transmission Protocol
CSMA	Carrier Sense Multiple Access
cTCP	concurrent TCP
CWND	Congestion Window
DBCCA	Delay-Based Congestion Control Algorithm
DEFT	Delay-Equalized FAST
DEAM	Delay-Energy-quality-Aware MPTCP
DMPTCP	Dynamic MPTCP
ECN	Explicit Congestion Notification
eNMCC	Extended Normalized Multi-flow Congestion Control

FD	Full Duplex
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
FDMP	FD based multi-path transmission control protocol
FEC	Forward error correction
FER	Frame Error Rate
FLRA	Flow Level Resource Allocation
GSAM	Gentle Slow Start Algorithm proposed for MPTCP
GMSK	Gaussian Minimum Shift Keying
GPRS	General Packet Radio Service
HD	Half Duplex
HoL	Head of Line
HSPA	High Speed Packet Access
HSR	Highest Sending Rate
IETF	Internet Engineering Task Force
IMT	International Mobile Telecommunications
IoT	Internet of Things
IoV	Internet of Vehicles
IP	Internet Protocol
LIA	Linked Increases Algorithm
LL	Lowest Latency
LOS	Line-of-Sight
LTE	Long-Term Evolution
LTS	Lowest Time/Space
LWS	Largest Window Space
M2M	Machine-To-Machine
MAC	Medium Access Control

mMTC	Massive Machine-Type Communication
MPTCP	Multi-Path Transmission Control Protocol
MPTCP-TOASF	MPTCP Transmission Optimization Algorithm for Short Flows
MPTCP-TSC	MPTCP-Traffic Split Control
NAT	Network Address Translation
NC	Network Coding
NMCC	Normalized Multi-flow Congestion Control
NUM	Network Utility Maximization
OBR	Offloading By Restriction
OFDMA	Orthogonal Frequency Division Multiple Access
OLB	Optimal Load Balancing
OLIA	Opportunistic Linked Increases Algorithm
OQPSK	Offset Quadrature Phase Shift Keying
OR	Opportunistic routing
OSI	Open Systems Interconnection
OSM	Open Street Map
PDR	Packet Delivery Ratio
PHY	Physical layer
PNC	Pipeline Network Coding
PPP	Poisson Point Process
PRR	Proportional Rate Reduction
PRN	Pass Route Number
PS	Packet Size
pTCP	Parallel TCP
QoS	Quality of Service
RED	Redundant Diversity scheduling

ReMP TCP	Redundant MPTCP
RF	Radio Frequency
RLC	Random Linear network Coding
RPC	Re-attempt Path Counter
RPS	Random Packet Spraying
RTC	Re-attempt Time Counter
RTO	Retransmission Timeout
RTT	Round-Trip Time
SB-CC	Shared Bottleneck based Congestion Control
SB-FPS	Shared Bottleneck based Forward Prediction packet Scheduling
SCFDMA	Single Carrier Frequency Division Multiple Access
SCTP	Stream Control Transport Protocol
S-EDPF	Stochastic Earliest Delivery Path First
SERO	Seamless Roaming
SI	Self-interference
SIS	Self-Interference Suppression
sRTT	Smoothed RTT
STTF	Shortest Transmission Time First
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TSC	Traffic Split Control
UDP	User Datagram Protocol
UE	User Equipment
UMW	Universal Control Policy

URLLC	Ultra-Reliable Low-Latency Communication
V2X	Vehicle to Everything
VANET	Vehicular Adhoc Network
VUE	Vehicle User Equipment
WANs	Wide Area Networks
WSN	Wireless Sensors Network

Mathematical Symbols:

P	Probability of successful packet delivery from source to destination
w	Congestion window size
w_n	Congestion window size of path n
w_{total}	Sum of congestion windows
RTT_n	Round-Trip Time on the n^{th} path
λ	Smoothing factor
$U_s(x_s)$	User utility at source s while the sending rate is x_s
c_l	Link capacity
S	Number of sources
L	Number of link
L_f	Number of wired link
L_w	Number of wireless link
p_s	Weight factor
$J(s)$	Number of transmission paths of source s
$L(s)$	Set of links used by source s
$S(l)$	Set of sources using link l
$J(l)$	Set of transmission paths using link l
x^i_{sj}	Data rate of source s, sub-flow j on link i
m_s	Minimal data rate
M_s	Maximal data rate
ε l	Rate-outage probability threshold of link l
P^L_r	Rate outage probability of link l

H_{sj}	Number of hops traversed by source s on path j
Pr	Probability of the event
b_h	Dimentional rate vector
b_h^l	Dimentional rate vector for link l
$f_{i,j}^k$	The i,j link capacity devoted to the flow to destination k
$f_{i,j,z}^k$	Aggregate capacity of all path
Z	Set of all path
x_i^k	Network flow that is produced at node i to destination k
m	Number of servers
λ_0	Arrival rate
μ_0	Service time
L_q	Queue size
$\bar{O}$	Average number of hops
v	Concurrent transmission ratio
G	Concurrent transmission capacity
$\bar{D}_i$	Queuing delay per hop
p_i^k	Lagrange multiplier for node i to destination k
$d_{i,j}$	Destination k differential congestion price over nodes i,j
γ	Positive scalar step size
DU^*	Optimal target
p^*	Optimal price
L^{und}	Set of undirected links
D^{und}	Set of undirected links weight
$e(t)$	State of the channel in time slot t
R	Feasible rate region
V	Lyapunov function
T	End-to-end delay

TD	Transmission delay
TR	Transmission rate
P_d^{bt}	Detection probability before transmission
P_f^{bt}	False alarm probability before transmission
P_d^{dt}	Probability of detection
P_f^{dt}	False alarm probability
Υ_{si}	Energy measurement of the Self-Interference (SI) signal
η	SI suppression factor

Table of contents

List of figures

List of tables

Author's Contributions

(*i*) M. Aljubayri, Z. Yang and M. Shikh-Bahaei, "Cross-layer multipath congestion control, routing and scheduling design in ad hoc wireless networks", *IET Communications*, vol. 15, no. 8, pp. 1096-1108, 2021. Available: 10.1049/cmu2.12145.

(*ii*) M. Aljubayri, T. Peng and M. Shikh-Bahaei, "Reduce delay of multipath TCP in IoT networks", *Wireless Networks*, vol. 27, no. 6, pp. 4189-4198, 2021. Available: 10.1007/s11276-021-02701-3.

(*iii*) J. Zang, M. Aljubayri, W. Ding and M. Shikh-Bahaei, "Feasibility Analysis of Full Duplex Technology in Enhancing MPTCP for URLLC Applications in B5G-IoV Networks", *IEEE Transactions on Vehicular Technology, under review*, 2021.

Chapter 1

Introduction

1.1 Evolution of Wireless Systems

Since the dawn of humanity, it was a target of mankind to be able to communicate over long distances. Smoke signals and pigeon post are early manifestation of this desire. With the formulation of the mathematical theory of electromagnetic waves by Maxwell in 1873, a journey of wireless communications started. Marconi made a historic milestone by realising radio wave transatlantic communications in 1901. Before the end of the 1940s, AM, FM and TV communication systems became available. Satellite communication, which started in the late 50s, was a major step in the evolution of wireless communications [2–4].

Probably the most important breakthrough for contemporary mobile communication systems was the introduction of the concept of cellular mobile communication by AT&T (American Telephone and Telegraph) Bell labs in the 1970s. This is now considered the first generation (1G) of cellular mobile communication systems, which are currently progressing towards 6G. The 1G and 2G systems provided only voice services. The evolved generation of 2G (also known as 2.5G) added packet data services. Nowadays, 5G data services dominate the wireless communication systems [5]. Table 1.1 summarizes the evolution of the wireless mobile communications systems over time, and shows the main technologies used in each system [2][3][4][5].

While the cellular mobile communication systems were developing, other types of wireless networks were also being developed. These data-oriented networks are listed in Table 1.2 [6]; namely Wi-Fi, Bluetooth, Zigbee and WiMAX [6]. Wi-Fi is a wireless local area network that is oriented towards low cost, small distance and high data rate. Bluetooth is oriented towards low power, small distance, low data rate and low-cost peer-to-peer networks. Zigbee is an industrial ad-hoc wireless sensor network that is designed for extremely low power and low data rate and very long battery life. WiMAX is a metropolitan area network that is intended for fixed or low mobility data applications and Internet services. WiMAX can probably be used to provide Internet service in isolated areas.

Currently, new wireless devices, such as laptop computers, smart phones and hand-held tablets have access to at least two of the above-mentioned wireless networks, therefore, they are *multihomed* devices. Multihoming is when the communication terminal has multiple network interfaces available for transmission [7]. By default, the device connects to one of the wireless networks and keeps the other network as a backup. Another access strategy is to let each wireless network serve a different running application in the device. In contrast to these conventional methods, the main theme of this thesis is the use of multiple wireless networks to provide multiple traffic flows simultaneously for the data of a specific application.

Table 1.1 : Evolution of the dominant cellular mobile communication systems

Gen	**Year intro**	**Name**	**Technology**	**Date rate (max, down link)**
1G	1970	AMPS, NMT and TACS	FM, FDD and FDMA	Voice only
2G	1990	GSM (Europe)	GMSK, FDD and TDMA	Voice only
2G	1993	IS-95A (US)	OQPSK, FDD, CDMA	Voice only
2.5G	2001	GPRS & EDGE	GMSK and 8-PSK, FDD and TDMA	172.2 kbps / 384 kbps
2.5G	1996	IS-95B (US)	OQPSK, FDD, CDMA	64 kbps
3G	2001	WCDMA (Europe)	QPSK, FDD, CDMA	2.3 Mbps
3G	2001	CDMA 2000 (US)	FDD, CDMA	2.457 Mbps
3.5G	2007	HSPA (Europe)	Up to 64-QAM, FDD, CDMA	56 Mbps
3.5G	2006	1x-EVDO (US)	Up to 16-QAM, FDD, CDMA	46.5 Mbps
4G	2010	LTE	OFDMA & SCFDMA	150 Mbps
5G	2020	5G New Radio	OFDMA & SCFDMA	More than 1 Gbps

Table 1.2 : Dominant data oriented wireless networks

Network	**Standard**	**Maximum data rate**
Wi-Fi	IEEE 802.11ac	1.73 Gbps
Wi-Fi	IEEE 802.11n	600 Mbps
Wi-Fi	IEEE 802.11g, IEEE 802.11a	54 Mbps
Bluetooth	IEEE 802.15.1	3 Mbps
Zigbee	IEEE 802.15.4	250 kbps
WiMAX	IEEE 802.16m	365 Mbps

1.2 Wireless IoT Networks

In the past few decades, the Internet has mainly been used for people to exchange news, information, literature, entertainment material and other data. Gradually, the Internet landscape has changed to include connectivity to everything, i.e., the Internet of Things (IoT). The reason for this change includes [8]:

- High speed Internet is available everywhere with a variety of wireless network types.
- Smart IoT devices are widely available, including smart phones, all types of smart sensors, life-saving and medical equipment, machines, actuators and even animals with injectable ID chips.
- Cloud-based information infrastructure is widely available, including cloud computing, cloud storage, security, data banks, display servers and many other cloud services.

The IoT is a network of physical components (i.e., things) like sensors, motors (large or small) smart phones, power switches, etc. These 'things' communicate together to achieve some functionality using the Internet as a communication platform. As a simple home application for IoT, one can imagine a refrigerator that senses the lack of a bottle of milk in the designated place on the shelf. The refrigerator is connected to the Internet through an IoT communication node employing Wi-Fi or Bluetooth. The refrigerator may send a message to the head of the household to remind him/her to buy milk. The refrigerator may also be configured to directly make the order to the local store. While the above example is a simple one for home use, the same concept is also used for more serious industrial applications. Sensors may sense mal-functioning parts, missing parts, parts that need maintenance or attention or even some hazardous situation. These sensors, which have Internet connection, send the information with the appropriate data to the relevant parties to take the needed actions.

Figure 1.1 shows an example of only some of the components of the wireless IoT network. As mentioned above, the IoT devices can be any device or sensor at home, office, street, vehicle, hospital, factory, etc. The IMT 2020 (International Mobile Telecommunications, targets for 2020) requires 5G to support one million device connections per square metre [9]. Consequently, one of the 3 major features of 5G is the support of massive Machine-Type Communication (mMTC) [10]. Therefore, the IoT is anticipated to generate a huge amount of data traffic in wireless networks. Many IoT applications are expected to have stringent Quality of Service (QoS) requirements in terms of throughput and latency. Examples of these applications include, but are not limited to, safety and health-related applications and high precision industrial automation applications. This thesis provides solutions to improve these QoS metrics through optimised utilisation of multiple parallel data flows system.

1.3 Protocol Layers

In data networks, the term 'protocol' refers to the set of rules and tasks that the source, destination and any possible intermediate devices (relays) should

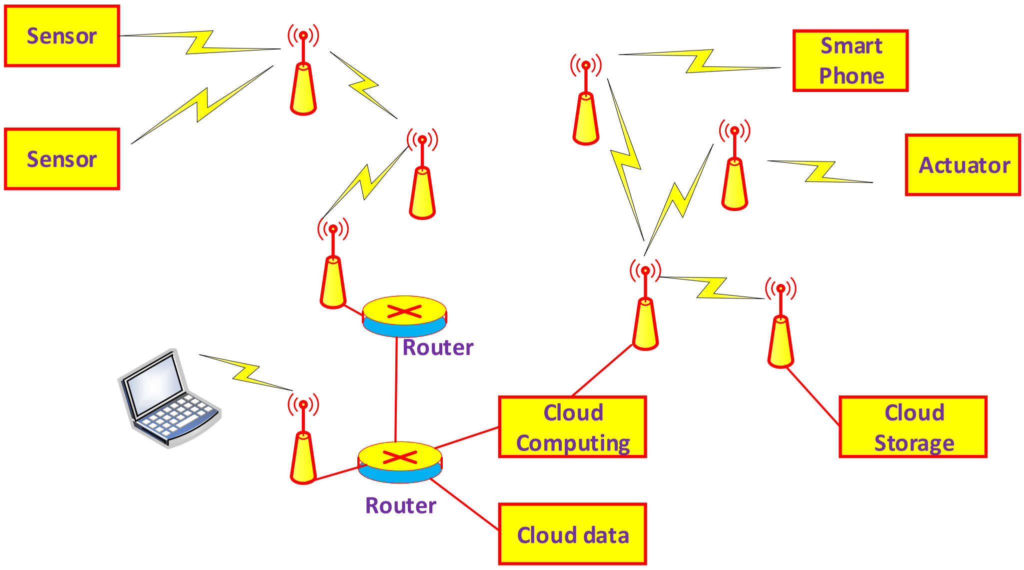

Fig. 1.1 Example of a wireless IoT network

follow so that the data packets are transmitted from the source to the destination successfully in the network. Since the process of data communication through a network is not simple, the task is divided between different layers, i.e., the protocol layers, where each layer has a specific responsibility. Two models of dividing the tasks into layers have been developed in the data communication community, the Open Systems Interconnection (OSI) model and the Transmission Control Protocol/Internet Protocol (TCP/IP) model. Figure 1.2 shows the protocol layers for the two models. The two models agree on the lowest 4 layers. The OSI model has two extra upper layers: the session layer and the presentation layer. The OSI model appeared after the TCP/IP model was already developed and put into practice. It was thought that the OSI model would replace the TCP/IP model. However, this did not happen since the OSI model did not show a high enough performance to justify the cost of the change [11]. The TCP/IP model is currently the dominant model in the wired and wireless data networks. Figure 1.3 shows an example of a small network with 3 hosts in a TCP/IP network. Interested readers are referred to [11] [12] to get the details of the tasks assigned to each layer. Here, this chapter only provides a brief description of the transport layer since it is the most relevant to the material in this thesis.

1.3.1 Transport Layer

The transport layer provides service to the application layer and receives services from the network layer. As shown in Figure 1.3, the transport layer provides application-to-application communication between the source and destination, i.e., it establishes a logical link between the two applications, one at the source and one at the destination. There are several types of transport layer protocols that are available for Internet applications. The most widely used are User Datagram Protocol (UDP), Transmission Control Protocol (TCP) and Stream Control Transmission Protocol (SCTP) [11]. This thesis focuses mainly on the TCP-based protocols. Hence, the next paragraphs briefly describe the main functions of TCP.

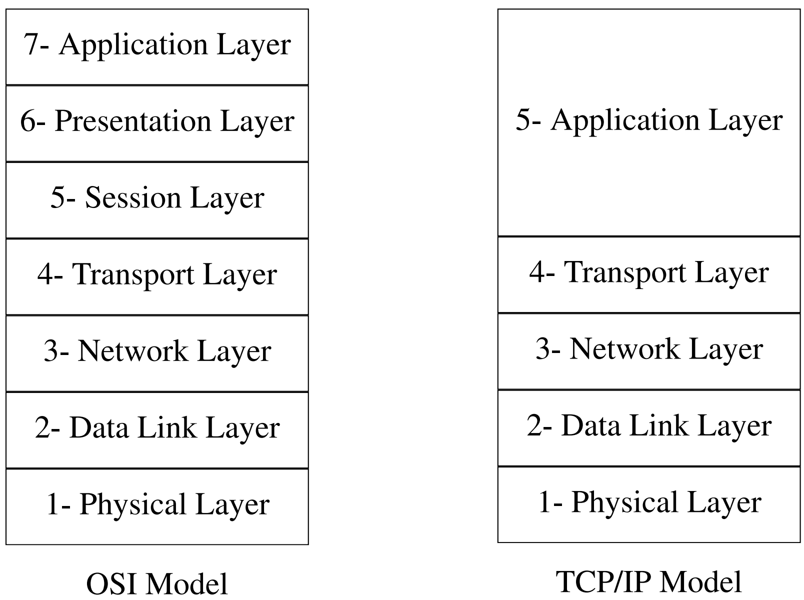

Fig. 1.2 TCP/IP and OSI models

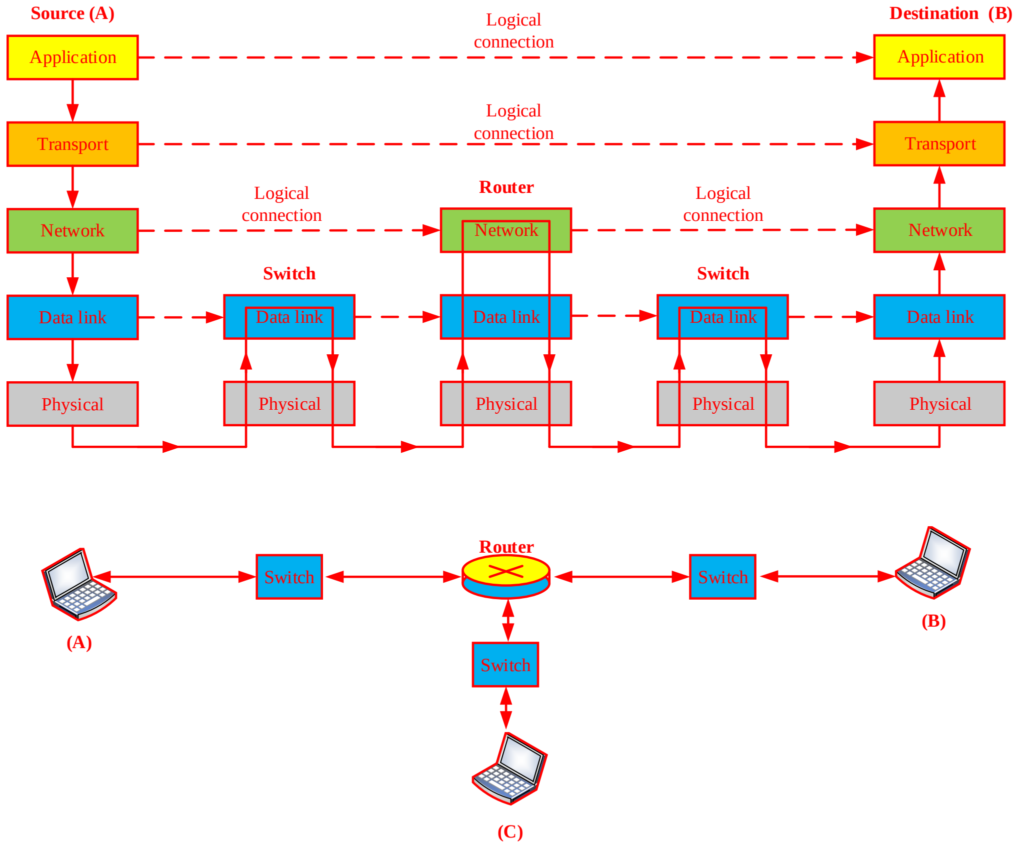

Fig. 1.3 Logical connections between protocol layers

Transmission Control Protocol (TCP)

The TCP is a connection-oriented protocol [11]. This means that the logical connection between the source and destination is established first (setup phase). After the logical connection is established data packets are transmitted (transmission phase). During transmission, the data packets are numbered and transmitted in the correct order. If they are not received in order, TCP at the destination must reorder them before providing them to the application layer. After the transmission phase is completed, the logical connection is terminated (teardown phase). Moreover, TCP provides flow control, i.e., matching the source transmission data rate to the receiving rate of the destination. Flow control aims to stop the source node from overwhelming the destination node with data packets. Also, TCP provides congestion control to avoid the occurrence of congestion of packets in the network. When the TCP source does not receive acknowledgement, or receives multiple acknowledgements for a sent packet, it knows that there is a congestion in the network, and it slows down packet transmission. Also, TCP provides reliable connection. This means that packets that arrive corrupted to the destination are dropped, and the TCP source side must resend them. Similarly, missed packets must be resent and duplicate packets are discarded.

TCP is a successful protocol that is widely used in most wired and wireless networks [12]. Hence, a significant amount of research has been devoted in the literature towards improving the performance of TCP further, since this can have a significant impact. Nevertheless, TCP still experiences a number of limitations. One of the shortcomings of TCP is the lack of a fault tolerance property which is important in next-generation communications [13]. When path failure occurs in the connection, data will be lost, thus, TCP has to reset a new connection [11] [12]. Conventional TCP does not exploit the availability of multiple networks in a single device for data transmission. In other words, TCP does not support multihoming. This motivated considerable research to introduce Multipath TCP (MPTCP), which is the main theme of this thesis. In the next section, the chapter provides a brief description MPTCP. Chapter 2 of this thesis presents a complete literature review of this topic. The contributions of this thesis in the field of MPTCP in wireless IoT networks are listed later in the current chapter.

1.4 Multipath Transmission Control Protocol (MPTCP) and Multihoming

Multipath TCP (MPTCP) is a recent extension to TCP. It allows sending the data of one application over several paths, or subflows (each subflow/path is basically a TCP connection), simultaneously [14]. For example, by using MPTCP, a wireless device can concurrently transmit a single TCP flow throughout a Wi-Fi and cellular network (i.e., it is multihomed to two networks). In fact, MPTCP is designed to support multihoming [15]. The scenario of MPTCP is shown in Figure 1.4, where an MPTCP client has access to 3 wireless networks (cellular, Wi-Fi and Wi-MAX). The MPTCP server also has multiple paths to communicate with the client. Using an MPTCP, both devices send and receive data of the same application over all interfaces.

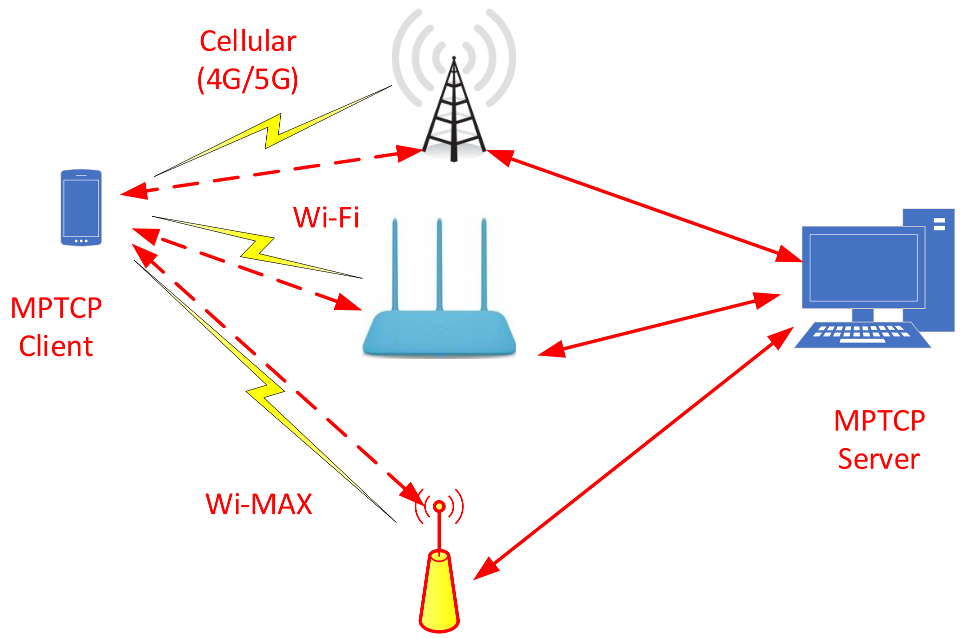

Fig. 1.4 an MPTCP with 3 sub-flows

In addition, the architecture of MPTCP in the TCP/IP model is shown in Figure 1.5, where an MPTCP-capable device sends data packets on different TCP connections (subflows/paths) [16] using the MPTCP packet scheduler. At the destination, these packets must be reassembled before providing them to the application layer. MPTCP can provide several advantages, including [17]:

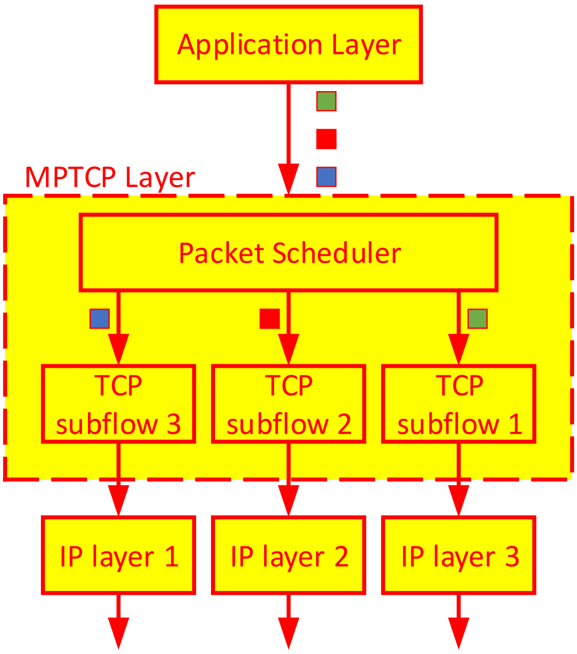

Fig. 1.5 The architecture of MPTCP in the TCP/IP model

- Improving network throughput: due to the concurrent use of multiple subflows.
- Improving resilience: If a path fails, packets can be sent or resent on different paths. In fact, MPTCP can seamlessly fall back to a normal TCP connection whenever required.
- Reducing congestion: the traffic can be moved from congested paths to less congested ones.

MPTCP protocols should be carefully designed to exploit the advantages of the available subflows. The packet scheduler should be correctly tuned, especially for wireless networks with imbalanced throughputs and delays. Otherwise, many packets may arrive at the destination out of order, which requires additional buffering and delay for reordering them [18]. MPTCP is the subject of considerable recent research [16], and several variants of MPTCP have been developed.

This thesis provides new novel algorithms that improve Multipath TCP using an optimised cross-layer design, the adaption of Opportunistic Routing [19] and the utilisation of the Full Duplex technology [20], as demonstrated in the upcoming chapters of the thesis.

1.5 Opportunistic Routing

Routing is a task of the network layer of the TCP/IP. It is crucial to the performance of all wired and wireless networks. The task of the network layer (which is called the IP layer in the TCP/IP protocol) is to select the best route to forward the packets of data from one node to the next in the network. The dynamic nature of the wireless networks makes routing a difficult task.

Traditional routing protocols in wireless networks pre-select the best neighbour node for routing before the packet transmission phase. This selected neighbour node is fixed during the transmission phase. This strategy may not be suitable for wireless networks where the state of the connection may dynamically change at a fast rate due to the dynamic behaviour of wireless channels, such as fading and blockage. Opportunistic Routing (OR) [21] is a strategy that benefits from the broadcast capability of wireless networks. The basic idea of OR is that the transmitted data packet is received by multiple neighbouring nodes. If at least one node successfully receives the packet, the forwarding task continues. A simple example that clarifies the benefit of OR is shown in Figure 1.6 [19]. Let us assume that the probability of successful relaying from the source to any neighbouring node is P (P = 20% in the figure), and the probability of successful relaying from any neighbouring node to the destination is 100%. In traditional routing, the probability of successful packet delivery from source to destination is P = 20%. However, with OR, this probability is $1-(1-P)^5 = 0.672$, which is much better than conventional routing. In Chapter 4, this thesis utilises OR in the network layer, together with MPTCP in the transport layer, to mitigate the MPTCP delay. Reducing delay is an important requirement in the next generation of IoT networks for which conventional MPTCP with traditional routing protocols may not be adequate.

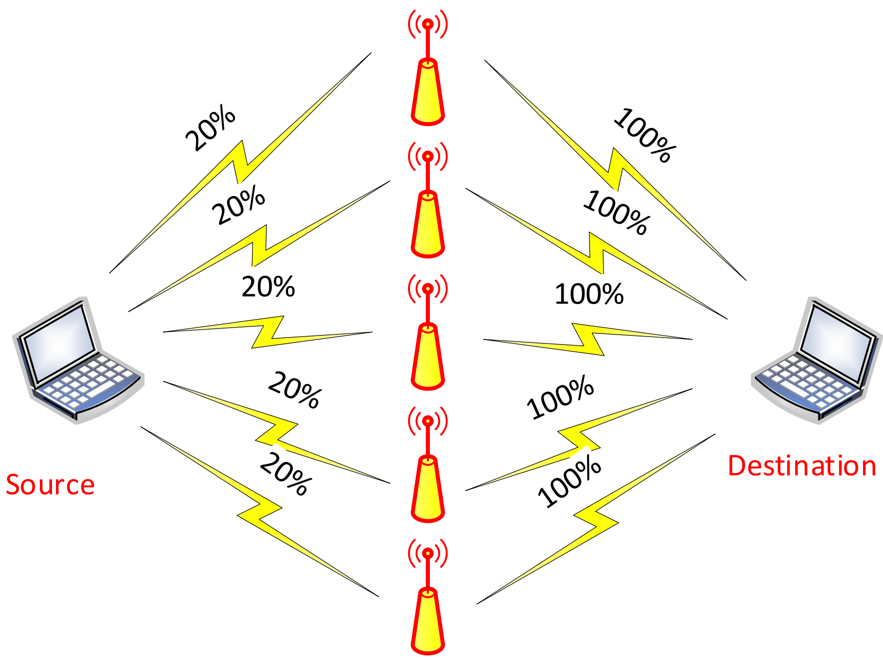

Fig. 1.6 Illustration of the benefits of Opportunistic Routing

1.6 Cross-Layer Optimisation in Wireless Networks

As outlined above, the TCP/IP communication protocol includes 5 layers, where each layer is assigned a specific set of tasks that are needed for successful end-to-end communication. In the current TCP/IP model, the exchange of data and services occurs between adjacent layers only. Each layer is allocated specific tasks that make this layer *a black-box* from the point of view of the adjacent layers.

Each layer does not need to know the internal details of any other layer, and each layer interacts with the inputs and outputs of the adjacent layer. In one hand, this black-box design of the TCP/IP protocol makes it easier to deploy. On the other hand, this strict design prevents interaction among different layers and causes a loss of performance that could have been realised if some level of flexibility (cross layer) had been allowed. In addition, cross layer design refers to the procedure where any layer can provide information and parameters to other non-adjacent layers, however, each layer retains its own functionality [22]. The goal of this interaction is to optimise the overall network performance. Figure 1.7 illustrates the concept of cross layer framework. Additionally, a lot of research has been dedicated to improve the TCP/IP model through cross-layer design [16]. Chapter 3 in this thesis contributes to this effort by presenting a cross-layer scheme in wireless networks to maximise user utility.

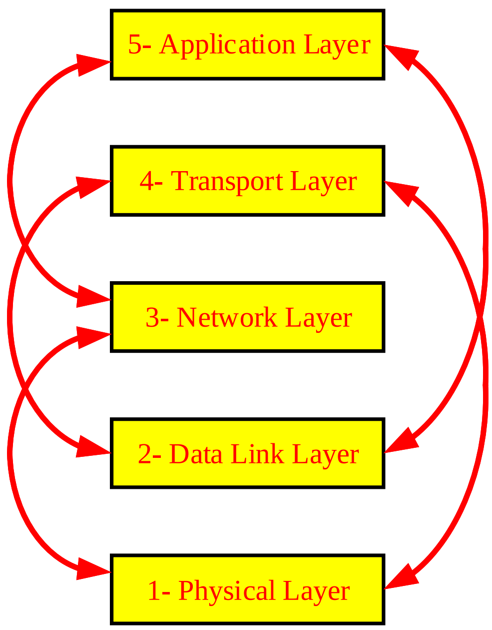

Fig. 1.7 Concept of cross-layer framework

1.7 Full Duplex Communication

Recently, there has been a significant interest in employing Full Duplex (FD) in wireless communications [20]. FD means that a wireless device can receive and transmit simultaneously, and at the same frequency band. All currently deployed wireless systems, including cellular systems, Wi-Fi and Bluetooth, employ either Frequency Division Duplex (FDD) or Time Division Duplex (TDD). Hence, conventional wireless devices receive and transmit on different frequency bands or different time slots. If FD can be successfully used, it has the potential of doubling the available frequency band.

However, fulfilling this promise of doubling the available frequency band is difficult to achieve. The main reason is the difficulty of suppression of the Self-Interference (SI) from the transmitter to the receiver. Although the transmitted signal is known at the receiver of the same device, the power difference between the transmitted and received signal makes Self-Interference Suppression (SIS) a difficult task [23]. Some prototypes of FD-capable wireless devices have been demonstrated lately with promising levels of SIS [20]. With the continuous research in this area, it is expected that in the next few years FD will be a reality. With this expectation in mind, Chapter 5 exploits FD technology to improve MPTCP in wireless IoT networks to considerably increase its reliability and reduce its latency.

1.8 Thesis Aim and Objectives

Aim:

The aim of this thesis is to enhance MPTCP in IoT wireless networks. The next generation wireless communications systems, such as 5G and beyond, open the door for high data rate, high reliability, low latency, and massive machine-type networks. Consequently, future IoT applications are expected to exploit these newly facilitated performance levels and elevate their requirements from wireless networks (i.e., requiring strict QoS). Traditional MPTCP is not able to satisfy these requirements due to several reasons. First, MPTCP works independently in the transport layer, however, if other layers are facing challenges (e.g., interference) and can not resolve them instantly, the performance of MPTCP degrades significantly [13] (this problem is addressed in Chapter 3). Second, the traditional routing method in MPTCP has a lot of drawbacks, such as suffering from a high number of transmissions [24] (this problem is addressed in Chapter 4). Last, MPTCP does not exploit FD communication that is a promising technology in wireless networks (this problem is addressed in Chapter 5). Thus, this thesis is proposed to improve the performance of MPTCP.

Objectives:

1. **Perform joint network utility maximisation of the source rates and link capacities to maximise the overall utility of multipath ad hoc wireless networks under delay constraints:**
   As a first attempt to develop MPTCP, Chapter 3 presents a cross-layer optimisation of the MAC (Medium Access Control) layer, network layer and transport layer. The objective is to select the joint optimum values of the source rates of the different data sources in the network, and the link capacities among the network nodes for each path. The cross-layer design attempts to maximise the overall utility of the network. Since delay is important in MPTCP, it is considered in this framework.

2. **Investigate and compare several proposed schemes of MPTCP:**
   Several multipath schemes are proposed in the literature to enhance the

performance of MPTCP. These schemes are originally intended to be employed for Wide Area Networks (WANs). Their performance in wireless IoT networks is questionable. Here, Chapter 4 applies the original MPTCP and other promising MPTCP schemes for IoT wireless networks and analyses and compares their performance. This study has not been done before. It important to note that these MPTCP schemes are potential candidates for future research in cross-layer optimisation, as has been mentioned in the previous objective.

3. **Adapt Opportunistic Routing algorithm to different MPTCP schemes:**
   Opportunistic Routing (OR) is a promising network layer protocol that takes advantage of the broadcast capability of wireless networks to optimise the routing link selection for source packets. OR has shown significant improvements in wireless networks. Thus, Chapter 4 adapts OR to different schemes of MPTCP to mitigate delay in IoT networks.

4. **Implement Full Duplex capability to enhance MPTCP in IoT wireless networks:**
   One of the recent advances in the physical and MAC layers of wireless networks is the FD Radio Frequency (RF) capability. An FD capable device can transmit and receive on the same RF bandwidth at the same time. The receiver side can perform self-interference suppression of the self-transmitted signal. Hence, it can separate and detect the received signal from other wireless nodes in the network. While this technology is employed in the physical layer, it is yet to be exploited by the upper layers of the protocol stack. In Chapter 5, this thesis designs an MPTCP protocol

that exploits the FD capability of a device to considerably reduce the IoT network latency and increase its reliability.

1.9 Thesis Contributions

As a first contribution in this thesis, the author proposed a novel cross-layer design for the communication protocol of ad-hoc networks in Chapter 3. To accomplish this goal, a utility maximisation problem is considered that results in a cross-layer design for Multipath TCP, involving congestion control, scheduling and routing. Utility maximisation is performed via an optimisation problem that includes source-rate, queuing delay and scheduling constraints. The problem is solved for both fixed channels and time-varying channels. The solution of the optimisation problem exploits the Lagrangian. In addition, the outcome of the optimisation problem is formulated in two possible methods of scheduling: perfect scheduling and distributed scheduling. Each method provides a different trade-off between performance and complexity. In perfect scheduling, all nodes in the communication network contribute to realise the optimum solution. In distributed scheduling, only the neighbour nodes can contribute, which makes the implementation simpler. The chapter analytically proves that the optimisation problem converges to a global optimum in all methods. The results are compared with other algorithms in the literature. This comparison revealed that the proposed algorithms outperform existing algorithms. This is due to the fact that the proposed algorithms take into account the queuing delay and the available multipath connections. Specifically, the proposed algorithms demonstrate a higher source rate, lower congestion price and faster optimisation convergence to the optimum solution.

The second contribution of this thesis is presented in Chapter 4, where the author compares several variants of MPTCP that are available in the literature for wireless IoT networks. The compared protocols are traditional MPTCP, MPTCP-TSC (MPTCP- Traffic Split Control) and ReMP TCP (Redundant MPTCP). MPTCP-TSC is a greedy protocol which splits traffic to minimise the cost (which is a weighted sum of the required transmission rates) while satisfying the delay constraints. On the other hand, ReMP TCP sends data redundantly over all available multiple paths in the network. These protocols were not examined for IoT wireless networks before this work. It is shown in the numerical results that ReMP TCP has the lowest number of transmissions and the lowest delay compared to other schemes. To improve the delay further, the chapter introduces the Opportunistic Routing (OR) protocol to all considered schemes (MPTCP, MPTCP-TSC and ReMP TCP). Reducing the delay is an important requirement in the next generation of IoT networks for which conventional MPTCP, and its variants, with traditional routing protocols may not be adequate. OR is a networking protocol in which the traffic is broadcast to all wireless nodes that can hear the transmission. Hence, the reliability of correct data transmission in a network is increased. The simulation results show that the OR-based MPTCP schemes reduce the number of transmissions and delay compared to the case not using the OR method. The results also reveal that the best protocol in the experiment is ReMP TCP with OR.

