Title
Traffic Offloading in Heterogeneous Mobile Networks: the Centrally Directed and User Collaborated Options

Author
Du, Pengyuan

Publication Date
2018

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Traffic Offloading in Heterogeneous Mobile Networks: the Centrally Directed and User Collaborated Options

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Pengyuan Du

2018

ABSTRACT OF THE DISSERTATION

Traffic Offloading in Heterogeneous Mobile Networks: the Centrally Directed and User Collaborated Options

by

Pengyuan Du
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2018
Professor Mario Gerla, Co-Chair
Professor Songwu Lu, Co-Chair

In recent years, we have witnessed the growth of smart devices and an ever-increasing number of users in the cellular system. Mobile applications such as video streaming and online gaming become popular ways of entertainment but also consume excessive network bandwidth. To meet the high bandwidth demand, the vision of future cellular system is to form a converged infrastructure network by integrating alternative radio access technologies such as WiFi, airborne Unmanned Aerial Vehicle (UAV) and satellites into the current cellular system. The converged infrastructure network increases total bandwidth capacity so that cellular traffic can be effectively offloaded to the alternative networks. On the users' side, as smart phones today are commonly equipped with multiple network interfaces, users can toggle between different connections in the heterogeneous network environment, or even concurrently use all the available networks and aggregate their bandwidth.

Nevertheless, mobile users may experience intermittent connectivity or fluctuated channel quality when using the infrastructure networks due to the nature of mobile, wireless communications. In this case, it is beneficial to establish an additional device-to-device (D2D) connection to nearby devices opportunistically and use them as temporary network access points. For delay-tolerant data such as emails, short messages, whose delivery is time insensitive, or for popular contents such as video and music chunks, they can even be deliv-

ered or shared purely using the D2D transmissions. Therefore, this kind of user collaborated network has the potential to offload cellular traffic as well, as it is ideal for the transmission of delay-tolerant data, and it also saves the redundant downloads of popular contents.

This dissertation discusses the methods to realize traffic offloading in a heterogeneous mobile network environment consisting two different but related mobile network scenarios: the infrastructure and opportunistic networks. In the infrastructure network, we developed a centrally directed multipath transmission system to solve the traffic offloading problem. Our system applies the Software-Defined Networking (SDN) paradigm to centralize network control and management. Moreover, it adopts a newly emerged transport layer protocol Multipath TCP (MPTCP) that is run on mobile devices to enable multipath data transmission. MPTCP initiates parallel TCP connections using multiple network interfaces. An SDN controller is designed to manage the MPTCP connections and regulate their bandwidth usage from each connected network. Therefore, when seeing network congestion in the cellular systems, the controller will direct MPTCP traffic to go to the other available networks such as WiFi and satellites. Additionally, our designed system has the capability of real-time topology monitoring and dynamic network reconfigurations leveraging its centralized design. The proposed system is proved to be efficient in both cellular-WiFi and tactical satellite communication networks. In the opportunistic network, while technologies such as MPTCP and WiFi Direct allow mobile users to conveniently collaborate, it is yet a question whether the network itself encourages user participation and whether it prevents non-cooperative behaviors. To this end, we study the cooperative behaviors of mobile users in opportunistic network using the Evolutionary Game Theory (EGT). Through both mathematical proof and numerical simulations, we discover that cooperation is sustainable in the opportunistic network. The emergence of cooperation in the opportunistic network confirms its potential for offloading cellular traffic.

The dissertation of Pengyuan Du is approved.

Danijela Cabric

Demetri Terzopoulos

Douglas S. Parker

Songwu Lu, Committee Co-Chair

Mario Gerla, Committee Co-Chair

University of California, Los Angeles

2018

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I would like to express my heartfelt gratitude to my advisor, Professor Mario Gerla, for his guidance throughout my PhD study, and also for all the freedom he gave me to pursue my research.

Next, I would like to thank Dr. Jae H. Kim, Dr. Ceilidh Hoffmann and Dr. Adam J. Brown from Boeing Research & Technology for providing me with Graduate Student Research (GSR) opportunities to work on the Software Defined Naval Battlefield Network (SD-NBN) project. Thanks also to the UCLA Computer Science Department for providing me with many years of Teaching Assistantship.

Thanks to everyone in the UCLA Network Research Lab (NRL) for their help on my research - Qi Zhao, Vince Rabsatt, Seunghyun Yoo, Noor Abani, Jorge Mena, Josh Joy, Ruolin Fan, Seungbae Kim, Tuan Le, You Lu, Xiao Li, Yu-Ting Yu, and the visitors, Prof. Torsten Braun of University of Bern, Prof. Feng Zhou of Harbin Engineering University, Prof. Susumu Ishihara of Shizuoka University, and Li Wei of Michigan Tech University.

I would also like to thank my girlfriend Lujin Luo for her sacrifice and spiritual support during the 5 years of long-distance relationship.

Last but not least, thanks to my parents, Yiping Du and Wenming Wang, for their meticulous care and encouragement. I would not have achieved this without their support.

VITA

2013	B.S. in Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China
2016	M.S. in Computer Science, University of California, Los Angeles, Los Angeles, California
2014-2018	Teaching Assistant/Graduate Student Researcher, Department of Computer Science, University of California, Los Angeles, Los Angeles, California

PUBLICATIONS

Pengyuan Du, Seunghyun Yoo, Qi Zhao, Muhao Chen, Mario Gerla. "Towards Opportunistic Resource Sharing in Mobile Social Networks: an Evolutionary Game Theoretic Approach." International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc Poster), 2018.

Qi Zhao, Pengyuan Du, Jorge Mena, Mario Gerla "A Multi-path TCP Solution for Software-Defined Military Heterogeneous Network." Wireless Communications and Networking Conference (WCNC Poster), 2018.

Pengyuan Du, and Mario Gerla. "An Evolutionary Multi-player Game Model for Two-hop Routing in Delay Tolerant Networks." International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 2017.

Pengyuan Du, Fan Pang, Torsten Braun, Mario Gerla, Ceilidh Hoffmann, and Jae H. Kim. "Traffic Optimization in Software Defined Naval Network for Satellite Communications." Military Communications Conference (MILCOM), 2017.

Pengyuan Du, and Mario Gerla. "Promotion of Cooperation in Public Goods Game by Socialized Speed-Restricted Movement." Wireless On-demand Network Systems and Services (WONS), 2017.

Pengyuan Du, Sobhan Nazari, Jorge Mena, Ruolin Fan, Mario Gerla, and Riten Gupta. "Multipath TCP in SDN-enabled LEO Satellite Networks." Military Communications Conference (MILCOM), 2016.

Sobhan Nazari, Pengyuan Du, Mario Gerla, Ceilidh Hoffmann, Jae H. Kim, and Antonio Capone. "Software Defined Naval Network for Satellite Communications (SDN-SAT)." Military Communications Conference (MILCOM), 2016.

Pengyuan Du, Mario Gerla, and Xinbing Wang. "Throughput and Delay Scaling of Cognitive Radio Networks with Heterogeneous Mobile Users." International Conference on Computer Communication and Networks (ICCCN), 2015.

Pengyuan Du, Xiao Li, You Lu, and Mario Gerla. "Multipath TCP over LEO Satellite Networks." International Wireless Communications and Mobile Computing Conference (IWCMC), 2015.

CHAPTER 1

Introduction

The rapid development of wireless and mobile technologies offers people levels of mobility and services never available before. The current cellular technologies (4G/LTE) enable ubiquitous communication anytime and anywhere, and have become the ground to provide connectivity through IP core network. They are designed to optimize the network performance, improve cost efficiency and facilitate the uptake of mass market IP-based services.

Nevertheless, the growing demand and the diverse patterns of mobile traffic place an increasing strain on cellular networks. A report by Cisco [Cis17] in 2017 shows that the global mobile data traffic grew 74% in 2016 and more than half is video embedded. It is predicted that mobile data traffic will increase nearly sevenfold by 2021 and 3/4 will be video embedded. The growth of mobile data usage is straining the capacity of airwaves and is likely to outstrip spectrum capacity in the near future. This trend is known as the "spectrum crunch" – there just isn't sufficient bandwidth capacity to satisfy the demands.

To cope with the large volume of traffic, future mobile system such as 5G should exploit the capacity of a converged heterogeneous communication networks, e.g. wireless access infrastructure in both licensed and unlicensed wireless spectrum, Wireless Local Area Network (WLAN), and satellite networks. Traffic that is originally carried by the cellular system now can be shifted to the other access networks to prevent network congestion. Additionally, with the increasing popularity of smart mobile devices, non-conventional techniques such as device-to-device (D2D) or mobile ad hoc communications may further alleviate the stress of cellular systems by providing the extra transmission opportunities based on mobile device collaborations. For example, users may share popular contents with nearby devices in order to save their redundant downloads using the cellular network.

In this dissertation, we define traffic offloading as the use of complementary wireless technologies (i.e. WiFi, satellites, or D2D communications) to transfer data originally going through the cellular network. It not only prevents the "spectrum crunch", but also benefits the mobile users by improving network coverage and enabling mobile collaboration capabilities. Depending on the scenario and the type of technology used for traffic offloading, this dissertation naturally breaks into two parts: (1) infrastructure-based offloading, where cellular traffic is offloaded to the alternative infrastructure networks such as WiFi and satellite systems; (2) opportunistic offloading, where mobile users transmit or share data using D2D communication links opportunistically. The two traffic offloading approaches complement each other but serve for the same purpose: to alleviate cellular system congestion and improve mobile service quality by forming a heterogeneous mobile network with increasing network access and data delivery methods.

1.1 An Overview of Traffic Offloading in Heterogeneous Mobile Networks

1.1.1 Infrastructure-based Traffic Offloading

Offloading traffic from cellular to other alternative radio access networks (RAN) attracts a lot of research attention as it is currently the most cost-effective solution that can be quickly deployed leveraging the existing technologies. Some promising technologies are WiFi and satellite networks. Major cellular providers such as AT&T, Verizon, and China Mobile have started to install WiFi networks in their cellular systems for additional capacity, since WiFi access point (AP) is cheap to deploy at scale. Although WiFi has shorter range than cellular base stations, it usually provides better throughput and shorter latency. However, due to the limited communication range, mobility or handover management is necessary in order to guarantee service continuity when the mobile users cross the boundary of WiFi networks. Satellite networks on the other hand have the advantage of global coverage. Satellite communication technology is also advancing rapidly to increase its link capacity.

Therefore it can play an important role in offloading the traffic of users who are not covered by the WiFi networks, e.g. in vehicular or tactical scenarios.

Infrastructure-based offloading schemes can be classified into the following categories:

Network selection This scheme assumes mobile applications use a single RAN connection for data transmission. Thus the key to the network selection problem is to devise an algorithm that optimizes the choice of access networks (e.g. LTE, WiFi, or satellite) given user requirements and network constraints. The algorithm can run on mobile devices [DSB14] and make the decision based on a local view of the network status. This approach however has the shortcoming of being greedy when choosing the networks, e.g., all the mobile users pick the same WiFi network, which may lead to network congestion. To mitigate the suboptimal network selection, in [CIH16, ACO13], the decisions are made by a centralized network intelligence based on a global view of the instantaneous network load, channel quality, or the priority of user data flows, leveraging the Software-Defined Networking (SDN) paradigm. However, the assumption of using only a single connection makes it challenging to handle vertical handover events, as the ongoing transmission will be severely disrupted during the network interface switching. The authors in [MVS14] proposed a network switching scheme that smoothly transfers the ongoing transmission to the alternative interface, but it only works for a limited set of applications.

Network aggregation Different from network selection, network aggregation schemes aim to concurrently use multiple RANs in order to enjoy the aggregated bandwidth and the flexibility in network switching. Existing network aggregation schemes include the 3rd Generation Partnership Project (3GPP) [LLR18] and the Multipath-TCP (MPTCP) [FRH11b] protocol. 3GPP is continuously releasing proposals to define the architectures for efficient and low cost LTE and WiFi aggregation. The Options are Licensed Assisted Access (LAA), LTE-WLAN Aggregation (LWA), and LTE-WLAN radio level integration with IPsec tunnel (LWIP), etc. These proposals are integrating LTE and WiFi networks at the IP and link layers, which requires significant modifications to the existing cellular and WiFi networks.

On the other hand, MPTCP serves as a transport layer solution to the multi-RAN integration, which is probably the most economic network aggregation scheme since no changes are made to the existing infrastructure.

Both network selection and network aggregation concern about how to optimally utilize the additional bandwidth brought by the alternative RANs. However, network aggregation in general provides more benefits than the network selection schemes. By concurrently using multiple RAN connections, mobile devices can potentially obtain higher throughput. Using multiple connections also guarantees more robustness against handover, channel quality fluctuations, or network failures. For example, LWA proposed the bearer split function in order to offload Packet Data Convergence Protocol (PDCP) layer packets to the WiFi network for seamless link layer integration. The diverged PDCP packets will be reordered at mobile device once they are received from the LTE and WiFi interfaces. MPTCP-based integration moves the above functionality to the transport layer. Both LWA and MPTCP achieve higher throughput and smoother transmission during handver compared to conventional single-path based offloading schemes.

1.1.2 Opportunistic Traffic Offloading

Opportunistic traffic offloading relies on the D2D connections between mobile users to transmit data that would otherwise be sent to the cellular system. Compared to infrastructure-based offloading, it does not follow the client-server communication model. Mobile users collaborate to relay each other's packets, or directly share contents to other users who have the same interests. Since it completely depends on mobile users to carry and propagate data upon encountering other users, both user movement and the duration of users contacts affect the volume of traffic that can be offloaded by the opportunistic network. Opportunistic offloading is ideal for propagating delay-tolerant type of data such as news, email, popular video chunks or pod casts, which do not have a strict deadline to be delivered.

Opportunistic traffic offloading has raised great interests in the research community. Several works have studied the feasibility of opportunistic offloading by investigating user

mobility and contact patterns revealed in real-world mobility traces [WCH14, HHK12]. Because users tend to contact a small group of people very frequently, and rarely contact random people, which resembles human behaviors in social networks, the opportunistic network formed by mobile users is also called Mobile Social Networks (MSN). The frequent and socialized contacts in MSN provide opportunities for data offloading. To initiate opportunistic offloading, first, a set of "seeding" nodes are selected to download some popular contents from the cellular networks. Next, they distribute the contents and delegate the forwarding power to nearby mobile users who have the same interests. This process continues until the content is propagated to every node in the network who needs the contents. The result of [WCH14] shows that the 20% carefully selected "seeding" nodes are able to distribute popular contents to at least 80% of interested users. This content distribution protocol has the potential to save at least 75% of the bandwidth for downloading the same contents from the cellular network. There are also applications that are completely infrastructure-less, e.g., the opportunistic packet forwarding in Delay-Tolerant Networks (DTN). In [MW11], an infrastructure-less packet forwarding scheme is developed. For every source user who wants to send a message to some destination user, it first attempts to deliver the message using DTN, i.e. it relies on the intermediate users to store, carry and forward the message when they encounter the destination. Only if the probability of delivering the message before the deadline is low, the source node will transmit it using the cellular network for immediate delivery. It has been shown that more than 50% of the packets can be successfully offloaded through DTN.