In Chapter 5, this thesis investigates the feasibility of FD technology in enhancing MPTCP for ultra-reliable low-latency communication (URLLC) applications in beyond 5G Internet of Vehicles (B5G-IoV) networks. First, the chapter presents the architecture of B5G-IoV wireless communication networks. After that, the chapter analyses the application of MPTCP in B5G-IoV networks, which involves studying the transport layer in such networks. As a result of this analysis, an

FD-based MPTCP protocol is proposed that is denoted as an FD-based multi-path transmission control protocol (FDMP). The proposed FDMP algorithm utilises the FD simultaneous transmission and sensing feature, and incorporates a modified scheduler and congestion control mechanism, a proactive ACK mechanism and a novel re-transmission strategy. The results show that the FDMP algorithm outperforms the benchmark MPTCP design, reduces latency, and increases the Packet Delivery Ratio (PDR) in dense network scenarios. This demonstrates that FDMP is feasible and more suitable for URLLC applications in B5G-IoV networks.

1.9.1 List of Publications

The thesis's novelties and contributions have been drawn from the following publications:

(*i*) M. Aljubayri, Z. Yang and M. Shikh-Bahaei, "Cross-layer multipath congestion control, routing and scheduling design in ad hoc wireless networks", *IET Communications*, vol. 15, no. 8, pp. 1096-1108, 2021. Available: 10.1049/cmu2.12145. **(Chapter 3 forms this article)**

(*ii*) M. Aljubayri, T. Peng and M. Shikh-Bahaei, "Reduce delay of multipath TCP in IoT networks", *Wireless Networks*, vol. 27, no. 6, pp. 4189-4198, 2021. Available: 10.1007/s11276-021-02701-3. **(Chapter 4 forms this article)**

(*iii*) J. Zang, M. Aljubayri, W. Ding and M. Shikh-Bahaei, "Feasibility Analysis of Full Duplex Technology in Enhancing MPTCP for URLLC Applications in B5G-IoV Networks", *IEEE Transactions on Vehicular Technology, under review*, 2021. **(Chapter 5 forms this article)**

The first publication, that is (*i*), is based on Chapter 3 where a cross layer design for MPTCP is proposed. The second publication, which is (*ii*), is related to Chapter 4. In this work, opportunistic routing-based MPTCPs are introduced to the literature. Finally, (*iii*) is formed by Chapter 5 where a new MPTCP algorithm is presented using the FD communication system.

1.10 Thesis Organisation

The thesis is organised as follows. The first chapter (this chapter) presents an introduction to this thesis. Chapter 2 provides a literature review of the conventional MPTCP and the state-of-the-art research targeting the development of the protocol in wireless networks in general, and wireless IoT specifically. Chapter 3 introduces the first contribution of the thesis. In this chapter, the author presents an analytical solution of a cross-layer design for a utility maximisation problem to maximise user utility with respect to the source rates and link capacities in multipath ad hoc wireless networks. Chapter 4 investigates different variants of the MPTCP for wireless IoT and determines the best variant. Then, this best variant of MPTCP, as well as the other schemes, are supported with Opportunistic Routing in the network layer to provide superior QoS. Chapter 5 includes the FD capability in a new innovative design of MPTCP. This design can exploit the characteristics of the FD to the benefit of MPTCP in the transport layer. Finally, Chapter 6 presents the conclusion of the thesis and makes suggestions for future research.

Chapter 2

Literature Review

2.1 Introduction

This thesis presents novel improvements for Multipath TCP scheme. The MPTCP by itself is a development of the well-known TCP that is currently dominating the Internet. While TCP is a single path protocol, MPTCP can use different sub-flows (or multipath) for a single transport layer connection. Nonetheless, MPTCP is still in its early stages, compared to other transport layer protocols, and a lot of research effort is still needed to improve its performance and alleviate its drawbacks. The current thesis contributes to this effort.

In this chapter, the author provides a background and literature review of the important topics and research progress that is related to the designs presented in this thesis. Chapter 2 is organized as follows:

- Section 2.2 provides a short background on IoT.
- Section 2.3 presents a brief description of TCP. Also, the section explains the motivations that led to the development of multipath transmission protocols.

- Section 2.4 introduces the Stream Control Transmission Protocol (SCTP) [25] in addition to its variant the Concurrent Multipath Transfer Using Stream Control Transmission Protocol (CMT-SCTP) [26] since they are common multipath schemes in the literature.

- Section 2.5 provides a description of the MPTCP [14] [15] [17] that was standardized in 2013 by the Internet Engineering Task Force (IETF). The section explains the goals of MPTCP, its challenges and the research needed to tackle these challenges.

- Section 2.6 reviews the implementation of MPTCP in wireless networks and IoT.

- Section 2.7 provides a description of the cross layer-based Network Utility Maximization (NUM) model that is used to optimize the resource utilization, particularly in wireless networks [27].

- Finally, Section 2.8 summarises the chapter.

2.2 Internet of Things

The Internet is one of the most impactful innovations in human history. It permits the interconnection of all traditional computing devices that use the TCP/IP model throughout the world [28]. In the past few decades, the Internet has mainly been used for people to exchange news, information, literature, entertainment material and other data. Gradually, the Internet landscape has changed to include connectivity to everything, i.e., the Internet of Things (IoT). The IoT is a network of physical components (i.e., things) like sensors, motors (large or small) smart phones, power switches, etc. These 'things' communicate together to achieve some functionality using the Internet as a communication platform. As a simple home application for IoT, one can imagine a refrigerator that senses the lack of a bottle of milk in the designated place on the shelf. The refrigerator is connected to the Internet through an IoT communication node employing Wi-Fi or Bluetooth. The refrigerator may send a message to the head of the household to remind him/her to buy milk. The refrigerator may also be configured to directly make the order to the local store. While this example is a simple one for home use, the same concept is also used for more serious industrial applications. Sensors may sense mal-functioning parts, missing parts, parts that need maintenance or attention or even some hazardous situation. These sensors, which have Internet connection, send the information with the appropriate data to the relevant parties to take the needed actions.

There is no exact definition for IoT in the literature as it can be seen from different perspectives. "Important work is being done by the IEEE Internet initiative in order to find a conceptual IoT definition. ITU defines IoT as being an infrastructure that will connect physical and virtual devices. IETF defines IoT as being the Internet that considers TCP/IP and Non-TCP/IP suites at the same time and the things as being "objects" identified by unique addresses. IEEE, in its special report on Internet of Things, defines it as a network that connects devices having sensing capabilities. The IEEE Internet initiative gives its own definition as follows: The Internet of Things is a network that connects uniquely identifiable virtual and physical devices, using existing or new communications protocols. These Things are dynamically configurable and have interfaces that must be accessible distantly through the Internet"[28].

The term 'Internet of things' is first introduced by Kevin Ashton in 1999 [29]. However, the concept of IoT was first discussed in 1982 and the first application of smart devices was a Coca Cola machine presented at the Carnegie Melon University [30]. This vending machine was able to explain the availability of drinks as well as whether the drink was cold or not. "In 1990, John Romkey created a toaster that could be turned on and off over the Internet. WearCam was also another Internet device invented in 1994 by Steve Mann. It had a near-real-time performance using a 64-processor system. Paul Saffo gave the first brief description about sensors and their future course of action in 1997. In 2000, businesses and companies such as LG became interested in IoT and offered commercial IoT products. By 2013, IoT had become a popular concept and gained a lot of interest from academia and industry" [31].

IoT is a structural system that is divided into three layers [32]. The first layer is described as the perception layer. This includes sensors and actuators. Let us take the refrigerator example above. In this instance, sensors are responsible for detecting the milk level and transmitting the collected data to the sink node. The second layer of the IoT system is the network layer. This layer basically explains the connectivity and data forwarding of the IoT framework. It involves routers and gateways. Going back to the refrigerator example again, this layer describes carrying the sensing information from the gateway to other devices and eventually to the household device. The last layer is the application and storage unit. Clouds, servers and smart phones are the common equipment used for this layer. Receiving, visualizing the message and providing a service to the household represent the application layer in the aforementioned example of the refrigerator. Fig 2.1 shows the overall architecture of IoT.

Wireless networks are technologies used to connect all IoT equipment [33]. They are IoT connectivity solutions such as 4G/5G cellular networks, Low Power Wide Area Networks (LPWANs), Wi-Fi, Bluetooth and Zigbee. These wireless networks are different from each other in terms of data rate, energy consumption and coverage. The success of IoT systems depends on selecting the wireless

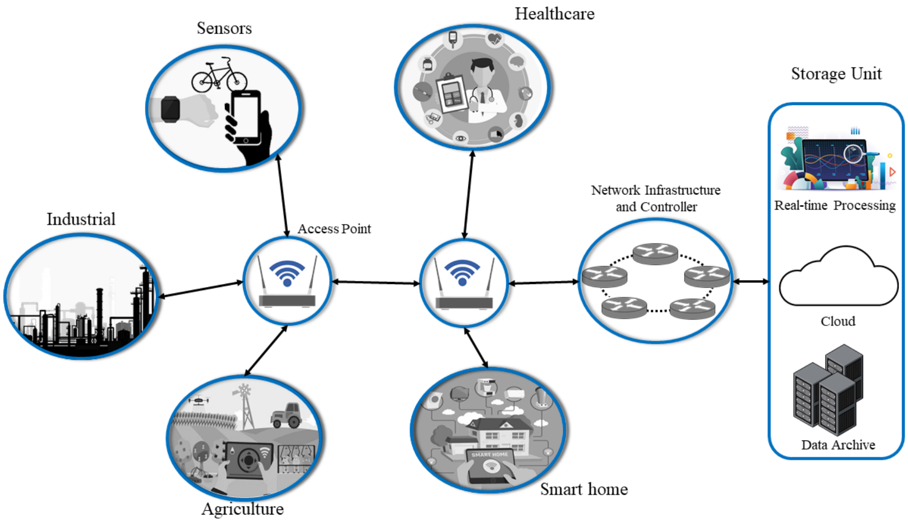

Fig. 2.1 Internet of things (IoT) architecture

technology that satisfies the needs. In this thesis, 4G/5G and Wi-Fi are chosen as wireless technologies to form IoT wireless networks.

However, MPTCP, that is the main subject of this thesis, is not optimised for IoT wireless networks. The next generation of IoT applications are expected to have strict Quality of Service (QoS) requirements. Traditional MPTCP is not able to satisfy such requirements, therefore, there is an opportunity to improve the protocol in IoT wireless networks. Before discussing MPTCP, the next section provides a short background about TCP and the emergence of multipath transmission protocols.

2.3 TCP and The Rationale for Multipath Transmission

TCP is a part of the TCP/IP protocol stack that was first introduced by the IETF in the 1980s [34–37]. TCP is a connection-oriented and reliable transport protocol, which can be used with applications that need a guaranteed delivery between the client and server sides (the two communicating endpoints). In addition, TCP is a full duplex protocol. This refers to the fact that the transmission between the two endpoints can go in both directions simultaneously. Besides, TCP, which is considered the standard transport layer protocol on the Internet, is located in the 4^{th} layer of the TCP/IP suite; and provides its services to the corresponding application layer. It also receives services from the lower layers. Additionally, it has a logical connection between the two communicating endpoints without detailed consideration of the individual hops between the endpoints [11].

In the last few years, modern communication systems such as Long-Term Evolution (LTE), 5G and Wi-Fi have been significantly developed to provide much higher specifications. They provide high data rates, approaching the giga bits per second region with low latency, to widely available wireless devices with strong processing capabilities, such as smart phones, PCs and laptops [16]. This increase in communication capabilities attracted new kinds of application that support real time multimedia with stringent QoS. Hence, the role of the TCP, which directly provides services to these applications, became more challenging [16]. Considerable research has been devoted to advancing the TCP layer performance [16] [38] which led to several major milestones in the TCP design. However, TCP is a single path protocol and does not support the connection resilience that is required in modern communication systems [13]. If the TCP connection dies out for any reason, which is particularly the case in wireless networks, TCP would re-connect to the remote endpoint, resulting in a new connection being established [38]; which encouraged the research community to find novel methods to overcome this drawback. One of these methods is the employment of multipath transmission [39]. Multipath transmission attempts to *turn the curse into a blessing*. This is done by exploiting the same heterogeneous communication networks that caused the pressure on TCP. These heterogeneous wireless networks

(LTE, 5G, Wi-Fi, etc.) are exploited to provide the solutions to the transport layer by providing multiple paths for the connection between the endpoints. In fact, multipath transmission can be employed in all layers of the communication protocol [40]. However, the focus of this thesis is only on the transport layer. In the next sections, two major transport layer protocols with multipath capabilities, namely, CMT-SCTP and MPTCP are presented. MPTCP is the baseline protocol for the remaining chapters of this thesis.

2.4 Concurrent Multipath Transfer with Stream Control Transmission Protocol (CMT-SCTP)

Similar to TCP, Stream Control Transmission Protocol (SCTP) [25] is designed as a connection-oriented, reliable transport layer protocol, which supports multihoming at either or both endpoints of the connection. Multihomed terminals have more than one network interface that can be utilized to enhance the network connection. For example, a smart phone could be connected to the Internet using 5G and Wi-Fi. Additionally, SCTP supports multihoming for resilience and fault tolerance purposes. It enables each endpoint of the connection to provide the other endpoint with a list of IP addresses, by which it can be identified and reached. However, since SCTP is designed with reliability as the main goal, simultaneous transmission on the different networks is not allowed [25]. Only one network is employed at any point in time.

To provide a simultaneous multipath transmission through SCTP, a modification is proposed in [26], which is denoted as Concurrent Multipath Transfer with Stream Control Transmission Protocol (CMT-SCTP). While CMT-SCTP is expected to increase the network throughput, it also has side effects that need to be resolved [26]. To improve the performance of CMT-SCTP, network coding (NC) is proposed [41], [42]. Using a low-complexity NC approach in [41], the authors were able to avoid data reordering and the subsequent buffer blocking. Hence, the coding capability of error correction compensates for the lost packets and reduces the number of retransmissions. Similarly, the NC approach in [42] exploits the cooperation among the multipaths of an SCTP-CMT connection to completely eliminate the problem of receiver buffer blocking. Random linear network coding (RLC) with machine learning is used to guarantee that the receiver can retrieve the originally transmitted packets. In addition, the multipath capability of CMT-SCTP has attracted the attention of real-time streaming video transmission research. The research efforts in [43] and [44] propose solutions to employ CMT-SCTP for video transmission over heterogeneous wireless networks using two different forward error correcting codes. Although CMT-SCTP is a promising scheme, it is difficult to implement in real-life networks, and consequently it is not widely deployed. The MPTCP has therefore emerged in the literature as an alternative. This is the subject of the next section.

2.5 MPTCP: Goals, Challenges and Solutions

The idea of extending TCP, particularly, to be a multipath protocol is not completely new. The first attempt was introduced by Huitema in 1995 [39]. Nevertheless, the protocol required additional amendments to TCP such as changing the header and the segment format. There were many schemes introduced to the literature to convert TCP to a multipath protocol such as Parallel TCP (pTCP) in 2002, mTCP in 2004 and concurrent TCP (cTCP) in 2007 [45]. However, all of these schemes had fairness and deployment issues. The concept of resource pooling, which inspired MPTCP, was proposed in 2008 [46]. It basically suggested gathering different resources to act as a single source. In 2009, MPTCP was proposed to the literature, however, it was not fully standardised by the IETF until 2013 [17].

MPTCP is promising protocol because it is fully backward compatible to the conventional (i.e., single-path) TCP and backward compatible to existing applications that rely on conventional TCP [38]. Non-MPTCP-aware entities (like middleboxes) in the network see it as a conventional TCP [14], thus, MPTCP does not require any modification to the current TCP/IP suite which makes the protocol attractive. Those are the reasons behind considering MPTCP in this thesis.

MPTCP transmits data through multiple paths simultaneously. Similar to SCTP and CMT-SCTP, MPTCP is designed to support multihoming but with better compatibility to the current network's infrastructure. Figure 2.2 shows the messages exchanged between two endpoints to establish an MPTCP session. First, the source node informs the destination node that it is a multipath capable device. This is done by sending the synchronisation packet (SYN) along with the MP_CAPABLE option [15]. An option is a field in the TCP header created for different purposes. MPTCP uses this field to establish a connection [47]. If the destination is multipath capable, it acknowledges the packet (SYN+ACK+MP_CAPABLE). Then, the source acknowledges the destination's packet (ACK+MP_CAPABLE). To this end, the first path is initiated [48]. To add another path to the connection, the source node sends an acknowledgment packet with another option called MP_JOIN. Next, the destination acknowledges the packet (SYN+ACK+MP__JOIN) and the source can start sending data [49]. This is just a brief scenario of how MPTCP works.

MPTCP is currently widely deployed [16] [39] [49]; and is expected to realize the following goals:

- Improve throughput: The throughput of MPTCP should be the same or higher than the throughput of TCP on its best path.

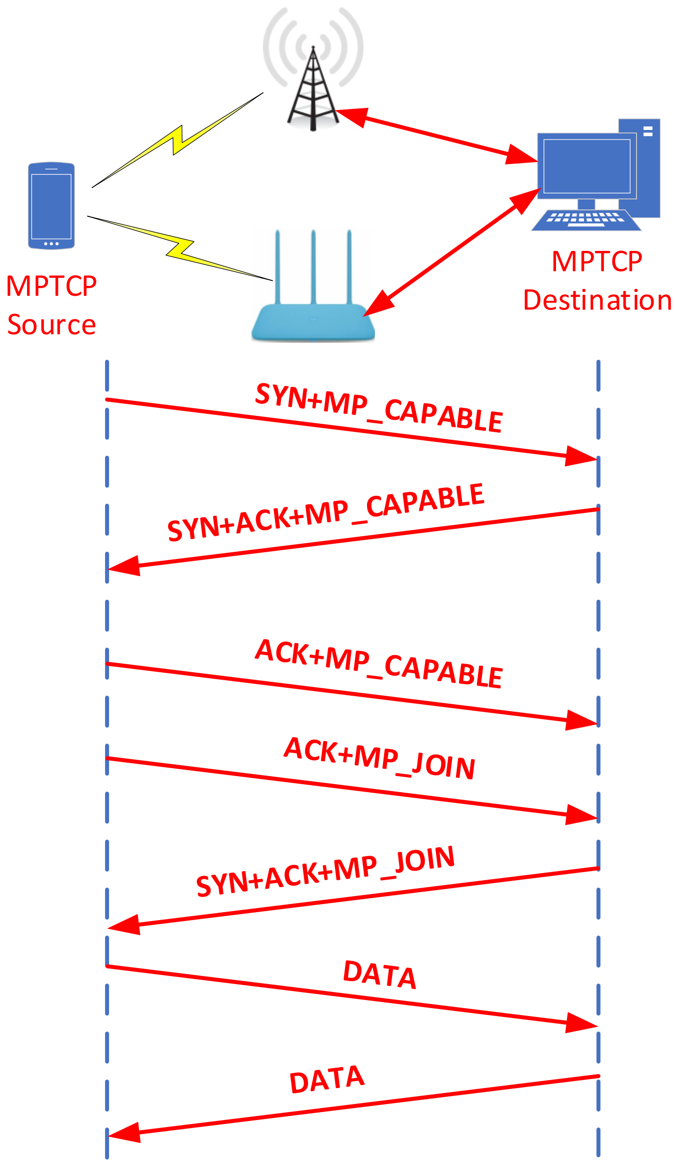

Fig. 2.2 Establishing an MPTCP session

- Improve reliability: MPTCP resilience should be the same or higher than the reliability of TCP.
- Mitigate delay, latency and jitter: In addition to increasing the throughput, MPTCP should reduce latency for applications that accept moderate throughput but require low latency and jitter. This includes for example interactive and real time multimedia applications.
- Balance traffic: As a result of fulfilling the previous goals, MPTCP should improve the overall network utility by directing the traffic load away from congested paths and exploiting the less loaded paths. Hence, network resources are optimally utilized and wasted resources are minimized.
- Retain network compatibility: MPTCP should be able to work with the presence of network 'middleboxes'. Examples of middleboxes include firewalls, Network Address Translators (NATs) and proxies.
- Retain fairness: MPTCP should not harm other conventional TCP users. It should not take more capacity than a single path TCP.
- Retain network security: MPTCP should be as secure as the conventional TCP.

To achieve these goals in different wired and/or wireless network scenarios, MPTCP has triggered considerable research in the past few years. Nevertheless, MPTCP faces many challenges, and the most common ones are *Congestion Control*, *Packet Scheduling* and *Latency* [50] [51] [52]. In the next subsections, the chapter reviews these challenges and the proposed solutions that are applicable for wired and wireless networks.

2.5.1 Congestion Control Algorithms:

The process of congestion control is a main function in TCP as well as MPTCP. However, it is more important in MPTCP [16] [52]. To understand the importance of congestion control, let us consider a network with a number of nodes and a number of links. Assuming some of these links have a limited capacity, they become bottlenecks in the network. When multiple sources generate their data flows, multiple flows of data may share a bottleneck link. Hence, this link becomes congested and data packets are delayed or even dropped. Consequently, many source(s) do not receive an acknowledgement (ACK) within the Retransmission Time-Out (RTO) interval. In the absence of a *Congestion Control algorithm*, these sources retransmit their data packets, causing further congestion. On the other hand, with a proper Congestion Control algorithm, these sources realize that there is congestion, reduce their own data rate and alleviate further congestion. Congestion is typically controlled by adjusting the congestion window (denoted as CWND). The CWND is increased when ACK is received for a packet, and decreased when a packet is lost. The most widely used method for CWND update is the Additive Increase Multiplicative Decrease (AIMD) principle as follows [16] [52]:

$$w \leftarrow \begin{cases} w + \frac{\alpha}{w} & \text{when ACK is received (Additive increase)} \\ \beta w & \text{when packet is lost (Multiplicative decrease)} \end{cases} \tag{2.1}$$

where w denotes the congestion window size. In the NewReno (a name for a congestion control algorithm) single-path TCP version, the parameters α and β take values $\alpha = 1$ and $\beta = 0.5$ [53]. The NewReno congestion control algorithm is a single-path TCP congestion control that responds to 'partial acknowledgement'.

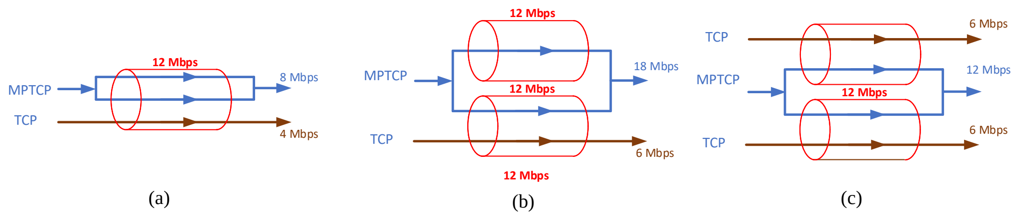

Fig. 2.3 Unfair bandwidth sharing between single-path TCP and MPTCP with independent congestion control per path of the MPTCP.

Partial acknowledgement refers to the case when the new ACKs cover new data, but some of the outstanding data is not ACKed yet. NewReno is designed to improve the fast recovery phase of the congestion control algorithm [53].

Turning the attention to MPTCP, the first version of MPTCP congestion control algorithm employed the same NewReno AIMD, but independently for each path. However, this choice led to unfair treatment by MPTCP to other single-path TCPs [52] [54]. This is illustrated in Figure 2.3 [52]. In illustration (a) the MPTCP shares a 12 Mbps link with a single-path TCP. Since each path is independently controlled, the MPTCP share is twice that of the single-path TCP. In illustration (b) one path of the MPTCP flows uses a 12 Mbps link, while the other path shares a 12 Mbps link with a single-path TCP. Again, the MPTCP gets three times the data rate of the single-path TCP. In illustration (c) with one MPTCP and 2 separate single-path TCPs, the MPTCP gets twice the rate of any of the 2 single-path TCPs.

To solve the fairness problem, *coupled* congestion control should be used (i.e., coupling all paths in the MPTCP connections). A *coupled* congestion control algorithm, denoted as Linked Increases Algorithm (LIA), is presented in [15]. The algorithm pertains only to the increased part in the CWND for path n when a packet is acknowledged. In LIA, the window is increased as:

$$w_n = w_n + \min\left(\frac{\alpha}{w_{total}}, \frac{1}{w_n}\right) \tag{2.2}$$

where w_n is the size of the congestion window for path n of the MPTCP, and w_{total} is the sum of the congestion windows in all paths (sub-flows of MPTCP). The parameter α is given by [54] as:

$$\alpha = w_{total} \frac{\max_n \left(\frac{w_n}{RTT_n^2} \right)}{\left(\sum_n \frac{w_n}{RTT_n} \right)^2} \tag{2.3}$$

where RTT_n is the Round-Trip Time on the n^{th} path of MPTCP. A practical implementation of this algorithm is standardized in [15]. This algorithm (i.e., LIA) is considered the default algorithm implemented in the Linux operating system [55]. While the LIA algorithm improves the fairness with TCP, it does not deal with the Head-of-Line (HoL) Blocking problem. The HoL problem is when the lost packet blocks other packets and prevents them from being delivered to the application layer.

The LIA algorithm is improved by the Opportunistic Linked Increases Algorithm (OLIA) [56]. In [56] the authors indicate that LIA is still not fair enough to single-path TCP users in some scenarios. Moreover, when some TCP users change to MPTCP, they may cause a reduction to the throughput of other TCP users, while they are not getting any benefit themselves. The OLIA algorithm [56] modifies the formula for CWND increase in case of packet loss. This modification is shown to resolve these problems. Similar to its predecessor (LIA), OLIA does not deal with the HoL problem.

Another algorithm that is widely implemented is the Balanced Linked Adaptation (BALIA) [51]. BALIA further improves the fairness between MPTCP and single-path TCP users while providing good responsiveness to the network congestion without causing a problem of congestion window oscillation. Hence, BALIA aims to make a balance between the properties of LIA and OLIA by providing TCP friendliness criteria. Similar to LIA and OLIA, BALIA does not deal with the HoL problem.

All the algorithms above fall under the category of loss-based algorithms, since they rely on acknowledgement or loss of packets to decide that congestion is presenting in the network. Another category is the delay-based algorithms, where large delay is used as an indication of congestion instead of packet loss. A delay-based congestion control algorithm that is proposed in [57] for MPTCP is denoted by the name DBCCA. This algorithm employs an optimization problem to minimize the difference in delay among the MPTCP paths. This typically happens in heterogeneous networks with different traffic characteristics. However, fairness to other single-path TCP users is not studied. Moreover, DBCCA is purely delay-based congestion control and does not give attention to packet loss. Probably a hybrid solution could provide better performance.

In contrast to most previous congestion control algorithms, the research in [58] does not depend on packet ACK and packet loss to estimate congestion. This research exploits the Explicit Congestion Notification (ECN) signal that is available in new operating systems such as Linux and Windows. With ECN, the source has a clear and explicit indication of the presence of congestion in the sub-flow. The authors present their design of a Shared Bottleneck based Congestion Control (SB-CC) algorithm to deal with the case when congestion bottlenecks are shared among multipaths. In SB-CC, the sub-flows that pass through a shared bottleneck congested link are grouped together into one set. Coupled congestion control is implemented on each set. However, the ECN may not be available in all network configurations.

Recently, a new congestion control algorithm that is intended for single-path TCP has been introduced by Google Inc. This algorithm is named Bottleneck Bandwidth and Round-trip propagation time (BBR) [59]. The BBR is a delay-based algorithm. It estimates the round-trip time (RTT) delay at the congestion bottleneck of the sub-flow. Hence, BBR is able to better characterize the bottleneck. Relying on the estimated value of the RTT, the BBR algorithm decides the transmission data rate of this sub-flow to avoid congestion at the bottleneck. Leveraging this BBR algorithm, the research in [60] presents a coupled congestion control algorithm for MPTCP using the concepts of the single-path TCP. This new congestion control algorithm, named BBR-Based Congestion Control and Packet Scheduling (BCCPS), can improve the overall throughput while realizing better fairness with single-path TCP flows. As a new algorithm, it needs to be thoroughly tested. However, the work in [60] performed some basic test scenarios that are not sufficient to gain confidence. More complex scenarios are reserved for future research, as outlined in [60].

The work in [61] focused on improving the congestion control algorithm of MPTCP for short packet flows. The authors noticed that the main source of performance loss in short flows is the aggressive increase of the congestion windows in each individual sub-flow at start-up. In short packet flows, the MPTCP is at start-up most of the time. To solve this problem, the Gentle Slow start algorithm (GSAM) was proposed for MPTCP. This algorithm makes the aggressive increase in the congestion windows smoother during the start-up phase of MPTCP congestion control. It is illustrated by the authors that GSAM decreases the completion time for short flows considerably. However, the benefit for long flows is much less.

Another research that is interested in the performance of short flows is presented in [48]. In [48] a new algorithm called MPTCP Transmission Optimization Algorithm for Short Flows (MPTCP-TOASF) is proposed. This algorithm groups the paths of MPTCP according to their RTT. Then, different control procedures are applied for each group. MPTCP-TOASF relies on the Veno algorithm [62] for long flows, which improves the performance for both the long and the short flows. It is noted in the numerical results of [48] that the comparisons with other MPTCP congestion control algorithms in diverse scenarios are limited. The authors put this point for further study.

Table 2.1 summarizes the congestion control algorithms reviewed in this section with their advantages and disadvantages.

Table 2.1 : Summary of the reviewed MPTCP congestion control protocols

Protocol	**Concept**	**Advantage**	**Disadvantage**
NewReno [53]	Responses to partial acknowledgments received during fast recovery	Improves fast recovery in TCP	MPTCP becomes unfair to other single-path TCPs
LIA [15]	Increases CWND based on coupled weights	Better fairness to TCP users	HoL problem still exists
OLIA [56]	Modified the CWND increase of LIA	Optimal fairness	HoL problem still exists
BALIA [51]	Balances the properties of LIA and OLIA	Balancing among responsiveness to congestion, friendliness, and window oscillation	HoL problem still exists
DBCCA [57]	Minimizes the difference in delay among different paths	Optimizes transmission in heterogeneous networking	No consideration of packet loss
SB-CC [58]	Employ ECN (Explicit Congestion Notification)	Better estimation of the congestion degree of each subflow.	Requires the network to support ECN
BBR [59]	Designed for single-path TCP	Better characterization of the bottlenecks to match traffic flow accordingly	Not suitable for MPTCP as it is for single-path TCP
BCCPS [60]	Enhances BBR for MPTCP	Adaptively controls the transmission rate of each sub-flow	Only simple scenarios were tested. Needs to be examined in more complex scenarios
GSAM [61]	Improves transmission performance for short flows	Smooth the congestion window growth during slow-start phase	Less optimization of long flows due to congestion window growth
MPTCP-TOASF [48]	Enhances transmission performance for short flows	Groups paths based on their RTTs to improve performance of small flows	Limited comparison with other known MPTCP congestion control in diverse scenarios

2.5.2 Packet Scheduling Algorithms:

Another factor that affects the MPTCP performance is the scheduling algorithm [63]. Figure 2.4 shows the relation of the packet scheduling and the congestion control in MPTCP [63]. As explained earlier, the congestion control algorithm decides a congestion window w_n for path n, which influences the transmission rate of this path. Now, when a packet is to be transmitted from the send buffer, some paths may have their congestion window full. The paths whose congestion windows have room are considered available. The packet scheduler selects one of the available paths for transmission based on the scheduling criteria. It turns out that packet scheduling criteria have a significant effect on the performance of MPTCP with respect to throughput and delay [64].

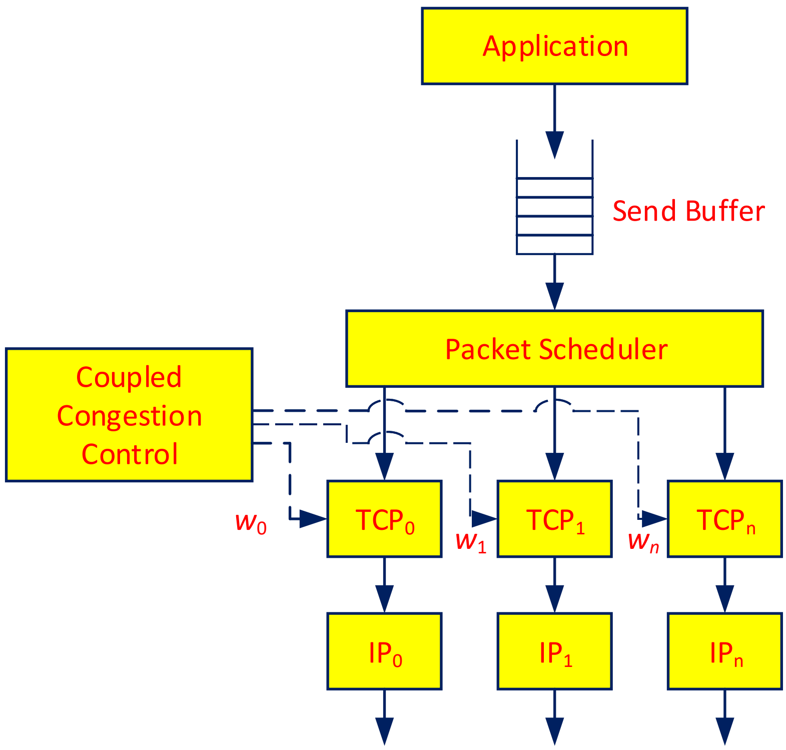

Fig. 2.4 MPTCP packet scheduler with coupled congestion control

The MPTCP packet scheduler in Linux follows the Lowest-RTT-first [63][64], i.e., a packet is sent on the available path with the lowest *smoothed* RTT (sRTT). This is also referred to as the Lowest Latency (LL) scheduler. RTT is the time duration between the transmission of a packet and reception of the ACK of this packet. The smoothed RTT is the operation of smoothing (i.e., filtering) successive RTTs on the same path to get a smooth or filtered value as follows:

$$sRTT = (1-\lambda) \times sRTT + \lambda \times RTT \tag{2.4}$$

The typical value of the smoothing factor λ is $\lambda = 1/8$ [63]. While the algorithm presented in [63] is very simple, it does not take into consideration other important factors that may affect the scheduling decision; namely, the congestion on the path being scheduled and the packet size being transmitted. Other algorithms provide better performance when these points are taken into consideration in the design.

In [64], it was shown that for small size packets, the LL scheduler is not efficient. It is more efficient for the scheduler to wait till the fastest path becomes available and allocate the small packets to it. In [1], three alternative heuristic schedulers are proposed. These are: 1) The Highest Sending Rate (HSR), 2) The Largest Window Space (LWS) and 3) The Lowest Time/Space (LTS). The 3 proposed scheduling algorithms outperform the Linux LL scheduler. The scheduling prioritization is in Table 2.2 below.

Table 2.2 : Description of the work in [1]

Algorithm	**Scheduling prioritization criteria**
HSR	The ratio between the current CWND measured in bytes, and the instantaneous RTT.
LWS	Paths with larger CWND and fewer unacknowledged packets
LTS	Lowest ratio between the current smoothed RTT and window space. The window space is defined as the difference between the current CWND and the current number of packets in-flight

The LWS algorithm is best when the paths have a varying transmission rate. On the other hand, the HSR algorithm is suitable when the paths have varying delays. Lastly, the LTS algorithm is best for high loss and varying delay paths. However, HSR showed worse performance when the paths changed their transmission rates. Also, LWS does not perform well with homogeneous transmission rates in all paths. The LTS showed best performance due to its bilateral decisions. LTS scheduling prioritizes paths based on both the lower latency and the larger window space, which adds complexity to the scheduler.

In [65], an analytical derivation is given for the theoretical limit of the total throughput that can be achieved in MPTCP. The study shows that this limit can be realized by the optimal load balancing (OLB) scheduling algorithm. The study also provides a design for the OLB and performs simulations to demonstrate that the newly proposed OLB is superior to the Linux default LL scheduler. OLB finds the optimum scheduling weights for each sub-flow, where 'optimum' means minimizing the RTT in MPTCP. With the improved scheduling, the HoL problem at the destination side is reduced. However, as outlined in [65] the research considered only two sub-flows, LTE and W-Fi. To gain confidence in the OLB scheduler, more sub-flows should be taken into consideration.

Along with the congestion control algorithm presented in [58], which is discussed in the previous subsection, the same authors presented a new scheduling algorithm for heterogeneous networks with large difference between sub-flows in MPTCP. The scheduling algorithm named Shared Bottleneck based Forward Prediction packet Scheduling (SB-FPS). As outlined above, this algorithm relies on ECN signal to detect groups of sub-flows with shared congestion bottlenecks. The SB-FPS estimates the future behaviour of each sub-flow and pre-schedules the packets on each path according to changes of window size in each sub-flow.

Similar to [58], the research in [66] presented a new congestion control algorithm named MPTCP-BBR which is outlined the previous subsection. In addition, the same work proposed a scheduling algorithm relying on the same formulations of the congestion control algorithm, named BBR-Based Congestion Control and Packet Scheduling (BCCPS). This packet scheduling algorithm has two phases. During the first phase, it sends the packets in a redundant way through all paths. This phase is beneficial for small flows since it requires less delay when the transmission rates vary widely in each path. In the second phase, the regular MPTCP packet scheduler is retained (i.e., scheduling based on path quality) for large size packet flows. The work performed some basic scenarios, but more complex scenarios are reserved for future research, as outlined by the authors.

Table 2.3 summarizes the MPTCP scheduling algorithms reviewed in this section with their advantages and disadvantages.

Table 2.3 : Summary of the reviewed MPTCP scheduling protocols

Protocol	**Concept**	**Advantage**	**Disadvantage**
Lowest-RTT-first [63]	Sends packets on lowest available RTT path	Simple	No optimization based on path congestion or packet size
HSR [1]	Prioritizes paths with recent higher goodput	Better for varying delay paths	Slow response to network dynamics when paths change rates
LWS [1]	Prioritizes paths with larger CWND and fewer unacknowledged packets	Better for paths that exhibit varying transmission rates	Low performance with homogeneous transmission rates in all paths.
LTS [1]	Prioritizes paths with smallest ratio between the current smooth RTT and window space	Best performance for lossy paths and/or varying delay paths	More complexity due to more complex scheduling
OLB [65]	Provides an optimal load balancing between sub-flows	Reduces HoL problem	Only 2 paths are considered in the analysis and results
SB-FPS [58]	Distributes data based on the predicted changes of window size of each sub-flow	Accurately scheduling of data in shared bottleneck situations using ECN	Requires the network to support ECN
BCCPS [66]	Keeps delivery in-order by redundant scheduling, followed by regular scheduling	Best in heterogeneous wireless networks when different transmission rates	Only simple scenarios were tested. Needs more complex scenarios to test

2.5.3 Latency Algorithms:

An important performance aspect of data communications is the latency of the data packets. Low latency is critical in many applications that involve real time multimedia, Augmented Reality/Virtual Reality (AR/VR) content distribution [67], and safety applications such as Vehicle to Everything (V2X) applications. Considerable research has been conducted lately to exploit or modify MPTCP to mitigate latency as detailed below.

In [68], the authors argue that, based on their assessment, most existing MPTCP schemes do not perform well with asymmetric paths. Conventional MPTCP schemes are not able to exploit the aggregate capacity of the available paths to provide low latency. Two new algorithms are proposed for MPTCP in heterogeneous networks where the paths have different quality in terms of loss and delay (i.e., asymmetric paths). The first algorithm is the block estimation (BLEST) mechanism while the second is the shortest transmission time first (STTF) method. BLEST tries to reduce the receive buffer blocking by freezing the path which causes the blocking; while the goal of the STTF method is to predict and minimize the transmission time of packets. The BLEST scheme is a simple solution, but STTF is more computationally complex. However, both are able to considerably outperform the default MPTCP. Real world experiments are conducted with both BLEST and STTF, where STTF was more able to reduce latency and provide fast downloads than BLEST. Moreover, BLEST does not respond fast enough to time-varying link conditions.

In [69], an algorithm is presented to reduce the effect of jitter in MPTCP. This is particularly useful in jitter-sensitive real-time applications, such as gaming and video conferencing. The idea is to send the packets in an out-of-order manner over different sub-flows. The selection of the sub-flow is such that the packets arrive in order at the MPTCP receiver. This reduces the effect of jitter, which is very important for delay-sensitive real-time applications. However, the proposed algorithm was tested only in a small network with two routers and two laptops.

The four algorithms presented in [70], [68], [69] and [63] are compared in [71] in different real-life scenarios in heterogeneous networks. This includes different RTTs, different file sizes, different buffer sizes and different network topologies. The study provides guidelines for designing future MPTCP schedulers for 5G systems.

In [70], the authors distinguish between MPTCP short flows and long-lived flows. The article designs a dynamic MPTCP (denoted as DMPTCP) to dynamically adjust the sub-flows. The DMPTCP algorithm first estimates the latency on the path considered for transmission, as well as the data sent on the other paths at the same time. Using this information, the DMPTCP decides which sub-flow to use for the application to decrease the completion time for short flows, and at the same time realizing higher throughput for the long-lived flows. The outcome shows superior performance compared to the conventional LL Linux scheme for short flows as well as long flows. However, it is observed that some degradation occurs in case of common bottleneck competition among sub-flows.

Another study that is interested in moderating the latency for short flows is introduced in [72]. It is known that one of the requirements for MPTCP congestion control is to achieve fairness with TCP. Also, a good congestion control algorithm reduces the loss of network bandwidth and, hence, improves the resources utilization of this network. In [72] it is noted that most congestion control algorithms take a long time to achieve fairness with TCP (hundreds of seconds might be required). Hence, this goal is not reached in short flows. Previously, the authors proposed a new congestion control algorithm called Normalized Multi-flow Congestion Control (NMCC) which is designed to realize the fairness goal much faster than other congestion control algorithms. This is done through controlling the growth of the throughput rather than the throughput itself. The algorithm showed faster convergence. However, the algorithm is unfair to TCP users in the case of mild congestion. Thus, the authors extended it (eNMCC) to ensure TCP-friendliness during increased or decreased congestion windows phases. The resources (bandwidth of each sub-flow) utilization of the network is compatible with LIA. However, the utilization is still slightly lower.

In [73], an MPTCP scheduler that is optimized for reducing latency and jitter is proposed. While traditional schedulers send different data packets on different paths (based on certain criteria), the proposed redundancy MPTCP (ReMP TCP) scheduler sends each packet redundantly on all available sub-flows. The shortcoming of this approach is reducing the throughput. However, it also reduces latency due to the higher probability of correct delivery. Figure 2.5 shows the difference between the conventional MPTCP scheduler and the proposed ReMP TCP scheduler. The ReMP TCP scheduler guarantees that every packet is queued on the fastest paths available (in addition to other paths), which reduces latency and jitter at the cost of reducing throughput. This is attractive for applications that do not need high throughput but need low latency. Moreover, ReMP TCP is effective against packet drops since there is no retransmission when the packet is received and acknowledged on any sub-flow.

In [74], an MPTCP scheduler is proposed that employs the concept of redundancy to decrease latency and jitter. The authors propose the REdundant Diversity scheduling (RED) algorithm, whose main purpose is to prioritize the uncorrelated paths during packet replication. This algorithm can identify the paths that have high probability of being disjoint (uncorrelated in the article). The packets are duplicated on these disjoint paths only. Hence, the algorithm reduces latency and jitter at the cost of reducing throughput due to the redundant transmission. The result shows that this procedure considerably diminishes the latency of the system.

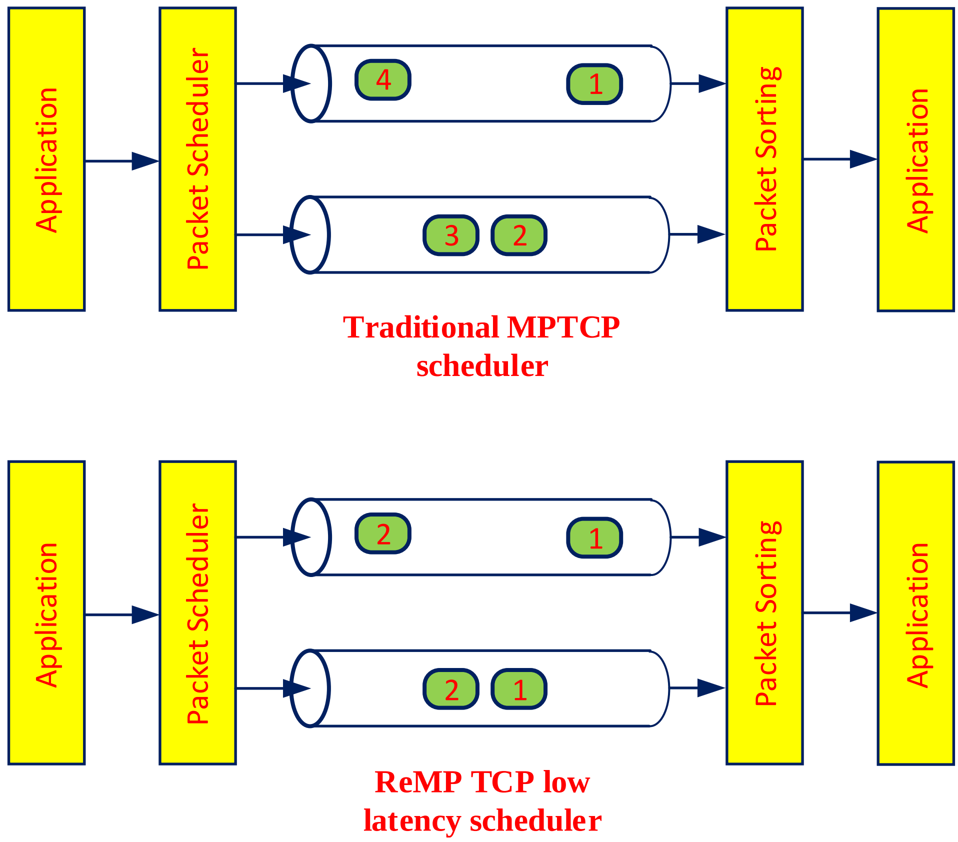

Fig. 2.5 ReMP TCP packet scheduler

The work in [75] focuses on minimizing the delay of MPTCP for Internet of Vehicles (IoV) applications. This work has two approaches: 1) improve the scheduling of the MPTCP over the dynamically changing path delays such that the load is balanced across the paths. 2) Employ Forward Error Correcting Code (FEC). The work analytically models the re-ordering delay in the MPTCP and develops a new algorithm denoted as MPTCP-IoV for IoV applications. The outcome shows that this algorithm significantly reduces the delay in an IoV system employing Wi-Fi and 4G networks. While the algorithm is proposed for IoV, it assumes reasonable mobility where the wireless environment changes more slowly than the algorithm convergence.

The research in [18] proposed another congestion control algorithm to be used for MPTCP. The target of this algorithm is to realize higher throughput and lower end-to-end delay for 5G networks employing mmWave technology. In mmWave frequency, the signal is subject to severe path loss and Line-of-Sight (LOS) must be available at the wireless link. The key novelty in the proposed algorithm is to realize a fast response to the possible variation in the fading channel condition, from LOS to non-LOS and vice versa. The congestion control algorithm is a development of an older algorithm named FAST [76], but with higher computational complexity. The proposed algorithm is denoted as delay-equalized FAST (DEFT) since it involves a procedure of equalizing the delay in the conventional MPTCP by equalizing the Round-Trip-Time (RTT) of all paths in MPTCP. This delay equalization is designed such that it minimizes the ordering delay in the receiver buffer. Simulation results demonstrate that DEFT has a superior performance to conventional congestion control algorithms.

Table 2.4 summarizes the MPTCP latency reduction algorithms reviewed in this section with their advantages and disadvantages.

2.6 MPTCP Implementation in Wireless Networks and IoT

Multipath TCP is designed to be used in all communication networks, wired or wireless. However, it is designed on the premise that the main source of packet loss is traffic congestion. Therefore, it behaves with the assumption that reducing the transmission rate automatically reduces congestion, and consequently reduces packet loss. However, when operating in wireless networks there are other sources of packet loss that pertain to the wireless environment. This includes channel fading, interference, mobility and handover effects [77]. Also, these effects may be changing rapidly in wireless environments. When a certain path is designated as a congested path (due to wireless effects, which are hidden from MPTCP),

Table 2.4 : Summary of the reviewed MPTCP latency reduction algorithms

Protocol	**Concept**	**Advantage**	**Disadvantage**
BLEST [68]	A scheduler to diminish buffer blocking	Lightweight and simple solution	Does not respond well to varying link conditions
STTF [68]	A scheduler to reduce the transmission time of each data packet	Fast download time in real life applications	Computationally Complex algorithm
Out-of-order Transmission [69]	Mitigates jitter by sending packets in an out-of-order manner on different sub-flows in a way that makes them arrive in-order at the receiver of MPTCP	Reduces jitter for real-time applications that need stable network delay performance	Tested in small networks only
DMPTCP [70]	Dynamically adjusts the sub-flows according to workloads of application	Decreases the completion time for short flows and increases the goodput of long-lived flows	Degradation when the sub-flows compete for the common bottleneck links
eNMCC [72]	TCP-fairness in both throughput growth and throughput reduction epochs	Accelerates throughput convergence to achieve fairness for short-lived flows	Resource under-utilization compared to LIA
ReMP TCP [73]	Redundant transmission on all sub-flows	Reducing end-to-end latency and jitter	Trading bandwidth for latency. Throughput is reduced due to redundancy
RED [74]	Redundant transmission on all sub-flows with prioritization	Considerable reduction to latency and jitter	Utilizing bandwidth for latency. Throughput is reduced due to redundancy
MPTCP-IoV [75]	Jointly uses forward error correction (FEC) and load balancing for coupled congestion control	Low delay which is suitable for safety applications in IoV	Situation where the network environment changes rapidly (as in IoV) requires further research
DEFT [18]	Delay-equalizing algorithm to minimize reordering delay in the receiver buffer in mmWave channels	Fast response to mmWave link change between line-of-sight and non-LOS	High computational complexity.

MPTCP reduces its rate on this path, which can be an unnecessary action because the wireless channel status could quickly change to a better condition. Therefore, MPTCP should be carefully designed in wireless networks. A large amount of research has been devoted recently to study, characterize and improve the performance of MPTCP in wireless networks and IoT.

In [78], a new congestion control algorithm called mVeno is introduced for wireless networks. The proposed mVeno controls the transmission rate on each sub-flow based on packet acknowledgements. Nonetheless, a mechanism is proposed to distinguish between packet losses due to errors in the wireless channel and packet losses that may occur due to congestion. The packet loss in the wireless channel is assumed to behave randomly. The mVeno algorithm can distinguish between the two cases, and the formula for updating the congestion window is different for each case. Fundamentally, the congestion window reduction in each sub-flow of the MPTCP is more when the packet loss is estimated to be due to congestion. The algorithm mVeno is able to provide better throughput in wireless environments than conventional congestion control algorithms.

The research in [79] presents an experimental study of three wireless network configurations, and employs three new congestion control algorithms. The first one is the CUBIC algorithm [16], which is a single-path TCP congestion control algorithm. The second and the third are LIA [15], and OLIA [56], which are described earlier in the chapter. The LIA and OLIA algorithms are designed for MPTCP. The experiments in the study are real-world scenarios (not just simulations). Wireless nodes in all cases are homed to two wireless networks: an LTE network and a Wi-Fi network. Several factors are changed including the receiver buffer size, the number of connections, the data size and the flow duration. The throughput performance is compared for all systems and for all configurations with the three congestion control algorithms. An important finding in this research is that the CUBIC congestion control, which is designed for single-flow TCP, outperformed the MPTCP congestion control algorithms. This shows that wireless networks add a new dimension to the designs of the congestion control algorithms that need to be taken into consideration.

The research in [80] studies a system of three users employing MPTCP. Each of the three users has access to two networks: a Wi-Fi access point and a 3G cellular base station. In this research, all the users are assumed to be stationary with a long-lived connection. In the cellular system, the users are allocated to three different channels, while for the Wi-Fi case they compete for the Wi-Fi channel using the CSMA-CA algorithm of the MAC layers. The network is modeled analytically and with computer simulation. The authors introduce an adaptive modification to the MPTCP LIA congestion control algorithm. This modification is such that the paths (i.e., the sub-flows) with longer one-way delay will have a smaller congestion windows increase than those in LIA when ACK is received. Consequently, high delay paths are less loaded. When applying the proposed algorithm on Wi-Fi and LTE paths, it takes into consideration the different loss and the delay characteristics of the Wi-Fi and the cellular connections. The new algorithm is adaptive to variations between Wi-Fi and LTE, as well as the variations for each of them, due to the wireless channel fading. However, the variations should be slow enough to allow convergence of the algorithm.

In [81], the same authors of [80] study the fairness between MPCTP users and single-path TCP users over the WiFi link. Single-path TCP users are added to the network studied in [80] to compete with the existing MPTCP users on the Wi-Fi connection. The research studies the fairness realized between the TCP and MPTCP users on the Wi-Fi. It is found by simulation that a considerable unfairness exists between the two types of users on the Wi-Fi link. Further, the research develops an Adaptive Loss Management (ALM) algorithm that is able to reduce the unfairness gap between the two types of user.

The research in [82] assumes a smart phone that is connected to the cellular network using LTE and 5G mmWave. As mentioned earlier, the mmWave signals typically require Line-of-sight (LOS) connection. In case of non-LOS, they suffer from high path-loss and blockage from objects, trees or even people. Hence, the network connection can suffer from sudden loss of the mmWave signal due to the sudden presence of an obstacle in the wireless link. When the LTE and 5G mmWave links are aggregated using MPTCP, considerable sudden performance degradation could occur due to the mmWave characteristics. The work in [82] proposes an Offloading by Restriction (OBR) algorithm which attempts to optimally offload the data packets from 5G to LTE when the 5G signal is blocked. The simulations show that the OBR algorithm outperforms other MPTCP congestion control algorithms in this scenario.

The research in [83] is interested in wireless IoT networks. This is due to the increasing demand for IoT networks as reported in [84]. To that end, the work in [83] considers a wireless system where IoT devices transfer their data to local Wi-Fi wireless gateways. The gateways relay the collected data from the different IoT devices to the corresponding management entities on the Internet through wireless networks. However, it is assumed that each gateway has access to both a cellular network and a wide coverage Wi-Fi network. Hence, the wireless gateways are multihomed to the cellular and Wi-Fi networks, and implement MPTCP to improve their throughput and reliability. In addition, the research measures the throughput with an actual testbed as well as with a simulated setup. The work concludes that the MPTCP LIA algorithm is unable to distribute the gateways' traffic efficiently over the two available wireless networks. Moreover, the work observed a large difference in throughput between the gateways when MPTCP is employed, which is not the case when regular TCP is employed. Consequently, it is concluded that further studies are needed on using MPTCP for wireless IoT networks.

In [85], an experimental setup is performed to study the performance of MPTCP in multi-hop wireless IoT networks. The first experiment includes double-homed devices with two hops Wi-Fi wireless networks. The second experiment includes one additional Wi-Fi router in one of the flows. Hence, one of the flows stays with two hops while the other flow employs 3 hops. The experiment includes varying the transmission power of the Wi-Fi routers and measuring the throughput. The experiment shows that reducing the signal strength does reduce the throughput (as expected). However, adding an additional router to compensate for the signal strength did not further improve the throughput.

The work in [86] employs wireless network architecture similar to [82]. A wireless device is multihomed to LTE and mmWave 5G networks and applies either TCP or MPTCP. The research evaluates the throughput in addition to the end-to-end delay at different scenarios and at different distances from the LTE and 5G mmWave base stations. The overall throughput increased due to the presence of the large bandwidth mmWave signal. However, the authors conclude that the current congestion control algorithm (BALIA) that is designed for MPTCP does not perform satisfactorily in mmWave environments. Similar to [82], this is due to the fact that the mmWave signal is subject to sudden drops in signal quality when LOS connection is lost. The authors also conclude that further research should be conducted with MPTCP in similar harsh wireless systems.

The research in [87] considers real-time video transmission over multihomed hand-held wireless device with several wireless access networks. In fact, the performance results assume that the device has three wireless networks: Wi-Fi, WiMAX and cellular LTE. The goal of the study is to design a scheduler that appropriately distributes the video stream over the 3 available MPTCP sub-flows. The hand-held device is assumed to be energy limited. Furthermore, the research presents a new algorithm named Delay-Energy-quality-Aware MPTCP (DEAM). This algorithm is developed through a Network Utility Maximization (NUM) optimization problem. Energy consumption of the hand-held device is the objective function to be minimized. The constraints are: 1) The target video quality; 2) the data rate per sub-flow; and 3) the target delay. In addition, the variable of optimization is the data rate on each sub-flow (Wi-Fi, WiMAX and cellular). It is shown that the new proposed algorithm provides superior performance in terms of video quality and energy conservation.

The research in [88] exploits MPTCP to support handoff in wireless mesh networks without additional hardware and with a backward compatible manner. Multihomed devices moving in a wireless mesh network employ MPTCP to get connected to multiple wireless networks. When moving, these devices perform seamless roaming (SERO) with layer-2 handoff (i.e., switching from one Wi-Fi access point to another) or vertical roaming (i.e., switching from Wi-Fi to cellular). With MPTCP, the device can drop one of its old connections (paths) and add a new one to perform handoff. The goal is to guarantee a minimum bandwidth during the handoff process. It is assumed that the MPTCP-enabled device has several Wi-Fi interfaces as well as a cellular interface. The device, as it moves, searches for better access points to associate with using MPTCP, or to associate with cellular when it cannot get the desired data rate from Wi-Fi.

In the previous paragraphs, the chapter presented a diverse range of research in MPTCP in wireless networks. Each study has its own goals and makes a unique solution to achieve these goals. Table 2.5 summarizes the wireless networks scenarios that are considered in the above-mentioned review. The table also outlines the main findings of each study.

2.7 Network Utility Maximization

The concept of Network Utility Maximization (NUM) in data networks was first introduced in [89] and [90]. The purpose of NUM is to optimize the flow control in the network. Consider a network (wired or wireless) which includes L links that are shared by S sources. Each source has a transmission rate x_s. Also, each link has a link capacity c_l. Consider a suitable utility function $U(x_s)$, where $U(x)$ is a concave function that is non decreasing and twice continuously differentiable. Each source $s \in S$ uses a set of links denoted by $L(s)$. Let $S(l)$ be the set of

Table 2.5 : Summary of the reviewed wireless networks research involving MPTCP

Reference	Wireless scenario considered	Main findings
[78]	Any number of concurrent wireless multipath streams.	New congestion control algorithm (mVeno) that can distinguish between wireless packet errors and congestion packet errors. Better throughput than conventional congestion control.
[79]	Wireless testbed with LTE and Wi-Fi. Implementing 3 congestion control algorithms: CUBIC, LIA and OLIA	MPTCP is clearly better than single-path TCP. However, in many scenarios the CUBIC (which is designed for single-path TCP) outperforms LIA and OLIA (which are designed for MPTCP). Hence, more attention needs to be given to congestion control in wireless networks.
[80]	Three MPTCP users with 3G cellular access (10 Mbps) and Wi-Fi (54 Mbps). The heterogeneity of the paths is considered.	New MPTCP design that is adaptive to wireless networks variation in throughput and delay due to wireless channel changes, as long as the variations are slower than the convergence speed of the algorithm.
[81]	Same model as [80] but with additional 3 TCP clients using a Wi-Fi network.	Unfairness between MPTCP users occurs. A design of Adaptive Loss Management in the Wi-Fi router is proposed to increase fairness.
[82]	One user equipment (UE) connected to LTE and 5G mmWave networks.	The mmWave network is vulnerable to channel dynamics. Proposed an Offloading algorithm that adaptively controls the congestion by shifting the traffic from 5G to LTE when 5G is in a bad state. Considerable throughput improvement achieved.
[83]	Testbed with several IoT gateways that are multihomed to LTE and Wi-Fi. Gateways implement MPTCP with OLIA congestion control. Several IoT devices connect to the gateways via Wi-Fi.	OLIA does not fairly and efficiently utilize the two wireless paths. Gateways have a large difference in terms of throughput. Concludes that MPTCP still needs further optimization in wireless networks.
[85]	An MPTCP source and destination are connected to two Wi-Fi paths. Each path consists of multi-hop routers. Two scenarios are considered: 1) each path has 2 hops, 2) one path has 2 hops and the other has 3 hops.	Decreasing the signal level negatively affects throughput. Adding one more hop to compensate for signal level reduction does not help. Further research on MPTCP in wireless networks is needed.
[86]	MPTCP wireless device is multihomed to LTE and 5G mmWave networks. BALIA congestion control is used.	Overall throughput increased when adding mmWave connection. However, the BALIA congestion control provides unsatisfactory performance with mmWave signal due to its fast variations in channel fading condition. Further research for MPTCP with mmWave is needed.
[87]	A Multihomed wireless device, with LTE, WiMAX and Wi-Fi connections, is considered to transmit real-time video.	Proposed the Delay-Energy-quality-aware MPTCP (DEAM) algorithm to provide good energy and quality trade-off. DEAM is better than existing algorithms
[88]	Employing MPTCP to investigate handoff when a wireless node is moving in Wi-Fi networks with a cellular connection.	MPTCP can provide a means for handoff that does not require additional hardware and it is backward compatible to existing networks. Sufficient throughput can always be achieved with optimized handoff decisions.

sources that use a specific link l. The goal is to find the set of rates x_s that are able to maximize the sum of the utilities $\sum_{s \in S} U(x_s)$ subject to the constraints of the capacities of the links in the network. This can be written as a NUM problem as follows:

$$\max_{s_x} \sum_{s \in S} U(s_x) \tag{2.5}$$

Subject to:

$$\sum_{s \in S(l)} x_s \leq c_l \quad , l = 1, 2, 3, \cdots, L$$

The set of L constraints indicate that the sum of rates crossing any path l, $l = 1, 2, .., L$, should not exceed the capacity of this path.

The above NUM problem forms the basis for more advanced cases that may include modifications to the objective function, the variable(s) of the maximization and the constraints. When considering wireless networks, more network resources can contribute to the maximization problem. The work in [27] provides a survey of the employment of the NUM concept for congestion control over wireless networks. Since the NUM may include several resources of the wireless networks that are typically managed by a different layer in the five layers of TCP/IP, the solution of the NUM is expected to lead to a cross-layer design. Here, this chapter focuses *as much as possible* on cross-layer based NUM systems that are MPTCP-enabled in their transport layer. It should be noted that the research in this area remains scarce. Therefore, this thesis is devoting Chapter 3 to mitigate this issue.

The authors in [91] extends NUM to a multi-path NUM model and considers MPTCP. The optimisation problem is solved after approximation using Jensen's inequality. The authors consider the case when the different paths have different RTTs. If each source has R_s available multi-paths, the original NUM problem becomes:

$$\max_{x_s} \sum_{s \in S} U \left(\sum_{i=1}^{R_s} x_{s,i} \right) \tag{2.6}$$

subject to:

$$\sum_{s \in S_{(l)}} \sum_{i: l \in R_{s,i}} x_{s,i} \leq c_l, \quad l = 1, 2, 3, \cdots, L$$

Since this problem is non-convex, it is approximated using Jensen's inequality and solved using the dual decomposition technique. The authors also propose a set of other MPTCP algorithms that are compatible with the regular TCP.

The work in [92] proposes a hop-by-hop algorithm for rate control (as opposed to end-to-end rate control) for wireless multipath transmission, since a hop-by-hop solution is expected to be more stable. This is particularly important for wireless links. The main optimization problem takes the form:

$$\max_{x_{sj}^{i}} \sum_{s=1}^{S} U_s \left(\sum_{j=1}^{J_s} x_{sj}^{H_{sj}+1} \right) \tag{2.7}$$

subject to:

$$m_s < x_{sj}^{i} < M_s$$

$$x_{sj}^{i+1} = x_{sj}^{i} \left(1 - P_r^{(i)}\right), i = 1 : H_{sj}, \forall S, J \in J(s)$$

$$Pr \left(\sum_{s \in S(l), j \in J(l)} x_{sj}^{l} > c_l \right) \leq \varepsilon_l, \forall l$$

The variables in the optimization problem are defined in the following table [92]:

S	Number of sources	c_l	Capacity of link l
L	Number of links	m_s	Minimal data rate
$J(s)$	Number of transmission paths of source s	M_s	Maximal data rate
$L(s)$	Set of links used by source s	ε_l	Rate-outage probability threshold of link l
$S(l)$	Set of sources using link l	P_r^L	Rate outage probability of link l
$J(l)$	Set of transmission paths using link l	H_{sj}	Number of hops traversed by source s on path j
x_{sj}^i	Data rate of source s, sub-flow j on link i	Pr	Probability of the event

The above optimization problem is not convex. Hence, it is approximated using Jensen's inequality and solved using the Lagrangian duality algorithm. The proposed algorithm proved to be better than single-path end-to-end rate control algorithms.

On the other hand, the research reported in [93] presents a design for an end-to-end congestion control algorithm for a system with a wireless last hop. The research in [93] argues that the conventional congestion control methods, which depend on packet loss and packet acknowledgement to adjust a congestion window, are not optimum for wireless channels. Instead, the authors considered packet queue length and queuing delays as a congestion signal. The main NUM problem is given by:

$$\max_{x_s} \sum_{s \in S}^{S} p_s log x_s \tag{2.8}$$

subject to:

$$\sum_{s \in S(l)} x_s \leq c_l, \forall l \in L_f$$

$$\sum_{s \in S(l)} x_s \leq r_l, \forall l \in L_w$$

$$r_l \in R, \forall l \in L_w$$

where p_s is a weight factor, c_l is the capacity of the wired links, r_l is the capacity of the wireless link, L_f is the number of wired links, L_w is the number of wireless links and R is the average wireless rate over all possible fading channel realizations. The problem is simplified and solved using the Lagrangian dual algorithm. An optimal window-based TCP congestion control is developed. A cross-layer design of TCP congestion control and the wireless scheduling of the MAC layer is provided. While this research considers single-path TCP, the authors conclude that an extension to MPTCP is possible.

The research reported in [94] is interested in wireless multi-hop mesh networks. It proposes a flow level resource allocation (FLRA) algorithm using cross-layer design. This algorithm maximizes the network utility and at the same time puts in a delay requirement. FLRA considers the MAC layer as well as the TCP layer. In the MAC layer, it optimizes the transmission scheduling using adaptive Carrier Sense Multiple Access (CSMA). At the transport layer, the data rate is adaptively controlled based on the path load and the required delay constraint. Simulation results demonstrate the effectiveness of the FLRA algorithm in terms of the convergence speed to the optimal solution.

In [95], the authors employ NUM to develop a cross-layer design for wireless communication networks under heterogeneous traffic scenarios. The traffic may include unicast, multicast or broadcast flows. The authors extend an old method denoted as universal control policy (UMW). In fact, UNW is applicable only to unicast flows. Hence, the new proposed algorithm is called UMW+. UNW+ is applicable to unicast, multicast or broadcast flows. The cross-layer design includes an optimized admission control as well as routing and scheduling policies. The

UMW+ algorithm is designed using the approach of a precedence-relaxed virtual queuing network. The authors present extensive simulation results to validate the analytical results of the new algorithm.

The work in [96] considers a multi-hop wireless ad-hoc network. Typically, for ad-hoc networks, the wireless devices have limited power resources. Examples of such devices are the wireless-sensor networks. In addition, the work formulates the NUM with power constraints and performs joint congestion control and power allocation. The solution of the optimization problem is based on a new concept of the primal-dual interior-point optimization method. It is shown that this new method is able to converge faster than the conventional methods employed in the literature. Moreover, simulation results demonstrate that the proposed algorithm increases the energy efficiency of the network.

Table 2.6 provides a summary of the NUM scenarios presented above. It should be noted that the research that exploits NUM techniques with MPTCP is limited. Hence, only [91] and [92] consider MPTCP. The other works have potential benefits to MPTCP that needs further research.

Table 2.6 : Summary of the NUM research involving MPTCP or potentially extensible to MPTCP

Reference	Description	Main findings
[91]	Introduce a multipath TCP NUM model.	Jensen inequality and dual decomposition technique are used to solve the NUM problem.
[92]	Introduce hop-by-hop rate control algorithm instead of end-to-end- rate control.	The optimization problem is solved using Jensen inequality and Lagrangian duality technique.
[93]	Single path TCP is considered. TCP end-to-end congestion control for wireless last hop. Packet queue length and queuing delays are taken as a congestion signal.	Find an optimal window-based TCP congestion control algorithm for wireless networks. Resulting in an optimal cross-layer design involving TCP and MAC layers.
[94]	Maximize network utility in wireless mesh networks with single-path TCP. Propose a distributed and adaptive Flow-Level cross-layer Resource Allocation (FLRA) algorithm.	FLRA dynamically optimizes the CSMA in the MAC layer as well the data rate of each source at the transport layer. The utility is maximized with the required delay constraints.
[95]	Consider NUM with various traffic types: unicast, multicast or broadcast flows. Also, propose cross-layer design that includes an optimized admission control as well as routing and scheduling policies.	A new algorithm UMW+ is analytically designed using the approach of precedence-relaxed virtual queuing network. Confirmed with simulation results.
[96]	Consider a multi-hop wireless ad-hoc network. Formulate NUM with power constraints and performs joint congestion control and power allocation.	Fast convergence achieved with the primal-dual interior-point optimization method. The model is energy efficient.

2.8 Summary

In this chapter, the author reviewed the IoT and how it relates to a wireless network and the work of the thesis. Also, a brief description of TCP that is widely used as a transport layer protocol in the TCP/IP model was provided. From this description, the chapter derived the motivations to Multipath TCP. Further, common multipath protocols, specifically SCTP and CMT-SCTP were presented. Their limitations were also highlighted. Moreover, the main characteristics and goals of MPTCP and its potential weakness points that need further research were described. Then, several key published works in the literature that attempt to address three key challenges, namely, congestion control, packet scheduling and latency were introduced. For each challenge, the key differentiating points for the algorithms that tackle this challenge as well as the advantages and disadvantages of each algorithm were presented. After presenting the key algorithms in general networks, the chapter turned attention to the implementation of MPTCP in wireless networks and IoT. This is because MPTCP is likely to be implemented in wireless networks, where the transmitting and receiving nodes can easily have access to several wireless connections simultaneously, such as 4G, 5G, Wi-Fi, WiMAX, etc. The chapter outlined several wireless implementations and scenarios that are presented in the literature, with the main characteristics, challenges and proposed solutions for each wireless scenario. Finally, the chapter introduced an important analytical tool that is widely used for optimizing the performance of data networks using cross-layer designs. The mathematical basis of this tool, which is known as Network Utility Maximization (NUM), was outlined first. This was followed by some of the implementations and utilizations of NUM in MPTCP to demonstrate its importance and effectiveness for MPTCP cross-layer design research. In the next chapter, the author introduces a cross-layer design as the first contribution in this thesis.

Chapter 3

Cross-layer Multipath Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks

This chapter presents a cross layer model for multipath ad hoc wireless networks. A utility maximization problem is considered with rate, scheduling and the end-to-end queuing delay constraints. The optimisation problem is solved using the Lagrangian and two scheduling algorithms (perfect and distributed algorithms) are proposed. The performance of the scheme is evaluated. Although the framework is designed for multipath ad hoc wireless networks, it is applicable to Multipath TCP (MPTCP) and Internet of Things (IoT) wireless networks. In addition, it should be mentioned that the author published the content of this chapter in *IET Communication*, as a journal article, and it is cited in [97].

The structure of the chapter is as follows. Section 4.1 provides an introduction. Section 3.2 reviews some background and related works available in the literature. In Section 3.3, the system model is shown. In Section 3.4, the cross-layer design in fixed channels is proposed and the algorithm convergence is derived. Section 3.5 further analyses the algorithm convergence based on the proposed cross-layer design in time varying channels. Section 3.6 reports the obtained results. Finally, the chapter is summarised in Section 3.7.

3.1 Introduction

With the recent advancements in wireless communication technologies and developments of the IoT, efficient multipath transmission protocols are gaining more interest [98]. A common example is the ubiquitous multihomed terminals which support wireless communications, data processing and rich multimedia [99]. These devices are very common and suitable benchmarks for cross layer interaction and multipath congestion control [100]. Another potential area is smart city which has played an important role in academic and industrial fields (e.g., see [101]) in the last few years. It involves the development and deployment of different IoT-based infrastructures and wireless sensor networks with available multipath connection [102].

In an effort to address such demands, there is a growing interest in efficient cross-layer designs. Cross layer design protocols have emerged to provide an integrated scheme to enable interaction, coordination and joint optimization among different layers [103]. For instance, the cross layer design employed in order to maximize user utility for internet congestion control in the wireless networks were studied in the literature [104–106]. In [107], physical (PHY) and medium access control (MAC) layers were taken into consideration to provide stable wireless connection among moving, portable and fixed stations. In [108], Rajesh and Gnanasekar studied congestion control in the heterogeneous wireless ad hoc networks by considering data priorities. In [109] [110], the authors studied a joint routing, power allocation policy and the influence of interference on multi-hop wireless network performance. Similarly, in [111] [112], a joint routing and scheduling problem was studied by excluding the transport layer. In [113], Chen *et al.* proposed a joint design of congestion control, routing and scheduling for ad hoc wireless networks by considering network utility maximization.

The above works considered only a single path connection, but ignored the availability of multi-path in the network. Multipath congestion control methods are usually developed as a joint design of transport layer and multipath routing at the network layer [52]. In [114], Mishra *et al.* proposed a packet scheduling algorithm and formulated the MPTCP problem, but they neglected the routing problem of the network layer.

In addition, it is practical to consider queuing delay when designing cross-layer protocols. Queuing delay is defined as the waiting time in a queue for a packet to be served [115]. The number of waiting packets in the buffer is counted by each separate node. This will allow an estimate of the expected queuing delay [116]. As a result, fewer packets in the queues leads to smaller end-to-end delays [117]. Reducing such delays can also be beneficial in load-balancing schemes to prevent congestion [118]. Thus, end-to-end queuing delay in multipath protocols plays an important role to provide a robust connection and improve the network performance.

In this chapter, a utility maximization problem is considered for multipath ad hoc wireless networks via joint cross layer congestion control, routing and scheduling [1]. The chapter considers the transport layer, network layer and MAC

[1] This cross-layer scheme is designed for multipath transmission protocols, thus, it is applicable to MPTCP. It is also applicable to IoT networks since they can be in a form of ad hoc networks.

layer for the following reasons. First, in wireless networks, the medium is shared, and each node competes to have radio access which result in, among other issues, contention and interference problems in the data link layer. Second, the default routing procedure does not take into account the conflict, if any, in the data link layer, which can degrade the network performance. Third, the transport layer presumes that the task in network and data link layers is well-maintained which is not always the case. Figure 3.1 shows the overview of the proposed cross layer scheme. In the transport layer, sending rate is adjusted based on calculated congestion price. The transport layer information is sent directly to the MAC layer and the MAC layer accordingly performs the scheduling task and allocates link capacity. Further, the MAC layer provides the information to the network layer and based on that, the best route is selected. In addition, this chapter extends the work in [113] to adapt multipath transmission and queuing efficiency. The formulation of the optimisation problem with respect to source rate, scheduling and queuing delay constraints in fixed and time varying channels are presented. The Lagrangian is employed in this study to solve the optimisation problem and two methods of scheduling namely, *Perfect Scheduling* and *Distributed Scheduling* are proposed. All nodes in the perfect scheduling algorithm participate in the scheduling process. In the distributed scheduling, neighbour nodes only contribute to the scheduling. Moreover, the global convergence is mathematically proved for the above methods. The results show that the proposed algorithms outperform existing schemes in terms of enhancing source rate and decreasing congestion price.

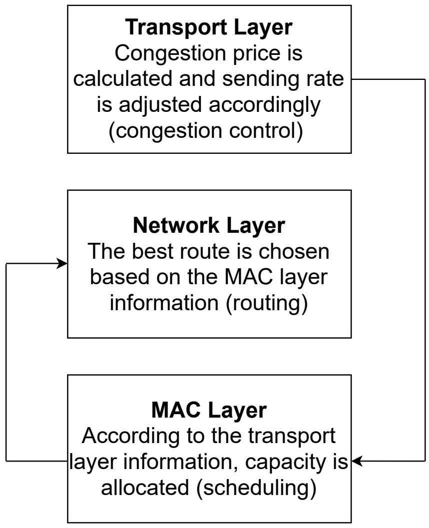

Fig. 3.1 The overall concept of the proposed cross layer design.

3.2 Background and Related Works

After Internet Protocol (IP) and Transmission Control Protocol (TCP) were introduced in the 1970s, they were investigated in the literature due to their major functionality of exchanging data packets and providing a reliable byte stream service in [119] [120]. However, as discussed in Chapter 1, single-path transport protocols such as TCP, suffer from several efficiency-related issues, e.g. failure to support multihoming and bandwidth aggregation [121]. Due to this motivation, MPTCP was proposed to provide efficient data transmission in such environments [122]. Also, Stream Control Transport Protocol (SCTP) was developed to adapt multipath transmission [25]. Multipath protocols adapt fault tolerance, i.e different paths can be assigned for different tasks. For instance, the fast paths can be used for data transmission while the slow paths could be utilised for a backup process [14]. Moreover, it has been shown that the connection-level throughput region of multi-path routing/congestion control algorithms can be larger than that of a single-path congestion control scheme [123] [124].

Various studies have been proposed to develop multi-path routing/congestion control problem including the Host Identity Protocol and Mobile IP [125]. However, these solutions usually have negative impacts on the congestion price, scheduling and the possible conflict with regular TCP for existing applications. The approaches to address the multipath congestion control issues can be divided into two major categories. First, the application-based approach, where software is developed and tested on an applied benchmark while neglecting the mathematical formulation, convergence and stability analysis of the algorithm. For instance, a Linux based software was developed in [122]. As another example, in [126], Cui *et al.* proposed a fountain code based MPTCP algorithm and they report the validation of the algorithm through simulations. The main idea behind this algorithm is to use the random nature of fountain code in transmitting data through different subflows. Further, Zhang et al. introduce a cross layer design which benefits from the exchange information between MAC and transport layer [127]. This scheme takes into account path condition before data transfer. The paths are evaluated according to the Frame Error Rate (FER) method and the best path is considered first for transmission. This algorithm is verified over some simulation scenarios. In [128], a Reward and Penalize window adaption flow control mechanism for video delivery is reported. To estimate the wireless channel quality, effective signal-to-noise (ESNR) is calculated at the data link layer and is reported to the transport layer. At the transport layer, the average rate and efficiency of each path is estimated. It is worth mentioning that parallel transferring and fairness towards other TCP flows are also considered. In [13], Sharma et al. propose a cross layer design for MPTCP. Based on MAC layer information, the protocol employs a flexible policy for transferring data. The main objectives of this study are improving throughput and end-to-end delay which are verified through simulation. There are also some experimental studies associated with MPTCP in [129] which can be considered within the application-based category.