Opportunistic traffic offloading has the potential to save the tremendous bandwidth currently used to transmit delay-tolerant type of data in the cellular network. However, this is assuming everyone is cooperative. Being cooperative means users must agree to share the contents downloaded using their own cellular connections, or to act as relay and forward packets received from other users. In fact, it is apparent that volume of traffic that can be offloaded by MSN is positively correlated with the fraction of cooperative users. The offloading results promised by the previous content distribution and DTN applications are only achievable if the cooperation rate is 100%. Although a lot of research efforts have been

spent on developing opportunistic traffic offloading schemes, users' cooperative behaviors so far have not been well studied. In this dissertation, we are particularly interested in user cooperative behaviors when they are involved in different opportunistic offloading protocols, as it directly affects the actual offloading performance that we can deliver in MSN. Although many incentive schemes have been proposed to stimulate user cooperation, the condition for cooperation to emerge in opportunistic traffic offloading is not well understood.

To investigate the emergence of user cooperation in opportunistic traffic offloading, we adopt the Evolutionary Game Theory (EGT) approach, which is a branch of game theory that models the evolutionary dynamics of user behaviors when they participate in group interactions with neighbors (co-players) and evolve their behaviors according to an evolutionary algorithm mimicking the natural selection. The EGT framework is mathematically tractable on static networks and can be conveniently simulated over dynamic networks. In order to simulate MSN, it is important to design mobility model that reproduces realistic contact patterns and statistics as observed in real-world MSN mobility traces.

1.2 Contributions

In this dissertation, we develop a software-defined and multipath-based network aggregation system for infrastructure-based traffic offloading, and we investigate the cooperation issues in opportunistic traffic offloading following the EGT theoretic approach. Specifically, we make the following contributions:

- We propose a network aggregation framework using MPTCP and SDN technologies for infrastructure-based traffic offloading. MPTCP is run on mobile devices to create TCP subflows over multiple network connections. An SDN controller is designed to direct MPTCP subflows for properly using network bandwidth. We formulate traffic offloading as a network optimization problem. Therefore, the SDN controller configures MPTCP flow rates in each RAN to optimize the network objective of load balancing and fairness. The devised controller is also responsive to network status change, and the centralized design facilitates network resource management and policy making.

- We implement a prototype of the infrastructure-based traffic offloading system using network emulator Mininet. Emulation results prove its efficiency in both satellite communication and cellular-WiFi networks.

- Motivated by the correlation between opportunistic traffic offloading and user cooperation rate, we investigate the emergence of cooperation in the opportunistic network using EGT theory. First, the evolutionary game models are devised to capture the interactions and payoff functions relevant to the opportunistic offloading protocols, i.e. DTN packet forwarding and content distribution network. Then we propose a realistic mobility model to simulate user movements in MSN. Through both mathematical proof and simulation verification, we discover the condition for cooperation to emerge and verify the achievable traffic offloading performance in the opportunistic network.

1.3 Roadmap of the Dissertation

The rest of the dissertation is organized as follows: Chapter 2 presents the background of heterogeneous mobile networks, and reviews previous research efforts related to infrastructure-based and opportunistic traffic offloading. Chapter 3 demonstrates the formation of our centrally directed multipath transmission system using MPTCP and SDN in satellite communication networks. In Chapter 4, we discuss the design and implementation of a infrastructure-based traffic offloading system named Software Defined MultiPath end-to-end Traffic OPtimization (`SD-MPTOP`). Chapter 5 presents our study on the emergence of cooperation in the opportunistic network and its impact on the performance of opportunistic traffic offloading. Finally, we conclude this dissertation in Chapter 6 and outline future work.

CHAPTER 2

Background and Related Work

2.1 Architecture of Heterogeneous Mobile Networks

Driven by the increasing demand of mobile bandwidth and more user-centric services, there are a number of key changes in the technology and structure of future heterogeneous mobile networks.

The first one is an increase in the number of WiFi hotspots and ubiquitous WiFi enabled devices. WiFi supports bandwidth up to 54Mbps at extremely low cost. Increasing the number of WiFi hotspots will encourage people to connect to the Internet via WiFi which is a relief to cellular system. Also all mobile devices except for mobile phones, will be WiFi enabled and have limited cellular connectivity.

The other change is the rise of small cells. The idea is to create a denser network and increase the area spectral efficiency by orders of magnitude. It solves the coverage, capacity and spectrum issues in the current macro-cell system. Small cells are available for a wide range of air interfaces including GSM, CDMA2000, LTE, etc. It further encompasses femto-cells, pico-cells and micro-cells depending on the coverage. These small cells offload traffic in 3G networks and are self-organizing heterogeneous networks in 4G/LTE networks.

The third change is introducing satellites to the terrestrial mobile system. Satellite systems have served mankind in a heterogeneous set of ways since its creation. With the feature of global coverage, they have been providing services such as TV broadcasting, digital messaging, GPS navigation and worldwide telecommunications for decades [Sax14]. Now 5G opens up new markets for satellite-based services by pushing the convergence between satellite and terrestrial systems. This is appealing because satellite can play an important

role in the following areas:

- **Coverage**

  Satellites can provide the wide coverage to complement and extend the terrestrial cells. This will bring ubiquitous coverage to reality.

- **Resilience**

  Satellite network can take over and keep alive the network in case of disasters.

- **Capacity**

  The satellite technology is advancing to take capacities from 100 Gbps to over a Terabit/s. In addition, high level frequency reuse and spotbeam technology will further increase the capacity.

Hybrid networks, in which satellite complements terrestrial cellular communications rather than compete, is in line with the key targets of 5G. Various use cases can be envisioned, e.g. high throughput connectivity in remote areas where terrestrial infrastructure is not available, cellular backhaul, and on-demand traffic offloading when the capacity of cellular system is exceeded. Although not the topic of this dissertation, the hybrid satellite-terrestrial network itself is an important issue to study.

A last change would be the inclusion of opportunistic communication techniques. Opportunistic networks enable D2D communications between mobile devices to route messages or share contents. With the increasing density of mobile devices and the evolution of wireless technologies [TKD17, CDM13], opportunistic communications have growing potential to carry more traffic and decongest the cellular networks. They also bring benefits to the mobile users. As mobile applications are nowadays more content-centric, users can discover and retrieve popular contents from nearby peers using faster and more stable opportunistic connections with zero cost.

2.2 Infrastructure-based Traffic Offloading

2.2.1 Software-Defined Networking (SDN)

SDN is an emerging concept that promises to overcome the bundling between control plane (that decides how to handle network traffic) and data plane (that forwards traffic according to the decisions made by the control plane) in traditional ossified [MAB08] IP networks by breaking the so-called vertical integration and removing the network control functions from the underlying routers and switches, thereby promoting the logical centralization of network control and introducing the network programmability [KRV15]. The Open Networking Foundation (ONF), a non-profit user driven organization, is dedicated to the promotion and the adoption of SDN through open standards development (e.g., OpenFlow).

OpenFlow is an open interface for remotely controlling the forwarding tables in network switches, routers, and access points. Upon this low-level primitive, researchers can build networks with new high-level properties. For example, OpenFlow enables more secure default-off networks, wireless networks with smooth handoffs, scalable data center networks, host mobility, more energy-efficient networks and new wide-area networks to name a few.

Centralized control, network programmability and the decoupling between control and data plane are the main principles that simplify network management, allowing administrators to dynamically manage network devices through abstraction layers, enabling innovations and changing the limitations of current infrastructures. By separating the control and data planes, network routers and switches become simple forwarding devices. The control logic is implemented in a logically centralized controller, simplifying policy enforcement and network (re)configuration, evolution and scalability [KF13]. Forwarding decisions made by the data plane are flow-based, instead of destination-based. A flow is defined as a set of packet field values acting as a match (filter) criterion and a set of actions (instructions). The flow abstraction allows the classification of network traffic based on predefined matching rules, thus unifying the behavior of different types of network devices. Flow programming enables unprecedented flexibility, limited only to the capabilities of the implemented flow tables inside the forwarding devices [MAB08].

When a flow is generated from an entry that was previously installed in the switch, no control traffic is generated. The packets immediately pass through the switch and a much lower delivery time is experienced for these packets of the flow. When the switch receives a packet from an unknown flow, it forwards the packet to the controller. The controller inspects the packet, decides what the appropriate action is and instructs the switch to execute this action for all packets of this particular flow.

2.2.2 Network Selection

Network selection provides intelligent interface switching services to mobile users in heterogeneous mobile networks where multiple RANs coexist. As different RANs have different policies and network characteristics, network selection is essentially the choice of a single RAN that trades off a set of requirements, e.g., throughput, latency, fairness, QoS, etc. In [DSB14], the authors proposed a transport layer network selection protocol called *Delphi*, to make the best network choice taking both user objectives and the social utility into considerations. [TCK13] on the other hand considered the backhaul capacity of WiFi networks and optimizes user fairness in the bandwidth usage. [CIH16] assumed the deployment of SDN in heterogeneous networks, and optimizes network selection based on the traffic type and the network conditions seen by the SDN controller. It shows that the SDN-based scheme migrates traffic transparently and immediately without disrupting the ongoing TCP connections. Other aspects of network selection problems studied in the context of heterogeneous networks include economy issues [LGC14], energy consumption, [NSD13], and system trade-off considering throughput efficiency, mobility management, and cost of deployment [MVS14].

2.2.3 Network Aggregation

2.2.3.1 The 3rd Generation Partnership Project (3GPP)

In response to the need for a converged cellular and WiFi heterogeneous network, 3GPP has defined various methods of LTE-WiFi integration ranging from LTE in Unlicensed/Licensed

Assisted Access (LTE-U/LAA), LTE-WiFi Link Aggregation (LWA), and LTE-WiFi integration with IPsec tunnel (LWIP) [Nug16]. LTE-U/LAA simply extends LTE carrier aggregation (CA) used in licensed bands to unlicensed bands. LTE-U/LAA is the same as LTE CA in that it uses all channels for LTE, but different from LTE CA in that it uses unlicensed $5GHz$. LTE-U/LAA requires installation of new $5GHz$ hardware. It should also resolve the possible issues raised by its coexistence with WiFi networks in the unlicensed band. Differently, LWA uses existing WLAN as the unlicensed band RAN. It is proposed to integrate LTE and WiFi at the link layer by splitting the LTE bearer into the two networks. PDCP packets are encapsulated into WiFi frames so that they are compatible with WiFi protocol. LWIP on the other hand integrates LTE and WiFi on the network layer based on IPsec tunnel. Both LWA and LWIP are more cost effective than LTE-U/LAA, since they do not require major changes to the existing unlicensed band protocols. User devices however require software/hardware upgrades in order to combine the traffic delivered from LTE and WiFi interfaces.

2.2.3.2 MPTCP

MPTCP [FRH11b] is a transport layer protocol that initiates multiple TCP flows, while presenting only one connection to the application layer. The multiple flows are called MPTCP subflows. Each MPTCP subflow is initiated using physically separated interface, i.e., for a mobile terminal that has interface to WiFi, LTE, and satellite networks, MPTCP is enabled to create multiple subflows that go through each of the networks via separate paths. Moreover, MPTCP maintains the subflows in an intelligent way rather than treating them as independent TCP flows. The latter case would be problematic because there is a lack of method to split the traffic, maintain sequence number and react to congestion of a particular flow. Suppose one of the connections is congested, the packets assigned to this flow would be delayed and TCP window is shrunk, while the rest packets are successfully delivered by other connections. Therefore the delivered packets have to wait for the lost packets to be retransmitted before they are sent to the upper layer. Even worse, there is no mechanism detecting and shutting down the slow connection. As a consequence the fast connections are

dragged down by the slow one until the congestion goes away.

MPTCP is designed to handle all those problems. The MPTCP *path manager* is in charge of establishing subflows using all available interfaces. Once established, packets are sent through different subflows. Each packet carries two sequence numbers: global and subflow sequence number. On MPTCP layer, global sequence numbers are maintained for each packet so that they can be reassembled at the destination. The subflow sequence is used only within the subflow through which the packet is delivered. To monitor and react to congestions, MPTCP runs the so called *coupled congestion control*. This coupled congestion increases the congestion window of each subflow according to its measured Round Trip Time (RTT). The goal is two-fold. First, MPTCP tries to exploit the bandwidth of all subflows, therefore the congestion window of subflow is increased addtively. Secondly, MPTCP tries to be fair to the competing flows, so that the increment of window size is upper bounded by the increment of a single path TCP using the same path. In the case of congestion, the coupled congestion will decrease the window. At the same time, MPTCP *packet scheduler* will avoid scheduling packets to this flow so that it will not slow down the transmission. Instead, packets will be scheduled to connections with smaller RTT and no congestion. A good amount of work has been devoted to MPTCP's coupled congestion control [PWL13][SXK13][KGP12][WRG11]. As a commonly accepted version, [WRG11] regulates traffic using a Linked Increase Algorithm (LIA). In congestion avoidance, assume a sub-flow i has an RTT of r_i, and a congestion wind of w_i. $\hat{w}$ denotes its value at equilibrium. Then upon reception of each Acknowledgement (ACK) on sub-flow i, increase the window by $\min(\frac{\alpha}{w_{tot}}, \frac{1}{w_i})$, where w_{tot} is the total congestion window across all sub-flows. Upon each loss on sub-flow i, reduce the window by $\frac{1}{w_i}$. The parameter α is determined based on the following formula:

$$\alpha = \hat{w}_{total} \frac{\max_i \frac{\hat{w}_i}{r_i^2}}{(\sum_i \frac{\hat{w}_i}{r_i^2})^2},$$

in order to satisfy the goal of fairness, load balancing and compensate different RTTs across the sub-flows.

The sophisticated and careful design of MPTCP makes sure that all the potential band-

width is exploited, but also avoids being greedy and unfair to other flows. Another advantage of MPTCP is it provides smooth handover in a highly mobile scenario. For example, a mobile phone has connection to both WiFi and LTE can experience handover when it moves out of the coverage of WiFi access point. Using single path TCP, the connection will be dropped and reconnected using LTE interface, so the application will be interrupted. However using MPTCP the subflow established using WiFi interface will be closed and all the traffic will be shifted to the LTE subflow gracefully, without experiencing interruption.

2.3 Opportunistic Traffic Offloading

Opportunistic traffic offloading is complementary to the infrastructure networks. It does not require mobile devices to connect to the cellular or WiFi networks, but is enabled by the use of opportunistic wireless connections between mobile devices. As the number of mobile devices increases, there will be a denser opportunistic network providing considerable supplementary bandwidth to offload the traffic of cellular system. The limitation of opportunistic traffic offloading however is that the D2D connections are intermittent due to the movement of mobile users. Therefore opportunistic traffic offloading is only ideal for the transmission of the type of data known to be delay-tolerant. Examples are news, advertisements, email, pod casts, short videos, etc. In this section, we will review some applications related to opportunistic traffic offloading. Another important aspect in the opportunistic network is the cooperation issue. We will review the ongoing research on the incentive schemes for opportunistic traffic offloading and the background of evolutionary game theory, the approach we are going to take for studying mobile cooperation in the opportunistic network.

2.3.1 Applications

2.3.1.1 Delay-Tolerant Networks (DTN) packet forwarding

DTNs are sparse mobile ad-hoc networks in which users connect with each other intermittently. Since DTNs allow people to communicate without using network infrastructure, they

are widely used in mobile and vehicular communications [TLB09, MW11]. To handle the sporadic connectivity of mobile nodes in DTNs, the store-carry-and-forward method is used. That is, messages are temporarily stored at a user until an appropriate communication opportunity arises. User mobility is exploited to let mobile users physically carry data as relays, and forward data opportunistically when contacting others.

Packet forwarding in DTN however is challenging as the network is sparse and there is typically no end-to-end path for message transmission. Therefore, much attention has been paid to the development of efficient forwarding algorithm in DTN. The first attempt is Epidemic routing [VB00], in which source replicates packets whenever it meets an intermediate relay. This scheme approaches the optimal delivery ratio and delay by flooding the entire network. To reduce the overhead, routing protocols that limit the number of message copies at source nodes are proposed. Spray-and-Wait [SPR05] is a well-known protocol in this category. In the spray phase, source node spreads all its message copies to the first few encountered nodes. In the wait phase, nodes who have a copy of the message deliver the message directly to the destination. The optimal performance and resource consumption trade-off is also extensively studied in [ABD10, DMC10, ZWW14, LE10, ZZG10, FYG15] etc. Most theoretical work is targeting at the two-hop routing case, as it simplifies the route selection and highlights the importance of choosing the right relay nodes.

2.3.1.2 Opportunistic content distribution

Opportunistic content distribution refers to the application of sharing popular contents to mobile users using opportunistic communications with the help of the infrastructure networks. A set of "seeding" nodes are carefully selected to download the initial content copies from the Internet using the infrastructure. Then the seeding nodes exploit the opportunistic links to share the content with local users. The content will be propagated until reaching all the interested users. Several works have proved the efficiency of this content distribution approach [WCH14, PSS17, HHK12]. In [PSS17], the authors investigated the feasibility of a content distribution network formed purely by mobile users for disseminating news, texts,

and video contents. The paper showed that with even only 5% of seeding (source) nodes downloading contents from the Internet, they are able to disseminate contents to almost 70% of users who show interests in the contents. The D2D communications between mobile devices are realized by WiFi Direct. It is also proved that the proposed content distribution network is energy efficient. Real-life implementations of such opportunistic content delivery applications have recently been developed as well [BLL13, TKD17].

2.3.1.3 Device-to-Device (D2D) communications

In the previous two applications, content or packet delivery is made through the D2D connections between mobile devices. In literature, D2D communications can also be an approach for opportunistic Internet access, i.e., mobile device first connects to a nearby device via D2D link, then accesses the Internet using this peer device as a relay. D2D communications provide several benefits to the existing cellular network: coverage extension [KS17], improving bandwidth utilization [IGH17, ZSM10], and multipath Internet access [ZS14], etc.

2.3.2 Incentive Schemes

A critical assumption in opportunistic traffic offloading is user cooperation. Therefore it is straightforward to study the incentives to encourage mobile users to participate in opportunistic network applications. Auction theories are adopted in [ZGC11] to encourage opportunistic traffic offloading. Pricing schemes are considered in [LZG16]. Different from them, [SKT13] proposed an incentive mechanism based on reward to encourage content forwarding in opportunistic networks. Incentive schemes in DTN fall into three categories: (1) credit-based incentive schemes which introduce virtual currency to regulate message forwarding in DTN; (2) reputation-based schemes which evaluate the cooperation levels of users, and selfish users get less service than cooperative users; (3) tit-for-tat (TFT) schemes where users contribute the same amount of resource as they obtain from other users. An informative review of these incentive schemes can be found in [RC15]. The challenge, however, lies in the cost of operating and securing the incentive systems. Some researchers are seek-

ing conditions for mobile users to behave cooperatively without external incentive schemes [FHB06, UBG03].

2.3.3 Evolutionary Game Theory (EGT)

EGT extends the concepts of game theory to evolving populations [Wei97]. It models an evolutionary (Darwinian) process where the more fit individuals pass on their strategies to more offspring and increase their representation in the population. Lots of effort is devoted to studying the *replicator equations* of evolutionary games, which characterize the strategy dynamics under the influence of natural selection [HS03]. The attractors (fixed points) of the equations are called Evolutionarily Stable Strategies (ESS), a refinement of Nash Equilibrium. A strategy is ESS if it cannot be invaded by any alternative strategies. The EGT model serves as a mathematical framework to understand cooperative behaviors in a wide range of biological and social games [NS04]. Early works typically assume pair-wise interactions in infinite and complete mixing populations.

More recently, population structures such as lattice and complex networks are introduced to the EGT model [NTA10, PGS13]. While the impact of spatially structured populations on evolutionary dynamics is mostly studied with the aid of computer simulations, a surprisingly simple analytical result, known as the "σ-rule" [OHL06], reveals that structured populations promote the evolution of cooperation. Multi-player games representing group interactions can be naturally embedded in the structured populations and extend two-player games to a generic form [GT10]. The σ-rule for multi-player games is developed in [PWT16, PWA16].

The EGT methodology has been applied to many networking research problems. To name a few, [CWL10a] formulated cooperative streaming as an evolutionary game and derives the ESS strategy of the game; [JCL14] proposed an EGT framework to model the information diffusion dynamics in social networks; [SJ15, DG17b] studied the cooperation in Public Goods Game under topological changes in vehicular and mobile social networks. EGT model for DTN routing protocol also exists in literature [EDK10]. It integrated the EGT model with credit-based incentive to motivate cooperation in packet forwarding games. The difference is

that [EDK10] assumes infinite and well-mixed populations, and its main focus is mechanism control with resource constraint.

CHAPTER 3

The Software-Defined Multipath Transmission System: an Integration of MPTCP and SDN

In this chapter, we develop a software-defined multipath transmission system framework which integrates MPTCP and SDN protocols in a single network architecture. The motivation of this system comes from issues like large transmission latency and frequent satellite handovers in Low Earth Orbit (LEO) and geostationary (GEO) satellite networks, which pose a challenge for the design of transport and routing protocols. In section 3.1, we present the formation of an MPTCP-SDN framework that solves these two problems. Then in section 3.2, we show how the framework is enhanced to solve the traffic optimization problem in naval satellite communication system.

3.1 MPTCP-SDN Framework in LEO Satellite Network

Satellite systems have served mankind in a heterogeneous set of ways since its creation. With the feature of global coverage, they have been providing services such as TV broadcasting, digital messaging, GPS navigation and worldwide telecommunications for decades [Sax14]. In recent years, the tremendous growth in mobile devices and demand for connectivity "aytime, anyplace" pushes the trend of satellite-terrestrial system integration. Satellite systems may become an important complement to the next generation cellular system (5G) [EOS15].

The telecommunication community has drawn attention to the satellite systems deployed at altitudes between $500km$ and $1500km$, which are known as Low Earth Orbit (LEO) satellite systems [PKP07]. A LEO satellite network is composed of polar orbits (planes), each with a certain number of satellites at low distances from the Earth as shown in Fig. 3.1.

Compared to geostationary (GEO) satellite networks, the propagation delay in LEO satellite systems is usually under $100ms$, while GEO has a propagation delay of $250ms$. Furthermore, LEO satellites have lower transmission power requirements. In many constellations, direct inter-satellite links (ISLs) between mutually visible LEO satellites are used to carry signaling and management traffic as well as data packets [WJL95].

However, the topology of LEO satellite networks changes over time due to constant rotation of the LEO satellites. This will result in frequent satellite handover whenever the current gateway (or serving) satellite fades out and the ongoing connection has to be transferred to the next satellite visible to the ground terminal [CAI06]. For example, in Iridium system, the visibility period of a satellite is usually 8-10 minutes. Without appropriate control and mitigation mechanisms, satellite handover can lead to route failures, drastic link quality change, call blocking and eventually result in severe performance degradation.

We classify the transmission state as stable state and handover state. Stable state corresponds to the period when the gateway satellite is constantly visible to the ground terminal, while handover state corresponds to the satellite handover period. Even though the network condition is relatively steady during stable states, it has been well known that the performance of TCP is seriously impaired in LEO satellite systems, due to long Round Trip Times (RTTs) and packet losses [CS00][AG00]. The RTT problem is even worse in LEO networks, since the end-to-end delay varies as satellites move along orbits.

Different high bandwidth delay product (BDP) TCP variants are proposed to deal with these issues [BCV07][HRX08][Kel03][MCG01][CF04]. Given their strengths in miscellaneous scenarios and applications, we notice that none of them are specially designed to protect against satellite handover. A transport layer solution for satellite handover is yet to be discovered.

We attempt to solve the problem using Multi-path TCP (MPTCP). With the rise of highly connected mobile devices which possess multiple wireless interfaces, MPTCP becomes an underway promising protocol matching the needs of today's multipath networks.

MPTCP is a new transport layer protocol that allows exploiting several Internet paths be-

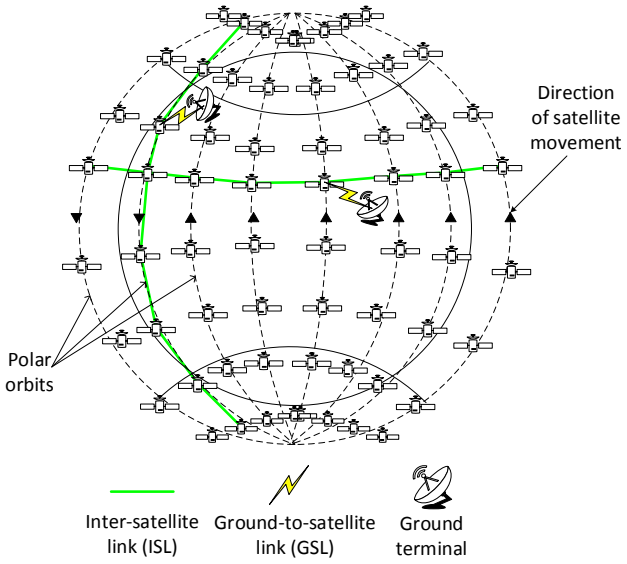


Figure 3.1: LEO satellite constellation.

tween a pair of hosts, while presenting a single TCP connection to the upper layer. Therefore the upper layer applications only need to deal with a single logical "master" TCP connection, while multiple sub-flows are running underneath, each of which is a conventional TCP connection. With the rise of highly connected mobile devices which possess multiple wireless interfaces, MPTCP becomes an underway promising protocol matching the needs of today's multipath networks.

MPTCP is well supported by multihomed devices, such as smart-phones that are equipped with both 3G and WiFi. With multiple network interfaces, these devices are allowed by MPTCP to establish sub-connections through multiple paths and increase performance. IETF has released a wealth of experimental standards for MPTCP implementation in the current Internet [FRH11b][Bag11][RHW11][FRH11a]. Additionally, various experiments have been conducted to verify that MPTCP smoothly transfers traffic on a breaking sub-flow to other flows during vertical handovers [PDD12]. These facts motivate us to investigate the advantage of deploying MPTCP in LEO satellite networks.

However in satellite networks multiple paths are not offered by the legacy routing algorithm. To exploit the best of MPTCP's capabilities, we propose an On-demand Multipath

Source Routing (OMSR) scheme which can efficiently discover multiple disjoint paths by leveraging the predictability of LEO satellite topology. After MPTCP sub-flows are initiated, they branch out from the gateway satellite into multiple disjoint paths discovered by OMSR. We define the necessary interactions between transport and routing layers to ensure that MPTCP sub-flows are split into disjoint paths. We also demonstrate the use case of such an MPTCP-OMSR joint protocol in both stable and handover states. Simulation results verify that the MPTCP-OMSR protocol improves throughput and delay performance and handles handover gracefully.

3.1.1 MPTCP-Routing Interaction

In this section, we introduce a framework for MPTCP-Routing design in LEO satellite network. We will start with the basic routing architecture in LEO system. Then we show how MPTCP packets are forwarded through a two-level routing table. We also demonstrate the use case in both stable and handover states.

3.1.1.1 Routing architecture in LEO satellite networks

The routing architecture we assume in this section follows the framework of Internet routing over LEO satellite networks proposed in [NCD98]. According to this framework, externally, LEO satellite network should work transparently with terrestrial IP routing, and internally use its proprietary routing protocol. For a ground terminal to connect to the system, it first needs to register to the border gateway of the Autonomous System (AS), which is the entry point satellite over head. This action will retrieve the IP address of the satellite gateway.

Next, the ground terminal can send packets to this IP address with final destination IP address IP_{dest}. The entry point will determine the exit gateway satellite by looking up its destination IP routing table, and will forward the packet to the exit point associated with IP_{dest} using proprietary routing protocol (as shown in Fig. 3.2). Essentially the LEO satellites are maintaining a two-level routing table, one on IP layer and the other on network layer.

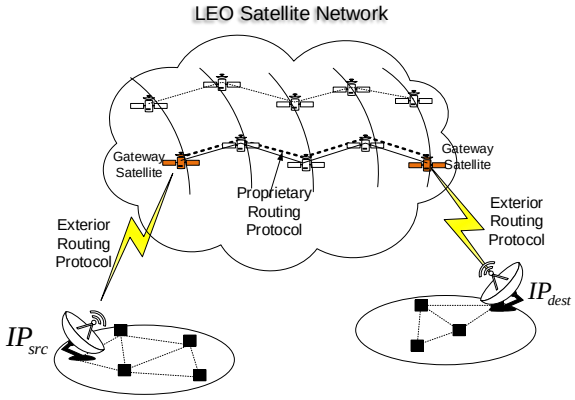


Figure 3.2: Routing architecture in LEO satellite network.

For our case, we propose an IP-OMSR based routing protocol. After the ground terminal registers and obtains the IP address of the LEO satellite network, it sends packets to the entry satellite using the default exterior routing protocol. Next, the entry satellite extracts the destination IP address, looks up the exit satellite in the IP routing table, and forwards the packets using OMSR routing scheme. To separate the IP and OMSR routing layers and avoid unnecessary routing updates, IP packets are tunnelled at the entry satellite as the payload of OMSR packets, and source routing options are injected in the packets' header to specify the route. This process is illustrated in Fig. 3.3. When OMSR packets reach the exit satellite, which is the end point of the tunnel, they are decapsulated and IP packets are reassembled. Then the next IP hop is determined based on the destination IP address. Eventually, all the IP packets are forwarded to this address, again, using external IP routing. For succinctness, we refer to this IP-OMSR routing protocol as OMSR protocol.

3.1.1.2 MPTCP-routing interaction

Now we show how the OMSR routing protocol supports MPTCP in LEO satellite networks. Several interactions will be defined to make this possible. Before we proceed, to simplify the discussion, there are a few assumptions on the ground terminal and the system.

1. The ground terminals represent the users in LEO satellite system. They must be

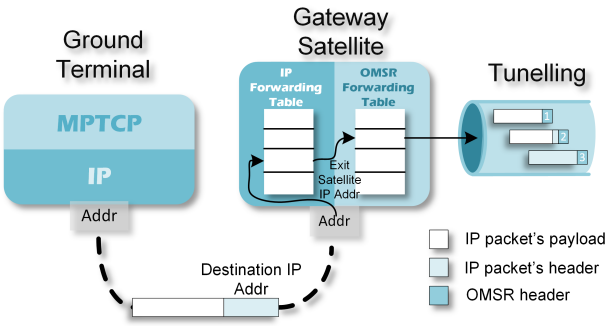


Figure 3.3: IP-OMSR routing.

MPTCP-capable for the system to run MPTCP;

2. To simplify the discussion, the ground terminals are assumed to have 2 interfaces and thus 2 IP addresses. Each address is configured to a network interface that can be used for a particular MPTCP sub-flow;
3. The gateway satellites support OMSR routing in order to allocate MPTCP sub-flows to different disjoint paths.