The second approach utilizes mathematical formulation, graph theory and optimization problems for a given network topology. As a result, the stability of the algorithm and its convergence to the optimal solution are mathematically analyzed and discussed. For example, in [113], a jointly optimal design of cross layer congestion control, routing and scheduling was proposed for single path connection in the context of ad-hoc wireless networks. In [130], Eryilmaz and Srikant analysed the stability of the congestion control by using LaSalle's invariant principle. In [131], Zhou *et al.* proposed a cross-layer design considering the rate control, the medium access control and the routing problem for the cooperative vehicular ad hoc network. Considering the flow rate allocation [132], Ploumidis *et al.* proposed a distributed scheme to maximize the aggregate flow throughput. The multi-hop networks utility maximization was studied in [115] [133] [134]. Nevertheless, this literature is mainly limited to single rate congestion control. To the best of the author's knowledge, there are few works focusing on multipath congestion control using the second approach. In [91], Vo *et al.* studied the congestion prices of a multipath TCP network, however, the joint congestion control, routing and scheduling is not considered. In this chapter, the author follows the second approach and considers joint congestion control, routing and scheduling design for multipath ad hoc wireless networks.

3.3 System Model

This chapter considers an ad-hoc wireless network including a set of N nodes and a set of L links. The chapter assumes that the network links are directed with symmetric connectivity, i.e., link $(j,i) \in L$ if and only if $(i,j) \in L$. Each link $l \in L$ has a fixed link capacity c_l [2]. It is further assumed that the nodes are randomly distributed in the network area while each node has z number of available channels for transmission.

Due to scarce resources in the wireless spectrum, connection links compete with each other to obtain access to the channel. This chapter considers the conflict graph (also called contention graph [135]) to observe contention relations between the links. In addition, the primary interference in a network, i.e. the links that share a node cannot transmit at the same time, is considered.

Fig. 3.2 illustrates an ad-hoc wireless network with five nodes and eight links and the corresponding conflict graph. In this graph, each vertex donates a link, and the edges represent the contention between the links. For a given conflict graph, a set of vertices that have no edges between each other constructs an independent set which are able to transmit simultaneously. Assume that H denotes the independent set. An independent set $h \in H$ can be represented by a $|L|$ dimensional rate vector b^h and for any link l, b_l^h is given by

$$b_l^h = \begin{cases} c_l \; if \; l \in h \\ 0 \; otherwise \end{cases} \tag{3.1}$$

Referring to [113], at the link layer, the feasible rate region R can be described by the convex hull of these rate vectors. Thus,

$$R := \{b : b = \sum_h a_h b^h, a_h \geq 0, \sum_h a_h \quad = 1\} \tag{3.2}$$

Suppose $k \in D$ represents the set of destination node of network layer flow. Also, let $f_{i,j}^k$ denote the capacity of link (i, j) devoted to the flow to destination k. Therefore, the aggregate capacity on link (i, j) can be expressed as $f_{i,j} = \sum_{k \in D} f_{i,j}^k$.

[2]This assumption will be relaxed in Section 3.5.

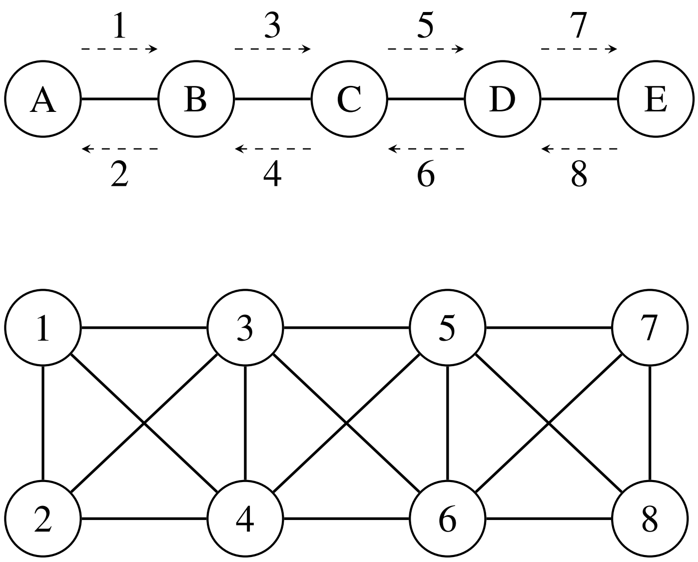

Fig. 3.2 An ad hoc wireless network with its conflict graph.

Considering the multipath opportunities, the aggregate capacity of all paths available to the destination k can be written as $f_{i,j,z} := \sum_{z\in Z}\sum_{k\in D} f^k_{i,j,z}$ where Z denotes the set of all paths. Based on this system model, $f := f_{i,j,z}$ should satisfy the first constraint

$$f \in R \tag{3.3}$$

The physical meaning of this constraint is that the capacity allocated to the flows f through all links (i, j) and available channels z to destination k should meet the feasible rate region's requirements at the MAC layer. Let x^k_i indicate the network flow created at node i to the destination k. The difference between the aggregate capacity for all its outgoing flows and the aggregate capacity for all its incoming flows should be greater than or equal to x^k_i. Mathematically speaking, this constraint can be expressed as

$$x^k_i \leq \sum_{z\in Z}\sum_{j:(i,j)\in L} f^k_{i,j,z} - \sum_{z\in Z}\sum_{j:(j,i)\in L} f^k_{j,i,z} \tag{3.4}$$

Multipath transmission and M/M/m queuing system as the queuing delay model (see Fig. 3.3) are considered in this chapter. In the M/M/m queuing system, m represents the number of servers allocated for customer service. This should satisfy the condition of $m > 1$. In this chapter, it is assumed that the number of servers m denotes the number of transmitters in each node. The packets' arrival in M/M/m follows a Poisson process with mean arrival rate of λ_0. In this model, $\lambda_0 = \sum_i x_j^i$ which is the sum of incoming flow rate to node i. Additionally, let P_u represent the equilibrium probability that there are u packets in the queuing system. Thus, equalizing the flow between $u-1$ and u transition states can be written as [136][137]

$$\sum_i x_j^i P_{u-1} = \min(u,m)\mu_0 P_u, \quad u = 1,2,... \tag{3.5}$$

where the service time μ_0 is exponentially distributed with the mean value of $1/\mu_0$. By iterating the above equation, it can be stated that [136][137]

$$P_u = \frac{(m\rho)^u}{u!} P_0, \quad u = 0,1,...,m \tag{3.6}$$

as well as

$$P_{u+m} = \rho^u P_m = \rho^u \frac{(m\rho)^m}{m!} P_0, \quad u = 0,1,2,... \tag{3.7}$$

where P_0 refer to the probability that there is zero packet in the system. Therefore, considering normalisation in [136][137] gives

$$\begin{aligned} P_0 &= \left(\sum_{u=0}^{m-1} \frac{1}{u!} (m\rho)^u + \frac{1}{m!} \frac{1}{1-\rho} (m\rho)^m \right)^{-1} \\ \rho &= \frac{\sum_i x_j^i}{m\mu_0} \end{aligned} \tag{3.8}$$

In queuing systems, Little's theorem defines $Y = \lambda_0 \,.\, t_s$ where Y and t_s is the average number of packets and the average amount of time a packet spends in the system, respectively [138][137]. Note that $\lambda_0 = \sum_i x_j^i$, thus,

$$t_s = \frac{Y}{\sum_i x_j^i} \tag{3.9}$$

Also, M/M/m queuing model confirms that $Y = L_q + m\rho$, therefore [138][137]

$$t_s = \frac{1}{\sum_i x_j^i}(L_q + m\rho) \tag{3.10}$$

where L_q and $m\rho$ denote the average number of packets in the queue and the average number of packets in the transmitters, respectively. Furthermore, M/M/m queue defines L_q as [138][137]

$$L_q = P_q \frac{\rho}{1-\rho} \tag{3.11}$$

where

$$P_q = \frac{(m\rho)^m P_0}{m!(1-\rho)}, \quad u > m \tag{3.12}$$

Considering equations (3.11) and (3.12), the mean stay time for a packet, which includes waiting and service times, in equation (3.10) can be expressed as [116][138]

$$t_s = \frac{1}{\sum_i x_j^i}\left(\frac{(m\rho)^m \rho}{m!(1-\rho)^2} P_0 + m\rho\right), \quad \rho < 1 \tag{3.13}$$

where

$$\begin{aligned} P_0 &= \left(\sum_{u=0}^{m-1} \frac{1}{u!}(m\rho)^u + \frac{1}{m!}\frac{1}{1-\rho}(m\rho)^m\right)^{-1} \\ \rho &= \frac{\sum_i x_j^i}{m\mu_0} \end{aligned} \tag{3.14}$$

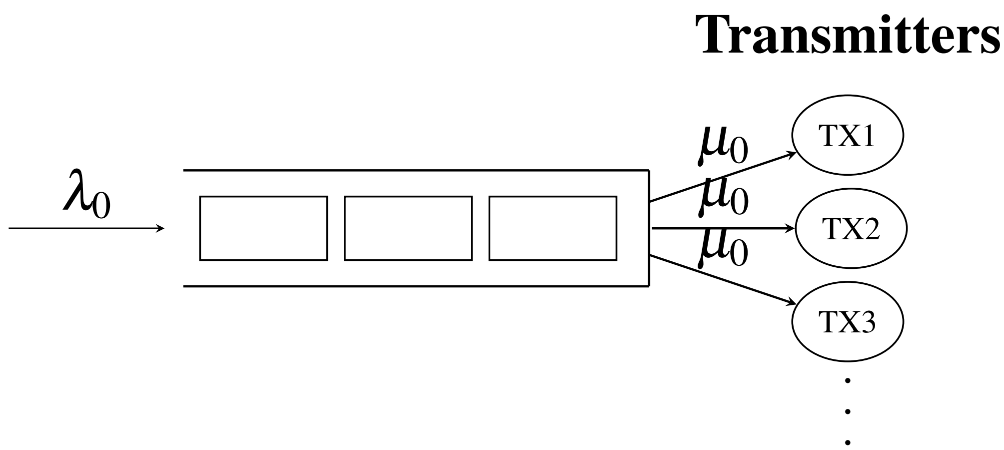

Fig. 3.3 Illustration of M/M/m queuing system.

As mentioned earlier, nodes have a number of channels available for transmission. Since multipath transmission is considered, the number of channels z is the number of all paths Z. Let $s \in S$ indicate the set of source nodes. Every node can be either a traffic source node s or a relay node sr that forwards packets to the required destination. The number of traffic flows that pass the node is defined as the Pass Route Number (PRN). Thus, the mean value of PRN can be defined as [116]

$$sr = \frac{Zn(\bar{o}-1)}{n} = Z(\bar{o}-1) \tag{3.15}$$

where $\bar{o}$ and n are the average number of hops from source to destination and the total number of nodes in the network, respectively. Furthermore, *concurrent transmission ratio (v)* is defined as the *concurrent transmission capacity (G)* over the total number of nodes in the network (i.e., n) and can be expressed as [116]

$$v = \frac{G}{n} \tag{3.16}$$

Therefore, the packets' arrival follows a Poisson process at each relay node with the arrival rate of $sr\sum_i x_j^i$. However, if the node is a destination node, the arrival of packets also follows a Poisson process but with the arrival rate of $Z\sum_i x_j^i$, where packets are transmitted to the destination through different Z paths simultaneously. Hence, in general, the packet's arrival is following a Poisson process with parameter $sr\sum_i x_j^i + Z\sum_i x_j^i = Z\bar{o}\sum_i x_j^i$. Since $v = G/n$, all nodes can be split into $1/v$ sets and each set is given a time slot to ensure concurrent transmission. Hence, the service time follows an exponential distribution and the mean service time is $1/\mu v$ for each node transmitting once every v time slots [116]. In addition, the mean queuing delay per hop $(\bar{D}_i)$ can be defined as

$$\bar{D}_i = \frac{1}{Z\bar{o}\sum_i x_j^i}\left(\frac{(m\rho)^m\rho}{m!(1-\rho)^2}P_0 + m\rho\right),\ \rho < 1 \tag{3.17}$$

where

$$\rho = \frac{Z\bar{o}\sum_i x_j^i}{m\mu v} \tag{3.18}$$

Since $\bar{o}$ denotes the average number of hops from source to destination, thus, the end-to-end queuing delay can be expressed as $\sum_i^{\bar{o}} \bar{D}_i$.

Let $U_s(x_s)$ denote the user utility at source s while the sending rate is x_s. The utility maximization problem can be written as

$$\begin{aligned}
&\max_{x_s \geq 0, f_{i,j,z}^k \geq 0} \sum_s U_s(x_s) \\
\text{s.t.}\quad & x_i^k \leq \sum_{z\in Z}\sum_{j:(i,j)\in L} f_{i,j,z}^k - \sum_{z\in Z}\sum_{j:(j,i)\in L} f_{j,i,z}^k,\ \ \forall i,k \\
& f \in R \\
& \bar{D}_i \leq t_{th}
\end{aligned} \tag{3.19}$$

where $i \in N, k \in D, i \neq k, x_i^k = 0$ if $(i,k) \notin S \times D$.

The delay constraint in (3.19) can be shown to be equivalent to a linear constraint, i.e., $\sum_i x_j^i \leq E_i^k$, where E_i^k is a function of t_{th}. Note that all the constraints in (3.19) are linear. Further, the utility function is concave, thus, the objective function in (3.19) is concave as well. As a result, problem (3.19) is a convex function, which can be solved to its optimality via the Lagrangian.

3.4 Cross-layer Design in Fixed Channels

The congestion control problem plays a critical role in the multipath transport protocols. The optimality and stability of the proposed algorithm for solving the congestion control problem is an important issue that should be considered. It should be noted that the design of the multipath congestion control algorithm depends on the problem under consideration. In fact, the utility function $U_S(x_s)$ is not a function of the flows therefore, they can be separated. The maximization over x_s and $f^k_{i,j,z}$ can be performed independently in the primal problem (i.e., (3.19)). Thus, the problem using the Lagrangian can be expressed as

$$\min_{p\geq 0} Du(p) \tag{3.20}$$

where p denotes a set of consensus variables and p^k_i is Lagrange multiplier for node $i(i \in N)$ and destination $k(k \in D)$.

To perform the congestion control and joint routing and scheduling, consider the following Lagrangian relaxations

$$\begin{aligned} Du(p) = & \max_{x_s \geq 0, f^k_{i,j,z} \geq 0} \Bigg\{ \sum_s U_s(x_s) \\ & - \sum_{i \in N, k \in D, i \neq k} p^k_i \Bigg(x^k_i \\ & - \sum_{z \in Z} \sum_{j:(i,j) \in L} f^k_{i,j,z} \\ & - \sum_{z \in Z} \sum_{j:(j,i) \in L} f^k_{j,i,z} \Bigg) \\ & - \sum_{i \in N, k \in D, i \neq k} q^k_i \left(\sum_i x^i_j - E^k_i \right) \Bigg\} \end{aligned} \tag{3.21}$$

s.t. $\quad f \in R,$

Considering that a flow generated in the system can only be emitted from components of source nodes set, (3.21) can be written as

$$Du_1(p) = \max_{x_s \geq 0} \left\{ \sum_s U_s(x_s) - \sum_s \left(p_s + \sum_i q^k_i \right) x_s \right\} \tag{3.22}$$

where $p = \left\{ p_s, \Sigma_i q^k_i \right\}$ and p_s denotes the multipliers p^k_i if $[i,k] \in S \times D$, and

$$Du_2(p) = \max_{f^k_{i,j,z} \geq 0} \left\{ \sum_{i,k} p^k_i \left(\sum_{z \in Z} \sum_{j:(i,j) \in L} f^k_{ij,z} - \sum_{z \in Z} \sum_{j:(j,i) \in L} f^k_{j,i,z} \right) \right\} \tag{3.23}$$

s.t. $\quad f \in R,$

Here, decomposition method is employed to perform cross-layer optimization. Therefore, according to (3.22), the congestion control can be expressed as

$$x_s(p) = {U'}_s^{-1}\left(p_s + \sum_i q_i^k\right) \tag{3.24}$$

in which a unique maximizer regarding the queuing delay is subjected. It should be noted that in (3.23), (i,j) is a bilateral node pair in L. In other words, if $(i,j) \in L$, consequently $(j,i) \in L$ and vice versa. Further, each link capacity is assigned if and only if sequence of (i,j) and z are specified. Hence, (3.23) can be written as

$$\begin{aligned} Du_2(p) = & \max_{f_{i,j,z,k} \geq 0} \sum_{z \in Z} \sum_{i,j} f_{i,j,z}\left(p_i^k - p_j^k\right) \\ \text{s.t.} \quad & f \in R \end{aligned} \tag{3.25}$$

In order to undertake scheduling, the first step is to search through destinations. The optimal destination for each joint scheduling can be written as $k^* \in \arg\max_k \left(p_i^k - p_j^k\right)$. Let $d_{i,j}$ denote the destination k differential congestion price over nodes i,j. Considering k^*, the differential price can be $d_{i,j} = (p_i^{k^*} - p_j^{k^*})$. Therefore, the scheduling demands can be expressed as

$$\tilde{f} \in \arg\max_{f \in R} \sum_{z \in Z} \sum_{(i,j) \in L} d_{i,j} f_{i,j,z} \tag{3.26}$$

Choosing an extreme point maximizer, each routing link launches packets toward the optimal destination as follows

$$f_{i,j,z}^k = \begin{cases} \tilde{f}_{i,j,z} & \text{if } k = k^* \\ 0 & \text{if } k \neq k^* \end{cases} \tag{3.27}$$

Noting that (3.25) is a piece-wise linear and not differentiable, hence, the Lagrangian function in (3.20) is not differentiable either. By introducing the following function

$$
\begin{aligned}
\vartheta_{i,z}^{k}(p) = &\left(\sum_{z}\sum_{j} f_{i,j,z}^{k}(p) - \sum_{z}\sum_{j} f_{j,i,z}^{k}(p) - x_{i}^{k}(p)\right) \\
&- \sum_{i\in N,k\in D,i\neq k} q_{i}^{k}\left(\sum_{i} x_{j}^{i} - E_{i}^{k}\right)
\end{aligned}
$$

as a subgradient of the Lagrangian function and utilizing the subgradient method in [139], the congestion price of each node can be expressed as a non-negative real value of

$$
\begin{aligned}
p_{i,z}^{k}(t+1) = &\Bigg[\Bigg(p_{i}^{k}(t) + \gamma\Bigg(x_{i}^{k}(p(t)) \\
&- \Bigg(\sum_{z\in Z}\sum_{j:(i,j)\in L} f_{ij,z}^{k}(p(t)) \\
&- \sum_{z\in z}\sum_{j:(j,i)\in L} f_{j,i,z}^{k}(p(t))\Bigg) \\
&- \sum_{i\in N,k\in D,i\neq k} q_{i}^{k}\Bigg(\sum_{i} x_{j}^{i} - E_{i}^{k}\Bigg)\Bigg)\Bigg)\Bigg]^{+}
\end{aligned}
\tag{3.28}
$$

where γ is a positive scalar step size and "+" is the projection of $\Re^{+}$ set [113]. Therefore, the problem for the joint optimal design of cross layer congestion control, scheduling and routing is solved. The pseudo-code of the proposed solution for perfect scheduling is outlined in Algorithm 1.

Algorithm 1 : Perfect Scheduling over Network with Fixed Channels

for each time t **do**
    Update price for each node i to destination k according (3.28).
    Each node sends its price information to its neighbours
    for each source s **do**
        $x_s(p) = {U'}_s^{-1}\left(p_s + \sum_i q_i^k\right)$
    end for
    Find $k^* \in \arg\max_k \sum_{z\in Z}\left(p_i^k - p_j^k\right)$ for each node according to the adjacent nodes and obtain $d_{i,j} = \sum_z \left(p_i^{k^*} - p_j^{k^*}\right)$
    Assign respective capacity to each link according to $\tilde{f} \in \arg\max_{f\in R} \sum_{z\in Z} \sum_{(i,j)\in L} d_{i,j}(t) f_{i,j,z}$
    According to assigned capacities, generate the routing at each link toward the considered destination.
end for

3.4.1 Algorithm Convergence

In this section, the chapter investigates the convergence of the proposed algorithm. Suppose $\bar{p}(t) := \frac{\sum_{\tau=1}^{t} p(\tau)}{t}$ is the average congestion price at time t. The convergence can be obtained by proving

$$\limsup_{t\to\infty} Du(\bar{p}(t)) - Du^*(p^*) < \varepsilon : \text{ for any positive real value of } \varepsilon \tag{3.29}$$

where Du^* and p^* denote the optimal target and optimal price, respectively. Recalling the subgradient method, for a small enough step size γ, gives

$$\begin{aligned} \|p(t+1) - p^*\|_2^2 \le \|p(t) - p^*\|_2^2 - 2\gamma\vartheta(p(t))^T \|p(t) - p^*\| \\ + \gamma^2 \|g(p(t))\|_2^2 \end{aligned} \tag{3.30}$$

In fact, (3.30) can be written as

$$\begin{aligned}\|p(t+1)-p^*\|_2^2 \leq \|p(t)-p^*\|_2^2 \\ -2\gamma(Du(p(t))-Du^*)\,\|p(t)-p^*\| \\ +\gamma^2\|g(p(t))\|_2^2\end{aligned} \tag{3.31}$$

As $\|p(t+1)-p^*\|_2^2 > 0$ and according to subspace in each reclusive duration along the corresponding space, (3.31) can lead to

$$Du(\bar{p}(t)) - Du^*\,(p^*) \leq \frac{\|p(t)-p^*\|_2^2}{2\gamma t} + \frac{\gamma \sum_{n=1}^{t} \|\vartheta(p(n))\|_2^2}{2t} \tag{3.32}$$

If $\|\vartheta(p(n))\|_2 \leq Q$, and since Du is a convex function, by Jensen's inequality, it can be concluded that $Du(\bar{p}(t)) - Du^*\,(p^*) \leq \frac{\gamma Q^2}{2}$. In other words, it is proved that the algorithm converges statistically to within $\frac{\gamma Q^2}{2}$ of the optimal value. Hence according to (3.24), convergence of the congestion price is guaranteed.

Proof of algorithm convergence:

Let p^* and Q denote the optimal price and the upper bound, respectively. By considering [140] and recalling equation (3.28) in section 3.4, it can be stated that

$$\begin{aligned}\|p(t+1)-p^*\|_2^2 = \left[\|p(t)-\gamma\vartheta(p(t))\right]^+ - p^*\big\|_2^2 \leq \|p(t)-\gamma\vartheta(p(t))-p^*\|_2^2 \\ = \|p(t)-p^*\|_2^2 - 2\gamma\vartheta(p(t))^T(p(t)-p^*) \\ +\gamma^2\,\|\vartheta(p(t))\|_2^2 \leq \|p(t)-p^*\|_2^2 \\ -2\gamma(Du(p(t))-Du(p^*)+\gamma^2\,\|\vartheta(p(t))\|_2^2\end{aligned} \tag{3.33}$$

The last inequality in (3.33) is based on subgradient definition. When the inequality is applied repetitively, the result is

$$\begin{aligned} &\|p(t+1)-p^*\|_2^2 \le \|p(1)-p^*\|_2^2 - 2\gamma\sum_{\tau=1}^{t}(Du(p(\tau))-Du(p^*))+ \\ &\gamma^2\sum_{\tau=1}^{t}\|\vartheta(p(\tau))\|_2^2 \end{aligned} \tag{3.34}$$

Considering $\|p(t+1)-p^*\|_2^2 \geqslant 0$, it can be expressed that

$$\begin{aligned} 2\gamma\sum_{\tau=1}^{t}(Du(p(\tau))-Du(p^*)) &\le \|p(1)-p^*\|_2^2 + \\ \gamma^2\sum_{\tau=1}^{t}\|\vartheta(p(\tau))\|_2^2 &\le \|p(1)-p^*\|_2^2+\gamma^2 tQ^2 \end{aligned} \tag{3.35}$$

Thus, (3.35) can be written as

$$\frac{1}{t}\sum_{\tau=1}^{t}(Du(p(\tau))-Du(p^*)) \le \frac{\|p(1)-p^*\|_2^2}{2\gamma t}+\frac{Q^2\gamma}{2} \tag{3.36}$$

Because *Du* is a convex function, by Jensen's inequality, the following is true

$$Du(\bar{p}(t))-Du(p^*) \le \frac{\gamma Q^2}{2}+\frac{\|p(1)-p^*\|_2^2}{2\gamma t} \tag{3.37}$$

Hence,

$$\limsup_{t\to\infty} Du(\bar{p}(t))-Du(p^*) \le \frac{Q^2\gamma}{2} \tag{3.38}$$

Therefore, Algorithm 1 converges to within $\frac{Q^2\gamma}{2}$ of the optimal value.

3.4.2 Distributed Scheduling

Distributed scheduling in a multipath network can play a significant role in the robustness of the wired or wireless network, specifically when traffic shifting to a better path is required [15]. For example, a flow which operates along multiple paths at the same time, could be designed using traffic scheduling mechanisms which allow the shifting of traffic to those paths with good delivery conditions (e.g., high bandwidth, low delay and packet loss rate) [121].

The scheduling demands in (3.26) are equivalent to maximising the weighted sum of the capacity of links. Equivalently, this problem can be solved by a weighted matching problem rather than finding the maximum weight of all independent sets in the conflict graph. In a graph, a subset of links which do not share common nodes, is referred to as a matching set. Maximum weighted matching problems have polynomial time complexity (see, e.g., [141]) with centralized implementation requirement, i.e., each node should announce its weight to a central node to construct the weighted graph of the network topology. As an alternative method, this chapter utilizes a sequential greedy algorithm introduced in [142]. In order to convert (3.26) to a maximum weighted matching problem, it is considered that $d_{i,j} > 0$ if $d_{j,i} < 0$ for all $z \in Z$. From each directed bilateral link pair, an undirected link $\langle i,j \rangle$ is extracted and its weight can be expressed as

$$d_{\langle i,j \rangle} = \max\left\{ w_{\langle i,j \rangle} c_{\langle i,j \rangle}, w_{\langle j,i \rangle} c_{(j,i)} \right\}$$

Let L^{und} and D^{und} denote the set of undirected links and its set of weight, respectively. Now, this problem is equivalent to a maximum weighted matching problem over the graph $G^{\text{wtd}} = \left(N, L^{und}, D^{und}\right)$ [115].

While the scheduling process in ad hoc networks usually proceed in each time step, the time complexity and generation of multi-hop scheduling for a deterministic solution are controversial [143]. On the other hand, the greedy behaviour of wide domination of wireless interactions requires the study of detection in compromised nodes and overall utilization of advanced statistical methods [144][145]. Moreover, developing the dynamic of channel residual reaction in steady state modes should be considered to expand efficiency of ad hoc networks scheduling [146][147]. Here, this chapter considers the concepts represented in [113][147][148] for the distribution scheduling algorithm which is further detailed in Algorithm 2.

***Remark*:** The scheduling algorithm needs to cope with the hierarchy of search along paths with the difference of node-to-node price in a region starting from the source set and ending up in the destination set. In addition, seeking an optimal solution would be done according to the greediness of the algorithm and the bearable congestion threshold. Furthermore, being subject to such an optimization problem, it is necessary to make a balance between the greediness of the algorithm and the absolute optimality of the system with respect to computational complexity. When the primal problem is broken down into the aforementioned sub-problems in the cross-layer design which is a generic sequence throughout requisite solutions for (3.25) and (3.22), it is important to avoid an exponential increase of iteration steps. Hence, in this scheduling problem, it is recommended to make a balance between the greedy algorithm and the dynamic programming presented in [149].

3.5 Cross-layer Design in Time Varying Channels

Multichannel networks involve multi-rate devices. In such cases, the allocated capacity for each link is time variant. This chapter assumes that the state of the channels is fixed within a time slot. It also assumes that the channel states

Algorithm 2 : Distributed Scheduling over Network with Fixed Channels

1: **for** each time t **do**
2: Calculate $d_{(i,j)}$ for all edges where $d_{(i,j)} = \max\left\{w_{\langle i,j\rangle}c_{\langle i,j\rangle}, W_{\langle j,i\rangle}c_{(j,i)}\right\}$
3: **for all** $z \in Z$ **do**
4: find node j^* with free neighboor j.
5: **end for**
6: **if** each i received a matching request from corresponding j^*, **then**
7: $\langle i, j^*\rangle$ is a matched link and node i send a matched reply to j^*. Consequently, i will drop all other messages.
8: **end if**
9: **if** i did not receive any message **then**
10: send a matching request to j^*
11: **end if**
12: **for all** $z \in Z$ when receiving a matching request from neighbour j **do**
13: **if** $j = j^*$ **then**
14: choose the corresponding link as a matched link and node i sends a matched reply to j^*. Consequently, i will drop all other messages.
15: **else**
16: saves the message.
17: **end if**
18: **end for**
19: **for all** $z \in Z$ **do**
20: when receiving a matched reply from neighbour j, link $\langle i, j^*\rangle$ is confirmed by node i to be a matched link and a drop message is sent to all free neighbours.
21: **end for**
22: **for all** $z \in Z$ **do**
23: when receiving a matched reply from neighbour j, node i removes node j form the list of free neighbours.
24: **end for**
25: Repeat steps 3 to 8 unless node i is matched or no free neighbours are available.
26: All matched links are ready for transmission. The corresponding nodes will use $d_{(i,j)}$ to schedule the directed links for transmission.
27: **end for**

change independently among different time slots. Let $e(t)$ denote the state of the channel in time slot t. The capacity of link l, at each time slot t is denoted by $c_l(e)$, therefore, (3.1) can be revised as

$$b_l^h(t) := \begin{cases} c_l(e): & \text{if } l \in h \\ 0 \quad : & \text{otherwise} \end{cases} \tag{3.39}$$

Recalling the earlier discussion in Section 3.3, the feasible rate region at link layer is identified as a convex hull of the rate vectors. Let $R(e)$ denote the feasible rate region. Thus, using a finite channel state assumption and a finite channel process in the identical distribution $u(e)$, the mean rate region can be defined as

$$\bar{R} := \left\{ \bar{b} : \bar{b} = \sum_{e} u(e) b^{h}(e), b^{h}(e) \in R(e) \right\} \tag{3.40}$$

Following the steps provided in Section 3.4, the Lagrangian of the problem statement in (3.19) and (3.20) can be given by

$$\begin{aligned} Du(p) = \max_{x_s \geq 0, f_{i,j,z}^{k} \geq 0} & \left\{ \sum_{s} U_s(x_s) \right. \\ & - \sum_{i \in N, k \in D, i \neq k} p_i^k \left(x_i^k - \sum_{z \in Z} \sum_{j:(i,j) \in L} f_{i,j,z}^{k} \right. \\ & \left. - \sum_{z \in Z} \sum_{j:(j,i) \in L} f_{j,i,z}^{k} \right) \\ & \left. - \sum_{i \in N, k \in D, i \neq k} q_i^k \left(\sum_{i} x_j^i - E_i^k \right) \right\} \end{aligned} \tag{3.41}$$

$$\text{s.t.} \qquad f \in \bar{R},$$

3.5.1 Perfect Scheduling over Time Varying Channels

The generated stream from the source set will comply accordingly from the utility function as congestion price measures linked sets of channels. This time variation in the system model requires stability evaluation for the proposed congestion control method. Here, for simplicity, the floor function is employed which returns the largest integer less than or equal to the actual value of the price.

Algorithm 3 : Perfect Scheduling over Network with Time Varying Channels

for each time t **do**

Update each node's price toward destination k

$$p_{i,z}^{k}(t+1) = \left\lfloor \left[\left(p_i^k(t) + \gamma \left(x_i^k(p(t)) - \left(\sum_{z \in z} \sum_{j:(i,j) \in L} f_{i,j,z}^{k}(p(t)) - \sum_{z \in z} \sum_{j:(j,i) \in L} f_{j,i,z}^{k}(p(t)) \right) - \sum_{i \in N, k \in D, i \neq k} q_i^k \left(\sum_i x_j^i - E_i^k \right) \right) \right) \right]^{+} \right\rfloor$$

Each node sends its price information to its neighbours.

for each source s **do**

performs congestion control as

$$x_s(p) = {U'}_s^{-1} \left(p_s + \sum_i q_i^k \right)$$

end for

Find $k^* \in \arg\max_k \sum_{z \in z} \left(p_i^k - p_j^k \right)$ for each node according to the adjacent nodes and obtain $d_{i,j} = \sum_z \left(p_i^{k^*} - p_j^{k^*} \right)$

Assign respective capacity to each link according to

$$\tilde{f} \in \arg \max_{f \in \bar{R}(e(t))} \sum_{z \in Z} \sum_{(i,j) \in L} d_{i,j}(t) f_{i,j,z}$$

Generate the flow through the selected links toward the selected destination as scheduling demands.

end for

For each node, it is necessary to have the information of its neighbour nodes. In a long enough coherence time, with the small time step, channel states can be considered as fixed during the time slot.

In the proposed algorithm, at time t, each node i updates its price with respect to destination k and channel z as follows

$$
\begin{aligned}
p_{i,z}^{k}(t+1) = \Bigg\lfloor \Bigg[\Bigg(p_i^k(t) + \gamma \Bigg(x_i^k(p(t)) \\
- \Bigg(\sum_{z \in z} \sum_{j:(i,j) \in L} f_{i,j,z}^{k}(p(t)) - \sum_{z \in z} \sum_{j:(j,i) \in L} f_{j,i,z}^{k}(p(t)) \Bigg) \\
- \sum_{i \in N, k \in D, i \neq k} q_i^k \Bigg(\sum_i x_j^i - E_i^k \Bigg) \Bigg) \Bigg) \Bigg]^{+} \Bigg\rfloor
\end{aligned}
$$

and announces the price $p_{i,z}^{k}$ to all of its neighbour nodes. Here, $\lfloor\ \rfloor$ denotes the floor function. In the congestion control, each source node s modifies its sending rate for t based on the local congestion price which can be written as

$$
x_s(p) = U'^{-1}_{s} \left(p_s + \sum_i q_i^k \right)
$$

Node i, after collecting the congestion price information from its neighbour node j, identifies the destination such that $k^* \in \arg\max_k \sum_{z \in z} \left(p_i^k - p_j^k \right)$ and obtain $d_{i,j} = \sum_z \left(p_i^{k^*} - p_j^{k^*} \right)$. Then it will assign respective capacity to each link according to $\tilde{f} \in \arg\max_{f \in \bar{R}(e(t))} \sum_{z \in Z} \sum_{(i,j) \in L} d_{i,j}(t) f_{i,j,z}$. In addition, perfect scheduling over time varying channels is summarized in Algorithm 3.

It should be noted that the distributed scheduling algorithm in time-varying channels is straightforward where it can be obtained by converting step 8 in Algorithm 3 to a maximum weighted matching problem. Similar to what has been done in subsection 3.4.2, the problem can be solved by using the greedy approach proposed in [142].

3.5.2 Stability and Convergence

Since the global parameters are fixed, it is necessary to point out the stability. Considering the optimization problem in (3.41) and using Markov chain as the stochastic model for p, the price can be considered as a random variable. The aim is to prove that the Markov chain is non-explosive and stable. Consider the following function of p at each time slot where the conditional expected value by any Lyapunov function V is given by [113]

$$\mathrm{E}[V(p(t+1)) - V(p(t))|\mathrm{p}(t)] \tag{3.42}$$

In addition, for each $\mathrm{p}(t)$ during the time slot t is constant, $p(t+1) = p(t) - \gamma\vartheta(p(t))$ should be considered. The conditional expected value can be written as:

$$\begin{aligned} E[V(p(t+1)) - V(p(t))|p(t)] &= E[V([(p(t) - \gamma\vartheta(p(t)))]) \\ &\quad - V(p(t))|p(t)] \\ &\leq E[V(p(t) - \gamma\vartheta(p(t))) \\ &\quad - V(p(t))|p(t)] \end{aligned} \tag{3.43}$$

Consider the Lyapunov function $V(p(t)) = \|p(t) - p^*\|_2^2$ and also define $\bar{\vartheta}$ as the subgradient of (3.42), thus

$$\begin{aligned} E[V(p(t+1)) - V(p(t))|p(t)] =& \mathrm{E}\left[-\gamma\vartheta(p(t))^T\left(2\left(p(t) - p^*\right)\right) - \gamma\vartheta(p(t))\right. \\ & - \gamma\vartheta(p(t))^T\left(2\left(p(t) - p^*\right)\right) \\ & + \gamma^2\mathrm{E}\left[\|\vartheta(p(t))\|_2^2|\mathrm{p}(t)|\right] \end{aligned} \tag{3.44}$$

Since it has been already proved that the stated subgradient is bounded, it could be expressed that $\bar{\vartheta}$ is also bounded. Consequently, E$[V(p(t+1)) - V(p(t))|p(t)]$ is bounded and the stability based on Foster's criterion can be obtained [150]. In an infinite approach of p during time, $\overline{Du}(E[p(\infty)]) - \overline{Du}(p(t))$ is also bounded or, in other words, the referred Markov chain guarantees non-explosive behaviour and convergence to a boundary near the optimal value [151]. Therefore along with the congestion process, recursive time variant scheduling can generalised. The configuration of the network which forms the presented method can maintain the system with a practical evaluation and command. Moreover, the implemented effect of delay and considering the ambient space in both (3.41) and (3.21) would provide a more stable model as shown in the results section.

Proof of algorithm stability:

Let the function in (3.41) represented by $\overline{Du}(p)$ where the optimal congestion price is p^* and $\overline{\vartheta}(p)$ is the subgradient. When the Lyapunov function [152] $V(p) = \|p - p^*\|_2^2$ is taken into account, it can be given that

$$\begin{aligned} E[\Delta V_t(p(t))|p(t)] &= E[V(p(t+1)) - V(p(t))|p(t) = p] \\ &= E[V(\lfloor [p(t) - \gamma\vartheta(p(t))]^+ \rfloor) - V(p(t))|p(t) = p] \\ &\leq E[V(p(t) - \gamma\vartheta(p(t))) - V(p(t))|p(t) = p] \\ &= E[-\gamma\vartheta(p(t))^T(2(p(t) - p^*) - \gamma\vartheta(p(t))|p(t) = p] \\ &= 2\gamma\overline{\vartheta}(p)^T(p^* - p) + \gamma^2 E[\|\vartheta(p(t))\|_2^2 |p(t) = p] \\ &\leq 2\gamma\overline{\vartheta}(p)^T(p^* - p) + Q^2\gamma^2 \end{aligned} \tag{3.45}$$

where $\vartheta(p(t))$ is assumed to be upper bounded by Q. As $\overline{Du}(p)$ is a convex function, it can be stated that

$$E[\Delta V_t(p(t))|p(t)] \leq 2\gamma(\overline{Du}(p^*) - \overline{Du}(p)) + Q^2\gamma^2 \tag{3.46}$$

Suppose that

$$\varepsilon = \max_{\overline{Du}(p^*) - \overline{Du}(p) \leq \gamma Q^2} \|p - p^*\|_2 \tag{3.47}$$

and

$$B = \{p : \|p - p^*\|_2 \leq \varepsilon\} \tag{3.48}$$

The following inequality is obtainable

$$E[\Delta V_t(p)|p] \leq -Q^2\gamma^2 I_{p\in B^c} + Q^2\gamma^2 I_{p\in B} \tag{3.49}$$

where I is the index function. By considering the method (3.1) in [151], $p(t)$ is stable.