3.1.1.3 Use case in stable state

Recall that by definition, ground terminals can see only one gateway satellite in the stable state (i.e. no imminent handover). In spite of the fact that a ground terminal has 2 interfaces, all the MPTCP sub-flows will have to share one common entry satellite (through different physical channels). This resembles the single next-hop router case in [HRB09]. To maintain multipath in the satellite constellation, one must use source-address-based routing in the router as reported in [BPB11]. In Linux, the source route configuration can be supported using the "policy routing" concept. The basic idea is that, for hosts who have multiple addresses, the router will select different routing tables to forward packets based on the observed source address. We use a similar approach to distribute MPTCP traffic to disjoint paths.

Suppose a pair of users A and B are communicating with each other using MPTCP, where A is the source and B is the destination. A has two IP addresses A_1 and A_2, while B has B_1

and B_2. Let us assume that all the addresses have been registered, and A established 2 TCP sub-flows from A_1 to B_1 and A_2 to B_2. We denote them as $\langle A_1, B_1\rangle$ and $\langle A_2, B_2\rangle$. Then A starts to transmit data to the entry satellite S_1. S_1 will receive packets with IP address pair $\langle A_1, B_1\rangle$ and $\langle A_2, B_2\rangle$. As we introduced in the routing architecture, first S_1 will determine the exit satellite based the destination IP address. Suppose that B is currently registered at satellite S_2, so S_2 will be the exit satellite in this case. The first interaction between MPTCP and OMSR is:

Interaction 3.1. *MPTCP should notify OMSR the number of disjoint paths it needs. We denote this number as* m_0.

In our case, $m_0 = 2$. But in general, m_0 is equal to the number of sub-flows from A to B. Next OMSR sends RREQs to S_2 with m_0 included, and S_2 selects m_0 disjoint paths from $\mathcal{C}$ and returns them to S_1. Suppose OMSR discovers path P_1 and P_2 and stores them in the network level routing table. Before S_1 tunnels the packets and forwards them to S_2, we use source-address-based routing to split the traffic (suppose $\langle A_1, B_1\rangle$ uses P_1 and $\langle A_2, B_2\rangle$ uses P_2). Instead of using different routing tables, we use different entries to look up the outgoing interfaces. This process is illustrated in Fig. 3.4.

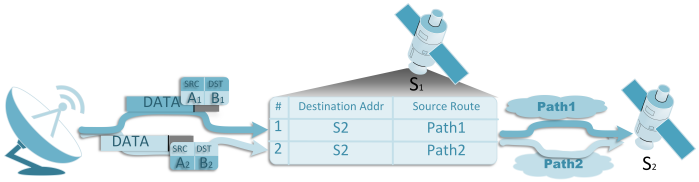


Figure 3.4: Source-address-based routing in stable state.

When the packets reach S_2, the payload will be extracted and forwarded to address B_1 and B_2. In this way, we split two TCP sub-flows to two disjoint paths from entry satellite to exit satellite. The ACK packets from B to A are forwarded in a similar way along the reverse paths.

3.1.1.4 Use case in handover state

In the handover state (i.e. imminent handover), the ground terminal can see two gateway satellites, thus can distribute TCP sub-flows over the two different links to the two satellites. Since the source and destination ground terminal enter handover state independently (from stable state), there are in total 3 combinations of source and destination states: 1) Source in handover state and destination in stable state; 2) Source in stable state and destination in handover state; 3) Both source and destination are in handover state. We will discuss the first case in more detail, and briefly summarize the other 2 cases.

Let us consider the same example where A is transmitting to B through entry satellite S_1 and S_2 using P_1 and P_2. Suppose now S_1 is fading out and a new satellite S_1' comes up. At this point, A hears the advertisement from S_1' and will allocate sub-flow $\langle A_2, B_2 \rangle$ to this new entry satellite based on its path delay measurement (assume P_1 is the shortest path so P_2 has a longer delay than P_1 as shown in Fig. 3.5. The new entry S_1', upon receiving the packets from $\langle A_2, B_2 \rangle$, will execute route discovery to accommodate $\langle A_2, B_2 \rangle$ with a disjoint path to S_2.

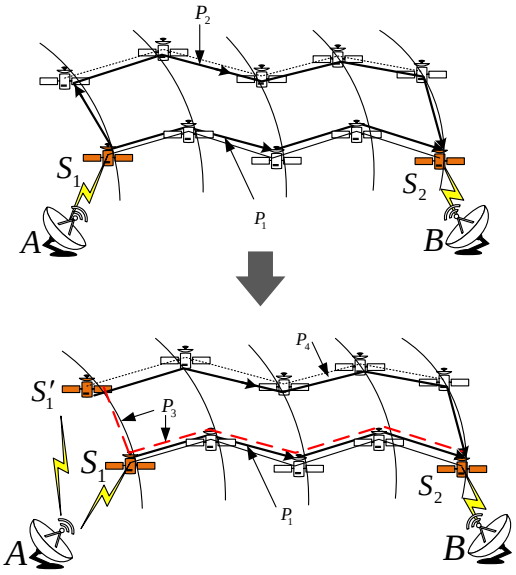


Figure 3.5: Potential route conflict.

Route conflict can arise if S_1' simply asks for 2 disjoint paths to S_2 and randomly chooses

one for $\langle A_2, B_2 \rangle$. Consider Fig. 3.5, where S_1' discovers 2 disjoint paths P_3 and P_4, and assigns $P3$ to $\langle A_2, B_2 \rangle$. Obviously P_1 and P_3 are not disjoint paths. To solve the problem, we define the second interaction as follows:

Interaction 3.2. *When MPTCP allocates m_2 out of m_0 sub-flows to a new entry satellite S_1', it splits m_0 into $m_1 = m_0 - m_2$ and m_2, and notifies S_1' to discover m_2 disjoint paths. In the meantime, MPTCP signals S_1 to issue a special packet we call route-reference (RREF). RREF carries the m_1 route vectors that are still being served by S_1.*

In this way, upon receiving the RREQ from S_1' and RREF from S_1, S_2 selects m_2 paths that are mutually disjoint and also disjoint from routes in RREF. S_2 then replies the m_2 paths to S_1'.

Now we briefly talk about the rest two cases. The second case, where A stays in stable state, and B moves to handover state, is the mirror image of the first case.

For the third case, the first terminal that moves to handover state executes the procedure described above. When the other ground terminal enters handover state, $\langle A_1, B_1 \rangle$ and $\langle A_2, B_2 \rangle$ are using entry satellites S_1, S_2, S_1', S_2'. RREF will be sent from S_1 to S_2' and RREQ will be sent from S_1' to S_2'.

3.1.2 Performance Evaluation of MPTCP in LEO System

In this subsection, we present the simulation results of the MPTCP-OMSR framework and evaluate its performance. First we talk about the basic simulation setup. Then we will demonstrate the test scenario and present simulation results. Finally, we evaluate the performance of the MPTCP-OMSR protocol based on its comparison with two traditional "single-path" TCP variants and MPTCP with single shortest path routing.

In this subsection, we choose Iridium constellation as a representative of LEO satellite networks. We use Network Simulator 2 (NS-2) as the simulator, which has long been widely used for research and education on Internet and other networking systems. In our simulation, we adopt the default settings for Iridium system. Important parameter configurations are listed in Table 3.1.

Table 3.1: NS-2 Iridium system parameters

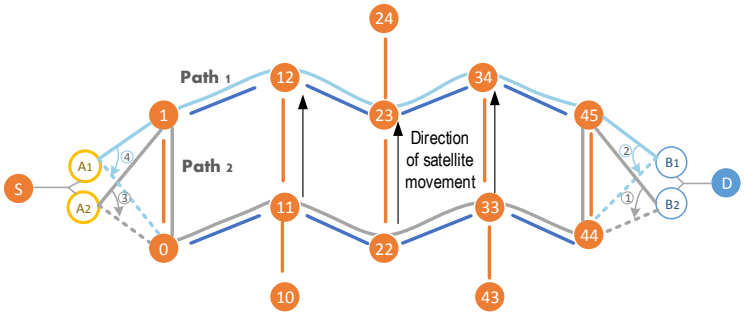
Parameter	Value
Altitude	$780km$
Planes	6
Satellites per plane	11
Inclination	86.4°
Inter-plane separation[1]	31.6°
Seam separation [2]	22°
Elevation mask[3]	8.2°
ISLs per satellite	4
ISL bandwidth	$9Mb/s$
Up/downlink bandwidth	$2Mb/s$
Packet size	$1000Bytes$

To test the MPTCP-OMSR protocol in NS-2, we design a test scenario with both source and destination executing handovers at the same time. Fig. 3.6a shows a graph illustrating the scenario. The source node S is located at position ($15.51^{\circ}N$, $1^{\circ}W$) and has two IP addresses A_1, A_2. The destination D is located at position ($17.4^{\circ}N$, $127.55^{\circ}E$) and has B_1 and B_2. S and D establishes two sub-flows $\langle A_1, B_1\rangle$ and $\langle A_2, B_2\rangle$. Initially both of the two sub-flows share the entry satellite S_1 and exit satellite S_{45}. However OMSR selects two disjoint paths P_1 and P_2 to diverse the route of the two sub-flows internally. 1 When D sees the upcoming satellite S_{44}, it connects B_2 to it. $\langle A_2, B_2\rangle$ is moved to the new disjoint path P_2' as earlier described. 2 After a while, handover at D is triggered and both sub-flows are now routed via S_{44}. P_1 is changed to P_1'. 3 Next, when handover at S happens, A_2 connects to S_0 first and P_2' is changed to P_2''. 4 A_1 connects to S_0 shortly and P_1' is changed to P_1''.

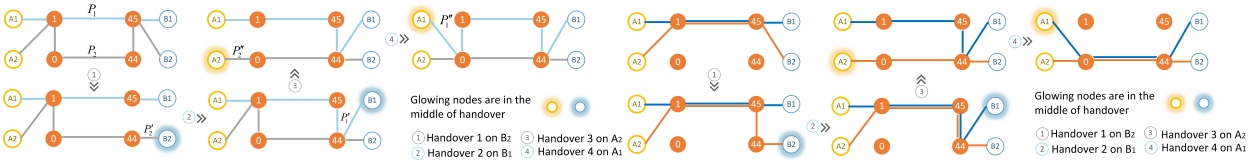
[1]The co-rotating planes are spaced 31.6° apart.

[2] The counter-rotating planes (orbit 1 and 6) are spaced 22° apart.

[3]The minimum elevation angle for an earth station is defined as elevation mask.



(a) Network topology of the scenario.



(b) Path changes of MPTCP-OMSR during handovers.

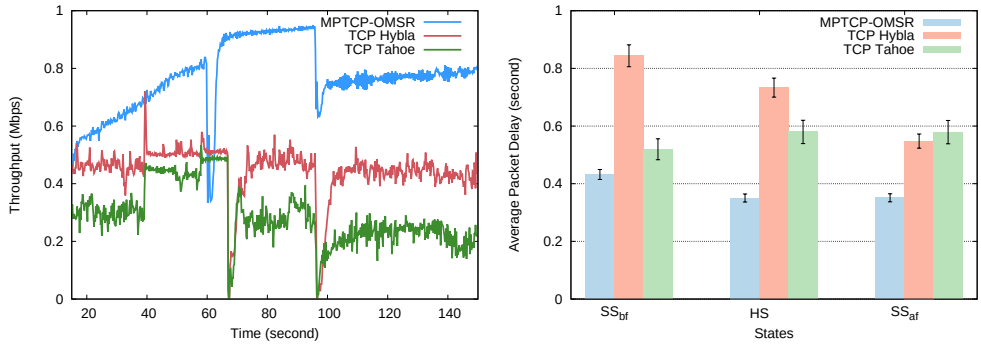
(c) Path changes of MPTCP-SSP during handovers.

Figure 3.6: Test scenario with dual handovers.

This process is illustrated in Fig. 3.6b. We note that the whole test scenario can be divided to three states: stable state before handover $\mathcal{SS}_{bf}$, handover state $\mathcal{HS}$, and stable state after handover $\mathcal{SS}_{af}$. Therefore the test scenario includes all the use cases discussed in 3.1.1 and can thus evaluate our protocol comprehensively.

For the sake of comparison, we also test two traditional "single-path" TCP variants, TCP Tahoe and TCP Hybla, in the same dual handover scenario. TCP Tahoe is widely used in the current Internet which implements a standard congestion control algorithm introduced in [Jac88]. Compared to TCP Tahoe, TCP Hybla is more suitable for high BDP scenarios such as LEO satellite constellation. As reported in [CFL09], Hybla achieves better throughput performance in presence of long RTT and high packet loss rate than most TCP variants. Comparing MPTCP-OMSR with TCP Tahoe and Hybla will give us a representative view of the performance gain MPTCP can bring for existing TCP applications running on LEO satellites.

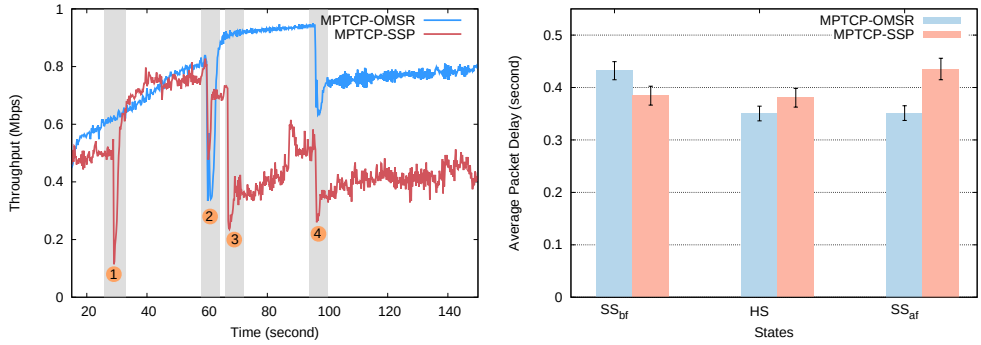
In the simulation, we randomly distribute 48 TCP source destination pairs within the area of $(5^{\circ}S \sim 35^{\circ}N, 10^{\circ}W \sim 130^{\circ}E)$ as background traffic. This simulates a regionally congested



(a) Throughput performance comparison.

(b) Packet delay performance comparison.

Figure 3.7: Performance comparison between MPTCP and single path TCP.



(a) Throughput performance comparison.

(b) Packet delay performance comparison.

Figure 3.8: Performance comparison between MPTCP-OMSR and MPTCP-SSP.

LEO satellite network where each TCP flow will have a share of $0.36Mbps$ bandwidth on average. We run 10 independent simulations with different random seeds and obtain the following throughput and packet delay results (with Confidence Intervals included). For MPTCP-OMSR, $\mathcal{SS}_{bf}$ duration: 15-30 second; $\mathcal{HS}$ duration: 30-96 second; $\mathcal{SS}_{af}$ duration: 96-150 second. For "single-path" TCP, $\mathcal{SS}_{bf}$ duration: 15-66 second; $\mathcal{HS}$ duration: 66-96 second; $\mathcal{SS}_{af}$ duration: 96-150 second.

As shown in Fig. 3.7a, TCP Tahoe approximately reaches the average bandwidth share ($0.36Mbps$) during stable state $\mathcal{SS}_{bf}$. However its throughput drops to zero during handover

state. TCP Hybla achieves higher throughput than Tahoe during $\mathcal{SS}_{bf}$. But again, its throughput drops to zero when handover happens. Because of its aggressiveness, Hybla grabs more bandwidth even when the path quality fluctuates during $\mathcal{HS}$. That is why Hybla maintains its throughput at almost the same level as its normal performance.

Finally, MPTCP-OMSR outperforms the other two variants in term of throughput. First, it has the highest throughput during $\mathcal{SS}_{bf}$. This is not surprising since MPTCP-OMSR uses two TCP flows going through disjoint paths. Second, when handover happens, the throughput of MPTCP-OMSR does not drop to zero, since the connection is never interrupted for the life time of the transmission. This verifies that MPTCP can indeed be utilized for "soft" handover. Third, the result shows that during $\mathcal{HS}$ and $\mathcal{SS}_{af}$, the throughput is even higher than during $\mathcal{SS}_{bf}$. This is due to the fact that OMSR guarantees that both TCP sub-flows are provided with shortest paths during $\mathcal{HS}$, while only one of the flows is using the shortest path during $\mathcal{SS}_{bf}$. We will explain this issue in more detail later when comparing with another MPTCP implementation.

The packet delay result is shown in Fig. 3.7b. We can observe that for all the three states, MPTCP-OMSR has shorter packet delay.

An interesting question to ask is how much improvement does the multipath routing scheme OMSR bring to MPTCP. In this section, we present the comparison between MPTCP with OMSR routing and MPTCP with the single shortest path (SSP) routing. The SSP routing computes the shortest path between satellites using Dijkstra's algorithm. During $\mathcal{SS}_{bf}$ and $\mathcal{SS}_{af}$, MPTCP-SSP chooses the same shortest path for $\langle A1, B1\rangle$ and $\langle A2, B2\rangle$, while during $\mathcal{HS}$, shortest paths between different gateway satellites are computed independently. Results are shown in Fig. 3.8a and 3.8b, where $\mathcal{SS}_{bf}$: 15-30 second; $\mathcal{HS}$: 30-96 second; $\mathcal{SS}_{af}$: 96-150 second. The numbering labels represent 1: Handover on B_2; 2: Handover on B_1; 3: Handover on A_2; 4: Handover on A_1.

It can be seen that MPTCP-SSP has lower throughput than MPTCP-OMSR during $\mathcal{SS}_{bf}$. Such result makes sense since in this case, both of its two TCP sub-flows share the same shortest path, so MPTCP uses its coupled congestion control algorithm to restrict the data

rate and be fair to other users. On the other hand, MPTCP-OMSR uses two disjoint paths during this state, so its throughput is slightly higher. This also implies that the longer disjoint path will not bring too much throughput gain since it has a longer RTT. When it comes to $\mathcal{HS}$ and $\mathcal{SS}_{af}$, MPTCP-OMSR has substantially higher throughput and slightly smaller delay than MPTCP-SSP. Additionally, as we mark in Fig. 3.8a, compared to MPTCP-SSP, MPTCP-OMSR is not affected by the handovers of $\langle A_2, B_2\rangle$. To see why MPTCP-OMSR performs so well during $\mathcal{HS}$ and can maintain the performance after handover, we demonstrate the path selection of OMSR and SSP during $\mathcal{HS}$ with Fig. 3.6b and 3.6c. From the figures we see that, even though the first and last ground-to-satellite link of $\langle A_1, B_1\rangle$ and $\langle A_2, B_2\rangle$ keep changing when link layer handover happens, OMSR always retains path segment $S_1 \to S_{45}$ for $\langle A_1, B_1\rangle$ and $S_0 \to S_{44}$ for $\langle A_2, B_2\rangle$. In contrast, SSP blindly selects shortest path after each handover and causes a larger scope of path changes, e.g. from $A_2 \to S_1 \to S_{45} \to S_{44} \to B_2$ to $A_2 \to S_0 \to S_{44} \to B_2$, and $A_1 \to S_1 \to S_{45} \to S_{44} \to B_1$ to $A_1 \to S_0 \to S_{44} \to B_2$. As a consequence, MPTCP-SSP is vulnerable to frequent handovers. MPTCP with OMSR scheme provides more robustness in path selection and achieves much higher throughput and shorter packet delay.

In addition to the throughput and delay performance, we also evaluate the traffic distribution of MPTCP-OMSR and MPTCP-SSP. Specifically, we measure the fraction of traffic on the two sub-flows $\langle A_1, B_1\rangle$ and $\langle A_2, B_2\rangle$ for the three states SS_{bf}, HS and SS_{af}. As shown in Fig. 3.9, when MPTCP-SSP is used, the traffic on the two sub-flows is more evenly distributed. This is due to the fact that SSP routing tends to select the same paths for $\langle A_1, B_1\rangle$ and $\langle A_2, B_2\rangle$, and their path characteristics are similar. Therefore MPTCP coupled congestion control allocates approximately half of the traffic to each sub-flow. In contrast, OMSR selects disjoint paths for the two sub-flows. As a result, the path characteristics are very different. For example, during SS_{bf} P_2 is longer than P_1 by 2 hops and introduces a difference in RTTs of about $40ms$. MPTCP coupled congestion control can perceive the difference and distribute the traffic unequally (in favor of the shorter path).

In summery, MPTCP-OMSR achieves a higher throughput by distributing traffic to disjoint paths, thus utilizes satellite bandwidth more efficiently. However, path disjointness

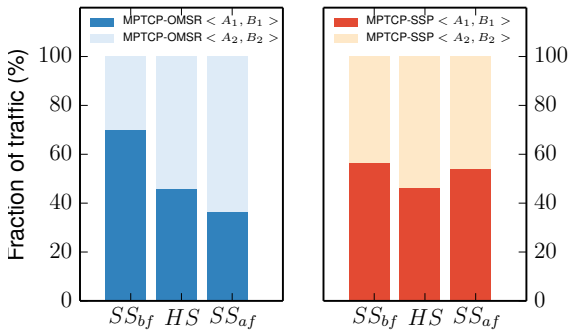


Figure 3.9: Traffic distribution of MPTCP-OMSR and MPTCP-SSP.

introduces difference in path characteristics. Unbalanced traffic distribution is an observed indication of this fact. The same phenomenon has been discovered in wired and wireless terrestrial networks [CLG13][AGC14], and it is pointed out that path characteristics do have an impact on MPTCP performance.

3.1.3 Limitations of the Distributed OMSR Routing Protocol

In the last subsection, an MPTCP and Multipath routing joint strategy is proposed. While MPTCP maintains parallel, simultaneous transport layer connections (subflows) between ground terminals, an On-demand Multipath Source Routing (OMSR) protocol works concurrently to provide the necessary link-disjoint paths. This framework handles handover gracefully by using multiple physical interfaces and significantly increases throughput compared to previous solutions.

In spite of the benefits, we identify the following drawbacks of the OMSR protocol: 1) OMSR is a distributed routing protocol. Control overhead will inevitably be introduced during the path discovery in order to fully exploit the constellation topology, which consumes limited bandwidth. 2) OMSR runs on gateway satellites independently. When the ground terminal connects to two different gateway satellites during handover, there is a chance that OMSR fails to split MPTCP subflows to disjoint paths due to the lack of inter-satellite communications. 3) It is possible to address the previous issues by devising a more sophisticated routing algorithm. For example, we can leverage the idea of Geo-routing to control

the diffusion of RREQ messages, thus reduce the overhead, or allow signallings between gateway satellites to prevent assigning overlapped paths to MPTCP subflows. However, this obviously increases the complexity of OMSR and requires more on-board processing ability in LEO satellites. To improve the routing efficiency while keeping the system simple, clean and flexible, we centralize the routing intelligence of LEO satellite networks using software defined networking (SDN).

Introducing SDN to satellite system is not a new idea. In [BZY14], the authors present the design of software-defined satellite network which divides satellite channels into control and data plane. The ground controller manages flows that are going through the satellites and provides fine-grained control. [Sax14] further shows with SDN, greater flexibility as well as reduction of operational expenses can be anticipated in satellite networks.

In this subsection, we assume that SDN is enabled in LEO satellite systems. All the satellites are turned into SDN switches that executes simple forwarding rules, so the on-board processing is minimized. The forwarding rules come from the SDN controller that locates on the ground. Based on this assumption, we design an SDN controller that is able to identify MPTCP subflows and compute disjoint paths for each subflow. The interplay between the SDN controller and a properly configured MPTCP connection is presented. We run simulations to verify the design and demonstrate how this framework improves throughput and accomplishes "soft" handover.

3.1.4 An MPTCP-SDN Framework in LEO System

We present an MPTCP-SDN framework in LEO satellite networks. Specifically, the SDN controller needs to identify the subflows that belong to the same MPTCP session, and assign disjoint paths accordingly, during both steady and handover states. We first describe the SDN-enabled LEO satellite network assumed in this subsection. Then we present the detailed framework and use cases in LEO satellite networks.

3.1.4.1 SDN architecture in LEO satellite networks

Unlike [BZY14], we consider a stand-alone LEO satellite system and assume the following architecture for SDN-enabled LEO satellite networks.

In the LEO satellite networks there is a ground control center (GCC) that monitors the satellite topology and sends control commands to satellites through several ground gateways (GG). The GCC serves as the Network Operations and Control Center (NOCC) since it has access to the full topology and is able to launch applications such as route calculation, resources utilization and mobility management. To simplify the model, we assume that the SDN controller is integrated in the NOCC. The control plane is built on top of the channels set up by GGs, bridging NOCC and LEO satellites. Control messages such as topology updates and flow management are sent here. Finally, the LEO satellite constellation as a whole forms the data plane, where data messages from users are forwarded. Since LEO constellation is a mesh network, multiple disjoint paths exist between any pair of satellites which gives reason to run MPTCP.

3.1.4.2 Communication model

The ground terminals possess 2 access interfaces to their registered LEO satellite system, and run MPTCP on transport layer. Consider satellite communications between user A and B. Suppose A has IP addresses A_1 and A_2, and B has B_1, B_2. Each IP address is associated with a different interface. During steady state, ground users have a unique entry satellite. In [DLL15], user A and B creates 2 MPTCP subflows $\langle A_1, B_1 \rangle$, and $\langle A_2, B_2 \rangle$. They connect to the same entry and exit satellite S_1, S_2 but go through disjoint paths between S_1, S_2. Starting from the latest version of MPTCP (Version 0.90.), one can create an arbitrary number of subflows between the same pair of IP addresses instead of one. Therefore we consider a different MPTCP configuration. Now only one pair of IP addresses $\langle A_1, B_1 \rangle$ is active, and MPTCP creates 2 subflows denoted as $sf^1_{A_1}$and $sf^2_{A_1}$. The other address pair stays idle in this state.

3.1.4.3 SDN controller design

When A connects to the satellite network, MPTCP initiates $sf^1_{A_1}$ using normal TCP handshake, except there is an `MP_CAPABLE` option carried in the TCP header to verify MPTCP capability and exchange authentication information. The signaling process is shown in Fig. 3.10. To establish $sf^2_{A_1}$, A sends an additional `SYN` message using the same IP address but with option `MP_JOIN`. In addition, the `SYN` message includes a token that is generated from the keys exchanged in the `MP_CAPABLE` handshake. This way B can identify the token and know that this is an additional subflow joining the current MPTCP connection. More details can be found in [FRH].

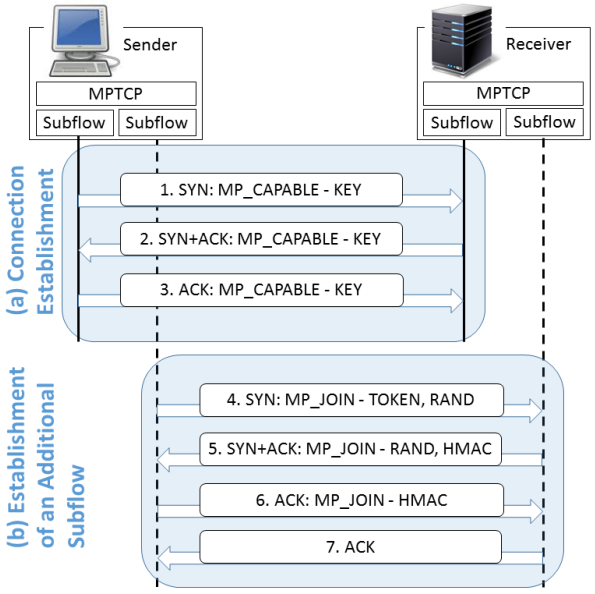


Figure 3.10: Subflow establishment in MPTCP.

According to the architecture in 3.1.4.1, entry satellite S_1 and S_2 now are SDN switches that receive and forward packets based on some forwarding rules installed by the controller. When S_1 and S_2 receive `MP_CAPABLE` messages from A and B, there is no rule installed yet. They will direct this packet to the controller through the control panel. The controller will simply assign $sf^1_{A_1}$ the shortest path P_1 between S_1 and S_2. The controller also keeps track of the keys exchanged in `MP_CAPABLE` messages and the shortest path by maintaining a hashtable $\mathcal{MP}$. In $\mathcal{MP}$, an entry is created for the MPTCP session between A and

B to store the authentication keys and paths assigned so far. Pseudocode for processing MP_CAPABLE packets is shown in Algorithm 3.1. We also create a table $\mathcal{TC}$ that contains temporary variables in MP_CAPABLE messages (line 6 and 12). The keys obtained and paths computed will then be used to create a new entry in $\mathcal{MP}$ for this MPTCP session mp (line 16-20). Specifically, mp is used as hash key of this entry and the value is a nested hash table, whose hash keys are authentication keys $SendKey$ and $RecvKey$. The bidirectional paths between S_1 and S_2 as well as the 4-tuple, IPs and Port numbers, are mapped to the entries accordingly. The keys are used to identify additional subflows that will join mp in future and the paths are used to compute disjoint paths. IP and Port number of the subflow will be installed to the switches along the computed paths (line 7 and 13).

For the subsequent MP_JOIN message, the pseudocode run by the controller is shown in Algorithm 3.2. If it is SYN message that comes from A, the controller first finds the MPTCP session that it belongs to by comparing the token with authentication keys stored in $\mathcal{MP}$ [4] (line 3). Once identified, a new path is computed from entry satellite S_1^* to exit satellite S_2^*, but should be disjoint to all other paths in the same entry (line 4). Otherwise if it is SYN+ACK message that comes from B, the controller identifies its MPTCP session using Hash-based Message Authentication Code (HMAC)[5] (line 10) and assigns a new disjoint path from S_2^* to S_1^* (line 11). Again the IP addresses and port numbers of these flows will be installed to the switches along the computed paths (line 8 and 15).

In this way the controller distinguishes $sf_{A_1}^1$and $sf_{A_1}^2$and split them to disjoint paths in steady state.

When handover approaches, the ground terminal locates on the boundary of two adjacent satellites' footprints, thus is covered by both of them. The advantage of having multiple physical interfaces is that the ground terminal can access both satellites and execute "soft" handover. Suppose S_1 is moving out of the sight of A and the next entry S_1' appears. The unused interface associated with A_2 now is activated and ready to establish new subflows to

[4]Token should be the most significant 32 bits of sha-1 hash of $RecvKey$.

[5]HMAC-B = HMAC(Key=(Key-B+Key-A), Msg=(R-B+R-A)).

B_1 via S_1', while subflows $sf_{A_1}^1$and $sf_{A_1}^2$are still carrying the whole data traffic.