3.6 Numerical Results

In this section, the chapter presents numerical results of the proposed cross layer design. A simple ad hoc-IoT wireless network is considered as shown in Fig. 3.4 [3]. In general, IoT wireless networks can be centralised systems, such as cellular networks, or decentralised systems, such as ad hoc networks. The latter is taken into account in this chapter. The chapter considers 6 ad hoc-IoT wireless nodes as

[3] To have a fair comparison, the ad hoc network under consideration is the same as the one in [113].

illustrated in Fig. 3.4. The connection in the network is assumed to be a multi-peer communication system (e.g., C can communicates with F through D using Wi-Fi P2P communication) and node E is assumed to be connected to the Internet via a sink node (e.g., Wi-Fi access point). Each node implements the proposed design in fixed and time-varying channel scenarios. Also, each node is assumed to be a multi-homed device which support multipath transmission technology. In other words, each node is capable of transmitting data through multiple paths simultaneously. In addition, the link capacities are supposed to be between 1 and 3 units. The demonstrated network in Fig. 3.4 is chosen to examine the proposed protocol in a minimum network with multi-homing and multiple possible paths with link contention possible. In the ad hoc-IoT network, it is assumed that there are two network layer flows; A to F and B to E with same utility function $U_s(x_s) = \log(x_s)$. The reason for selecting the log function as utility is to provide some sort of saturation effect as traffic is increased. The step size of the presented algorithms is set as $\gamma = 0.1$ because the chapter takes into account the subgradient of the function and a small enough step size in the subgradient is shown to be convenient to obtain the optimal value [113]. As mentioned in the system model, all the links are assumed to be bidirectional and able to carry the traffic in opposite directions. For the convenience of the reader, the network parameters are summarized in Table 3.1. As a bench mark, the numerical results include the analysis of algorithms introduced in [113] (denoted as single path 'SP') and [91] (denoted as Multi-path 'MP'). It should be noted that the results in this chapter are obtained from formulations. The formulations used for 'SP' are numbers 1 to 18 as well as 24 along with 25, and the formulations used for 'MP' are numbers 1 to 5 as well as 13 to 20. Moreover, the platform used to plot the results is MATLAB 2020.

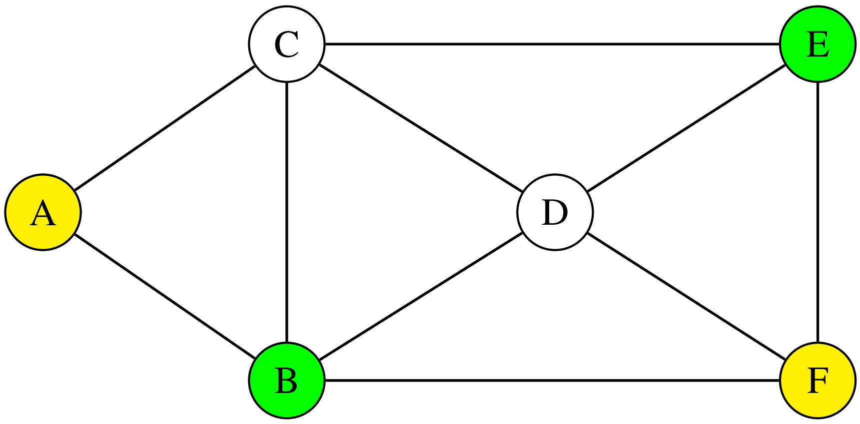

Fig. 3.4 Configuration of the studied network and the pairing of the two network layer flows.

Table 3.1 : Network Parameters

Parameter	Value
$U_s(x_s)$	$\log(x_s)$
γ	0.1
$\|\|N\|\|$	6
Link capacity	1-3 units

3.6.1 Perfect Scheduling over Fixed Channels

In this sub-section, the performance of the proposed perfect scheduling algorithm (i.e., Algorithm 1) over fixed time channel conditions is studied. It is assumed that the capacity of links $C \to E$, $E \to C$, $B \to F$, and $F \to B$ are one unit and the rest of links have two units of capacity. This chapter considers two distinct flows; AF and BE. Fig. 3.5 illustrates the performance of the presented algorithm compared with the existing schemes in terms of normalised source rate and normalised congestion price. It is worth mentioning that the oscillating behaviour is caused by the non-differentiability of the function in (3.20). The algorithm introduced in [91], always makes an effort to balance the congestion prices among its paths. In

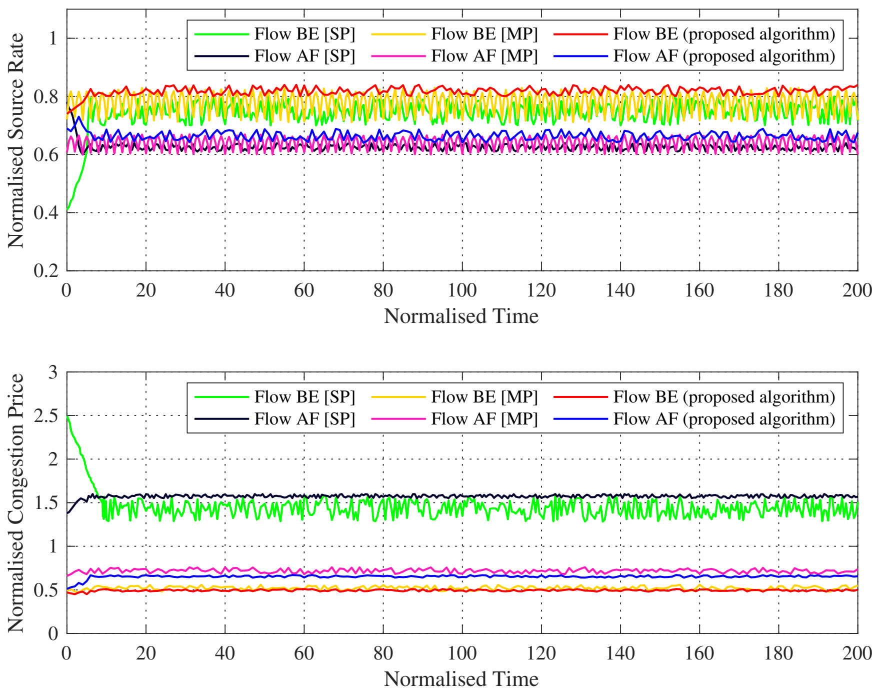

Fig. 3.5 Normalised source rate and normalised congestion price for perfect scheduling over fixed time channels.

case of high congestion in one path, the allocated rate to this path will be reduced to minimise the congestion. However, it can be observed from Fig. 3.5 that the proposed design outperforms the existing methods (i.e., [113] and [91]) for both flows in terms of source rate. This is due to the fact that the algorithm benefits from all the available multi-path in the network during the scheduling process. Furthermore, as shown in Fig. 3.5, the proposed algorithm presents more stability and it quickly converges to the optimal value in comparison with other schemes. Besides, the proposed algorithm demonstrates lower normalised congestion price compared with other methods. The reason for such behaviour lies in the fact that the algorithm takes into account the queuing delay in the communication system.

Moreover, Tables 3.2 and 3.3 show summary statistics of the results in Fig. 3.5. Specifically, they show the mean and standard deviation of the normalised traffic and normalised congestion price for each algorithm along with the relevant flow respectively. It can be observed that the proposed algorithm improved the mean traffic throughput by at least 6.3% compared to the other algorithms. Also, it reduced the mean congestion price in comparison to the others by 4.1% at the minimum.

Table 3.2 : The mean and standard deviation of the normalised source rate in Fig. 3.5.

Flow/Algorithm	Mean	Standard deviation
Flow BE (SP)	0.750	0.040
Flow AF (SP)	0.620	0.011
Flow BE (MP)	0.760	0.050
Flow AF (MP)	0.630	0.015
Flow BE (Proposed algorithm)	0.810	0.012
Flow AF (Proposed algorithm)	0.670	0.013

Table 3.3 : The mean and standard deviation of the normalised congestion price in Fig. 3.5.

Flow/Algorithm	Mean	Standard deviation
Flow BE (SP)	1.400	0.100
Flow AF (SP)	1.600	0.002
Flow BE (MP)	0.511	0.002
Flow AF (MP)	0.700	0.003
Flow BE (Proposed algorithm)	0.490	0.001
Flow AF (Proposed algorithm)	0.650	0.001

Furthermore, Fig. 3.6 presents an end-to-end queuing delay comparison of the schemes. It can be seen that the introduced cross layer design outperforms the existing designs and achieved the minimum delay required in both *BE* and

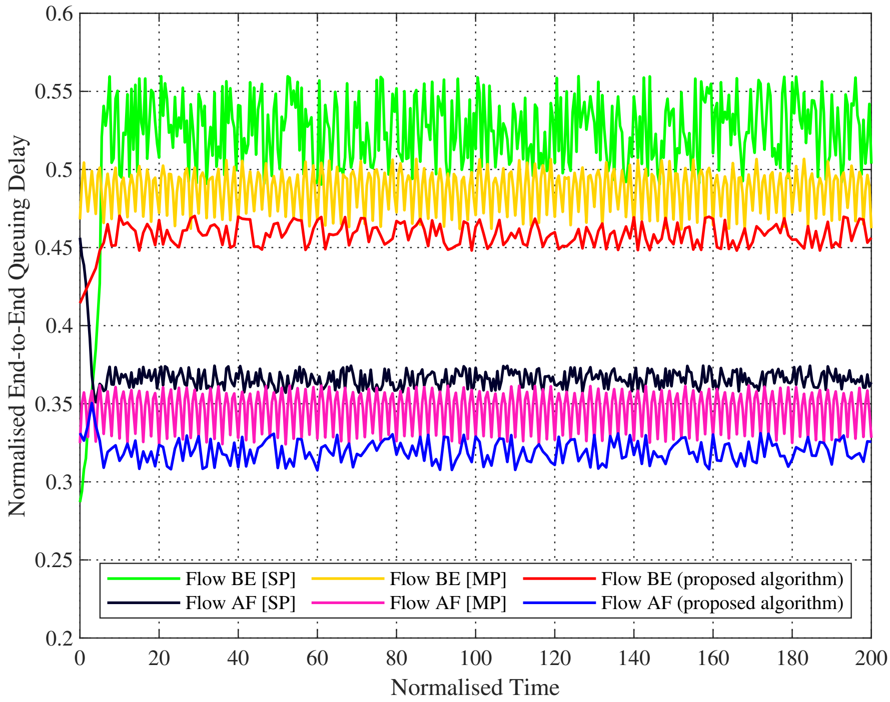

Fig. 3.6 Normalised end-to-end queuing delay for perfect scheduling over fixed time channels.

AF flows. Single path transmission in SP leads to a higher end-to-end queuing delay where multipath transmission in MP results in less delay, however, the MP congestion control mechanism is not ideal since shifting the traffic from congested paths to lower congested paths without considering paths characteristics, such as delay, increases end-to-end delay. The proposed protocol selects the paths with minimum delay for transmission. This technique brings the delay to a lower level compared to the others as demonstrated in Fig. 3.6.

3.6.2 Distributed Scheduling over Fixed Channels

In this scenario, the chapter investigates the performance of the introduced distributed scheduling algorithm (i.e., Algorithm 2) over the network under consideration with fixed time channels. Similar to the first scenario, it is assumed that the

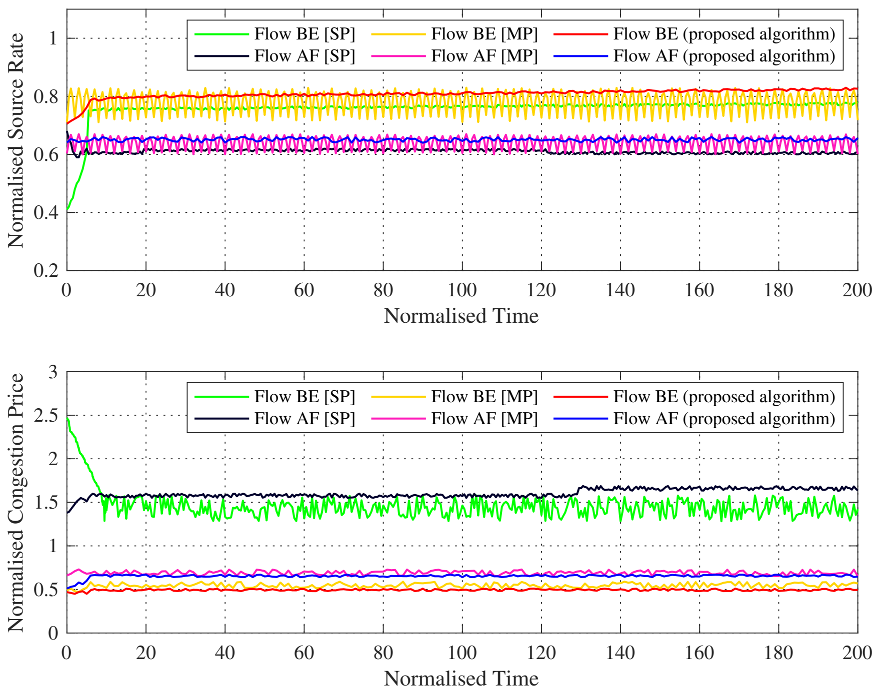

Fig. 3.7 Normalised source rate and normalised congestion price for distributed scheduling over fixed time channels.

capacity of links $C \to E$, $E \to C$, $B \to F$, and $F \to B$ are one unit and the rest of the links have two units of capacity. Fig. 3.7 illustrates the performance of the introduced algorithm compared with the existing schemes in terms of normalised source rate and normalised congestion price. It can be observed from Fig. 3.7 that the fluctuation pattern is lower than for the perfect scheduling algorithm. However, comparing Fig. 3.5 and Fig. 3.7 demonstrates that the distributed algorithm achieves lower normalised source rate. This comes from the fact that the feasible rate region shrinks to a smaller set under the approximate scheduling. It should be noted that in the source rate comparison, the algorithm in [91] performs better than the proposed algorithm at the beginning, nevertheless, the proposed algorithm outperforms it over time. Moreover, the introduced algorithm presents more stability compared to [91]. In general, the proposed distributed scheduling algorithm shows higher source rate and lower congestion price compared to both [113] and [91] methods. The main reasons behind the satisfactory performance are exploiting multipath transmission and considering an efficient queuing system in the cross-layer design.

Table 3.4 : The mean and standard deviation of the normalised source rate in Fig. 3.7.

Flow/Algorithm	Mean	Standard deviation
Flow BE (SP)	0.740	0.005
Flow AF (SP)	0.600	0.010
Flow BE (MP)	0.760	0.050
Flow AF (MP)	0.620	0.020
Flow BE (Proposed algorithm)	0.800	0.005
Flow AF (Proposed algorithm)	0.650	0.005

Table 3.5 : The mean and standard deviation of the normalised congestion price in Fig. 3.7.

Flow/Algorithm	Mean	Standard deviation
Flow BE (SP)	1.410	0.100
Flow AF (SP)	1.610	0.004
Flow BE (MP)	0.520	0.003
Flow AF (MP)	0.680	0.003
Flow BE (Proposed algorithm)	0.499	0.001
Flow AF (Proposed algorithm)	0.650	0.001

Besides, Tables 3.4 and 3.5 present summary statistics of the results in Fig. 3.7. The tables show that the proposed cross layer design enhanced the mean traffic throughput by no less than 4.8% compared to the others, and decreased the mean congestion price by at least 4.0%.

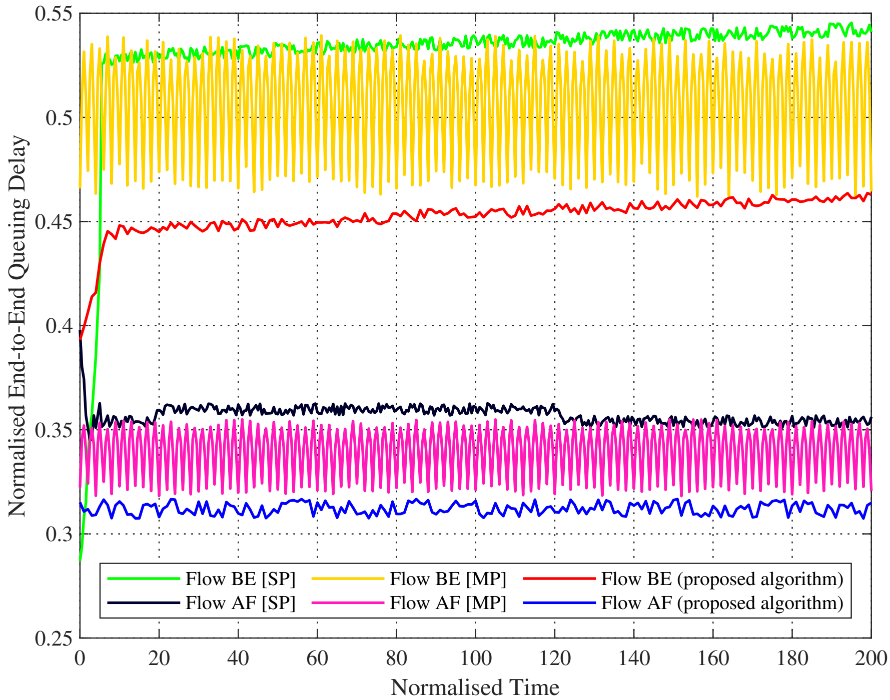

Fig. 3.8 Normalised end-to-end queuing delay for distributed scheduling over fixed time channels.

Moreover, Fig. 3.8 shows the results of end-to-end queuing delay. It can be observed that the proposed algorithm attained the lowest delay for both flows in comparison to SP and MP algorithms due to the efficient queuing system used. It can be noticed that flow *BE* in MP has greater fluctuation in delay compared to the others. This is due to the fact that MP does not differentiate between the low and high delay paths especially when the feasible rate region decreases in distributed scheduling. In addition, SP has a much higher end-to-end queuing delay compared to the presented algorithm. Thus, the proposed method is more effective than the others in ad hoc-IoT wireless networks.

3.6.3 Perfect Scheduling over Time Varying Channels

In this sub-section, the chapter explores the performance of the proposed perfect scheduling algorithm (i.e., Algorithm 3) over time varying channels condition. Here, it is assumed that the capacity of links $C \rightarrow E$, $E \rightarrow C$, $B \rightarrow F$, and $F \rightarrow B$ are uniformly distributed with 0.5, 1 and 1.5 units, whereas the rest of the links are uniformly distributed over 1, 2 and 3 units of capacity. Fig. 3.9 shows the performance of the algorithm with regard to the normalised source rate and normalised congestion price. As can be observed, considering the time varying condition leads to more fluctuations for both *BE* and *AF* flows. This is due to variation of channel states in this condition. However, the algorithm demonstrates a more stable pattern and lower congestion price. In addition, the source rate of the introduced algorithm is better for *AF* flow in comparison with the other schemes. Although *BE* flow in [113] achieved higher source rate, the algorithm shows lower oscillating behaviour. Further, the proposed algorithm outperforms the algorithm in [91] in both source rate and congestion price. Thus, the source rate of the proposed design is better than MP and only slightly worse than SP which has much poorer delay, and which will be explained later in this sub-section. It should be noted that the benefit of the available multi-path becomes more obvious in time varying conditions. As it can be observed, the achievable sending rate for each flow is larger compared to the fixed channel condition.

This chapter further summarizes the results of Fig. 3.9 in Tables 3.6 and 3.7. It is shown that the proposed method improved the mean traffic throughput by a minimum of 7.1% compared to the other methods. Also, it diminished the congestion price in comparison to the others by no less than 3.1%.

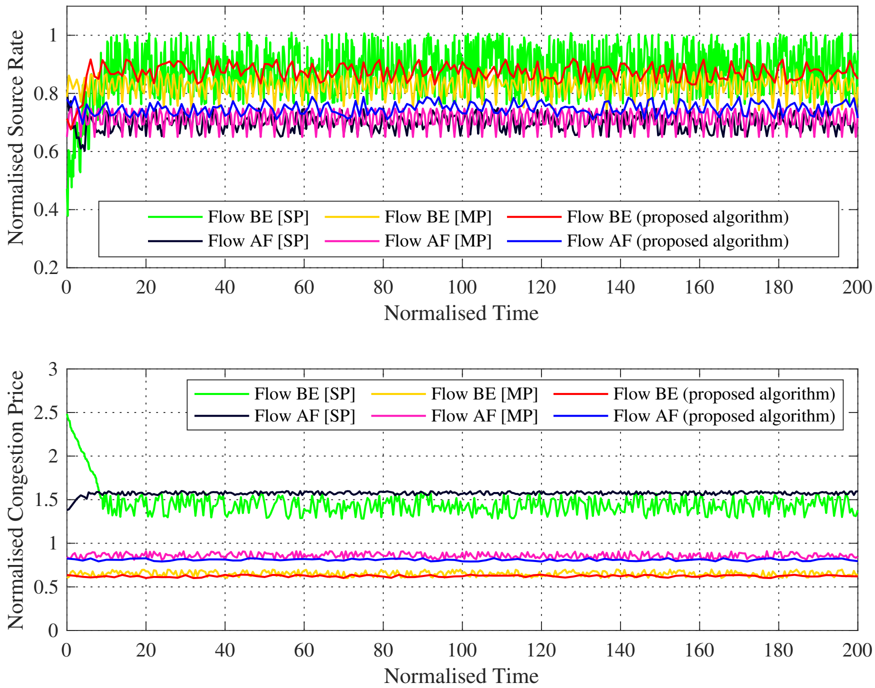

Fig. 3.9 Normalised source rate and normalised congestion price for perfect scheduling over time varying channels.

Table 3.6 : The mean and standard deviation of the normalised source rate in Fig. 3.9.

Flow/Algorithm	Mean	Standard deviation
Flow BE (SP)	0.910	0.080
Flow AF (SP)	0.680	0.020
Flow BE (MP)	0.820	0.030
Flow AF (MP)	0.700	0.030
Flow BE (Proposed algorithm)	0.880	0.040
Flow AF (Proposed algorithm)	0.750	0.030

Table 3.7 : The mean and standard deviation of the normalised congestion price in Fig. 3.9.

Flow/Algorithm	Mean	Standard deviation
Flow BE (SP)	1.407	0.101
Flow AF (SP)	1.602	0.002
Flow BE (MP)	0.640	0.006
Flow AF (MP)	0.790	0.007
Flow BE (Proposed algorithm)	0.620	0.001
Flow AF (Proposed algorithm)	0.760	0.001

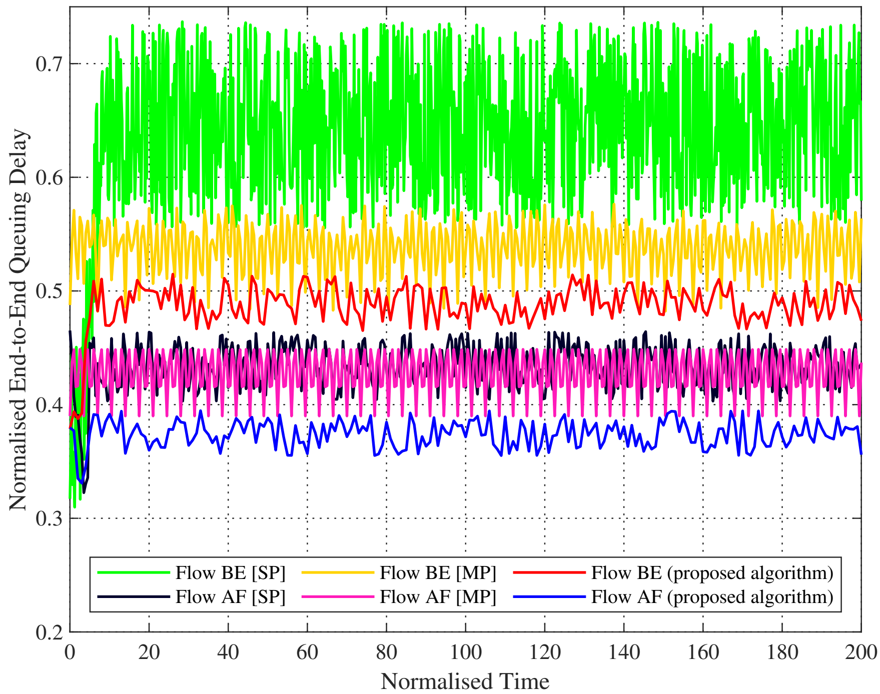

Fig. 3.10 Normalised end-to-end queuing delay for perfect scheduling over time varying channels.

Additionally, all methods in terms of end-to-end queuing delay are compared. Fig. 3.10 illustrates that the introduced cross layer protocol outperforms the existing protocols in both flows. Although flow *BE* in SP achieved higher source rate than the proposed design in Fig. 3.9, it has much higher delay as shown in Fig. 3.10. Also, it has greater oscillating behaviour compared to the others. This is because the link capacity is changing in time-varying channels which leads to changes in delay especially in the single path scenario where there is no possibility to split the traffic into multiple flows to reduce end-to-end queuing delay. The proposed algorithm demonstrated lower delay and greater stability compared to SP and MP algorithms.

3.6.4 Distributed Scheduling over Time Varying Channels

In this scenario, the performance of the proposed algorithm for distributed scheduling (i.e., Algorithm 3 with a weighted matching approach) under time varying channel conditions is studied. Similar to the previous scenario, it is assumed that the links' capacity for $C \rightarrow E$, $E \rightarrow C$, $B \rightarrow F$, and $F \rightarrow B$ are uniformly distributed with 0.5, 1 and 1.5 units, whereas the rest of links are uniformly distributed over 1, 2 and 3 units of capacity. Fig. 3.11 illustrates the performance of the presented algorithm compared with the existing schemes in terms of normalised source rate and normalised congestion price. The method in [113] is better than the proposed method as well as [91] in terms of source rate for *BE* flow. However, the introduced algorithm outperforms the existing methods in *AF* flow as well as congestion price measurement. Also, it shows greater stability. It can be observed that the source rate of the distributed scheduling algorithm in time varying channels is larger than the source rate in fixed channels. This is due to variation of channel capacities.

The corresponding mean and standard deviation of the results in Fig. 3.11 are summarized in Tables 3.8 and 3.9. It is illustrated that the presented cross layer scheme enhanced the mean traffic throughput by at least 5.2% in comparison to the others and reduced the congestion price by 3.6% at the minimum.

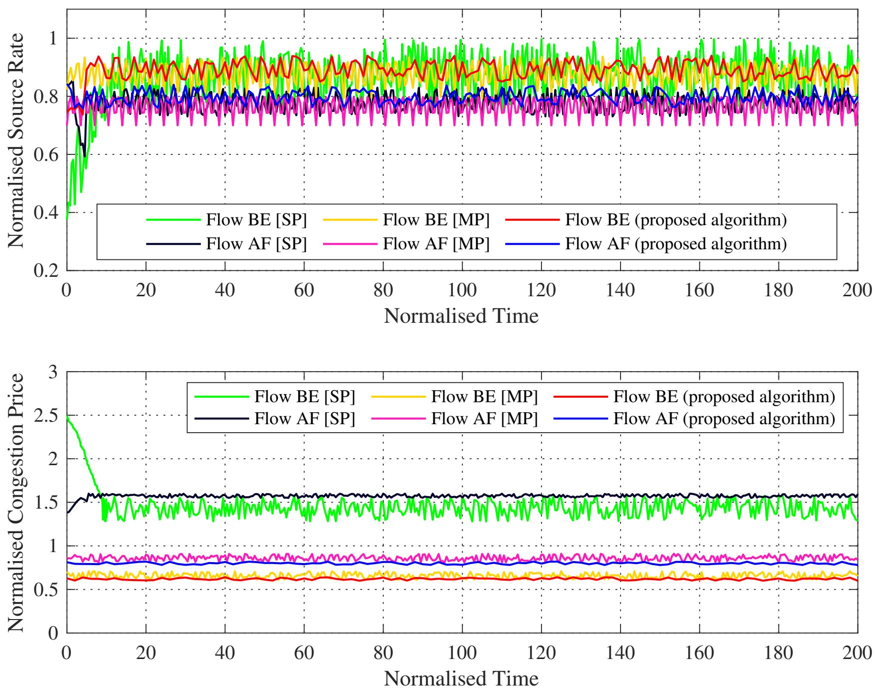

Fig. 3.11 Normalised source rate and normalised congestion price for distributed scheduling over time varying channels.

Table 3.8 : The mean and standard deviation of the normalised source rate in Fig. 3.11.

Flow/Algorithm	Mean	Standard deviation
Flow BE (SP)	0.900	0.090
Flow AF (SP)	0.760	0.020
Flow BE (MP)	0.820	0.025
Flow AF (MP)	0.760	0.020
Flow BE (Proposed algorithm)	0.870	0.028
Flow AF (Proposed algorithm)	0.800	0.020

Table 3.9 : The mean and standard deviation of the normalised congestion price in Fig. 3.11.

Flow/Algorithm	Mean	Standard deviation
Flow BE (SP)	1.406	0.101
Flow AF (SP)	1.600	0.002
Flow BE (MP)	0.680	0.006
Flow AF (MP)	0.780	0.007
Flow BE (Proposed algorithm)	0.655	0.001
Flow AF (Proposed algorithm)	0.750	0.001

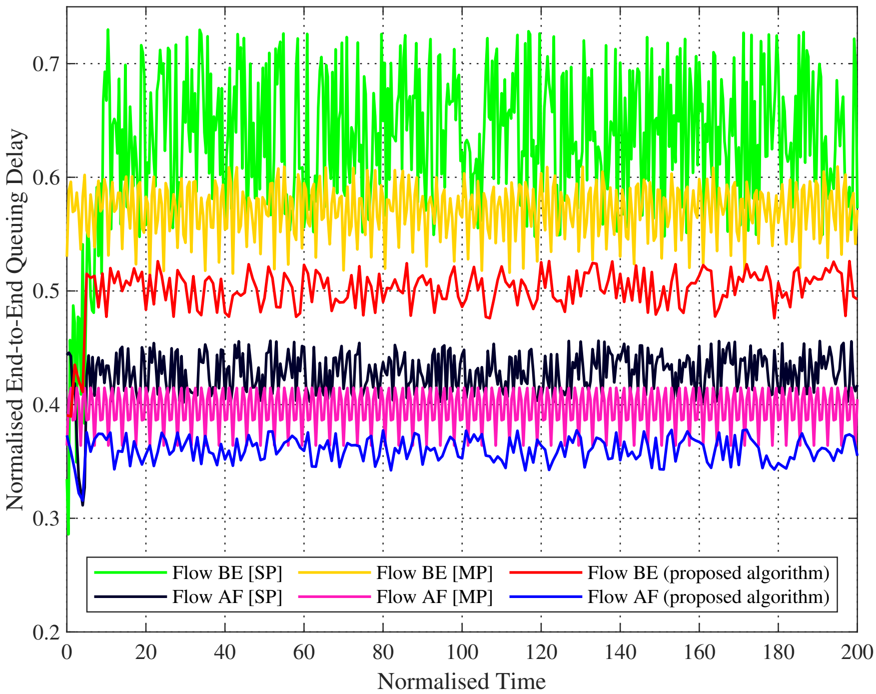

Fig. 3.12 Normalised end-to-end queuing delay for distributed scheduling over time varying channels.

Furthermore, Fig. 3.12 presents an end-to-end queuing delay comparison of the schemes. It can be seen that the cross layer design outperforms the existing designs and achieved the minimum delay required in both *BE* and *AF* flows.

Similar to the previous sub-section, SP is the worst case with higher delay in comparison to the other methods. The proposed protocol selects the paths with minimum delay for transmission. Therefore, it is a more effective system in ad hoc-IoT wireless networks.

3.7 Summary

Demands for stable and efficient transmission are increasing due to the ongoing expansion of global wireless data traffic. One of the most promising ways to moderate burden of traffic load and decrease the queuing delay is to design cross-layer protocols. In this chapter, a utility maximization problem was considered for multipath ad hoc wireless networks via joint cross-layer congestion control, routing and scheduling design. In the formulated utility maximization problem, rate, scheduling and queuing delay constraints were involved under the condition of fixed channels and time varying channels. To solve this utility maximization problem, two scheduling methods were proposed, i.e., perfect scheduling and distributed scheduling. In perfect scheduling, all nodes in the network contribute to the scheduling process, while the distributed method only takes into account adjacent nodes. For both methods, the global convergence was proved. The results showed that the proposed algorithms outperform existing cross layer protocols in terms of increasing source rate and reducing congestion price. In the following chapter, the author introduces the second contribution of this thesis, that is the use of Opportunistic Routing (OR) in MPTCP.

Chapter 4

Reduce Delay of Multipath TCP in IoT Networks

This chapter explains the introduction of Opportunistic Routing (OR) to MPTCP. First, the chapter compares a number of MPTCP protocols namely, traditional MPTCP, MPTCP-Traffic Splitting Control (MPTCP-TSC) and Redundant MPTCP (ReMP TCP) in an IoT wireless environment. After that, OR is adapted for the above schemes and the protocols are evaluated in terms of number of transmissions, end-to-end delay and start-up delay. Additionally, this chapter is published in *Springer Wireless Network* as a journal article and it is cited in [153].

The chapter is organized as follows. Section 4.1 introduces the chapter. In Section 4.2, related work is presented. The proposed design is illustrated in Section 4.3. The system model is introduced in Section 4.4. The results of the experiment are shown in Section 4.5. Finally, Section 4.6 summarises the chapter.

4.1 Introduction

Next-generation networks such as the Internet of Things (IoT), smart healthcare and intelligent transportation systems generate a massive amount of traffic with stringent Quality of Service (QoS) requirements (i.e., high throughput, low delay and packet-loss) [154]. Regular TCP was originally designed for wired networks and adapted in wireless networks which gives rise to some issues such as high packet loss and delay [155]. This is due to the characteristics of wireless channels such as high error rate, interference, fading, obstructions etc [13]. In addition, a lot of algorithms have been proposed to enhance TCP performance. For instance, FAST TCP which is designed to consider queuing delay as a congestion signal rather than packet loss probability [76]. Moreover, TCP Fast Open that significantly boosts the connection speed of TCP by avoiding the three-way handshake process [156]. Further, Proportional Rate Reduction (PRR) which is a novel TCP method introduced to improve recovery procedure [157]. However, as explained in Chapter 2, TCP lacks a fault tolerance property which is important in next-generation communications [13]. When path failure occurs in the connection, data will be lost, thus, TCP has to reset a new connection. Furthermore, TCP does not support multihomed terminals such as smartphones, tablets and laptops which are equipped with multiple heterogeneous interfaces available for transmission. This motivated the Internet Engineering Task Force (IETF) to introduce MPTCP that enables concurrent traffic forwarding on multiple paths using multiple network interfaces (e.g., Wi-Fi, 5G/LTE and Ethernet). The aim of MPTCP is to increase bandwidth and throughput while maintaining a reliable communication between two ends [17]. In fact, MPTCP is originally designed to satisfy three objectives: i) increase throughput; ii) load balancing and iii) do not harm [158]. Throughput increment can be achieved by bandwidth aggregation; and load balancing is at-

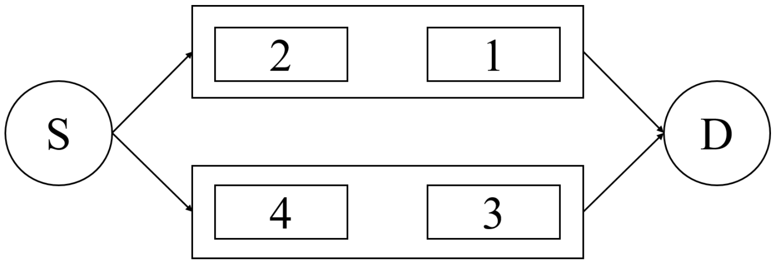

Fig. 4.1 An illustration of the MPTCP scenario, S is the source and D is the destination.

tained by moving the traffic from congested paths while ‘do not harm’ means that MPTCP must be fair to other TCP flows. Figure 4.1 shows the packet transfer scenario of MPTCP communication in the transport layer. MPTCP splits data traffic and sends it on different paths. In addition, MPTCP utilises numerous TCP paths within one connection, where each is called a subflow. This offers the opportunity of path diversity that is not available in conventional TCP [129]. Although MPTCP exploits path diversity, it uses a single TCP socket from the perspective of the application, therefore, the application layer in MPTCP remains unmodified. MPTCP is widely deployed and has no issues with middle boxes such Network Address Translations (NATs), firewalls and proxies since it is compatible with regular TCP [49].

MPTCP has attracted a lot of research in industry and academia due to its improvements in network throughput. For business purposes, MPTCP is being used in Siri, the digital assistance provided by Apple. It is supported in iOS version 7.0 and later versions. Moreover, Samsung supports MPTCP in its products such as Galaxy S6 to offer seamless communication and better service [13]. In academia, MPTCP was implemented and evaluated in [54] and demonstrated better performance than conventional TCP. In [159], the authors proposed MPTCP-TSC (Traffic Split Control) as a cost minimization problem subject to end-to-end delay constraint. A greedy heuristic is presented that allocates the transmission rate for every sub-flow. The simulation results showed enhanced performance in comparison with MPTCP. The authors in [73] proposed a scheme known as ReMP TCP to reduce latency. The scheme sends redundant packets over parallel sub-flows to diminish the packet drop probabilities and increase reliability. An experiment was conducted using a simulation as well as a real-world mobile scenario, and it was concluded that the ReMP TCP outperforms regular MPTCP. Nevertheless, the existing MPTCP schemes [54], [159] and [73] have a high number of transmissions which contribute to high delay. Number of transmissions refers to the number of times a packet is transmitted/re-transmitted from source to destination [160]. The reason for the high number of transmissions in the above MPTCP protocols is the use of traditional routing where a lot of re-transmissions can occur particularly in lossy links [161]. To address this challenge, the chapter exploits the Opportunistic Routing (OR) technique.

OR is a routing model used to increase the delivery rate and reliability of data transmission in wireless networks [160]. It selects the next routing path at the transmission time by using the broadcasting method. Moreover, in OR, the source node creates a list of forwarders (relays) and any adjacent node, which hears the data transmission, can be considered as a forwarder. The source node then can prioritise the forwarders according to certain conditions. In contrast, traditional routing defines the routing path before transmission which can be insufficient particularly in highly dynamic networks. Figure 4.2 presents the scenario of OR as well as traditional routing in which the source S forwards the traffic to the destination D. In the traditional forwarding mechanism, the route S-R1-D is selected before sending data. All traffic must use this route until the last packet is successfully delivered to D. In OR, the source S can select R2 or R3 as a forwarder at the transmission time. Therefore, either S-R2-D or S-R3-D route can be utilised.