A first sends an `ADD_ADDR` message to inform B of the new address A_2 from which upcoming subflows will be initiated. Next A sends 2 `MP_JOIN` messages from A_2 to B_1 to establish 2 subflows $sf_{A_2}^1$and $sf_{A_2}^2$. The messages will arrive at S_1' and be forwarded to the controller. Algorithm 3.2 is run to handle these messages. Note that when computing new

Algorithm 3.1: Pseudocode in controller that handles `MPTCP_CAPABLE` message

Input: `MP_CAPABLE` message *msg*

Output: Install path and create entry for new MPTCP connection in $\mathcal{MP}$ table

```
1  if msg has MP_CAPABLE option then
2      if SYN then
3          SendPath=path(S1,S2);
4          SendKey=msg.key;
5          SendAddr=msg.IPs:msg.Ports;
6          Add SendPath, SendKey, SendAddr to TC;
7          Install flow SendAddr along SendPath;
8      else if SYN/ACK then
9          RecvPath=path(S2,S1);
10         RecvKey=msg.key;
11         RecvAddr={msg.IPs,msg.Ports};
12         Add RecvPath, RecvKey, RecvAddr to TC;
13         Install flow RecvAddr along RecvPath;
14     else
15         Retrieve variables from TC and clear;
16         Create entry mp in MP ;
17         Add (SendAddr,SendPath) to
18                 MP [mp][RecvKey];
19         Add (RecvAddr,RecvPath) to
20                 MP [mp][SendKey];
```

Algorithm 3.2: Pseudocode in controller that handles `MPTCP_JOIN` message

Input: `MP_JOIN` message msg

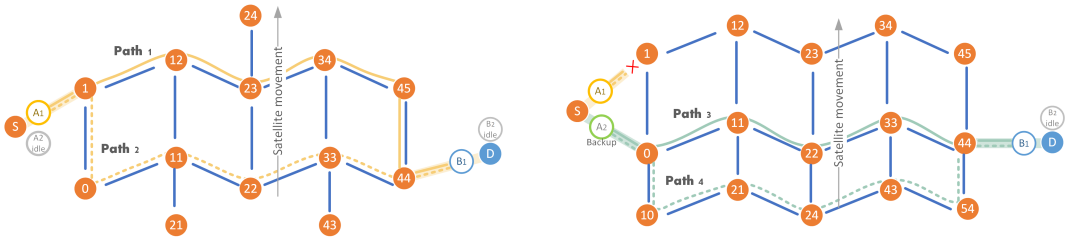
Output: Install path and update $\mathcal{MP}$ table

```
1  if msg has MP_JOIN option then
2      if SYN then
3          Find msg.token matches MP [mp].RecvKey;
4          SendPath=path(S1*,S2*) that differs MP [mp][RecvKey].Paths;
5          SendAddr={msg.IPs,msg.Ports};
6          Add (SendAddr,SendPath) to
7              MP [mp][RecvKey];
8          Install flow SendAddr along SendPath;
9      else if SYN/ACK then
10         Find msg.HMAC matches MP [mp];
11         RecvPath=path(S2*,S1*) that differs MP [mp][SendKey].Paths;
12         RecvAddr={msg.IPs,msg.Ports};
13         Add (RecvAddr,RecvPath) to
14             MP [mp][SendKey];
15         Install flow RecvAddr along RecvPath;
```

paths between S_1' and S_2 (line 4), we want it to be not only disjoint from the other existing paths under the same entry, but also as short as possible. This is because S_1' will be the entry satellite in the next steady state, while S_1 is fading out and the paths in use will break soon after handover. Therefore the controller computes 2 shortest but disjoint paths from S_1' to S_2 for $sf^1_{A_2}$and $sf^2_{A_2}$, and rearrange the paths of $sf^1_{A_1}$and $sf^2_{A_1}$if they are in the way. After establishment, $sf^1_{A_2}$and $sf^2_{A_2}$are turned to backup mode [PDD12], i.e. data traffic will not be transmitted on these subflows until other subflows become unavailable. After handover, the link between A and S_1 is down, which leads to the failure of $sf^1_{A_1}$and $sf^2_{A_1}$. Then data will be sent through $sf^1_{A_2}$and $sf^2_{A_2}$immediately to prevent packets drop. Later on A enters a new steady state with S_1' being the entry satellite, and $sf^1_{A_2}$and $sf^2_{A_2}$being the working subflows.

Similarly, when handover happens on B's end, i.e. S_2 is moving out of scene and S_2' comes up, it sends `ADD_ADDR` message informing A of its new address B_2. A_1 then establishes two new subflows to B_2 that run in backup mode. After handover the new subflows take over the old ones smoothly and MPTCP connection is never interrupted.

3.1.5 Performance Evaluation of the MPTCP-SDN Framework in LEO System



(a) Network topology before handover.

(b) Network topology during handover.

Figure 3.11: Test scenario in Iridium satellite network.

The test scenario is shown in Fig. 3.11a and 3.11b. Initially, the source S is connecting to entry satellite S_1 and destination D is connecting to exit satellite S_{44}. S first sends `MP_CAPABLE` and establishes $sf^1_{A_1}$using A_1. The ground controller then installs this flow along $Path_1$. Subsequently, $sf^2_{A_1}$is established and assigned to $Path_2$, which is disjoint to $Path_1$ between S_1 and S_{44}. The other address A_2 is disabled for MPTCP at this moment. As the satellites move, the next entry S_0 becomes visible and S establishes two backup subflows $sf^1_{A_2}$and $sf^2_{A_2}$using A_2. At the same time, the controller installs $Path_3$ and $Path_4$. When handover happens, the link between S and S_1 breaks and all data traffic of this MPTCP session is shifted to $sf^1_{A_2}$and $sf^2_{A_2}$, as shown in 3.11b. $sf^1_{A_2}$and $sf^2_{A_2}$will take charge of the transmission until the next handover.

For the sake of comparison, we also test two other protocols. The first is a "single path" TCP variants called TCP Hybla. TCP Hybla is suitable for high BDP scenarios such as LEO satellite constellation. As reported in [CFL09], Hybla achieves better throughput performance in presence of long RTT and high packet loss rate than most TCP variants. The other protocol we test is MPTCP with Equal-Cost Multipath (ECMP) routing [Hop00].

ECMP is a simple but widely deployed routing scheme for MPTCP in datacenters. It can quickly select multiple shortest paths and split the subflows using random hashing. The disadvantage is that ECMP only detects shortest paths and might assign subflows to the same path because of random hashing. In the simulation, we randomly create 48 flows

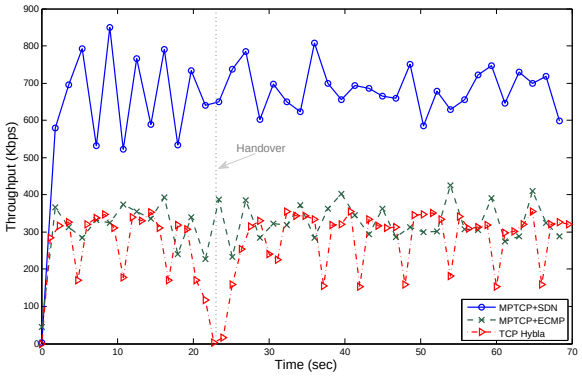
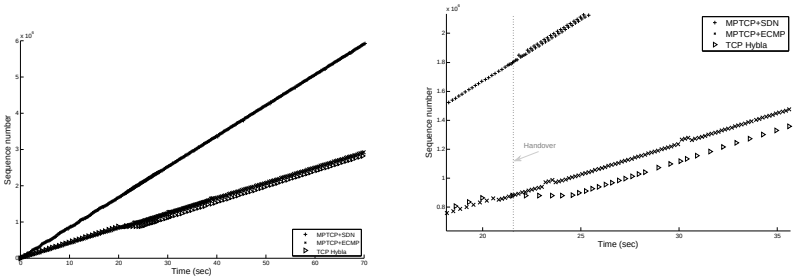


Figure 3.12: Throughput performance.



(a)

(b) Zoomed in time sequence graph during handover.

Figure 3.13: Time sequence graph.

in the simulated area as background traffic causing a regional congestion in LEO satellite network. Each TCP flow will have a share of $0.36Mbps$ bandwidth on average. We run each of the three tested protocols on the same pair of users and obtain the throughput results in Fig. 3.12. Obviously our MPTCP-SDN framework achieves a much higher throughput throughout the transmission. In fact MPTCP-ECMP has similar average throughput as TCP Hybla, since it only detects multiple shortest paths and fails to split the subflows to disjoint paths, in which case MPTCP behaves like a "single path" TCP. On the contrary,

the SDN framework designed in this subsection is MPTCP-aware and is able to compute disjoint paths with unequal cost. By utilizing disjoint paths across subflows, MPTCP-SDN has throughput twice as much as the other two schemes. On the other hand, when handover happens (marked in the Fig. 3.12) the throughput of TCP Hybla drops to zero. However for both MPTCP-SDN and MPTCP-ECMP, the throughput is not affected by handover. This is because after handover the traffic will be transferred to the two backup subflows and achieves a smooth failover. We also plot the progress of global sequence number for the three protocols. Fig. 3.13a verifies that the rate of acknowledged packets for MPTCP-SDN is twice the rate of MPTCP-ECMP and TCP Hybla. In Fig. 3.13b, we observe that after handover, the sequence number of TCP Hybla freezes for few seconds before it starts increasing, meaning the connection of "single path" TCP will be severely interrupted by satellite handover.

3.2 MPTCP-SDN Framework for Traffic Optimization in Naval Satellite Communication Network

The naval fleet relies on a combination of various government and commercial satellite communication systems (SATCOM) for both line-of-sight and beyond-line of sight communication with its ship peers and shore and airborne nodes. Service reliability issues such as network outages and high packet loss rates due to capacity saturation and/or non-optimal flow balancing are ameliorated by leasing additional commercial SATCOM capacity or by investing in larger capacity satellite terminals for certain individual vessels.

Navy's combined SATCOM network is mostly composed of geosynchronous (GEO) satellites with the inclusion of low earth orbit (LEO) satellite links on some vessels. In the current shipboard architecture, different satellite terminals provide segregated connections to different user classes on the ship. The current shipboard network consists of several subnetworks which are created by various satellite terminals and each subnet works independently. Consequently, a user can experience a congestion situation due to the overloading of the traffic in the subnet assigned to one satellite terminal, while the capacities in adjacent satellite termi-

nals may be under-utilized. Therefore, although today's navy Wide Area Network (WAN) is equipped with several satellite systems with the described architecture, it can only provide a single path access for the clients, which is limited to a best-effort internet connection without any guaranteed network services for especial users/flows. In essence, each shipboard users bandwidth is restricted to just one satellite terminal communication capacity. Moreover, in the current navy fleets network architecture, there is no mechanism to probe into real-time or historical traffic statistics due to strict security policies guarding information flow across the red-black side boundary. Knowledge of this side-channel information of the network can lead to better decision-making, especially in critical situations such as battlefield operations where real-time course corrections are necessary for improved network performance and overall mission success.

In addition, current navy's shipboard networks are also vertically integrated. It means that the control plane and the data plane are tightly coupled inside network devices. Therefore, the network flexibility is extremely reduced, hindering the innovation and evolution of the network infrastructure as well as its maintenance once it has been deployed.

To tackle all the above-mentioned restrictions, we propose Software Defined Networking for naval SATCOM (`SDN-SAT`), an MPTCP and SDN based network management architecture that allows flexible utilization of available resources and optimization of network operation based on a global view. It provides a global SATCOM network view to the navy network operator. Furthermore, this approach will bring the advantage of efficient bandwidth allocation, multiple survivable paths, and guaranteed quality of service for the users.

In this section, firstly, coexistence of these two approaches and their contribution in synergizing a production network will be addressed. Afterwards, we elaborate on the advantages of such an approach implemented over a naval ship network connecting to multiple satellite communication systems.

3.2.1 The SDN-SAT Architecture

In the proposed naval network architecture, each of the ships in the naval fleet will be equipped with an SDN-enabled switch. All satellite terminals - in particular the link layer modem subsystem - in a ship will have direct control plane communication access to the SDN-enabled switch. The modem of each satellite terminal has two main functional responsibilities: First, it interacts with the SATCOM network controller to broker and request SATCOM bandwidth based on the utilization statistics provided by its network switch. Second, it communicates with its network switch to relay SATCOM bandwidth statistics that are achievable in the current or near-future time frames. The local area networks (LANs) inside the ships or generally the subnets that previously had been connected to just one satellite terminals are connected to different ports of the SDN-enabled switch. Fig. 3.14 illustrates a symbolic shipboard network architecture in the `SDN-SAT` architecture.

It is recommended that control traffic between SDN switches and their controller is transported via SATCOM links that are reliable and of high capacity such that they meet the Southbound Application Programming Interface (API) protocol requirements as defined in OpenFlow.

The SDN controller can reside on the ground or on a ship as long as it has connectivity to the same SATCOM links. Another design choice is the total number of SDN controllers for the entire system. At a minimum a single controller can communicate with all shipboard SDN switches. At the maximum an SDN controller is used for each SATCOM service, thereby creating a more reliable and redundant network of SDN controllers at different geographic locations. In this second option, each SATCOM-specific controller only needs to maintain a single SATCOM link that it is servicing, hence enabling the possibility of co-locating the SDN controller with the ground station of the SATCOM service provider. It is assumed that in the case of multiple SDN ground controllers, they are all inter-connected via reliable terrestrial links. To run global optimization algorithms across all SATCOM systems, a global controller must be designated, either via an election process amongst them or as a stand-alone unit with interconnection to other controllers.

3.2.1.1 SDN control data exchange

There are two control APIs: SDN switch-controller and SDN controller-controller. The former is known as Southbound interface while the latter is East-West interface. If all controllers are ground terminals, then East-West control overhead cost is negligible since data is transported via high speed terrestrial links. If a controller is co-located at the SATCOM provider's ground station, then control message propagation delay between SDN switch and controller is equal to twice the ground-to-space delay (540 msec). This delay is doubled if the controller at a remote site.

SDN switch-controller data exchange uses over-the-air SATCOM bandwidth. In general, bandwidth overhead is proportional to the number of switches (i.e. ships) in the network - currently at 100, maximum - and the frequency of control exchange. Due to one-way propagation delay of at least 540 msec, switch-controller control exchange frequency is expected to be every second or longer. Each SDN switch reports router ingress and egress traffic statistics as well as SATCOM link capacities. Based on our estimates, we predict SATCOM bandwidth overhead per ship to be roughly 50 bytes/sec. In return, the controller sends back load-balancing instructions as well as SATCOM capacity requests, which together also amounts to roughly 50 bytes/sec. Given their small byte count, control data messages can be multiplexed with regular user data frames.

3.2.1.2 MPTCP for SDN-SAT

Multipath TCP allows each shipboard LAN user the option of maintaining several subflows within one TCP session. The number of subflows between each IP addresses pair can be set inside the MPTCP kernel of each LAN user host. The number of subflows for each individual user must be at least two. However, we claim that the optimum number of subflows for a LAN user on `SDN-SAT` with N satellite terminals to be N. Operational details of an `SDN-SAT` controller is as follows: First, it inspects incoming packets to a port of the SDN switch and assigns each MPTCP subflow of an MPTCP session to a different satellite link. This mechanism provides a great robustness for the navy clients since the connection

maintains through different SATCOM links. Even if the vessel loses one of its satellite assets, other connections will remain stable by the controller without any interruption. Moreover, according to the congestion status of the satellite assets of a vessel, its clients can achieve the maximum available throughput which is the sum of all the satellite terminals throughput.

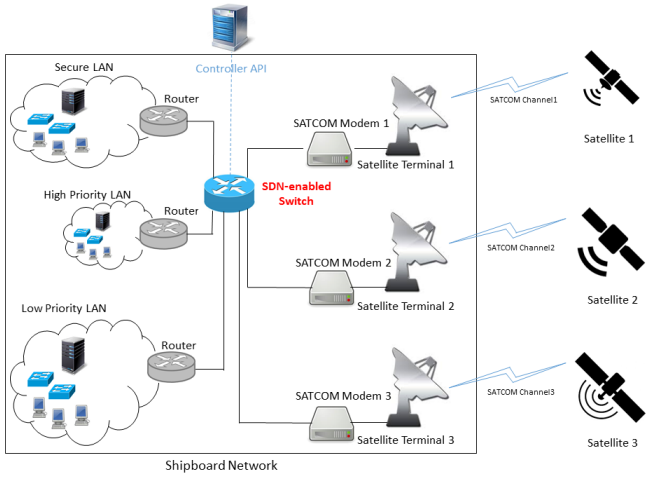


Figure 3.14: SDN-SAT shipboard network architecture.

3.2.2 Traffic Optimization as a Multi-Commodity Flow Problem

In previous subsection, we described architectural details of Software Defined Networking (SDN) implementation for the naval fleet network with multi-SATCOM connectivity, which we termed `SDN-SAT`. In `SDN-SAT`, onboard LAN hosts use either UDP or Multipath TCP (MPTCP) transport protocols for their LAN traffic flows. MPTCP splits a packet stream into multiple TCP subflows and the global SDN controller assigns subflows to multiple SATCOMs. The SDN solution also breaks the tight integration of control and data plane functions that is commonly seen in the shipboard hardware routers, enabling the adoption and dynamic reconfiguration of innovative networking protocols.

The problem not yet addressed is the network-wide traffic optimization among `SDN-SAT` networks. We assume that every SATCOM system has limited bandwidth that is assigned to `SDN-SAT` and every ship has a desired transmission rate (traffic demand) that can be estimated based on metered LAN traffic statistics. We also assume that the assigned SAT-

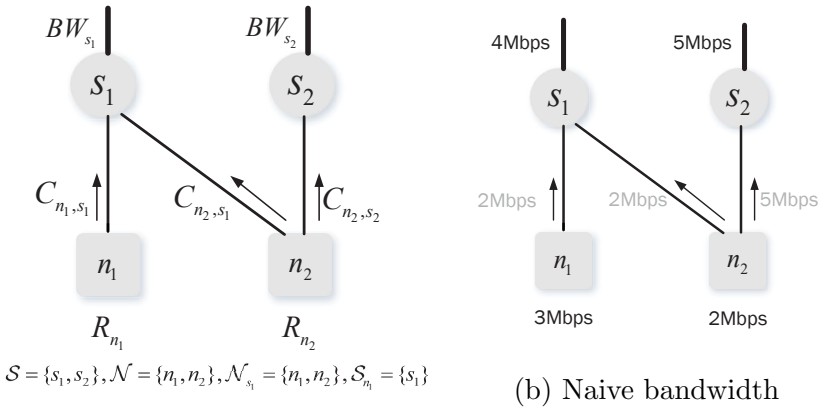
COM capacities are known to the global SDN controller and ships faithfully report their traffic demand to the global controller. Without a global bandwidth broker, navy ships can at best rely on default scheduling mechanisms (policy-based routing, MPTCP, greedy load balancing algorithms etc.) to share multiple SATCOM resources. In the MPTCP case the local scheduler is greedy and unaware of demands from other flows, resulting in suboptimal load balancing outcomes.

In this subsection, we propose an optimal load balancing solution by viewing the multi-SATCOM link utilization task as a Multi-Commodity Flow (MCF) optimization problem. Our formulation is closely related to the "Routing Assignment" problem in [FGK73], whose solution is given by the Flow Deviation Method (FDM). We propose using FDM as a network-wide optimal load-balancing solution that maximizes total throughput and minimizes traffic flow delay and jitter. Load balancing operation is carried out within the SDN framework via the SDN controller serving as the centralized traffic broker for `SDN-SAT`. The broker accepts traffic demands from navy ships and has knowledge of SATCOM capacities. In turn it instructs each ship in selecting a set of SATCOM channels and in configuring bandwidth demands on these channels. To validate network performance results and the feasibility of SDN implementation in current naval SATCOM systems, our proposed optimization algorithms and SDN protocols are tested in Mininet-based SDN emulation platform.

3.2.2.1 Bandwidth allocation as a multi-commodity flow problem

In `SDN-SAT` the set of navy ships (nodes) is denoted by $\mathcal{N}$ and the set of SATCOM systems as $\mathcal{S}$. Network topology is characterized by $\mathcal{N}_s$: the set of ships attached to any SATCOM system $s \in \mathcal{S}$, and $\mathcal{S}_n$: the set of SATCOM systems accessible by any ship $n \in \mathcal{N}$, respectively. Since each ship is bound to a switch which controls flows from this ship, for notational convenience, we also use n to denote the switch on ship n. Certain ships have full connectivity to all SATCOMs, i.e. $\mathcal{S}_n = \mathcal{S}$ while others only have partial connectivity, i.e. $\mathcal{S}_n \subset \mathcal{S}$. Each ship-satellite connection (i.e. SATCOM channel) is composed of an uplink and a downlink.

To avoid ambiguity in terminology and usage, we define the following: Each SATCOM system s allocates total data rate BW_s (in bits per second)—which we call SATCOM "capacity"—that is shared among SDN-SAT nodes. Each SATCOM channel can carry per-ship data rate (i.e. SATCOM "bandwidth"), denoted as $C_{n,s}$ for SATCOM s and ship n. The LAN user set of ship n is $\mathcal{L}_n$. Each LAN user's traffic demand is called user "bandwidth" and denoted as $r_{l,n}$, $l \in \mathcal{L}_n$. Each ship n has total bandwidth demand $R_n = \sum_{\forall l} r_{l,n}$ from its LAN users to destinations on another ship, shore nodes or external hosts switched via a SATCOM gateway. In this model we assume a rate-stationary system where both demands and supplies do not change over time. Our algorithms can be readily adapted to dynamic systems by running periodic updates. A 2-node, 2-SATCOM SDN-SAT system is given in Fig. 3.15a to illustrate our notational nomenclature. Finally, without loss of generality, we assume total traffic demand per ship is an aggregate of all LAN flows such that there is an entry per flow in the flow table in an SDN switch for selective treatment such as SATCOM link selection, rate metering, etc.



(a) Access network

(b) Naive bandwidth allocation

Figure 3.15: Bandwidth allocation in SDN-SAT

Problem Formulation The task of the SDN controller as a traffic broker is to provide instruction sets to SDN switches for their respective bandwidth allocations $C_{n,s}$ for $n \in \mathcal{N}_s, \forall s \in \mathcal{S}$, given $\mathcal{S}, \mathcal{N}, \mathcal{S}_n, \mathcal{N}_s, R_n, BW_s$. A naive approach is to distribute BW_s equally among the connected ships in $\mathcal{N}_s$. In Fig. 3.15b, 2 Mbps is assigned to each ship on

SATCOM s_1 while the entire capacity 5 Mbps of s_2 is available to n_2 since n_1 cannot access s_2. In this scenario if both n_1 and n_2 were to utilize their assigned SATCOM bandwidths, s_1 would be saturated, resulting in large packet delay and jitter. Our optimization objective is to balance traffic demands versus assigned SATCOM bandwidths across all nodes and SATCOM systems such that both SATCOM capacity and per-ship bandwidth saturation cases are avoided—provided that SDN-SAT network is under-saturated; i.e. total SATCOM capacity supply is greater than total node traffic demand.

In an under-saturated service-demand traffic model such as M/M/1 or M/D/1, packet delay D can be expressed as the inverse of residual bandwidth:

$$D \propto \frac{L}{BW_s - \sum\limits_{n \in \mathcal{N}_s} C_{n,s}}$$

where L is the packet size. As traffic demand approaches SATCOM bandwidth, packet delay becomes excessive ($D \to \infty$). We formulate the bandwidth allocation problem in SDN-SAT in the following MCF form, which is an adaptation to the "Routing Assignment" problem in [FGK73]:

$$\min_{C_{n,s}} T = \frac{1}{\gamma} \sum_{s \in \mathcal{S}} \frac{L \cdot f_s}{BW_s - f_s} \tag{3.1}$$

$$s.t. : f_s = \sum_{n \in \mathcal{N}_s} C_{n,s} \leq BW_s, \forall s \in \mathcal{S}$$

$$\sum_{s \in \mathcal{S}_n} C_{n,s} = R_n, \forall n \in \mathcal{N},\ C_{n,s} \geq 0$$

where $\gamma = \sum\limits_{n \in \mathcal{N}} R_n$. This formulation minimizes total (average) packet delay T of all flows while satisfying both capacity and bandwidth demand constraints.

The Flow Deviation Method The Flow Deviation Method (FDM) [FGK73] is a gradient-based iterative algorithm that converges to the solution of the MCF problem 3.1. Details of the algorthm as pseudo-code is provided in Algorithm 3.3. For successful execution, a complete network topology must be specified. In general end-to-end data delivery over a SATCOM system is either 2 or 4 communication hops. If the destination node is external (non SDN-SAT), then its trafic is forwarded via the ground hub gateway to external

networks—thus two hops with uplink access only. For SDN-SAT node-to-node data delivery, four hops are required:

$$\text{node} \rightarrow \text{satellite} \rightarrow \text{ground hub} \rightarrow \text{satellite} \rightarrow \text{node}$$

Here, we consider uplink bandwidth optimization only;[6] Based on this model all source nodes have a common destination: the SATCOM ground hub as shown in Fig. 3.16. Our model also assumes that the feeder link between the satellite and its ground hub has enough capacity to accommodate total uplink data rate BW_s. We denote the resulting network topology as graph $\mathcal{G}$. The FDM algorithm splits ships' total bandwidth demand into flows and assigns them to multiple paths (SATCOM channels from different SATCOM systems), aiming to balance the load and minimize packet delay. Once converged, FDM returns the flow assignment on each path from every source-destination pair. For the example in Fig. 3.15b, FDM-based allocations are $C_{n_1,s_1} = 3$ Mbps, $C_{n_2,s_1} = 0$ Mbps and $C_{n_2,s_2} = 2$ Mbps. Details of FDM, proof of optimality (in the limit) and convergence rate are provided in [FGK73][Ker93].

3.2.2.2 System implementation

Now we discuss the design and implementation details of the SDN-SAT traffic optimization algorithms and associated SDN protocols on the SDN network emulator Mininet [OSS14]. In summary, our SDN emulation testbed uses Mininet 2.1.0, Floodlight 1.2 (for the controller), OpenFlow 1.3 Software (for the switch) and MPTCP Linux Kernel Implementation v0.90. The MPTCP congestion control is set as Linked Increase Algorithm `lia`, the path manager is `fullmesh`. Experiments are run on a Linux Ubuntu 16.04 machine. Performance analysis is limited to TCP flows, highlighting unique features of MPTCP and SDN; however, our optimization methods and emulation procedures are equally valid for UDP flows.

Mininet-based SDN Environment Mininet provides a virtual network environment that consists of hosts, SDN switches, and SDN controller. Each host is a virtual represen-

[6]Joint uplink-downlink optimization research is currently ongoing.

tation of the physical machine. The hosts enable Multipath TCP (MPTCP) protocol to access available bandwidth from multiple SATCOMs. An element in OpenFlow 1.3 Software Switch that is essential to load-balancing is the *Meter*. A Meter is used to measure and control the rate of flow. It triggers a meter band if the packet rate passing through exceeds a predefined threshold—hence, also referred to as the rate limiter. Floodlight controls the network by communicating with OpenFlow switches via the OpenFlow protocol.

Algorithm 3.3: Pseudocode of FDM algorithm

Input: Graph $\mathcal{G}$ resulted from the SDN-SAT network.

Output: The optimal routing assignment.

Define: f_l^k the traffic on link l in iteration k; F_k the traffic profile in iteration k whose elements are f_l^k; iBW_l^k the inflated capacity of link l in iteration k; d_l^k the length (cost) of link l in iteration k.

Initialize: Link length $d_l^0 = \frac{\partial T}{\partial f_l^0} = \frac{L}{\gamma} \frac{BW_l}{(BW_l - f_l^0)^2}$, $f_l^0 = 0$, and BW_l is the link capacity.

while F_k *does not converge* **do**

  for *Link* l **do**

    $f_l^{k+1} \leftarrow 0$;

  for $n \in \mathcal{N}$ **do**

    Find the shortest path π_n from n to destination;

    for *Link* l *along* π_n **do**

      $f_l^{k+1} \leftarrow R_n$;

  Binary search $x^* \in [0, 1]$ such that $F_{k+1} \leftarrow x^* \cdot F_{k+1} + (1 - x^*) \cdot F_k$ yields the optimal F_{k+1} that minimizes T;

  $iBW_l^{k+1} \leftarrow \alpha \cdot BW_l$, where $\alpha = \max(1, \max_l \frac{f_l^{k+1}}{BW_l})$;

  for *Link* l **do**

    $d_l^{k+1} \leftarrow \frac{L}{\gamma} \frac{iBW_l^{k+1}}{(iBW_l^{k+1} - f_l^{k+1})^2}$;

Floodlight Implementation We implement the FDM-based traffic optimization in the Floodlight controller. Three main functionalities we add to the Floodlight controller are *MPTCP-aware forwarding*, *the FDM algorithm* and *bandwidth metering*.

FDM Algorithm *FDMCalculator* implements the FDM algorithm and populates the bandwidth limiter on target links. The input to this module is the up-to-date global topology as well as the traffic demands and SATCOM capacities. Similar to the FM, the FDM module queries TM to retrieve links and nodes in current topology. The current active paths (flows) will be stored in the FDM module in order to instruct FDM computation. As for the demand and bandwidth resource input, we implement REST API, which accepts user-specified configuration at runtime.

The *FDMCalculator* module has several components. The *FDMTopology* class is a topology builder that translates the actual network topology (built from Mininet script) into a logical topology for the ease of FDM computation. At the same time, the topology builder reads the demand and capacity passed by the REST API. These will be the input of the FDM algorithm. The second component is the *FlowDeviationMethod* class, which implements the main FDM algorithm. Finally, the *FDMCalculator* and *IFDMCalculatorService* classes wrap up the module and publish it as a Floodlight service.

Meter As noted earlier, we use OpenFlow switch meter to limit bandwidth demand of the links. We wrap our meter related functions into a class *DropMeter*. It implements the creation of meters on switches and flow-meter binding functions. The first time a flow arrives at a particular switch, the switch either consults the controller or relies on its local look-up table for an instruction list regarding this flow. Besides forwarding instructions, Floodlight controller also creates a meter (with rate obtained from FDM service) and binds the flow with the meter on this switch.