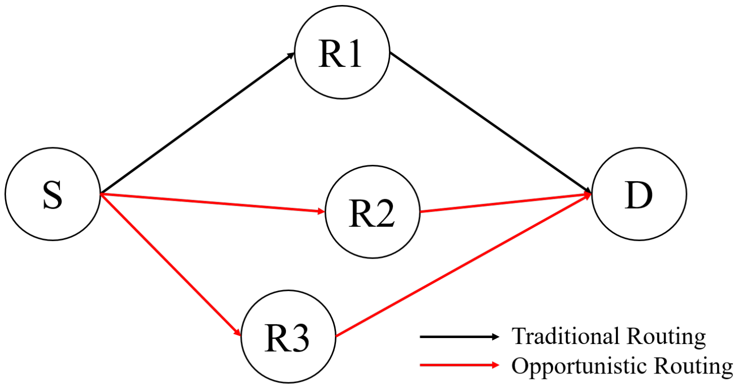

Fig. 4.2 An illustration of Opprtunistic Routing and Traditional Routing procedure.

The main idea behind OR is to utilize the broadcast nature of the wireless medium for a reliable transmission [162]. Consider the wireless network shown in Figure 4.3. The source wants to transmit a data packet to the destination. In OR, the source broadcasts the packet to the neighbouring nodes which are in its transmission range. If the packet is not received by any node, the source re-transmit it. Otherwise, the nodes that received the packet successfully can be considered as a candidate to forward it. These candidates cooperate with each other and choose the best relay node based on a specific metric such as wireless link quality, expected energy consumption, number of hops, security, data rate and queue length [163]. The candidates are grouped in a list called Forwarder Priority List (FPL), thus, if the best relay fails to transmit the packet, the second best relay takes over the transmission and so forth [160]. The broadcast and the Opportunistic Routing continue in the network until the packet is successfully received by the destination.

In this chapter, OR is introduced to the protocols MPTCP [54], MPTCP-TSC [159] and ReMP TCP [73] to reduce the number of transmissions required to successfully deliver a packet to the destination. This leads to delay reduction and efficient use of network resources in the IoT scenario. In addition, to the best of

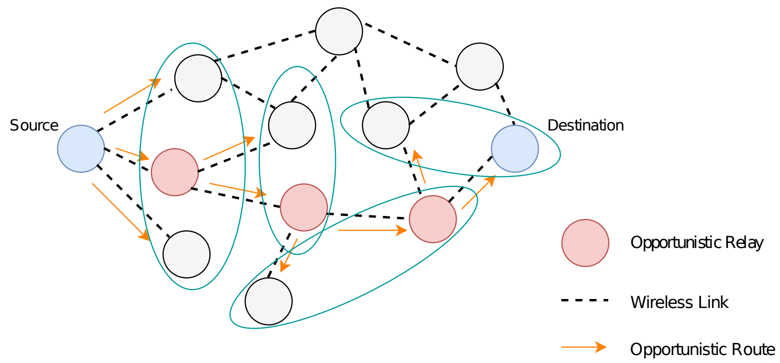

Fig. 4.3 An example of packet forwarding in a wireless network using Opportunistic Routing.

the author's knowledge, this is the first novel work which combines MPTCP with OR. The key contributions of this chapter are as follows:

1. TCP, MPTCP, MPTCP-TSC and ReMP TCP are compared in an IoT wireless network to determine the best protocol in the scenario.
2. OR is adapted to TCP, MPTCP, MPTCP-TSC and ReMP TCP in order to reduce the number of transmissions compared to the existing schemes.
3. End-to-end delay is decreased in the existing schemes by exploiting the OR technique.
4. Multimedia data transmission is considered and the newly proposed schemes reduce the start-up delay by requiring a smaller number of transmissions for the first video frame.

The simulation results illustrate that ReMP TCP has the lowest number of transmissions, lowest end-to-end delay and start-up delay compared to the other protocols. Furthermore, the selected schemes have shown better performance when using the OR technique in the IoT network.

4.2 Related Work

Wireless communication has experienced exponential growth over last decade. The advancements in next-generation communication networks, such as IoT, have led to a huge increase in the use of smart devices (i.e. smartphones, tablets and smart-watches). Therefore, there is a need for a promised protocol, such as Multipath TCP, to control the traffic. MPTCP is well-studied in the literature and numerous research have been proposed to develop the protocol. The authors in [124] studied MPTCP and concluded that it has higher throughput compared to regular TCP. In [164], the researchers proposed an MPTCP scheme based on the round-trip time (RTT) policy that forwards packets on the path with the lowest RTT. However, the approach is not efficient in the case of lossy networks. The authors in [64] introduced an MPTCP-based scheme which forwards the packets on the fastest path instead of waiting for the lowest RTT route. The results show that forwarding the packet on the fastest paths improved the performance of the protocol compared to waiting for the lowest RTT paths when the difference between RTTs is large. In addition, the literature in [1] explores the heuristics based on transmission rate, loss rate and delays and concluded that this approach performs better than the lowest RTT approach.

The authors in [165] explored the challenges of deploying MPTCP. Firstly, MPTCP can reduce the throughput of the network due to packet re-ordering issues. Moreover, the number of sub-flows can cause a buffer overflow problem at the receiver side. In [166], MPTCP- Pipeline Network Coding (MPTCP-PNC) was proposed to solve the problem of packet reordering. MPTCP-PNC introduced cheap coding coefficient rules that reduce encoding and decoding delay by progressively coding and decoding packets without waiting for the whole block (group of packets) on both sides (sender and receiver). This means when the transport layer receives data from the upper layer and converts it into blocks or groups, whichever packets come to the encoder are encoded and sent without waiting for the whole group. In addition, packets are assigned to paths according to path quality. On the receiver side, the same process occurs where the packets are decoded and sent to the application layer regardless of the other packets of the same group. Furthermore, in [167] an algorithm was proposed, called Stochastic Earliest Delivery Path First (S-EDPF), which combines scheduling and random linear coding using forward error correction (FEC). The authors provided end-to-end performance measurement and the result obtained demonstrates that the delay impact of reordering and packet loss is mitigated. However, the coding and decoding model introduces computational complexity to the systems.

In [168], the impact of latency in MPTCP in mobile-based applications was analysed. The results show that performance degrades in the case of high traffic generation where modern schedulers are required. Intelligent delay-aware packet scheduling (DAPS) was proposed to send future and current packets at the same time [169]. This means that some sup-flows will forward current packets where others simultaneously forward future packets. As the authors explained, applying this approach achieved a 77% reduction in the receiver's buffer, 63% decrease in the application delay and increased the overall throughput by 42 %. In [170], the authors presented an algorithm to minimize head of line blocking problem in MPTCP. The scheme dynamically blocks the flows that can cause the problem which leads to better performance according to the simulation results. Nevertheless, in some cases the sup-flows in [169] and [170] become idle. The lengthy idle period resets the congestion window (cwnd), hence, underutilises the network resources. This issue was addressed by [171] where the completion time of sub-flows was considered and the problem of underutilisation of flows was solved by reducing the long idle periods. In [70], the researchers introduced a new data transmission model by considering the path symmetry. The model adjusts the path selection for different data flows while predicting the volume of data. This technique achieved short completion time of the MPTCP sub-flows according to the simulation results. In [65], the authors proposed an MPTCP-based weighted round robin scheduler that integrated a load balancing approach and achieved optimal performance; however, the proposed scheme has a fairness issue. The authors in [172] proposed an MPTCP based approach for satisfying the QoS requirement of the network. The traffic is forwarded on the path with low queuing delay, RTT and congestion state. The results show that higher QoS is achieved compared to other schedulers.

The authors in [173] used MPTCP for resource pooling and a new metric that expects throughput was calculated for each sub-flow. The results showed higher throughput and lower network congestion. In addition, the authors in [174] dynamically adjusted the congestion window according to the requirement of the application. This method results in lower packet loss and better throughput. The authors in [175] designed a framework operating at the receiver-side to optimally allocate a transmission rate for each sup-flow in order to reduce the application-level delay. The receiver sends 3-duplicate acknowledgments (ACKs) to the sender to adjust the congestion window if the current cwnd size is greater than a threshold. However, this process is not practical in IoT networks. The authors in [176] proposed an MPTCP congestion control algorithm that adaptively decouples sub-flows which are not sharing the bottleneck to maximize the throughput.

The existing MPTCP performs well for traditional routing-based networks. However, such a routing method is shown to have a higher number of transmissions compared to OR, which result in delay [177]. In next-generation networks such as IoT, low delay is a significant requirement, thus, traditional routing-based MPTCP schemes cannot fully satisfy the need. Therefore, a research gap exists to improve the protocol. OR is a beneficial routing model particularly in wireless networks. It is shown that OR can increase packet delivery rate and reliability of data transmission by using broadcast technique in wireless communication channel [163]. Multiple relays could receive the same message and forward it, thus, the probability of successful packet delivery from source to destination is significantly increased. Consequently, this increases network throughput which is a key metric in wireless communication systems [178]. It also can increase goodput in wireless networks if the channels are in good condition. More importantly, OR could reduce delay since unnecessary re-transmission is avoided [179]. Re-transmission only occurs when the message is not received by any node. However, one of the issues of OR is overhead [160]. The communication between the relay nodes in the coordination process may cause network collision or congestion. The solution to this concern can be carrying out the OR coordination by using the fixed network bearer between wireless base stations, and not over the wireless bearer between the network base stations. Another challenge is the degraded performance of OR when interacting with TCP [180]. This due to the fact that OR requires more packets to transmit than a typical TCP window [177]. The solution to this issue could be isolating OR from TCP (e.g., using proxy). In this chapter, the opportunity is taken to adapt OR in MPTCP and enhance the IoT wireless sensors network. To the best of the author's knowledge, this is the first novel work which combines Multipath TCP with Opportunistic Routing.

4.3 Proposed Design

In TCP/IP system, the application layer generates data traffic in a specific rate and passes it to the transport layer for processing. The traffic reaches the transport layer in a form of stream [12]. It is known that the traffic cannot be sent as a stream, thus, the transport layer strips it into segments (packets) [11]. In this design, the MPTCP receives the application layer's stream, divides it into packets and stores them in a shared sender buffer. This buffer is shared between all the paths. Each path of the MPTCP broadcasts the packet that is assigned to it. The broadcast data packet includes a unique sequence number for the data packet and an identification number for each of the source and the destination nodes. In contrast to the traditional OR, the source node in this design does not have a forward list. In fact, it does not contribute to the routing task. This reduces the complexity of the presented system. The forwarding procedure is left to the neighbour nodes. When the packet is successfully received by the neighbours, they cooperate with each other to decide which node forwards it. Only one is allowed to forward the packet. The chapter assumes that the nodes are aware of the distance between each other, therefore, the priority is for the closest node to the destination. All other nodes discard that packet. If all neighbour nodes have the same distance to the destination, a random forwarder is selected. The process is repeated among the nodes until the source packet reaches the destination. If the destination node successfully obtains the packet, it broadcasts an acknowledgment. When the acknowledgment packet is received by the neighbours, they drop the current data packet and forward the acknowledgment packet back to the source node using the same OR technique. If the source node does not receive an acknowledgment within the Re-transmission Timeout (RTO) period that is established by MPTCP, it automatically rebroadcast the packet.

4.4 System Model

The components of the IoT networks are identified as follows: (i) Hardware – which includes sensors and communication equipment, (ii) Middleware – that refers to data storage and analysis (i.e., Internet gateways), and (iii) Presentation – which means data visualisation (i.e., applications) [181]. Figure 4.4 illustrates the general IoT network architecture. In this chapter, only the network between the sensors and the middleware is studied. The chapter considers a Wireless Sensors Network (WSN) with 10 wireless nodes distributed uniformly as shown in Figure 4.5. The sensors are deployed in a network area of 180 X 180 m^2. The S node is the source sensor and D is the destination node. The D node in the proposed model is acting as an IoT gateway. It can be a smartphone that is connected to the Internet using Wi-Fi or 5G cellular communication. S is installed with a camera which senses an object (for any purpose), records a video and sends it to D to take an action. The D node acts accordingly by processing the video and sending it to the remote server for further processing. Nevertheless, only the communication between the source sensor and the gateway is considered, thus, the study of the network afterward is outside the scope of this chapter. In the IoT network, only the S and the D nodes are MPTCP capable which means they can transmit or receive data through different paths simultaneously. Both the source

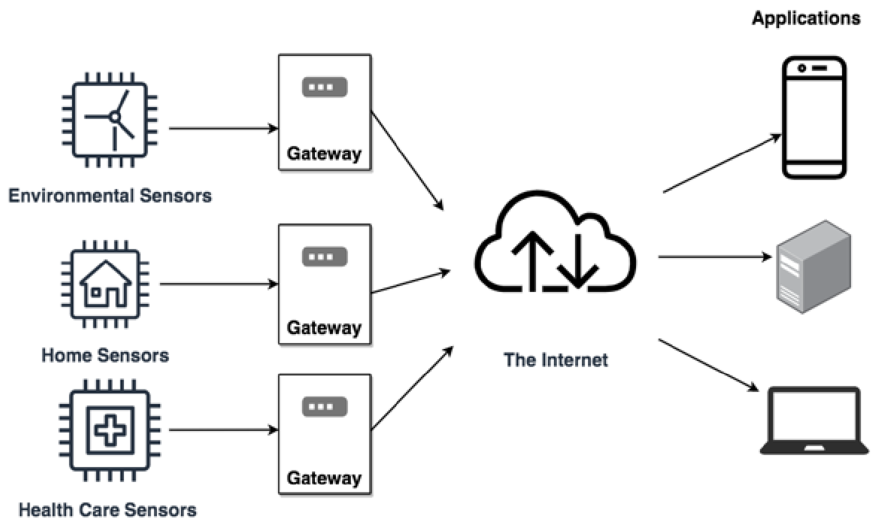

Fig. 4.4 IoT architecture consists of hardware (sensors,communication stack), middleware (gateways) and presentation (applications).

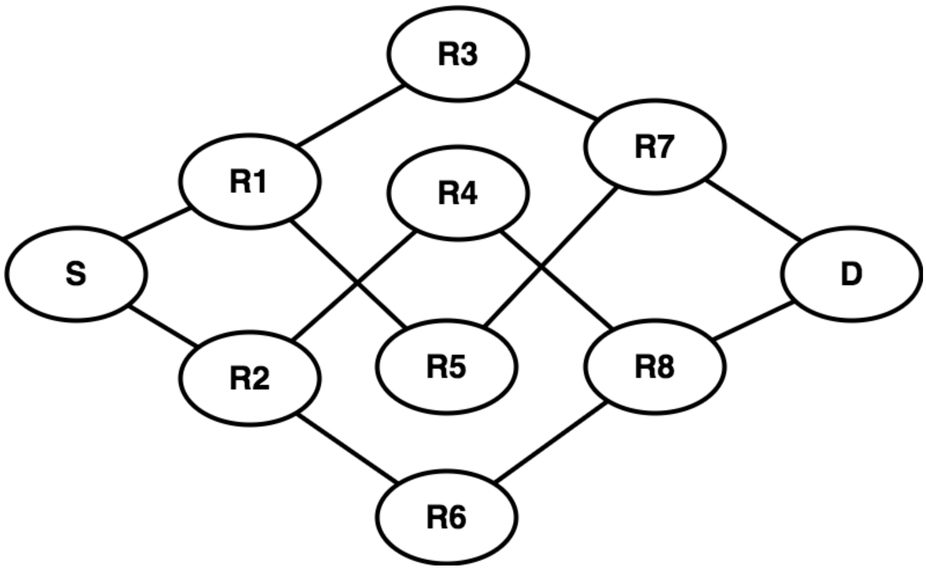

Fig. 4.5 An illustration of the considered IoT network.

and the gateway have 2 Wi-Fi interfaces ready for transmission. This is because Wi-Fi is a short distance network with high data rate which is convenient in this scenario. In addition, all other nodes act as relays. Since this chapter aims to examine the proposed design in a lossy network, all links in the network have a loss probability of 0.4.

In this work, it is assumed that there is no interference between the sensor nodes (e.g., using OFDMA communication channels). As previously mentioned, the number of transmissions refers to the number of times a packet is transmitted/re-transmitted from source to destination. Taking this into consideration, end-to-end delay T can be defined as

$$T = \sum_{n=1}^{N} NT_n.TD \tag{4.1}$$

where N is the total number of packets, NT_n is the number of transmissions for the nth packet and TD is the transmission delay. Transmission delay is known as the time taken by a packet to be successfully transferred into the communication link. Thus, TD can be expressed as

$$TD = \frac{PS}{TR} \tag{4.2}$$

where PS and TR are the packet size and the transmission rate respectively.

In addition, the startup delay considered in this chapter is evaluated based on the number of transmissions. The chapter assumes that 45 packets are required to display the first video frame (the first picture in a video stream).

4.5 Preliminary Experimental Results

In this section, Figure 4.5 is considered as the network topology. The simulation tool used for evaluating the performance of the proposed OR-based Multipath TCP is MATLAB-NS3 2020 [182]. The existing TCP and MPTCP in MATLAB-NS3 2020 are used, however, the author developed MPTCP-TSC and ReMP TCP protocols. Specifically, the scheduler of standard MPTCP is changed to implement the aforementioned schemes. In sub-section 4.5.1, TCP/IP stack is utilised to compare the selected protocols since they are compatible with the existing TCP/IP model. However, the stack is changed in sub-section 4.5.2 to adapt the OR method. This change includes introducing an adaption layer, called OR_MPTCP, which runs on top of the IP layer. The purpose of this layer is to hide the opportunistic routing from the transport layer. The reason is that exposure of OR to the transport layer may result in out-of-order packet arrivals, hence, hiding OR is a better solution [180]. The IP layer uses broadcasting when forwarding the packets. In addition, this chapter compares the selected schemes with respect to total number of data transmissions, delay and number of useful packets received. A total of 1000 packets are being transmitted to the destination. The simulation parameters can be seen in Table 4.1. The values of the parameters

Table 4.1 : Simulation Parameters

Simulation Parameter	Value
Simulation tool	MATLAB-NS3 2020
Coverage area	180 x 180 m^2
Transmission range	45 meters
Loss probability	0.4
Congestion control for MPTCPs	Coupled
Congestion control for TCP	NewReno
RTT	100 ms
RTO	200 ms
Number of packets (N)	1000
Number of paths (in the MPTCP)	2
IEEE protocol	Wi-Fi 802.11
Transmission rate (TR)	6 Mbps
Packet size (PS)	1 KB

are chosen according to [54], [159] and [73] for fair comparison. In the following sub-sections, the simulation results will be explained.

4.5.1 Schemes Without OR

In this sub-section, the chapter compared MPTCP, MPTCP-TSC, ReMP TCP, as well as regular TCP in an IoT wireless network. The protocols are simulated without exploiting the Opportunistic Routing technique to analyse the schemes and determine the best case in the experiment. Figure 4.6 illustrates the comparison of the benchmark schemes in terms of the total number of transmissions (represented by Y-axis) and the number of packets usefully received (represented by X-axis). It can be seen from Figure 4.6 that TCP has the worst performance compared to all MPTCP schemes where it required a total of 6752 transmissions to receive 1000 packets successfully at the destination. The reason is that TCP is a single path protocol and this results in a higher number of re-transmissions. The MPTCP performance is better compared to the TCP protocol as the single flow can be split into multiple parallel flows and the number of transmissions reduced to 6332

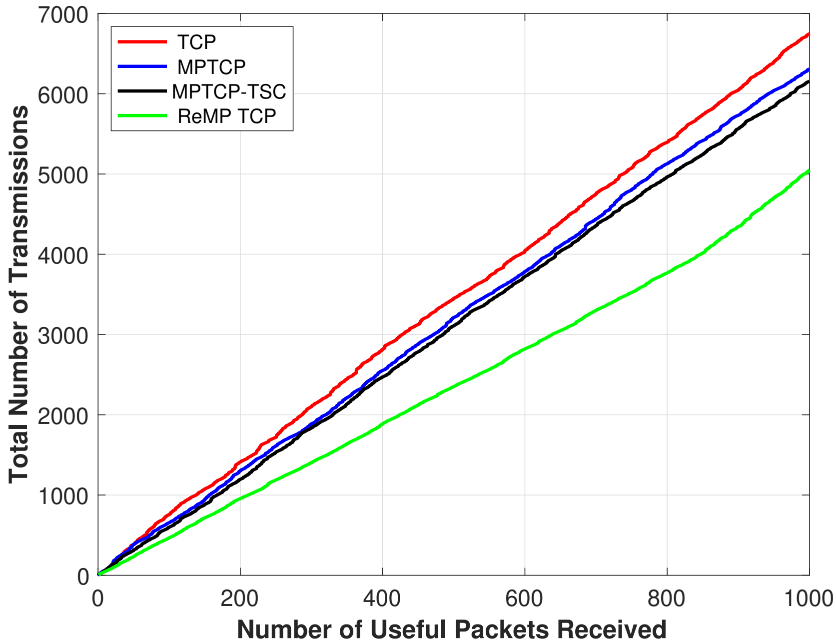

Fig. 4.6 A comparison of benchmark schemes in total number of transmissions versus number of packets usefully received.

in the experiment. The performance of the MPTCP-TSC has shown good results and achieved an acceptable total number of transmissions, particularly 6100, in comparison with TCP and MPTCP. This is because the MPCP-TSC allocates a transmission rate to each sub-flow taking into account the delay constraints. The ReMP TCP has the best performance of all the schemes with only 5056 packet transmissions. The previous Multipath TCP schemes use all available paths for data transmission where ReMP TCP uses some paths for redundant data to reduce re-transmission. Therefore, ReMP TCP is the best scenario.

Furthermore, the performance of each protocol with respect to end-to-end delay is evaluated. Figure 4.7 illustrates the comparison of the benchmark schemes in terms of number of packets usefully received (represented by X-axis) and end-to-end delay (represented by Y-axis). It can be observed from Figure 4.7 that TCP achieved the highest delay where it required 1.12 seconds to receive 1000 packets successfully at the destination. MPTCP and MPTCP-TSC protocols perform

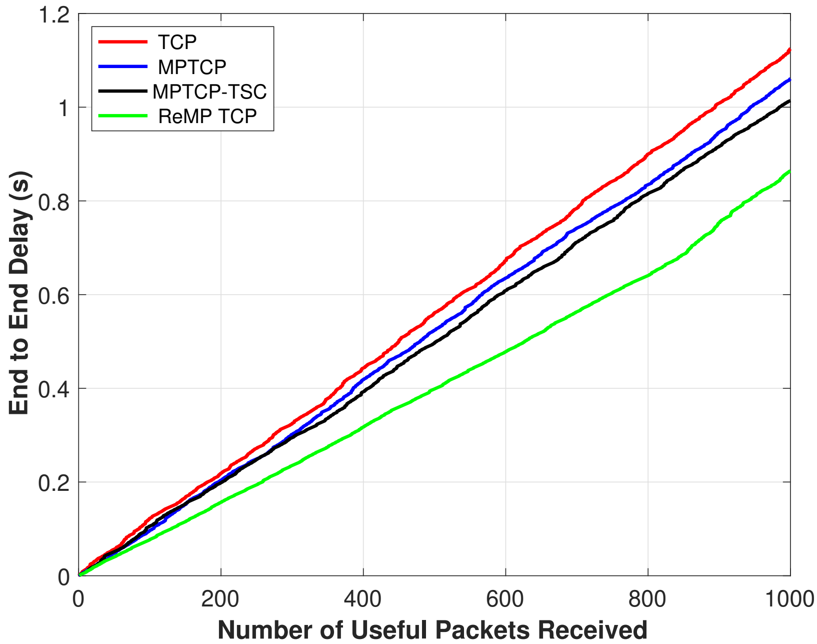

Fig. 4.7 End-To-End delay evaluation (in seconds) for each protocol with respect to number of useful packets received without exploiting OR technique.

similar to TCP at the beginning because the number of packets is too small to illustrate the difference between the two schemes. However, when the number of packets reaches about 300, MPTCP-TSC demonstrates a lower delay than MPTCP. The reason is that MPTCP-TSC considers path characteristics before transmission, therefore, unnecessary delay can be avoided. MPTCP needed 1.06 seconds to receive the traffic correctly and MPTCP-TSC required 1.01 seconds. In contrast to MPTCP and MPTCP-TSC, ReMP TCP duplicates packets and sends them over all the available paths simultaneously. Although this approach results in bandwidth underutilisation, it achieved the lowest delay in the experiment with 0.86 seconds.

4.5.2 Schemes with OR

In this sub-section, the protocols are improved by considering the OR method. Nevertheless, before integrating OR to the existing schemes, the chapter first simulated OR alone without any transport layer protocol to evaluate its performance. Then, each OR-based MPTCP scheme was evaluated. Figure 4.8 presents the results of the experiment where the X-axis represents the number of packets usefully received and the Y-axis represents the total number of transmissions. It was found that OR was the worst scenario in the simulation. It required a total of 6090 transmissions to successfully receive 1000 packets at the destination node. This is because OR does not consider flow control between the sender and receiver, which leads to frequent packet drops, therefore, frequent re-transmissions. Further, transport layer schemes improved significantly. TCP with OR reduced the transmission count by 12.7% in comparison to the previous result in Figure 4.6. OR-based MPTCP achieved a 12.4% reduction in the number of transmissions compared to plain MPTCP. The MPTCP-TSC scheme using OR decreases total transmissions by 10% in comparison with existing MPTCP-TSC. Lastly, when ReMP TCP was combined with the OR, the number of transmissions dropped by 8.4% compared to ReMP TCP without OR. Additionally, it is shown that ReMP TCP outperforms the aforementioned schemes with and without OR. Table 4.2 demonstrates a straightforward comparison between existing MPTCP protocols (Figure 4.6) and OR-based MPTCP protocols (Figure 4.8) in terms of the number of packets transmitted and the number of packets usefully received. It should be mentioned that the number of packets usefully received is the same for both scenarios. The OR-based MPTCP protocols show reduction in packet transmissions compared to the existing MPTCP protocols by at least 8.4%.

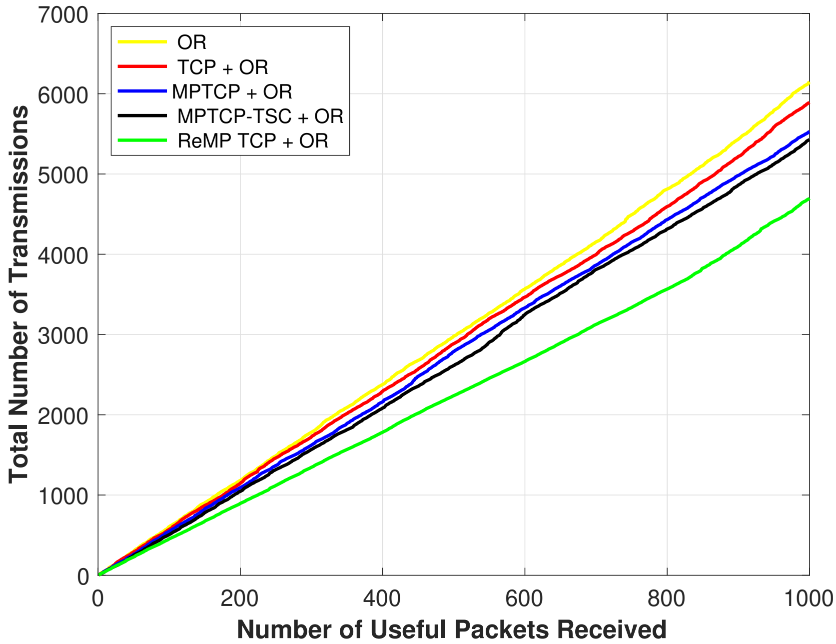

Fig. 4.8 A comparison of the OR-based MPTCP protocols in total number of transmissions versus number of useful packets received.

Table 4.2 : Summary of the results in Fig. 4.6 (existing schemes) and Fig. 4.8 (OR-based MPTCP schemes).

Schemes	**The number of packets transmitted (existing schemes)**	**The number of packets transmitted (OR-based schemes)**	**The number of packets usefully received**
ReMP TCP	5056	4630	1000
MPTCP-TSC	6100	5430	1000
MPTCP	6332	5546	1000
TCP	6752	5890	1000
OR	——	6090	1000

Moreover, this chapter evaluated end-to-end delay for OR itself and for each OR-based MPTCP scheme as shown in Figure 4.9. The X-axis represents the number of packets usefully received and the Y-axis represents the end-to-end delay. It is illustrated that OR was the worst case, which is similar to the result in the previous sub-section. It needed 0.99 second to receive the traffic correctly at the destination. However, in this simulation, all other schemes required less than 0.99 second to receive 1000 packets. In comparison with Figure 4.7, the ReMP TCP delay dropped from 0.86 to 0.76 second (11.6% reduction) where

MPTCP-TSC measurement diminished from 1.01 to 0.90 second (10.8%). Also, MPTCP delay reduced from 1.06 to 0.92 second (13.2%) when exploiting OR technique and TCP performance improved to require 0.97 second instead of 1.12 (13.3%) in the previous result.

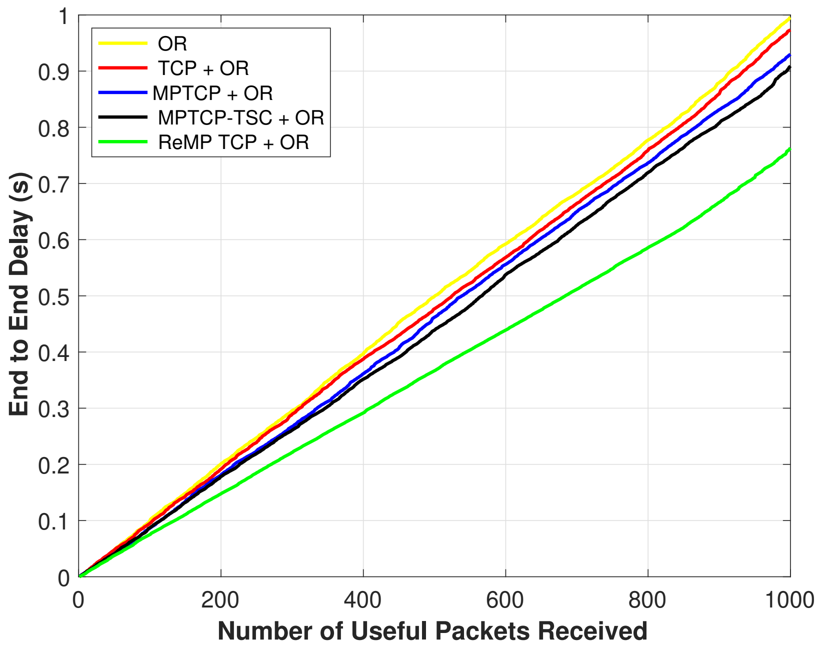

Fig. 4.9 End-To-End delay evaluation (in seconds) for each OR-based MPTCP scheme with respect to number of packets usefully received.

Table 4.3 presents summary statistics of the results in Figure 4.7 (existing MPTCP protocols) and Figure 4.9 (OR-based MPTCP protocols) with respect to the number of packets usefully received and the end-to-end delay. Similar to the statistics in Table 4.2, the number of packets usefully received is the same for both scenarios. The OR-based MPTCP protocols shows reduction in delay compared to the existing MPTCP protocols by at the minimum 10.8%. Therefore, the proposed schemes are demonstrated to be more effective and sufficient in terms of demanding a small number of transmissions and low delay in the IoT network.

Table 4.3 : Summary of the results in Fig. 4.7 (existing schemes) and Fig. 4.9 (OR-based MPTCP schemes).

Schemes	**The number of packets usefully received**	**End-to-end delay (existing schemes)**	**End-to-end delay (OR-based schemes)**
ReMP TCP	1000	0.86	0.76
MPTCP-TSC	1000	1.01	0.90
MPTCP	1000	1.06	0.92
TCP	1000	1.12	0.97
OR	1000	——	0.99

The work is further extended to demonstrate the total number of transmissions required for each OR-based MPTCP protocol to start-up a video at the destination node. As mentioned before, it is assumed that 45 packets are required to display the first video frame. Figure 4.10 shows the comparison between the schemes and it can be observed that TCP is the worst case. This is due to single path utilization where frequent packet loss occurred. MPTCP performs better than TCP with a reduction of 28 transmissions. This shows the advantage of splitting a single flow into several sub-flows. The MPTCP-TSC forwards the same traffic to the destination with a specific transmission rate allocated for each sub-flow. This resulted in requiring 17 transmissions less than MPTCP. Finally, the ReMP TCP achieved the lowest number of transmissions required to run the first frame with a 15 transmission reduction from MPTCP-TSC. In addition, Table 4.4 illustrates the total number of transmissions for each protocol. The newly designed schemes showed better performance than the schemes without OR and diminished the start-up delay by requiring a smaller number of transmissions for the first video frame.

To summarize the simulation results, ReMP TCP outperforms the other schemes in the IoT network. Its performance is improved further by using OR. Also, it can be observed that the OR technique provides considerable enhancement to the existing protocols and reduces the number of transmissions

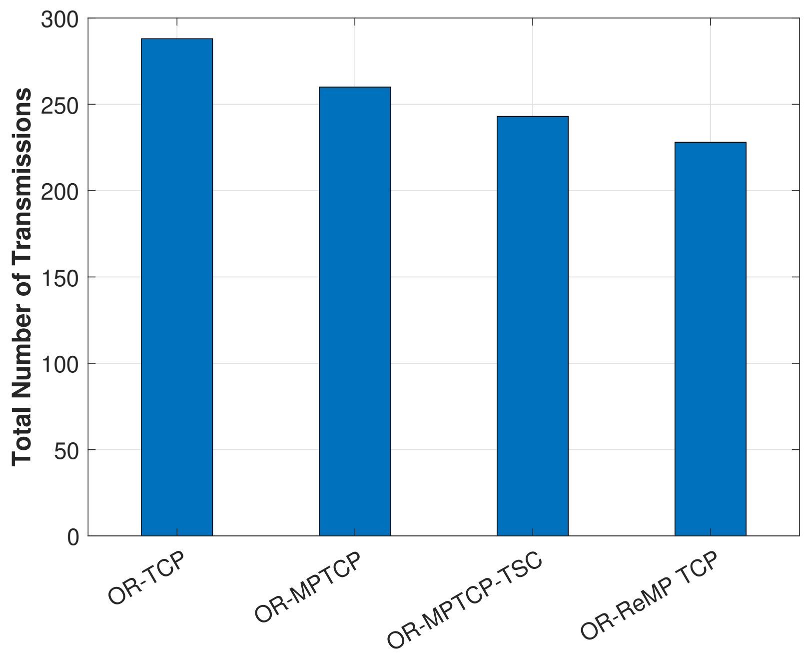

Fig. 4.10 Demonstration of the total number of transmissions required to display the first video frame (45 packets) at the destination for each protocol.

Table 4.4 : The total transmissions count for OR-based MPTCP protocols to successfully send the first video frame to the destination.

Schemes	Total number of transmissions
ReMP TCP	228
MPTCP-TSC	243
MPTCP	260
TCP	288

which consequently reduces delay. Thus, using OR with MPTCP protocols is a promising method for the next generation networks.

4.6 Summary

Although delay in MPTCP has been significantly enhanced in recent years, the high number of data transmissions remains an issue. In this chapter, MPTCP delay is reduced by reducing the number of transmissions using the OR technique.

OR is a routing model used to increase the delivery rate and reliability of data transmission in wireless networks by using the broadcasting method. This enables each subflow data to be received by multiple relays. The chapter first compared a number of MPTCP protocols namely, traditional MPTCP, MPTCP Traffic Splitting Control (MPTCP-TSC) and Redundant MPTCP (ReMP TCP) in an IoT environment. It was found that ReMP TCP outperforms other protocols. Then, OR was adapted to all the aforementioned protocols. The results showed that OR-based MPTCP schemes outperform existing schemes. The chapter further compared the OR-based MPTCP protocols in terms of startup delay. It was found that ReMP TCP is better than other schemes. In the next chapter, the author describes the last contribution of this thesis, i.e. Full Duplex communication and MPTCP.

Chapter 5

Feasibility Analysis of Full Duplex Technology in Enhancing MPTCP for URLLC Applications in B5G-IoV Networks

This chapter investigates the possibility of applying Full Duplex (FD) technology to Multipath TCP (MPTCP). It also describes a novel scheme, called FD-based multi-path transmission control protocol (FDMP), that is proposed for ultra-reliable low-latency communication (URLLC) in beyond 5G Internet of Vehicle (B5G-IoV) networks. In addition, the content of this chapter is submitted to *IEEE Transactions on Vehicular Technology* for publication and it is currently under review.

The chapter is structured as follows. The introduction is presented in Section 5.1. In Section 5.2, the chapter illustrates the considered system model. In Section 5.3, the chapter first analyses the shortcomings of MPTCP. Then, the proposed

FDMP is introduced. Moreover, a comparison between the two protocols is presented to elaborate the benefits and the cost of applying FD technology in enhancing MPTCP for URLLC applications in B5G-IoV networks. In Section 5.4, the performance of MPTCP and FDMP is evaluated and analysed. Finally, the chapter is summarised in Section 5.5.

5.1 Introduction

According to the latest frozen 3GPP Rel-16 standard, current vehicular communication technologies are only capable of supporting basic vehicle-to-everything (V2X) applications in Internet of Vehicle (IoV) networks, and future Quality of Service (QoS) requirements of advanced V2X applications will become more stringent [183]. Therefore, many new technologies are undergoing extensive investigation for enhancing the performance of beyond 5G IoV (B5G-IoV) networks. Full duplex (FD) technology is a promising solution for many existing networks. However, applying FD technology is not only a controversial question in general B5G networks, but it is also an even more worrying concern in IoV networks [184].

Besides the FD technology, future V2X applications also require support throughout Physical- (PHY-), Medium Access Control- (MAC-) and transmission control techniques. In the literature of V2X communications, many works focus on enhancing the PHY and MAC layer techniques. For example, in the PHY layer, the authors in [185] investigated the potential of multiple physical layer security techniques in improving the secure transmission in cellular-V2X networks. Also, in [186], the authors investigated FD collision detection and avoidance from the PHY layer perspective. In the MAC layer, the authors in [187] proposed a scheduling scheme based on the location and ordering of Vehicular

User Equipments (VUEs). In [188], the authors surveyed the latest developments in the standardisation of radio access technologies. On the other hand, researchers are also conducting extensive work in enhancing the transport layer transmission control, such as the conventional half duplex (HD) TCP and MPTCP. For example, the authors in [66] proposed a congestion control and scheduling mechanism, which is shown to outperform the MPTCP in heterogeneous wireless environments. However, to the best of the author's knowledge, B5G-IoV networks require transport layer support, but none of the works have investigated MPTCP.

Motivated by the lack of study in the transport layer for B5G-IoV networks, and the discussion on the feasibility of FD technology, this chapter applies the simultaneous transmission and sensing feature of FD technology to the MPTCP to address this concern in B5G-IoV networks. First, the communication architecture of B5G-IoV networks is introduced. Then, the chapter analyses the problems if MPTCP is applied to B5G-IoV networks without modification, because MPTCP was not designed for ultra-reliable low-latency communication (URLLC) applications. After that, the chapter proposes an FD-based multi-path transmission control protocol, named FDMP, aiming to improve the reliability and reduce the latency. Finally, the feasibility of FD technology in enhancing MPTCP for URLLC applications in B5G-IoV networks is analysed.

5.2 System Model

This chapter considers an IoV network in which VUEs are assumed to be equipped with the FD technology. Moreover, due to the fact that current 5G new radio V2X technology does not have backward compatibility, and it will be achieved through a dual-radio system [188], the chapter assumes that the 4G path and the 5G path coexist in the network, as shown in Fig. 5.1. In addition, the 4G and 5G paths are assumed to be the only two available sub-flows, since the work in [189] has shown that an MPTCP connection with two sub-flows is enough in most practical cases. Furthermore, it is also assumed that the VUEs are capable of sensing the channel status via the most widely-used energy detection technique, and autonomously executing the proposed protocol which is explained in detail in Section 5.3.

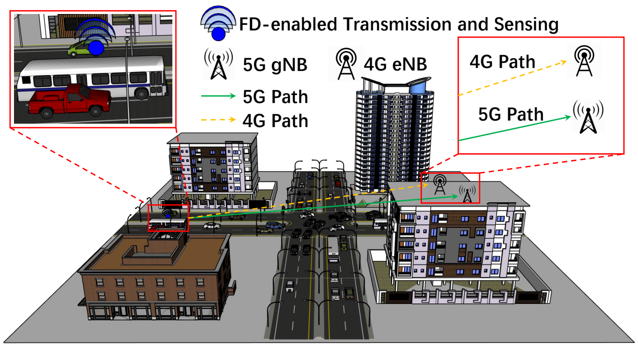

Fig. 5.1 An illustrative diagram of the IoV Network

Because of the rapid environment change and hardware imperfection, the deployed FD sensing technology is considered to be insufficient to provide perfect sensing results. Therefore, decisions in the proposed protocol are made with certain probabilities based on the measured signal strength. In this chapter, **detection probability** is expressed as the probability that a VUE or a base station correctly finds the occupancy of a time-frequency resource when the resource is actually taken by at least one other VUE. Also, **mis-detection probability** is defined as the probability that a VUE finds a resource free to be exploited when the resource is actually taken by at least one other VUE. Further, **false alarm probability** is explained as the probability that a VUE falsely finds the occupancy of a resource when the resource is actually free to be exploited.

Finally, it is assumed that the VUEs enjoy virtually endless energy and computational resource, so that the processing delay depending on the hardware calculation capability is ignored in this work. Moreover, the propagation delay is also ignored in this analysis due to the following two reasons. First, the distance between the transmitting VUE and its receiving UE varies from application to application. Second, the distance is relatively small in comparison to the speed of propagation of the electromagnetic wave. In other words, this chapter focuses on the reliability and latency analysis of the protocol execution performance.

5.3 Transmission Control

In this section, the chapter first analyses the shortcomings of the conventional MPTCP protocol. Then, it introduces how the FD sensing feature is exploited and integrated into the proposed FDMP design, and what additional modifications are made in order to enhance the performance for supporting URLLC applications in future B5G-IoV networks.

5.3.1 Problem Analysis

MPTCP is a standardised transport layer protocol that is designed based on the TCP protocol to increase the network throughput and robustness against path failure due to the unknown and fast-changing channel status [17]. However, MPTCP suffers from the following problems, if it is applied to B5G-IoV networks without modification. First, MPTCP design has a connection establishment phase (a.k.a. the handshake procedure), which enhances reliability, but increases latency, because packets cannot be transmitted until the connection has been established successfully. Second, MPTCP design incorporates a congestion control mechanism and a re-transmission mechanism, both of which depend on the passive reception of the acknowledgement (ACK) information. In other words, latency can be shortened if a novel proactive acknowledgement mechanism can be deployed. Third, MPTCP suffers from the re-transmission timeout control problem, which produces a large round-trip time (i.e. latency) as described and shown in [190]. Hence, MPTCP requires enhancement in the re-transmission mechanism design.

5.3.2 The Proposed FDMP Design

The core motivations of the FDMP design are as follows. The first motivation is to remove the connection establishment procedure, so that the connection time can be saved. The second motivation is to replace the passive acknowledgement strategy with a novel proactive acknowledgement strategy through FD sensing technology, so that the protocol does not require the reception node to send an acknowledgement packet, and the acknowledgement latency is shortened. The last motivation is to shorten the aforementioned re-transmission timeout control process, so that the overall latency can be further reduced for better supporting future stringent URLLC applications in B5G-IoV networks.

As shown in Fig. 5.2, the proposed FDMP design starts from a sensing phase. When a packet is generated at a generic VUE, V_x, it measures the energy level of the 5G path for resource occupancy status prior to the start of its transmission. The received signal, $r_{bt}[n]$, is expressed as [191]

$$r_{bt}[n] = \begin{cases} w[n]; \quad H_1 \\ s[n] + w[n]; \quad H_2 \end{cases} \tag{5.1}$$

where $w[n]$ indicates the noise signal, $s[n]$ indicates the transmitting signal from another VUE, H_1 refers to the hypothetical case in which the resource is free to be used, and H_2 refers to the hypothetical case in which the resource is occupied by another VUE and not available to be used. The energy detection test statistic,

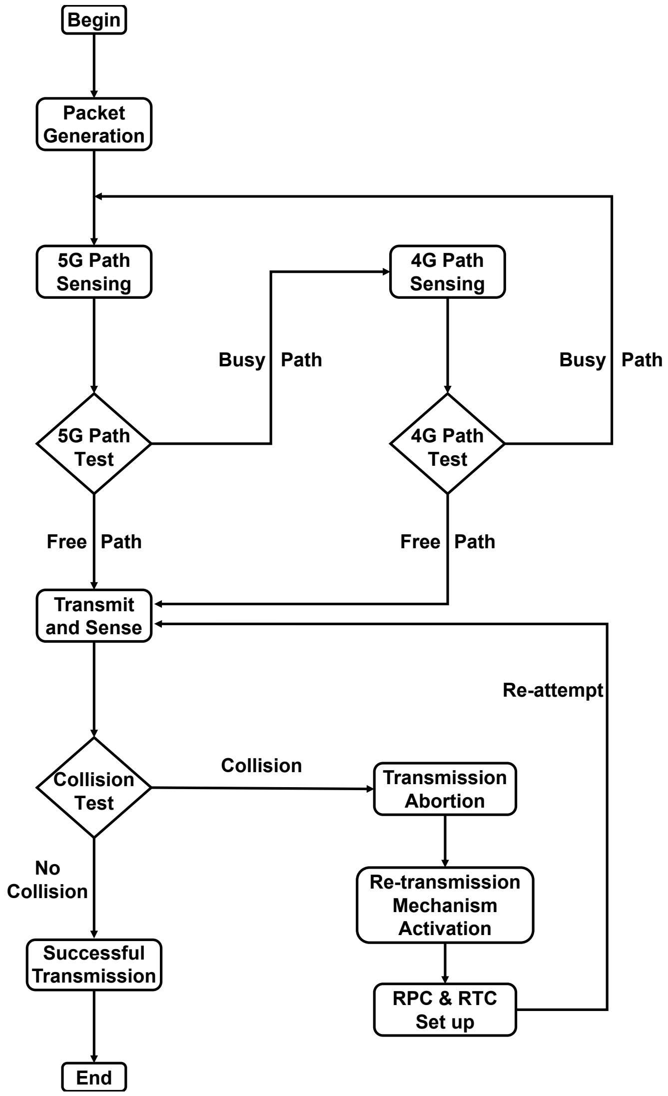

Fig. 5.2 The workflow diagram of FDMP design

En, is defined as

$$En = \frac{1}{N} \cdot \sum_{n=1}^{N} |r_{bt}[n]|^2, \tag{5.2}$$

where N refers to the total number of samples, which is given by $N = \tau \cdot f_s$, where τ denotes the sensing time, and f_s denotes the sampling frequency. Therefore, the detection probability before transmission, P_d^{bt}, is given by [192]

$$P_d^{bt} = Q((\frac{\varepsilon_{th_1}}{\sigma_w^2} - \Upsilon_s - 1) \cdot \sqrt{\frac{N}{2\Upsilon_s + 1}}), \tag{5.3}$$

where ε_{th_1} indicates the decision threshold, above which the resource is considered to be occupied by another VUE, σ_w^2 represents the noise variance, Υ_s indicates the energy measurement of the transmitting signal, $s[n]$, and $Q(.)$ is the Q function operator. In addition, the false alarm probability, P_f^{bt}, is given by [192]

$$P_f^{bt} = Q((\frac{\varepsilon_{th_1}}{\sigma_w^2} - 1) \cdot \sqrt{N}), \tag{5.4}$$

and the mis-detection probability is expressed as

$$P_m^{bt} = 1 - P_d^{bt}. \tag{5.5}$$

Afterwards, if a free 5G path is found by the V_x, it starts transmission immediately. Otherwise, V_x switches to the 4G path and senses for an available resource. If a free 4G path is found, V_x takes the 4G path to transmit. Otherwise, V_x switches between the 4G and 5G paths and keeps sensing for an available resource until a free path has been found. In addition to transmission, this continuous path switching and sensing phase can also be terminated by a new packet arrival. Then, V_x will re-execute the FDMP design from the beginning. The first motivation for proposing such a design comes from the frequent update requirement of URLLC applications. The second motivation comes from the congestion control perspective.

When a free path has been selected for transmission, V_x transmits in an FD manner. In other words, V_x is detecting for a potential collision caused by simultaneous path and resource selection whilst transmitting the packet. The received signal, $r_{dt}[n]$, is expressed as

$$r_{dt}[n] = \begin{cases} \sqrt{\eta} s_i[n] + w[n]; & H_3 \\ s[n] + \sqrt{\eta} s_i[n] + w[n]; & H_4 \end{cases} \tag{5.6}$$

where $s_i[n]$ indicates the self-interference (SI) signal, η indicates the SI suppression (SIS) factor. The SIS factor is the percentage of the residual SI after applying SI suppression [191]. The probability of detection, P_d^{dt}, is defined as [192]

$$\begin{aligned} P_d^{dt} =& Q((\frac{\varepsilon_{th_2}}{\sigma_w^2} - \Upsilon_s - \eta^2 \Upsilon_{si} - 1) \times \\ & \sqrt{\frac{N}{2\eta^2\Upsilon_{si} + 2\eta^2\Upsilon_{si}\Upsilon_s + 2\Upsilon_s + 1}}), \end{aligned} \tag{5.7}$$

where ε_{th_2} is the decision threshold for collision detection, above which the transmission is deemed to be in collision, and the packet is lost. Υ_{si} indicates the energy measurement of the SI signal, and η defines the SIS factor. The false alarm probability, P_f^{dt}, is given by [192]

$$P_f^{dt} = Q((\frac{\varepsilon_{th_2}}{\sigma_w^2} - \eta^2 \Upsilon_{si} - 1) \cdot \sqrt{\frac{N}{2\eta^2\Upsilon_{si} + 1}}), \tag{5.8}$$

and the probability of mis-detection is expressed as

$$P_m^{dt} = 1 - P_d^{dt}. \tag{5.9}$$

If a collision is not detected during transmission, the transmission is deemed to be successful. Otherwise, several VUEs have selected the same path simultaneously for the first attempt. In this case, all VUEs abort the current ongoing transmission immediately and activate a re-transmission mechanism. In this mechanism, each VUE differs its re-transmission in both time and path randomly. First, a generic collided VUE, V_x, sets up a re-attempt path counter (RPC) and a re-attempt time counter (RTC). Then, V_x randomly generates either 0 or 1 for the RPC, where 0 represents the 5G path and 1 represents the 4G path. Afterwards, V_x randomly differs its re-attempt time for α sub-frame(s) and stores this value in the RTC, where $\alpha \in [0, T_{max}]$, and T_{max} denotes the number of the next re-attempt. For example, $T_{max} = 1$ for the first re-attempt, $T_{max} = 2$ for the second re-attempt and so on. Same as the termination policy in the path switching phase, V_x discards the current packet when a new packet has been generated.

5.3.3 Comparison

The proposed FDMP method has modified the MPTCP design from a contention-free protocol to a contention-based protocol. Therefore, the reliability performance is a constraint and should be guaranteed, which is analysed in Section 5.4.

In particular, FDMP has introduced FD sensing feature, a modified scheduler and congestion control method, a proactive ACK mechanism and a sensing-based re-transmission strategy. Hence, the latency is reduced from the connection establishment, the ACK and the re-transmission activation procedures, which can happen in five cases, as shown in Fig. 5.3.

The first case refers to the connection establishment process. FDMP does not have a handshake procedure; therefore, there is no connection establishment latency. The second case refers to the ACK process. FDMP acknowledges a packet proactively through the FD sensing technology; thus, there is no ACK latency. The third case refers to the re-transmission activation according to the negative ACK (NACK) reception. FDMP does not rely on the NACK reception to trigger the re-transmission mechanism; hence, the re-transmission activation latency is reduced. The fourth case refers to the re-transmission activation according to the timeout mechanism. FDMP does not have a timeout mechanism and packets are re-transmitted according to the sensed collision information. Therefore, the re-transmission activation latency is reduced. The last case refers to the packet loss problem. A transmission abortion policy is introduced and integrated into

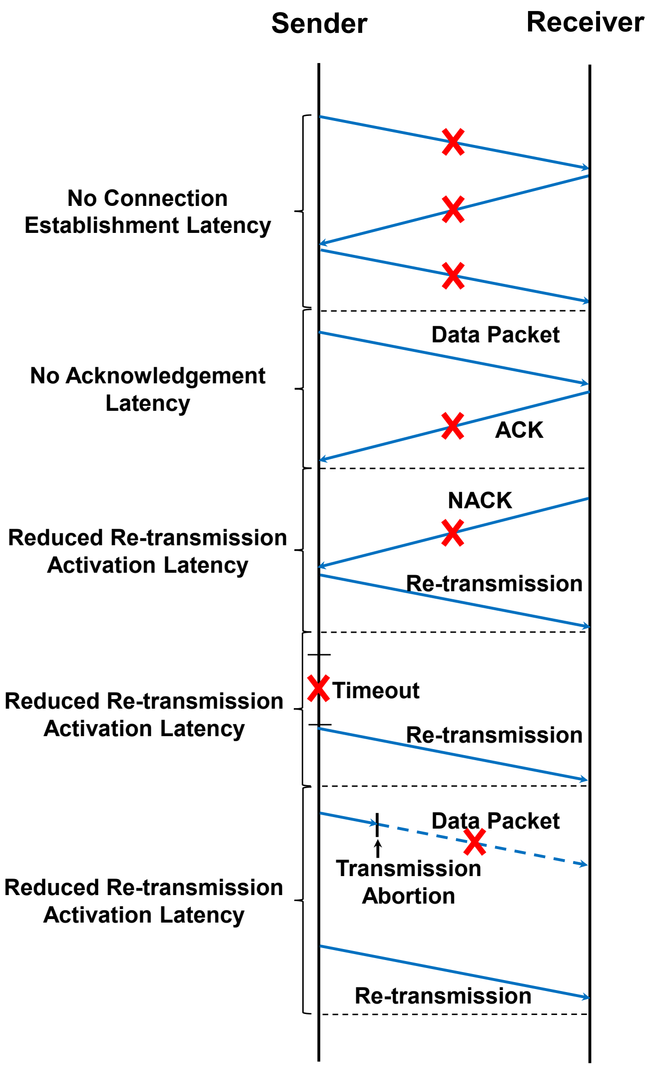

Fig. 5.3 Latency Reduction Comparison

FDMP. Therefore, the re-transmission activation latency can also be reduced when a packet loss is detected.

Whilst enjoying the above benefits, it is also meaningful to analyse the cost of applying FD technology in enhancing MPTCP. FDMP deploys FD technology for simultaneous sensing and transmission. Therefore, it suffers from the SI signal, and requires advanced SIS methods and more expensive hardware to support the SIS methods, whilst MPTCP deploys HD technology, and does not suffer from

SI. In addition, it is more important to find the SIS requirement against different parameters in different scenarios, which are found and analysed in Section 5.4.2.

5.4 Performance Evaluation

5.4.1 Simulation Environment

In this work, a customised B5G-IoV simulator using the *Veins* simulation tool [193] is developed, that combines the network simulator *OMNeT++* and the traffic simulator *SUMO*. Specifically, the author have developed the FD energy detection in the PHY layer, and the standard MPTCP and the proposed FDMP protocols in the transport layer.

In addition, the author selected a specific area for the simulation, called Strand, which is located in central London in the U.K as shown in Fig. 5.4. This is because it is a typical dense scenario. Road and traffic light entities are designed in *Open Street Map (OSM)* software, and then converted into a network using SUMO. Initial generation of VUEs follows the Poisson Point Process (PPP) model. The reason is that the VUEs distribution is proven to follow this model in the literature [194]. Furthermore, packet delivery ratio (PDR) and latency values are first arranged in *OMNeT++*. After that, they are imported into *MATLAB* for plotting and annotation.

5.4.2 Results and Analysis

In this section, the chapter analyses the feasibility of the proposed FDMP design by evaluating the PDR and latency against the SIS factor and network density. It also uses the MPTCP solution as the benchmark for comparison. Assumptions

Fig. 5.4 Simulation area: Strand, London, U.K.

and values are chosen from the 3GPP Rel-16 standard in [195], and listed in Table. 5.1.

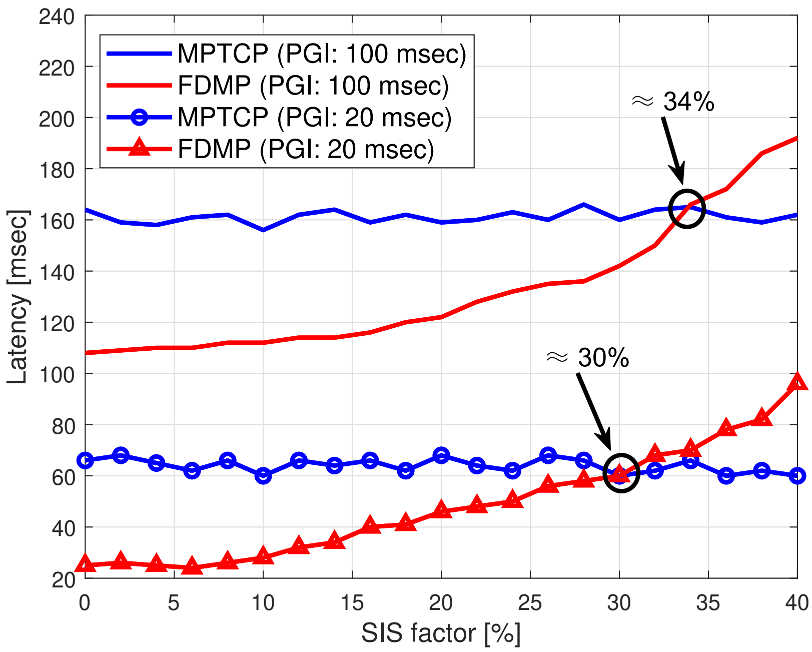

Fig. 5.5 Average latency with respect to SIS factor

Table 5.1 : Simulation Parameters

Parameter	**Value**
Duplex Technology	HD, FD
Target Detection Probability	90%
Channel Bandwidth	10 MHz
Coding and Modulation Method	MCS 9 (QPSK 0.7)
Packet Generation Interval (PGI)	20, 100 msec
Antenna Height	1.5 m
Multiple Access Mechanism	SC-FDMA, OFDMA
Transmission Power	20 dBm
Transmission Rate	6 Mbps
Transmission Control scheme	MPTCP, FDMP
Residual SI	0%-40%
Sensing Technique	Energy Sensing
Density of Vehicles	0-200 VUEs/km
Number of Lanes	2, 4 lanes
VUE Maximum Speed	40 miles/h
Available Sub-flows	4G Path, 5G Path
Packet Size	300, 500 bytes/pkt

SIS Requirement Analysis

In order to analyse the feasibility of applying FD technology to any technology, the first factor that has to be considered is the SIS requirement, because the state-of-the-art SIS method and hardware can only suppress the SI to the level of the noise floor [196]-[197].

Fig. 5.5 and Fig. 5.6 depict the SIS requirement from the latency and PDR perspectives, respectively. In Fig. 5.5, it can be seen that SI does not affect the performance of MPTCP, because MPTCP deploys HD technology, and there is no SI signal. On the other hand, FDMP is affected by the SI signal and the SIS factor. The latency of FDMP increases with the rise of residual SI. The reason is that higher residual SI degrades the detection accuracy, which in turn leads to more iterations to transmit a packet successfully. Furthermore, compared to MPTCP, Fig. 5.5 shows that FDMP can provide shorter latency if the SIS factor is smaller

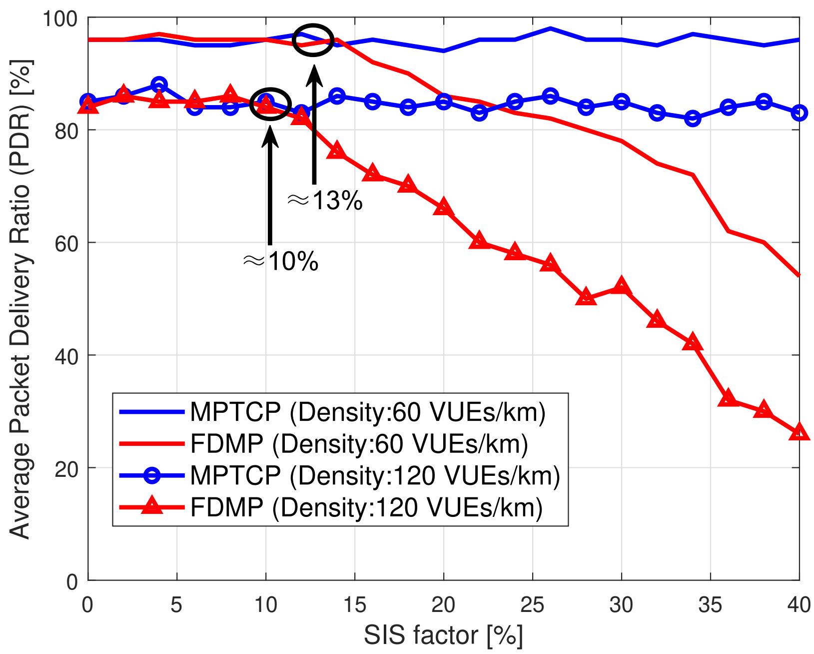

Fig. 5.6 Average PDR against SIS factor

than roughly 34% in the case where PGI is 100 *msec*. Also, FDMP can provide shorter latency if the SIS factor is smaller than about 30% in the case where PGI is 20 *msec*. Compared to the state-of-the-art SIS achievement, these numbers are achievable [196]-[197], which implies that the proposed FDMP is feasible.

Similarly, the feasibility is also shown by Fig. 5.6 in terms of PDR. First, it can be seen that the SIS requirement is about 10 % in the case where the density of the B5G-IoV network is 120 VUEs/km, and it is approximately 13 % in the case where the density is 60 VUEs/km, which is also achievable according to the state-of-the-art SIS techniques [196]-[197]. However, FDMP is not a more reliable transmission control protocol. When the amount of residual SI increases, the PDR of FDMP decreases accordingly. Even if the SIS requirement is satisfied, PDR cannot be further improved by simply deploying more advanced SIS techniques, because PDR also depends on other factors, such as collisions.

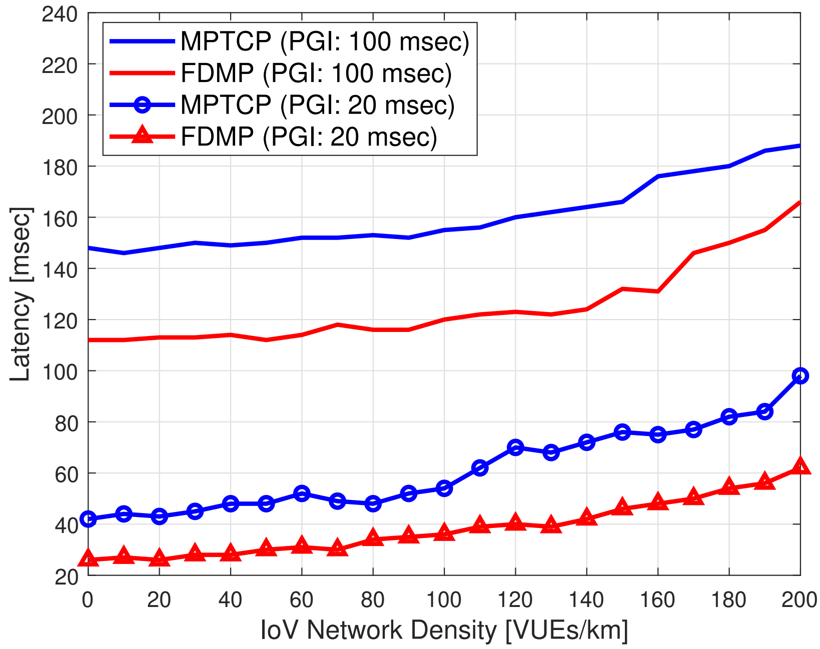

Fig. 5.7 Average latency against B5G-IoV network density

To summarise the results, it could be concluded that although some packet loss may occur, the self-interference suppression requirement for good application performance in FDMP is up to 30% (as shown in Fig. 5.5) considering the state-of-the-art SIS scheme [196]-[197]. The proposed FDMP protocol is feasible and does not require advanced SIS technologies or specific modifications to the PHY layer to be implemented. This is an important advantage and for an application up to 30% SIS requirement is likely to be attainable. Therefore, the proposed model can be useful for a wide range of B5G-IoV applications.

Latency and PDR Analysis

In addition to SIS requirement analysis, this chapter also evaluates the latency and PDR against the B5G-IoV network density, respectively, which are reported in Fig. 5.7 and Fig. 5.8.

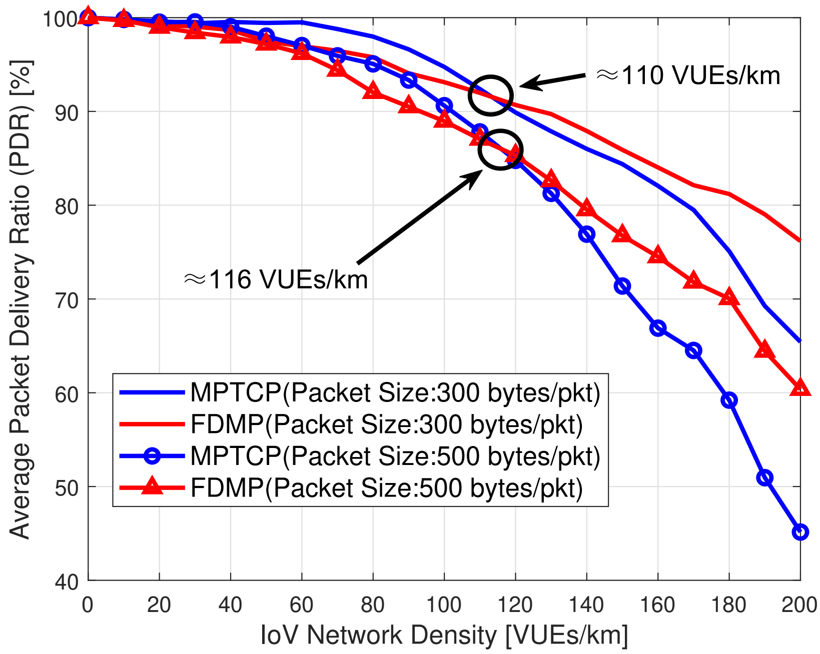

Fig. 5.8 Average PDR against B5G-IoV network density

In Fig. 5.7, it is shown that FDMP can significantly reduce the latency compared to the conventional MPTCP. In addition, when the density increases, the latency of both FDMP and MPTCP will also increase. The latency increment of FDMP is due to the competition between different VUEs, which leads to more iterations in executing the protocol. Whilst MPTCP is a contention-free protocol, the latency increment is due to the network congestion, which also leads to more iterations in executing the protocol.

In Fig. 5.8, it is shown that the PDR of both protocols decreases with the rise of the network density. Moreover, in the sparse network scenario, FDMP cannot provide better PDR performance compared to MPTCP. However, FDMP outperforms MPTCP in the dense network scenario, because the PDR of FDMP is higher when the density goes beyond approximately 110 to 116 VUEs/km, which refers to the dense network scenario. Therefore, it could be concluded that FDMP provides a slightly worse PDR performance in the sparse network scenario, whilst it can significantly enhance the PDR performance in the dense network scenario.

5.5 Summary

With recent developments on the FD technology and 5G vehicular communication technologies, a common and major concern is the feasibility of FD technology in B5G-IoV networks. In this chapter, the author proposed an FD based multi-path transmission control protocol (FDMP) for URLLC applications to address this concern. In particular, the simultaneous sensing and transmission feature of FD technology was exploited to enhance the MPTCP protocol. Three corresponding mechanisms were proposed and integrated into the proposed FDMP protocol, which were the path selection and switching policy, the congestion control mechanism and the re-transmission control design. The chapter focused on the reliability and latency performance as they are the two most significant metrics in evaluating B5G-IoV networks. It also compared the proposed FDMP method to the conventional HD MPTCP protocol. Simulation results showed that the proposed method outperforms the benchmark MPTCP design, FD technology is feasible in enhancing the performance of the MPTCP protocol for supporting URLLC applications in future B5G-IoV networks. In the next chapter, the author concludes this thesis and suggests future research points.

Chapter 6

Conclusion and Future Work

Advances in wireless technology over the past few decades have led to the proliferation of wireless network applications. Currently, wireless networks play a vital role in all aspects of life, be it business, industry, health or leisure. With the massive growth in the data traffic, most Internet of Things (IoT) wireless applications have stringent requirements on throughput and latency. In this thesis, the author contributes to the continuous efforts to find answers and solutions to such challenges. As outlined in the Introduction chapter, since the tasks required in each node in the communication network are enormous, this has led to dividing the tasks into different layers. The most widely implement model is TCP/IP. This thesis considers the fourth layer, specifically the transport layer, when implemented in IoT wireless communication networks. It is well known that the most common transport layer protocol is TCP.

TCP is a stream-oriented protocol. In conventional TCP, the connection between the source and destination in wireless networks involves a single path. However, most current wireless devices are multihomed, i.e. the devices can be associated with several network interfaces (e.g, Wi-Fi, 5G and Ethernet). These networks can be utilised concurrently to transmit data which is a limitation of TCP. This motivated the Internet Engineering Task Force (IETF) to introduce Multipath TCP (MPTCP), which is briefly outlined in Chapter 1. MPTCP has its own set of challenges that are discussed throughout Chapter 2, and several variants of MPTCP have been developed in the recent literature to address these challenges. This thesis contributes to the efforts of improving the MPTCP performance. In particular, the thesis maximises the user utility, improves reliability and decreases the latency of the MPTCP in IoT wireless networks.

In Chapter 3, a new paradigm is presented that jointly designs the congestion control, routing and scheduling algorithms for MPTCP in ad-hoc wireless communication networks. Each source has several network paths available for transmission. A utility maximization problem is investigated, where the utility function is conveniently selected to be the *log()* function. The objective of the optimization problem is to maximize the sum of the log of the source rates subject to a set of constraints. The constraints include the network flow rates, scheduling, and also the queuing delay (taken end-to-end). The variables to be optimized are the source rates and the link capacities among the network nodes on each available path. The resulting joint optimization problem is solved using the Lagrangian. In this chapter, two different algorithms are developed: perfect scheduling (all nodes contribute to the scheduling) and distributed scheduling (only neighbour nodes are allowed to contribute to the scheduling). Furthermore, the chapter considers fixed channels (each link has a time unvarying allocated capacity) and time varying channels (each link has a time varying allocated capacity). Also, the chapter proves analytically that these algorithms are stable and they converge to a global solution. The results are compared to other algorithms that were previously proposed in the literature. The comparison demonstrated that the proposed multipath algorithm is superior to the existing protocols. This can be attributed to the employment of the available multipath connections, and the queuing delay constraint that was imposed in its joint optimal design. The proposed algorithm is able to increase the source rates, expedite the convergence and reduce the congestion prices. Therefore, the presented techniques may be employed in many types of wireless ad-hoc networks, including Internet of Things (IoT) networks.

In Chapter 4, several variants of MPTCP that are available in the literature are compared in wireless IoT networks. The candidate protocols are the conventional MPTCP, MPTCP-TSC (MPTCP-Traffic Split Control) and ReMP TCP (Redundant MPTCP). MPTCP-TSC is a greedy scheme that splits traffic to minimize the cost (which is a weighted sum of the required transmission rates) while satisfying the delay constraints. On the other hand, ReMP TCP sends data redundantly over all available multiple paths in the network. These protocols were not examined for IoT wireless networks before this work. It is shown in the numerical results that ReMP TCP has the lowest number of transmissions and the lowest delay compared to other schemes. To improve the delay further, the chapter also introduces the Opportunistic Routing (OR) protocol to all considered Multipath TCP schemes (MPTCP, MPTCP-TSC and ReMP TCP). Opportunistic Routing is a strategy that benefits from the broadcast capability of wireless networks. The idea of OR is that the transmitted data packet can be received by multiple neighbouring nodes. If at least one node successfully receives the packet, it forward that packet to next hop. The simulation results show that the OR-based MPTCP schemes reduce the number of transmissions and delay compared to the case not using the OR method. The results also reveal that the best protocol in the experiment is ReMP TCP with OR.

In Chapter 5, the thesis investigates the feasibility of Full Duplex (FD) technology in enhancing MPTCP for ultra-reliable low-latency communication (URLLC) applications in Beyond 5G Internet-of-Vehicles (IoV) networks. Since IoV is a special case of IoT, the results obtained in this chapter also apply to IoT wireless networks. FD-capable devices can transmit and receive on the same Radio Frequency (RF) bandwidth at the same time. The receiver side of the device performs Self-Interference Suppression (SIS) of the self-transmitted signal from its own transmitted side. Hence, it can separate and detect the received signal from other wireless nodes in the network. In this chapter, an FD-based multi-path transmission control protocol, named FDMP, is proposed and evaluated. FDMP utilises the FD simultaneous transmission and sensing feature. It incorporates a modified scheduler and congestion control mechanism. FDMP employs a proactive ACK mechanism and a novel re-transmission strategy. Compared to the conventional MPTCP, the SIS requirement for deploying the FDMP design is first analysed. Results revealed that FDMP is feasible with the current state-of-the-art SIS techniques. In addition, FDMP is shown to be able to significantly decrease the latency. Interestingly, FDMP is also shown to perform slightly worse in terms of packet delivery ratio in sparse network scenarios, but it can provide a much better packet delivery ratio in dense network scenarios. As a conclusion, compared to MPTCP, FDMP is feasible and more suitable for URLLC applications in beyond 5G IoV and IoT wireless networks.

It worth mentioning that the main body of this thesis (Chapters 3, 4 and 5) considers separate but complementary aspects of Multipath TCP. That includes optimising throughput and congestion via cross layer design in Chapter 3, reducing delay by adapting a new routing method in Chapter 4, and enhancing latency and delivery rate utilising FD in Chapter 5.

Future Work

There are several research directions that can be followed to improve and extend the scope of the current research. Some of these points are outlined below:

- **Examine the number of network channels available to MPTCP for each node as a design parameter:**
  In the current research, the author assumes that the number of paths (number of channels) in MPTCP is fixed and equal at each node. All nodes in the network have access to multipaths. This may not be available in real life networks. The number of paths may be different at any node. The author's goal is to study the impact of the number of network channels available to each node on the design. This may be achieved by setting another optimization constraint on the number of available channels.

- **Consider separate discrete based simulations:**
  Simulating a real-world environment is one of the approaches, which is widely used in academia, to attain results. Solid research often supports theoretical analyses with simulations. In this thesis, some results are obtained from theoretical analysis only, hence, it would be worth carrying out separate discrete based simulations in future to validate the work.

- **Investigate large network topology:**
  Apart from Chapter 5, the network topologies considered in this thesis are small, typically fewer than 10 nodes. To gain more confidence on the scalability of the proposed solutions in this thesis, larger network sizes should be considered.

- **Utilising machine learning techniques:**
  One of the major necessities of 6G networks is the extensive employment of machine learning in the optimization process of the network parameters. This is due to the large number of optimization parameters in such systems. IoT devices generate massive amounts of data. To be able to analyse and exploit these data and adjust the network parameters, machine learning

techniques can be a solution. As a future work, the author aims to utilise machine learning techniques, such as the Deep Reinforcement learning technique, that can further enhance MPTCP design.

- **Adapting network slicing to MPTCP:**
  The 5G and 6G wireless networks are expected to widely exploit network slicing. Naturally, this will extend to IoT wireless networks. Network slicing provides the capability of dividing a physical network into several logical networks (i.e., slices). Each slice can have its own network characteristics and QoS requirements. It is yet to be investigated how the different variants of MPTCP can be adapted to network slices and what their performances are likely to be. This is an important research point for 6G wireless networks.

- **Combining the separate works of this thesis:**
  It is mentioned in this chapter that Chapters 3, 4 and 5 are separate but complementary aspects of MPTCP. In the future, the author aims to combine the chapters together where a cross layer design could be considered that optimises the MPTCP throughput taking into account OR in the network layer and FD in the MAC layer.

Appendix A

The average stream rate of flows *AF* and *BE* in Chapter 3

In an effort to have an insight into average stream rate, the following tables present the average stream rate of flows *AF* and *BE* employing our proposed algorithm in Chapter 3. The first column represents the source nodes and the first row represents the destination nodes. These tables provide the information about the paths that have been used for each flow when employing our perfect and distributed scheduling algorithms in both fixed and time varying channels scenarios.

Table A.1 : Average stream rate of AF flow using the perfect scheduling algorithm in fixed channel scenario

	A	B	C	D	E	F
A	0	0.176	0.525	0	0	0
B	0	0	0	0	0	0.176
C	0	0	0	0.263	0.263	0
D	0	0	0	0	0	0.263
E	0	0	0	0	0	0.263
F	0	0	0	0	0	0

Table A.2 : Average stream rate of BE flow using the perfect scheduling algorithm in fixed channel scenario

	A	B	C	D	E	F
A	0	0	0	0	0	0
B	0	0	0.211	0.431	0	0.210
C	0	0	0	0	0.210	0
D	0	0	0	0	0. 430	0
E	0	0	0	0	0	0
F	0	0	0	0	0.210	0

Table A.3 : Average stream rate of AF flow using the distributed scheduling algorithm in fixed channel scenario

	A	B	C	D	E	F
A	0	0.274	0.396	0	0	0
B	0	0	0	0	0	0.269
C	0	0	0	0.391	0	0
D	0	0	0	0	0	0
E	0	0	0	0	0	0.391
F	0	0	0	0	0	0

Table A.4 : Average stream rate of BE flow using the distributed scheduling algorithm in fixed channel scenario

	A	B	C	D	E	F
A	0	0	0	0	0	0
B	0	0	0.140	0.531	0	0.132
C	0	0	0	0	0.139	0
D	0	0	0	0	0.531	0
E	0	0	0	0	0	0
F	0	0	0	0	0.130	0

Table A.5 : Average stream rate of AF flow using the perfect scheduling algorithm in time varying scenario

	A	B	C	D	E	F
A	0	0.355	0.345	0	0	0
B	0	0	0	0.347	0	0
C	0	0	0	0.338	0	0
D	0	0	0	0	0	0.685
E	0	0	0	0	0	0
F	0	0	0	0	0	0

Table A.6 : Average stream rate of BE flow using the perfect scheduling algorithm in time varying scenario

	A	B	C	D	E	F
A	0	0	0.241	0	0	0
B	0.241	0	0.241	0.245	0	0.081
C	0	0	0	0.433	0.056	0
D	0	0	0	0	0.673	0
E	0	0	0	0	0	0
F	0	0	0	0	0.081	0

Table A.7 : Average stream rate of AF flow using the distributed scheduling algorithm in time varying scenario

	A	B	C	D	E	F
A	0	0.467	0.194	0	0	0
B	0	0	0	0.466	0	0
C	0	0	0	0.190	0	0
D	0	0	0	0	0	0.656
E	0	0	0	0	0	0
F	0	0	0	0	0	0

Table A.8 : Average stream rate of BE flow using the distributed scheduling algorithm in time varying scenario

	A	B	C	D	E	F
A	0	0	0	0	0	0
B	0	0	0	0.800	0	0.010
C	0	0	0	0	0	0
D	0	0	0	0	0.800	0
E	0	0	0	0	0	0
F	0	0	0	0	0.010	0

References

[1] B. Y. Kimura, D. C. Lima, and A. A. Loureiro, "Alternative scheduling decisions for multipath tcp," *IEEE Communications Letters*, vol. 21, no. 11, pp. 2412–2415, 2017.

[2] K.-L. Du and M. N. Swamy, *Wireless communication systems: from RF subsystems to 4G enabling technologies*. Cambridge University Press, 2010.

[3] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[4] R. L. Haupt, *Wireless Communications Systems: An Introduction*. John Wiley & Sons, 2019.

[5] S. Parkvall, Y. Blankenship, R. Blasco, E. Dahlman, G. Fodor, S. Grant, E. Stare, and M. Stattin, "5g nr release 16: Start of the 5g evolution," *IEEE Communications Standards Magazine*, vol. 4, no. 4, pp. 56–63, 2020.

[6] B. Wi-Fi, "Zigbee and wimax. h labiod, h afifi, c de santis," 2007.

[7] M. U. Mudassir and M. Baig, "Mfvl hcca: A modified fast-vegas-lia hybrid congestion control algorithm for mptcp traffic flows in multihomed smart gas iot networks," *Electronics*, vol. 10, no. 6, p. 711, 2021.

[8] P. Corcoran, "The internet of things: Why now, and what's next?," *IEEE consumer electronics magazine*, vol. 5, no. 1, pp. 63–68, 2015.

[9] K. Xu, Y. Qu, and K. Yang, "A tutorial on the internet of things: from a heterogeneous network integration perspective," *IEEE network*, vol. 30, no. 2, pp. 102–108, 2016.

[10] I. B. F. de Almeida, L. L. Mendes, J. J. Rodrigues, and M. A. da Cruz, "5g waveforms for iot applications," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2554–2567, 2019.

[11] B. Forouzan, "Global edition," *Data Communications and Networking*, pp. 126–129, 2013.

[12] W. Stallings, "Data and computer communications. isbn-10: 1-29-201438-5," 2014.

[13] V. K. Sharma, L. P. Verma, and M. Kumar, "CL-ADSP: Cross-layer adaptive data scheduling policy in mobile ad-hoc networks," *Future Generation Computer Systems*, vol. 97, pp. 530–563, 2019.

[14] A. Ford, C. Raiciu, M. Handley, S. Barre, J. Iyengar, *et al.*, "Architectural guidelines for multipath tcp development," *IETF, Informational RFC*, vol. 6182, pp. 2070–1721, 2011.

[15] C. Raiciu, M. Handley, and D. Wischik, "Coupled congestion control for multipath transport protocols," tech. rep., IETF RFC 6356, Oct, 2011.

[16] M. Polese, F. Chiariotti, E. Bonetto, F. Rigotto, A. Zanella, and M. Zorzi, "A survey on recent advances in transport layer protocols," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3584–3608, 2019.

[17] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, *et al.*, "TCP extensions for multipath operation with multiple addresses," tech. rep., RFC 6824, 2013.

[18] C. Lee, J. Jung, and J.-M. Chung, "Deft: Multipath tcp for high speed low latency communications in 5g networks," *IEEE Transactions on Mobile Computing*, 2020.

[19] H. Liu, B. Zhang, H. T. Mouftah, X. Shen, and J. Ma, "Opportunistic routing for wireless ad hoc and sensor networks: Present and future directions," *IEEE Communications Magazine*, vol. 47, no. 12, pp. 103–109, 2009.

[20] H. Alves, T. Riihonen, and H. A. Suraweera, *Full-Duplex Communications for Future Wireless Networks*. Springer, 2020.

[21] N. Chakchouk, "A survey on opportunistic routing in wireless communication networks," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2214–2241, 2015.

[22] B. Fu, Y. Xiao, H. Deng, and H. Zeng, "A survey of cross-layer designs in wireless networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 110–126, 2013.

[23] C. Campolo, A. Molinaro, A. O. Berthet, and A. Vinel, "Full-duplex radios for vehicular communications," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 182–189, 2017.

[24] Y. Zhang, J. Tourrilhes, Z.-L. Zhang, and P. Sharma, "Improving sd-wan resilience: From vertical handoff to wan-aware mptcp," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 347–361, 2021.

[25] R. Stewart, M. Tuexen, and M. Proshin, "Stream control transmission protocol: Errata and issues in rfc 4960," 2019.

[26] J. R. Iyengar, P. D. Amer, and R. Stewart, "Concurrent multipath transfer using sctp multihoming over independent end-to-end paths," *IEEE/ACM Transactions on networking*, vol. 14, no. 5, pp. 951–964, 2006.

[27] Q.-V. Pham and W.-J. Hwang, "Network utility maximization-based congestion control over wireless networks: A survey and potential directives," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1173–1200, 2016.

[28] O. Salman, I. Elhajj, A. Chehab, and A. Kayssi, "Iot survey: An sdn and fog computing perspective," *Computer Networks*, vol. 143, p. 221–246, 2018.

[29] B. Bekkai, H. Bendjenna, and I. Kitouni, "Internet of things: A recent survey," *2021 International Conference on Recent Advances in Mathematics and Informatics (ICRAMI)*, 2021.

[30] R. Sharma, M. Mogha, S. Tanwar, and A. Rana, "Revolution of iot in healthcare during covid-19," *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2021.

[31] P. Suresh, J. V. Daniel, V. Parthasarathy, and R. H. Aswathy, "A state of the art review on the internet of things (iot) history, technology and fields of deployment," *2014 International Conference on Science Engineering and Management Research (ICSEMR)*, 2014.

[32] M. N. Shad, M. Maadani, and M. N. Moghadam, "Gapso-svm: An idss-based energy-aware clustering routing algorithm for iot perception layer," *Wireless Personal Communications*, 2021.

[33] G. Gkagkas, D. J. Vergados, and A. Michalas, "The advantage of the 5g network for enhancing the internet of things," *2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, 2021.

[34] J. Postel *et al.*, "Transmission control protocol," 1981.

[35] J. Postel *et al.*, "Internet protocol," 1981.

[36] J. Postel *et al.*, "User datagram protocol," 1980.

[37] C. Kale and T. Socolofsky, "Tcp/ip tutorials," 1991.

[38] S. Habib, J. Qadir, A. Ali, D. Habib, M. Li, and A. Sathiaseelan, "The past, present, and future of transport-layer multipath," *Journal of Network and Computer Applications*, vol. 75, pp. 236–258, 2016.

[39] M. Li, A. Lukyanenko, Z. Ou, A. Ylä-Jääski, S. Tarkoma, M. Coudron, and S. Secci, "Multipath transmission for the internet: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2887–2925, 2016.

[40] M.-T. Suer, C. Thein, H. Tchouankem, and L. Wolf, "Multi-connectivity as an enabler for reliable low latency communications—an overview," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 156–169, 2019.

[41] C. Xu, Z. Li, L. Zhong, H. Zhang, and G.-M. Muntean, “Cmt-nc: improving the concurrent multipath transfer performance using network coding in wireless networks,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1735–1751, 2015.

[42] N. Arianpoo, I. Aydin, and V. C. Leung, “Network coding as a performance booster for concurrent multi-path transfer of data in multi-hop wireless networks,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 4, pp. 1047–1058, 2016.

[43] J. Wu, B. Cheng, M. Wang, and J. Chen, “Energy-aware concurrent multipath transfer for real-time video streaming over heterogeneous wireless networks,” *IEEE Transactions on circuits and systems for video technology*, vol. 28, no. 8, pp. 2007–2023, 2017.

[44] J. Wu, B. Cheng, and M. Wang, “Improving multipath video transmission with raptor codes in heterogeneous wireless networks,” *IEEE Transactions on Multimedia*, vol. 20, no. 2, pp. 457–472, 2017.

[45] M. Klapež, C. A. Grazia, and M. Casoni, “Reducing cellular networks power consumption: The role of resource pooling and cooperation,” in *2017 14th International Conference on Telecommunications (ConTEL)*, pp. 107–114, 2017.

[46] T. Khan, W. Tian, and R. Buyya, “Machine learning (ml)-centric resource management in cloud computing: A review and future directions,” 2021.

[47] A. Ford, C. Raiciu, M. J. Handley, O. Bonaventure, and C. Paasch, “TCP Extensions for Multipath Operation with Multiple Addresses.” RFC 8684, Mar. 2020.

[48] W. Tang, Y. Fu, P. Dong, W. Yang, B. Yang, and N. Xiong, “A mptcp scheduler combined with congestion control for short flow delivery in signal transmission,” *IEEE Access*, vol. 7, pp. 116195–116206, 2019.

[49] O. Bonaventure and S. Seo, “Multipath tcp deployments,” *IETF Journal*, vol. 12, no. 2, pp. 24–27, 2016.

[50] M. S. Kim, I. H. Jung, K. M. Han, J. Y. Lee, and B. C. Kim, “Performance enhancement of mptcp having a bufferbloat path using retransmission of hol blocking packets,” in *2018 International Conference on Electronics, Information, and Communication (ICEIC)*, pp. 1–2, IEEE, 2018.

[51] Q. Peng, A. Walid, J. Hwang, and S. H. Low, “Multipath tcp: Analysis, design, and implementation,” *IEEE/ACM Transactions on networking*, vol. 24, no. 1, pp. 596–609, 2014.

[52] C. Xu, J. Zhao, and G.-M. Muntean, “Congestion control design for multipath transport protocols: A survey,” *IEEE communications surveys & tutorials*, vol. 18, no. 4, pp. 2948–2969, 2016.

[53] T. Henderson, S. Floyd, A. Gurtov, and Y. Nishida, “The newreno modification to tcp’s fast recovery algorithm,” *RFC6582*, 2012.

[54] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, "Design, implementation and evaluation of congestion control for multipath tcp.," in *NSDI*, vol. 11, pp. 8–8, 2011.

[55] O. Bonaventure, C. Paasch, G. Detal, *et al.*, "Use cases and operational experience with multipath tcp," *RFC 8041*, 2017.

[56] R. Khalili, N. Gast, M. Popovic, and J.-Y. Le Boudec, "Mptcp is not pareto-optimal: Performance issues and a possible solution," *IEEE/ACM Transactions On Networking*, vol. 21, no. 5, pp. 1651–1665, 2013.

[57] H. Li, Y. Wang, R. Sun, S. Guo, and H. Wang, "Delay-based congestion control for multipath tcp in heterogeneous wireless networks," in *2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW)*, pp. 1–6, IEEE, 2019.

[58] W. Wei, K. Xue, J. Han, D. S. Wei, and P. Hong, "Shared bottleneck-based congestion control and packet scheduling for multipath tcp," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 653–666, 2020.

[59] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "Bbr: congestion-based congestion control," *Communications of the ACM*, vol. 60, no. 2, pp. 58–66, 2017.

[60] W. Wei, K. Xue, J. Han, Y. Xing, D. S. Wei, and P. Hong, "Bbr-based congestion control and packet scheduling for bottleneck fairness considered multipath tcp in heterogeneous wireless networks," *IEEE Transactions on Vehicular Technology*, 2020.

[61] P. Dong, W. Yang, K. Xue, W. Tang, K. Gao, and J. Huang, "Tuning the aggressive slow-start behavior of mptcp for short flows," *IEEE Access*, vol. 7, pp. 6010–6024, 2019.

[62] S. Patel, Y. Shukla, N. Kumar, T. Sharma, and K. Singh, "A comparative performance analysis of tcp congestion control algorithms: Newreno, westwood, veno, bic, and cubic," in *2020 6th International Conference on Signal Processing and Communication (ICSC)*, pp. 23–28, IEEE, 2020.

[63] B. Y. Kimura, D. C. Lima, and A. A. Loureiro, "Packet scheduling in multipath tcp: Fundamentals, lessons, and opportunities," *IEEE Systems Journal*, 2020.

[64] J. Hwang and J. Yoo, "Packet scheduling for multipath tcp," in *2015 Seventh international conference on ubiquitous and future networks*, pp. 177–179, IEEE, 2015.

[65] K. W. Choi, Y. S. Cho, J. W. Lee, S. M. Cho, J. Choi, *et al.*, "Optimal load balancing scheduler for mptcp-based bandwidth aggregation in heterogeneous wireless environments," *Computer Communications*, vol. 112, pp. 116–130, 2017.

[66] W. Wei, K. Xue, J. Han, Y. Xing, D. S. L. Wei, and P. Hong, "BBR-based congestion control and packet scheduling for bottleneck fairness considered multipath TCP in heterogeneous wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 914–927, 2021.

[67] F. Silva, D. Bogusevschi, and G.-M. Muntean, "A mptcp-based rtt-aware packet delivery prioritisation algorithm in ar/vr scenarios," in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 95–100, IEEE, 2018.

[68] P. Hurtig, K.-J. Grinnemo, A. Brunstrom, S. Ferlin, Ö. Alay, and N. Kuhn, "Low-latency scheduling in mptcp," *IEEE/ACM Transactions on Networking*, vol. 27, no. 1, pp. 302–315, 2018.

[69] F. Yang, Q. Wang, and P. Amer, "Out-of-order transmission for in-order arrival scheduling policy for multipath tcp," in *Proceedings of the 28th International Conference on Advanced Information Networking and Applications Workshops*, pp. 749–752, 2014.

[70] P. Dong, W. Yang, W. Tang, J. Huang, H. Wang, Y. Pan, and J. Wang, "Reducing transport latency for short flows with multipath tcp," *Journal of Network and Computer Applications*, vol. 108, pp. 20–36, 2018.

[71] P. Dong, J. Xie, W. Tang, N. Xiong, H. Zhong, and A. V. Vasilakos, "Performance evaluation of multipath tcp scheduling algorithms," *IEEE Access*, vol. 7, pp. 29818–29825, 2019.

[72] Y. Thomas, M. Karaliopoulos, G. Xylomenos, and G. C. Polyzos, "Low latency friendliness for multipath tcp," *IEEE/ACM Transactions on Networking*, vol. 28, no. 1, pp. 248–261, 2020.

[73] A. Frommgen, T. Erbshäußer, A. Buchmann, T. Zimmermann, and K. Wehrle, "Remp tcp: Low latency multipath tcp," in *2016 IEEE International Conference on Communications (ICC)*, pp. 1–7, IEEE, 2016.

[74] B. Felix, I. Steuck, A. Santos, S. Secci, and M. Nogueira, "Redundant packet scheduling by uncorrelated paths in heterogeneous wireless networks," in *2018 IEEE Symposium on Computers and Communications (ISCC)*, pp. 00498–00503, IEEE, 2018.

[75] S. R. Pokhrel and J. Choi, "Low-delay scheduling for internet of vehicles: Load-balanced multipath communication with fec," *IEEE Transactions on Communications*, vol. 67, no. 12, pp. 8489–8501, 2019.

[76] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, "Fast tcp: motivation, architecture, algorithms, performance," *IEEE/ACM transactions on Networking*, vol. 14, no. 6, pp. 1246–1259, 2006.

[77] M. U. Daigavhane and M. D. Chawhan, "Congestion control algorithm for tcp in wireless network," in *2018 4th International Conference on Recent Advances in Information Technology (RAIT)*, pp. 1–4, IEEE, 2018.

[78] P. Dong, J. Wang, J. Huang, H. Wang, and G. Min, "Performance enhancement of multipath tcp for wireless communications with multiple radio interfaces," *IEEE Transactions on Communications*, vol. 64, no. 8, pp. 3456–3466, 2016.

[79] M. Prakash, A. Abdrabou, and W. Zhuang, "An experimental study on multipath tcp congestion control with heterogeneous radio access technologies," *IEEE Access*, vol. 7, pp. 25563–25574, 2019.

[80] S. R. Pokhrel and M. Mandjes, "Improving multipath tcp performance over wifi and cellular networks: An analytical approach," *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2562–2576, 2018.

[81] S. R. Pokhrel, M. Panda, and H. L. Vu, "Fair coexistence of regular and multipath tcp over wireless last-miles," *IEEE Transactions on Mobile Computing*, vol. 18, no. 3, pp. 574–587, 2018.

[82] C. Lee, S. Song, H. Cho, G. Lim, and J.-M. Chung, "Optimal multipath tcp offloading over 5g nr and lte networks," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 293–296, 2018.

[83] M. Prakash and A. Abdrabou, "Performance insights on using multipath tcp for wireless multihomed iot gateways," in *2020 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pp. 1–5, IEEE, 2020.

[84] J. R. Foerster, X. Costa-Perez, and R. V. Prasad, "Communications for iot: Connectivity and networking," *IEEE Internet of Things Magazine*, vol. 3, no. 1, pp. 6–7, 2020.

[85] K. Izumi, Y. Ito, and K. Noda, "Qos evaluation of mptcp over a multi-hop wireless network for iot devices," in *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*, pp. 395–396, IEEE, 2018.

[86] M. Polese, R. Jana, and M. Zorzi, "Tcp and mp-tcp in 5g mmwave networks," *IEEE Internet Computing*, vol. 21, no. 5, pp. 12–19, 2017.

[87] J. Wu, R. Tan, and M. Wang, "Energy-efficient multipath tcp for quality-guaranteed video over heterogeneous wireless networks," *IEEE Transactions on Multimedia*, vol. 21, no. 6, pp. 1593–1608, 2018.

[88] C. Xue, W. Li, L. Yu, J. Shang, X. Chen, and S. Lu, "Sero: A model-driven seamless roaming framework for wireless mesh network with multipath tcp," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1284–1296, 2018.

[89] F. P. Kelly, A. K. Maulloo, and D. K. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research society*, vol. 49, no. 3, pp. 237–252, 1998.

[90] S. H. Low and D. E. Lapsley, "Optimization flow control. i. basic algorithm and convergence," *IEEE/ACM Transactions on networking*, vol. 7, no. 6, pp. 861–874, 1999.

[91] P. L. Vo, T. A. Le, S. Lee, C. S. Hong, B. Kim, and H. Song, "Multi-path utility maximization and multi-path tcp design," *Journal of Parallel and Distributed Computing*, vol. 74, no. 1, pp. 1848–1857, 2014.

[92] Q.-V. Pham and W.-J. Hwang, "Network utility maximization in multipath lossy wireless networks," *International Journal of Communication Systems*, vol. 30, no. 5, p. e3094, 2017.

[93] H. Huang, Z. Sun, and X. Wang, "End-to-end tcp congestion control for mobile applications," *IEEE Access*, vol. 8, pp. 171628–171642, 2020.

[94] T. Wang, Z. Yao, B. Zhang, and C. Li, "Adaptive flow-level resource allocation for wireless mesh networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 10121–10133, 2019.

[95] A. Sinha and E. Modiano, "Network utility maximization with heterogeneous traffic flows," in *2018 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pp. 1–8, IEEE, 2018.

[96] W. Feng, Y. Xu, X. Xu, M. Zhao, and Y. Yao, "An energy-efficient and fast convergent resource allocation algorithm in distributed wireless ad hoc networks," *IEEE Access*, vol. 7, pp. 17133–17148, 2019.

[97] M. Aljubayri, Z. Yang, and M. Shikh-Bahaei, "Cross-layer multipath congestion control, routing and scheduling design in ad hoc wireless networks," *IET Communications*, vol. 15, no. 8, pp. 1096–1108, 2021.

[98] C. Xu, S. Jia, L. Zhong, and G.-M. Muntean, "Socially aware mobile peer-to-peer communications for community multimedia streaming services," *IEEE Communications Magazine*, vol. 53, no. 10, pp. 150–156, 2015.

[99] E. Setton, T. Yoo, X. Zhu, A. Goldsmith, and B. Girod, "Cross-layer design of ad hoc networks for real-time video streaming," *IEEE Wireless Communications*, vol. 12, no. 4, pp. 59–65, 2005.

[100] Y. Cao, Z. Liu, and Y. Yang, "A centralized scheduling algorithm based on multi-path routing in WiMAX mesh network," in *2006 International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–4, IEEE, 2006.

[101] S. H. Alsamhi, O. Ma, M. S. Ansari, and F. A. Almalki, "Survey on collaborative smart drones and internet of things for improving smartness of smart cities," *IEEE Access*, vol. 7, pp. 128125–128152, 2019.

[102] D. Hosahalli and K. G. Srinivas, "Cross-layer routing protocol for event-driven M2M communication in IoT-assisted smart city planning and management: CWSN-eSCPM," *IET Wireless Sensor Systems*, vol. 10, no. 1, pp. 1–12, 2019.

[103] N. Zorba, C. Skianis, and C. Verikoukis, *Cross Layer Designs in WLAN Systems*. Troubador Publishing, 2011.

[104] V. Arun and H. Balakrishnan, "COPA: Practical delay-based congestion control for the internet," in *15th USENIX Symposium on Networked Systems Design and Implementation ({NSDI} 18)*, pp. 329–342, 2018.

[105] P. Goyal, A. Narayan, F. Cangialosi, S. Narayana, M. Alizadeh, and H. Balakrishnan, "Elasticity detection: A building block for internet congestion control," *arXiv preprint arXiv:1802.08730*, 2018.

[106] Z. Wang, X. Zeng, X. Liu, M. Xu, Y. Wen, and L. Chen, "TCP congestion control algorithm for heterogeneous internet," *Journal of network and computer applications*, vol. 68, pp. 56–64, 2016.

[107] IEEE Std. 802-11, *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*. IEEE, 1997.

[108] M. Rajesh and J. Gnanasekar, "Congestion control scheme for heterogeneous wireless ad hoc networks using self-adjust hybrid model," *International Journal of Pure and Applied Mathematics*, vol. 116, no. 21, pp. 519–536, 2017.

[109] K. Akçapınar, Ö. Gürbüz, and T. Ünlüyurt, "Power allocation and routing for full-duplex multi hop wireless networks under full interference," *Ad Hoc Networks*, vol. 82, pp. 91–99, 2019.

[110] H.-W. Kim and A. Kachroo, "Low power routing and channel allocation of wireless video sensor networks using wireless link utilization," *Adhoc & Sensor Wireless Networks*, vol. 30, 2016.

[111] B. Hajek and G. Sasaki, "Link scheduling in polynomial time," *IEEE transactions on Information Theory*, vol. 34, no. 5, pp. 910–917, 1988.

[112] M. Kodialam and T. Nandagopal, "Characterizing achievable rates in multi-hop wireless networks: The joint routing and scheduling problem," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, pp. 42–54, 2003.

[113] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, pp. 1–13, 2006.

[114] G. Mishra, S. Shailendra, H. K. Rath, and A. Pal, "An analytical cross layer model for multipath TCP (MPTCP)," in *2019 European Conference on Networks and Communications (EuCNC)*, pp. 289–294, IEEE, 2019.

[115] H. Xiong, R. Li, A. Eryilmaz, and E. Ekici, "Delay-aware cross-layer design for network utility maximization in multi-hop networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 5, pp. 951–959, 2011.

[116] X. Ma, F. Li, J. Liu, and X. Liu, "Throughput–delay tradeoff for wireless multichannel multi-interface random networks," *Canadian Journal of Electrical and Computer Engineering*, vol. 38, no. 2, pp. 162–169, 2015.

[117] J. So and N. H. Vaidya, "Load-balancing routing in multichannel hybrid wireless networks with single network interface," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 1, pp. 342–348, 2007.

[118] Y. Cao, M. Xu, and X. Fu, "Delay-based congestion control for multipath TCP," in *2012 20th IEEE International Conference on Network Protocols (ICNP)*, pp. 1–10, IEEE, 2012.

[119] S. Barré, C. Paasch, and O. Bonaventure, "Multipath TCP: From theory to practice," in *International Conference on Research in Networking*, pp. 444–457, Springer, 2011.

[120] C. Paasch and O. Bonaventure, "Decoupled from IP, TCP is at last able to support multihomed hosts," *ACM Queue (Print): tomorrow's computing today*, vol. 12, no. 2, 2014.

[121] C. Xu, J. Zhao, and G.-M. Muntean, "Congestion control design for multipath transport protocols: A survey," *IEEE communications surveys & tutorials*, vol. 18, no. 4, pp. 2948–2969, 2016.

[122] M. Zhu, L. Wang, Z. Qin, N. Ding, J. Fang, T. Liu, and Q. Cui, "BE-LIA: Bandwidth estimate-based link increase algorithm for MPTCP," *IET Networks*, vol. 6, no. 5, pp. 94–101, 2017.

[123] H. Han, S. Shakkottai, C. V. Hollot, R. Srikant, and D. Towsley, "Multipath TCP: A joint congestion control and routing scheme to exploit path diversity in the internet," *IEEE/ACM Transactions on networking*, vol. 14, no. 6, pp. 1260–1271, 2006.

[124] Y.-C. Chen, Y.-s. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, and D. Towsley, "A measurement-based study of multipath TCP performance over wireless networks," in *Proceedings of the 2013 conference on Internet measurement conference*, pp. 455–468, 2013.

[125] R. Moskowitz, P. Nikander, P. Jokela, *et al.*, "Host identity protocol (HIP) architecture," tech. rep., RFC 4423), May 2006.

[126] Y. Cui, L. Wang, X. Wang, H. Wang, and Y. Wang, "FMTCP: A fountain code-based multipath transmission control protocol," *IEEE/ACM Transactions on Networking*, vol. 23, no. 2, pp. 465–478, 2014.

[127] X. Zhang, G. Pujolle, *et al.*, "A cross-layer approach to optimize the performance of concurrent multipath transfer in wireless transmission," in *2009 2nd IFIP Wireless Days (WD)*, pp. 1–5, IEEE, 2009.

[128] C. Xu, Z. Li, J. Li, H. Zhang, and G.-M. Muntean, "Cross-layer fairness-driven concurrent multipath video delivery over heterogeneous wireless networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 7, pp. 1175–1189, 2014.

[129] C. Paasch, S. Ferlin, O. Alay, and O. Bonaventure, "Experimental evaluation of multipath TCP schedulers," in *Proceedings of the 2014 ACM SIGCOMM workshop on Capacity sharing workshop*, pp. 27–32, 2014.

[130] A. Eryilmaz and R. Srikant, "Joint congestion control, routing, and MAC for stability and fairness in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1514–1524, 2006.

[131] L. Zhou, B. Zheng, B. Geller, A. Wei, S. Xu, and Y. Li, "Cross-layer rate control, medium access control and routing design in cooperative VANET," *Computer Communications*, vol. 31, no. 12, pp. 2870–2882, 2008.

[132] M. Ploumidis, N. Pappas, and A. Traganitis, "Flow allocation for maximum throughput and bounded delay on multiple disjoint paths for random access wireless multihop networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 720–733, 2016.

[133] M. Chiang, "Balancing transport and physical layers in wireless multihop networks: Jointly optimal congestion control and power control," *IEEE Journal on selected areas in communications*, vol. 23, no. 1, pp. 104–116, 2005.

[134] R. Laufer, T. Salonidis, H. Lundgren, and P. Le Guyadec, "A cross-layer backpressure architecture for wireless multihop networks," *IEEE/ACM transactions on networking*, vol. 22, no. 2, pp. 363–376, 2013.

[135] H. Luo, S. Lu, and V. Bharghavan, "A new model for packet scheduling in multihop wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 76–86, 2000.

[136] D. P. Bertsekas and R. G. Gallager, *Data networks, second edition*. Prentice Hall, 1993.

[137] L. Kleinrock, *Queueing systems*, vol. 2. Wiley, 1976.

[138] L. Kleinrock, *Queueing systems*, vol. 1. Wiley, 1975.

[139] H. W. Kuhn and A. W. Tucker, "Nonlinear programming," in *Traces and emergence of nonlinear programming*, pp. 247–258, Springer, 2014).

[140] S. Guo, C. Wang, and Y. Yang, "Joint mobile data gathering and energy provisioning in wireless rechargeable sensor networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 12, p. 2836–2852, 2014.

[141] J. Kleinberg and É. Tardos, "Introduction to algorithms," *Cornell University Course Notes, COMS*, vol. 482, 2003.

[142] R. Preis, "Linear time 1/2-approximation algorithm for maximum weighted matching in general graphs," in *Annual Symposium on Theoretical Aspects of Computer Science*, pp. 259–269, Springer, 1999.

[143] S. Djahel, F. Naït-abdesselam, and D. Turgut, "Characterizing the greedy behavior in wireless ad hoc networks," *Security and Communication Networks*, vol. 4, no. 3, pp. 284–298, 2011.

[144] M. N. Mejri and J. Ben-Othman, “Detecting greedy behavior by linear regression and watchdog in vehicular ad hoc networks,” in *2014 IEEE Global Communications Conference*, pp. 5032–5037, IEEE, 2014.

[145] A. Aloisio, V. Izzo, and S. Rampone, “VLSI implementation of greedy-based distributed routing schemes for ad hoc networks,” *Soft Computing*, vol. 11, no. 9, pp. 865–872, 2007.

[146] S.-J. Lee and M. Gerla, “Dynamic load-aware routing in ad hoc networks,” in *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No. 01CH37240)*, vol. 10, pp. 3206–3210, IEEE, 2001.

[147] D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” in *Mobile computing*, pp. 153–181, Springer, 1996.

[148] D. N. Kanellopoulos, “Recent progress on QoS scheduling for mobile ad hoc networks,” *Journal of Organizational and End User Computing (JOEUC)*, vol. 31, no. 3, pp. 37–66, 2019.

[149] D. E. Kirk, *Optimal Control Theory: An Introduction*. Courier Corporation, 2012.

[150] S. P. Meyn and R. L. Tweedie, “Stability of Markovian processes iii: Foster-Lyapunov criteria for continuous-time processes,” *Advances in Applied Probability*, pp. 518–548, 1993.

[151] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” in *29th IEEE Conference on Decision and Control*, pp. 2130–2132, IEEE, 1990.

[152] L. Mo, X. Cao, Y. Song, and A. Kritikakou, “Distributed node coordination for real-time energy-constrained control in wireless sensor and actuator networks,” *IEEE Internet of Things Journal*, vol. 5, no. 5, p. 4151–4163, 2018.

[153] M. Aljubayri, T. Peng, and M. Shikh-Bahaei, “Reduce delay of multipath tcp in iot networks,” *Wireless Networks*, vol. 27, no. 6, p. 4189–4198, 2021.

[154] S. Verma, Y. Kawamoto, Z. M. Fadlullah, H. Nishiyama, and N. Kato, “A survey on network methodologies for real-time analytics of massive IoT data and open research issues,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1457–1477, 2017.

[155] Y. Tian, K. Xu, and N. Ansari, “TCP in wireless environments: Problems and solutions,” *IEEE Communications Magazine*, vol. 43, no. 3, pp. S27–S32, 2005.

[156] E. Sy, T. Mueller, C. Burkert, H. Federrath, and M. Fischer, “Enhanced performance and privacy for TLS over TCP fast open,” *Proceedings on Privacy Enhancing Technologies*, vol. 2020, no. 2, pp. 271–287, 2020.

[157] V. Mittal, V. Jain, and M. P. Tahiliani, "Proportional rate reduction for NS-3 TCP," in *Proceedings of the 10th Workshop on ns-3*, pp. 9–15, 2018.

[158] R. G. Usach and M. Kühlewind, "Implementation and evaluation of coupled congestion control for multipath TCP," in *Meeting of the European Network of Universities and Companies in Information and Communication Engineering*, pp. 173–182, Springer, 2012.

[159] S.-Y. Park, C. Joo, Y. Park, and S. Bank, "Impact of traffic splitting on the delay performance of MPTCP," in *2014 IEEE International Conference on Communications (ICC)*, pp. 1204–1209, IEEE, 2014.

[160] V. G. Menon, "A comprehensive survey on opportunistic routing protocols for MANETs: Issues, challenges and future directions," 2019.

[161] M.-H. Lu, P. Steenkiste, and T. Chen, "Video transmission over wireless multihop networks using opportunistic routing," in *Packet Video 2007*, pp. 52–61, IEEE, 2007.

[162] K. Bagirathan and A. Palanisamy, "Opportunistic routing protocol based epo–bes in manet for optimal path selection," *Wireless Personal Communications*, 2021.

[163] A. Parsa and N. Moghim, "Qos-aware routing and traffic management in multi-flow opportunistic routing," *Computers and Electrical Engineering*, vol. 94, p. 107330, 2021.

[164] D. Ni, K. Xue, P. Hong, and S. Shen, "Fine-grained forward prediction based dynamic packet scheduling mechanism for multipath TCP in lossy networks," in *2014 23rd International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–7, IEEE, 2014.

[165] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, and M. 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)*, pp. 399–412, 2012.

[166] C. Xu, P. Wang, C. Xiong, X. Wei, and G.-M. Muntean, "Pipeline network coding-based multipath data transfer in heterogeneous wireless networks," *IEEE Transactions on Broadcasting*, vol. 63, no. 2, pp. 376–390, 2016.

[167] A. Garcia-Saavedra, M. Karzand, and D. J. Leith, "Low delay random linear coding and scheduling over multiple interfaces," *IEEE Transactions on Mobile Computing*, vol. 16, no. 11, pp. 3100–3114, 2017.

[168] K.-J. Grinnemo and A. Brunstrom, "A first study on using MPTCP to reduce latency for cloud based mobile applications," in *2015 IEEE Symposium on Computers and Communication (ISCC)*, pp. 64–69, IEEE, 2015.

[169] N. Kuhn, E. Lochin, A. Mifdaoui, G. Sarwar, O. Mehani, and R. Boreli, "DAPS: Intelligent delay-aware packet scheduling for multipath transport," in *2014 IEEE International Conference on Communications (ICC)*, pp. 1222–1227, IEEE, 2014.

[170] S. Ferlin, Ö. Alay, O. Mehani, and R. Boreli, "BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks," in *2016 IFIP Networking Conference (IFIP Networking) and Workshops*, pp. 431–439, IEEE, 2016.

[171] Y.-s. Lim, E. M. Nahum, D. Towsley, and R. J. Gibbens, "ECF: An MPTCP path scheduler to manage heterogeneous paths," in *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*, pp. 147–159, 2017.

[172] K. Noda and Y. Ito, "Proposal of novel MPTCP congestion control to suppress QoS fluctuation for WebQoE improvement," in *2018 IEEE 8th International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*, pp. 1–3, IEEE, 2018.

[173] Y. Liu, X. Qin, T. Zhu, X. Chen, and G. Wei, "Improve MPTCP with SDN: From the perspective of resource pooling," *Journal of Network and Computer Applications*, vol. 141, pp. 73–85, 2019.

[174] T. Lubna, I. Mahmud, and Y.-Z. Cho, "D-LIA: Dynamic congestion control algorithm for MPTCP," *ICT Express*, 2020.

[175] S.-Y. Park, C. Joo, Y. Park, and S. Bahk, "Minimizing application-level delay of multi-path TCP in wireless networks: A receiver-centric approach," in *2015 IEEE 23rd International Conference on Network Protocols (ICNP)*, pp. 245–255, IEEE, 2015.

[176] S. Ferlin, Ö. Alay, T. Dreibholz, D. A. Hayes, and M. Welzl, "Revisiting congestion control for multipath tcp with shared bottleneck detection," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, IEEE, 2016.

[177] S. Biswas and R. Morris, "ExOR: Opportunistic multi-hop routing for wireless networks," in *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 133–144, 2005.

[178] X. Pang, M. Liu, Z. Li, B. Gao, and X. Guo, "Geographic position based hopless opportunistic routing for uav networks," *Ad Hoc Networks*, vol. 120, p. 102560, 2021.

[179] P. Chithaluru, R. Tiwari, and K. Kumar, "Performance analysis of energy efficient opportunistic routing protocols in wireless sensor network," *International Journal of Sensors, Wireless Communications and Control*, vol. 11, no. 1, p. 24–41, 2021.

[180] C. Zhang, Y. Chen, and C. Li, "Tcp adaptation with network coding and opportunistic data forwarding in multi-hop wireless networks," *PeerJ Comput. Sci. 2:e89*, 2016.

[181] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (IoT): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[182] K. Vamsi, "Matlab-ns3," *GitHub*, 2020.

[183] 3GPP, "Architecture enhancements for 5G system (5GS) to support vehicle-to-everything (V2X) services," *3GPP TS 23.287, Rel-16, V-16.5.0*, 2020.

[184] F. J. Martin-Vega, M. C. Aguayo-Torres, G. Gomez, J. T. Entrambasaguas, and T. Q. Duong, "Key technologies, modeling approaches, and challenges for millimeter-wave vehicular communications," *IEEE Communications Magazine*, vol. 56, no. 10, pp. 28–35, 2018.

[185] C. Wang, Z. Li, X. G. Xia, J. Shi, J. Si, and Y. Zou, "Physical layer security enhancement using artificial noise in cellular vehicle-to-everything (C-V2X) networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15253–15268, 2020.

[186] J. Zang, V. Towhidlou, and M. Shikh-Bahaei, "Collision avoidance in V2X communication networks," in *2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW)*, pp. 1–6, 2019.

[187] R. Molina-Masegosa, M. Sepulcre, and J. Gozalvez, "Geo-based scheduling for C-V2X networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 8397–8407, 2019.

[188] G. Naik, B. Choudhury, and J. Park, "IEEE 802.11bd & 5G NR V2X: Evolution of radio access technologies for V2X communications," *IEEE Access*, vol. 7, pp. 70169–70184, 2019.

[189] J. Lee, Y. Im, and J. Lee, "Modeling MPTCP performance," *IEEE Communications Letters*, vol. 23, no. 4, pp. 616–619, 2019.

[190] S. Shin, D. Han, H. Cho, J. Chung, I. Hwang, and D. Ok, "TCP and MPTCP retransmission timeout control for networks supporting WLANs," *IEEE Communications Letters*, vol. 20, no. 5, pp. 994–997, 2016.

[191] J. Zang and M. Shikh-Bahaei, "An adaptive full-duplex deep reinforcement learning-based design for 5g-v2x mode 4 vanets," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2021.

[192] J. Zang, V. Towhidlou, and M. Shikh-Bahaei, "A priority-based cross-layer design for future VANETs through full-duplex technology," *IEEE Transactions on Vehicular Technology*, vol. 69, pp. 7531–7544, May. 2020.

[193] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, pp. 3–15, Jan. 2011.

[194] X. Gu, B. Leng, L. Zhang, J. Miao, and L. Zhang, "A stochastic geometry approach to model and analyze future vehicular communication networks," *IEEE Access*, vol. 8, pp. 14500–14512, Jan. 2020.

[195] 3GPP, "Overall description of radio access network (RAN) aspects for vehicle-to-everything (V2X) based on LTE and NR," *3GPP TR 37.985, Rel-16, V-1.0.0*, Nov. 2019.

[196] D. Kim, H. Lee, and D. Hong, "A survey of in-band full-duplex transmission: From the perspective of PHY and MAC layers," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2017–2046, 2015.

[197] M. Chung, M. S. Sim, J. Kim, D. K. Kim, and C. Chae, "Prototyping real-time full duplex radios," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 56–63, 2015.