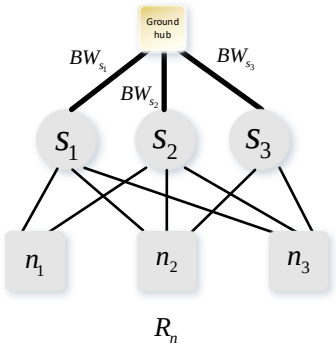


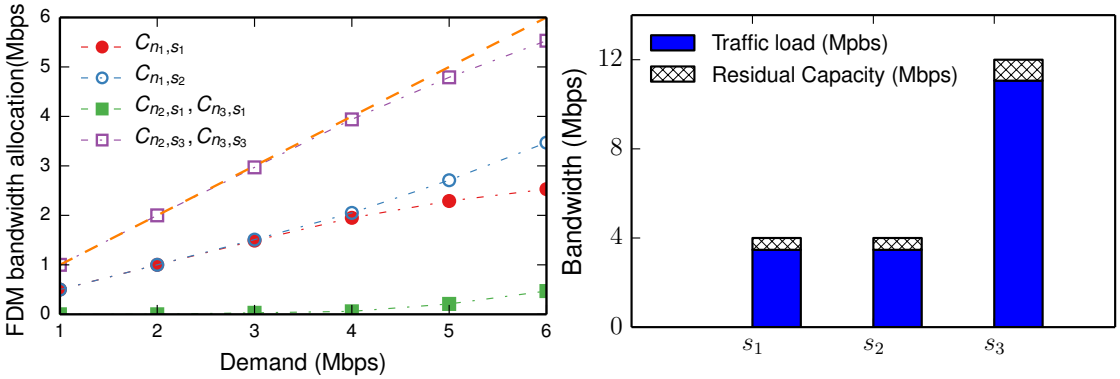
Figure 3.16: Emulation topology for SDN-SAT.

3.2.3 Performance Evaluation

The emulated `SDN-SAT` network topology is shown in Fig. 3.16. It consists of 3 ships and 3 SATCOM systems. This is the smallest network topology we can construct and still differentiate the outcomes of three data delivery options using pen-and-paper analysis. The following parameters are carefully chosen to elucidate asymmetric features among nodes and SATCOMs: SATCOM capacities $BW_{s_1} = 4.0$, $BW_{s_2} = 4.0$, $BW_{s_3} = 12.0$ (in Mbps); bandwidth demand of the ships R_n are equal, stepped up from 1 to 6 (Mbps) to generate over-saturated scenarios per link, per node or per SATCOM system. Ship1 (n_1) has no access to SATCOM 3 (s_3) while the other two have full SATCOM access. Traffic that arrives at a particular satellite on uplink channels is forwarded to the common ground hub before reaching its final destination.

If all ships use single path TCP (e.g. TCP Cubic [HRX08]), each ship relies on a default SATCOM system for data delivery. In the best case they each choose a different SATCOM that may or may not support their respective demands. In the worst case, each ship is unaware of demand from other ships—and without the coordination of SDN controller—some ships may select the same SATCOM system, resulting in poor throughput and delay statistics. MPTCP allows each ship to use multiple subflows but it alone cannot resolve network-wide bandwidth congestion problem since the underlying routing protocol uses single-path forwarding. With SDN and MPTCP-aware forwarding, every ship can be optimally allocated bandwidth on multiple satellites (i.e. if multiple SATCOM systems exist). This additional

agility of MPTCP-SDN-FDM allows each TCP subflow to use a different SATCOM link at optimized rate. This improves the individual TCP session throughput and delay jitter as well as the overall system bandwidth utilization via global load balancing offered by the FDM algorithm.



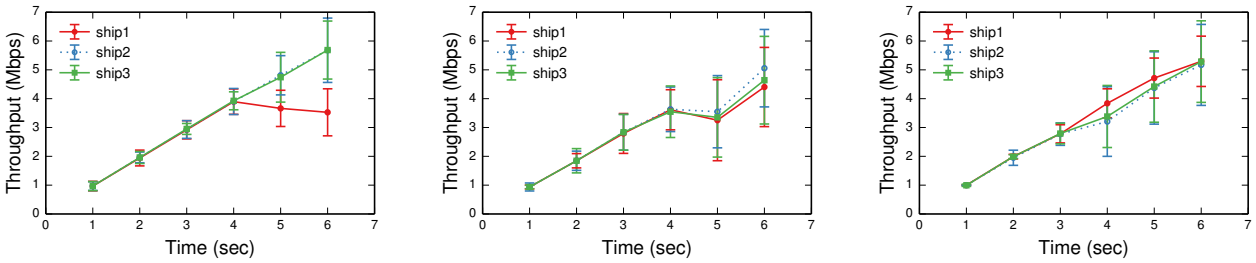
(a) $C_{2,2} = C_{3,2} = 0$.

(b) Traffic load of SATCOMs.

Figure 3.17: FDM bandwidth allocation.

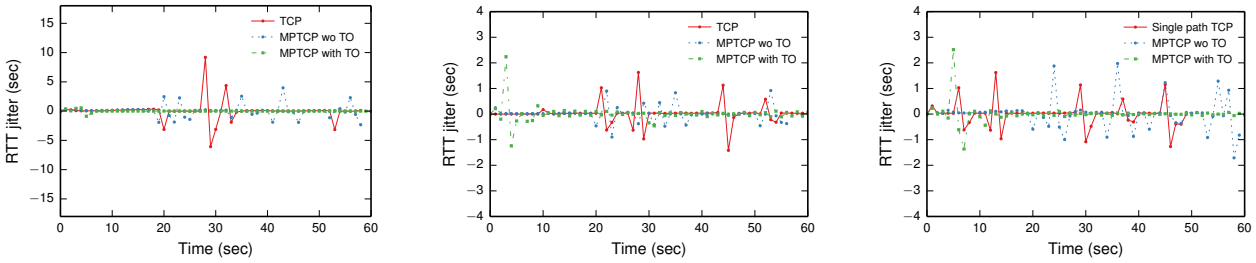
When all ships transmit using single path TCP, n_1 gets at most 4 Mbps. n_2 and n_3 either share s_3 or one of them chooses s_2. To maximize the transmission, we assume that n_2 and n_3 share s_3. Note that this is already the best configuration we can set up for single path TCP. On the other hand, MPTCP uses as many SATCOMs as possible (n_1 uses two, while n_2 and n_3 use all three). Since the total capacity is more than the demand, ideally all the ships should be satisfied if they use MPTCP. Moreover, FDM balances the load at the SATCOMs by adjusting the channel bandwidth between SATCOMs and ships. The allocation given by FDM is plotted in Fig. 3.17a when R_n increases from 1 to 6. Due to the symmetry of n_2 and n_3, their bandwidth allocation are the same. In addition, s_2 and s_3 are also symmetric. Therefore their total traffic load equal, i.e. $C_{1,1} + C_{2,1} + C_{3,1} = C_{1,2}$. Fig. 3.17b shows the total traffic load of three SATCOMs when $R_n = 6$ Mbps. Indeed, the load is well balanced by FDM and every SATCOM channel has residual bandwidth.

The average goodput (and its standard deviation) of three different schemes are shown in Fig. 3.18. In Fig. 3.18a, the goodputs of n_2 and n_3 increases linearly with their demands as they share s_3 which has abundant capacity. The goodput of n_1 suffers since its transmission



(a) Single path TCP (b) MPTCP-Baseline (c) MPTCP-SDN-FDM

Figure 3.18: Goodput performance.



(a) RTT of ship1. (b) RTT of ship2. (c) RTT of ship3.

Figure 3.19: Round Trip Time performance.

rate is limited by the capacity of s_1. In Fig. 3.18b and 3.18c, all ships have similar goodput performance as they employ MPTCP to share the SATCOMs. The difference is that, as the demand exceeds the saturation point 4 Mbps, goodput in Fig. 3.18b becomes unstable. The reason is that without intervention of FDM, the three MPTCP sessions try to optimize their packet scheduling independently. This results in greedy local traffic optimization which eventually saturates s_1 and s_2. The consequence is that subflows passing through s_1 and s_2 slow down the entire transmission [PFA14]. On the other hand, FDM globally optimizes bandwidth allocation and restricts the rate of subflows that access s_1 and s_2.

We also present the jitter of Round Trip Time (RTT) per ship under different schemes in Fig. 3.19—in this case, $R_n = 6$ Mbps. Using single-path TCP, all the SATCOMs are saturated and the delay jitter is large. Using MPTCP-Baseline, the delay jitter is almost equally large as the MPTCP scheduler greedily injects packets whenever the buffer of a subflow has free space. Therefore, s_1 and s_2 are saturated and packets that access these SATCOMs have large RTT jitter. Particularly for n_2 and n_3, the slow subflows that access

s_1 and s_2 also drag down the third subflow that goes through s_3 as a result of the head-of-line blocking issue, which is a well-know issue of MPTCP [PFA14]. Lastly, with the help of FDM traffic optimization, the RTT of MPTCP is small and smooth, as traffic load is well balanced and none of the SATCOMs are saturated.

As a final note, in our previous section, we have shown that another advantage of MPTCP compared to single-path TCP is that it improves system robustness. In the scenario of Fig. 3.16, suppose n_2 and n_3 are sharing s_3 to maximize their throughput. If s_3 is out due to jamming or intermittence, both ships will lose connection and have to be rerouted to other channels. In contrast, MPTCP maintains multiple concurrent connection to the SATCOMs and is able to smoothly shift traffic to other channels when s_3 fails.

CHAPTER 4

Infrastructure-based Traffic Offloading

In this chapter, we concentrate on traffic offloading in infrastructure-based mobile networks. The current cellular systems are facing an ever increasing number of mobile devices and applications. It has been reported in [Cis17] that the available bandwidth in cellular networks is insufficient to satisfy the demand of today's mobile users. Therefore, traffic offloading using alternative radio access technologies becomes a potential solution, since many existing networks (e.g., WiFi and satellite systems) can be directly utilized to relive the stress of the cellular systems. Among them, WiFi becomes one of the top candidates, as it has already been widely deployed for public and private internet access. In addition, WiFi provides fast wireless channel and short latency which significantly improves traffic offloading efficiency. Many network carriers such as AT&T and Verizon have attempted to build cellular-WiFi converged networks to enhance their services.

Developing an efficient traffic offloading scheme for the Heterogeneous cellular-WiFi Networks (HetNet) however is a challenging task. The design should take several features into account, e.g. traffic splitting and steering, mobility management, the provision of QoS, failover, etc. Existing solutions resolve only a subset of these problems. A class of network selection schemes [DSB14, AGG14, CIH16, ACO13, MVS14] chooses only the best single connection for mobile users, so the offloading solution can be greedy and it lacks connection continuity and failover. 3GPP [Nug16, LLR18, BZ] has released several proposals for LTE-WiFi integration in future 5G network. Its most recent proposal advocates the link layer aggregation of LTE and WiFi networks (LWA). However, this project is still ongoing and it requires both major changes to the existing infrastructure networks and software upgrades on mobile devices.

We introduce SD-MPTOP, a transport layer solution for LTE (cellular) and WiFi network integration. SD-MPTOP has two main components: (1) MPTCP subflows established between clients and servers over multiple RAN interfaces, which splits the traffic across multiple networks at the transport layer. (2) SDN centralized control over the heterogeneous RANs. Mobile devices are encouraged to enable MPTCP transmission in order to enjoy the benefits brought by SD-MPTOP such as optimized traffic splitting, mobility management, the provision of QoS, minimum transmission interruption, failover, etc. The software-defined multipath transmission framework we developed in the previous chapter serves as the base of SD-MPTOP. Our main contribution is that we develop a new mobility management mechanism in SD-MPTOP to support real-time user connectivity monitoring and dynamic flow reconfiguration. We also guarantee minimum interruption to the ongoing transmissions while system reconfiguration is in progress. Compared to existing solutions such as network selection schemes, 3GPP, or the stand alone MPTCP protocol, SD-MPTOP provides a wider range of services and better performance guarantees. It also strikes a good balance between system performance and deployment cost. We implement SD-MPTOP using Mininet-WiFi, an emulator for software-defined wireless network. Emulation results shows the advantage of SD-MPTOP.

4.1 Design of SD-MPTOP

The key design considerations of SD-MPTOP are summarized as follows:

(i) **Centralized control:** the separation of control and data plane in HetNet facilitates control logic centralization at network core and simplifies the functionality of edge devices. Centralized control brings the flexibility in designing or changing network control logic and also higher efficiency in policy making. To this end, SD-MPTOP employs an SDN architecture to decouple the control and data plane. The control plane is used for the communications (southbound interface) between the SDN controller and edge devices. For example, the SDN controller could collect information such as network topology, load of RAN, types of flows from mobile users, etc, using the control

plane. This gives the controller a global view of the network status in order to make the optimal network-wide decision. It could also be used by the controller to configure flow tables and QoS on edge devices. By adding or changing the control logic at the controller, network operator is able to conveniently manipulate HetNet operations (in the following, we use traffic steering as an example) or make necessary software upgrades. In practice, the network operators define applications or services (i.e. flow management, mobility management, security and privacy, etc.) which are converted to the control logic run by the SDN controller using the northbound interface.

(ii) **Traffic steering:** a critical part of the control logic in SD-MPTOP is traffic steering. This is the part that directly solves the traffic offloading problem. Given a set of mobile users and their connected networks, a traffic offloading solution essentially determines for each user, a set of networks to be used for its transmission, as well as the bandwidth it is allocated to from each chosen network. The SDN controller in SD-MPTOP relies on the traffic steering module to find the globally optimal traffic offloading solution. We formulate traffic offloading in HetNet as a network utility optimization problem. Instead of setting the goal as throughput maximization, in SD-MPTOP, we adopt the splittable MCF model. This formulation is suitable for SD-MPTOP because of the following reasons: (a) Unlike the throughput maximization MCF problem as adopted by [HKM13], the MCF model in SD-MPTOP associates each user with a desired transmission rate that can be inferred from its traffic type[1]. The desired rate of each user is termed demand or requirement. Compared to the throughput maximization formulation, the assumption of bandwidth demand differentiates users' requirements, leading to better fairness and higher bandwidth utilization. It is especially important to traffic offloading in HetNet as our goal is to satisfy as many users as possible given limited bandwidth resource. (b) MCF problem is classified into two categories, the unsplittable and splittable case. The unsplittable MCF problem is also called integer MCF problem, where every user is restricted to take only one path from source to desti-

[1]For example, YouTube video streaming prefers rate of $2Mbps$, while messaging requires much smaller rate say $50Kbps$.

nation. The splittable case on the other hand does not have this constraint. It has been shown in [EIS75] that integer MCF problem is NP-complete, so there exists only polynomial time approximated solutions. The splittable MCF problem is more appropriate for HetNet. First, using multiple paths per user increases path redundancy and is more reliable against network failure or mobility; second, removing the single-path constraint reduces the complexity of the MCF problem, so that the SDN controller can find the optimal solution in polynomial time. (c) In order to formulate traffic offloading in HetNet as a splittable MCF problem, it requires that the following information is made available to the controller: user demand, HetNet capacity, network connections of mobile users, RAN link quality, etc. User demand can be extracted based on deep packet inspection. HetNet capacity is accessible by querying the LTE and WiFi backbone networks. Network connectivity and RAN link information can be collected from the edge devices using the southbound interface.

(iii) **Network reconfiguration:** Most of the existing HetNet traffic offloading solutions rely on optimization algorithms to find the optimal bandwidth allocation in the system. Due to the network dynamics in HetNet, these algorithms need to be continuously running in order to update the bandwidth allocation. The update nevertheless could interrupt the ongoing transmission of existing users, especially when users are using multiple paths. This problem is rarely considered before. `SD-MPTOP` on the other hand takes it into account in the MCF problem. Specifically, we include an extra term in the objective function so as to minimize the interruption to the transmission of existing users when there is network update.

We note that the design of `SD-MPTOP` is in line with existing or planned traffic offloading proposals for HetNet, e.g. 3GPP LWA [LLR18], 5G, and HetSDN [CIH16].

4.2 Multi-Commodity Flow Modeling

In SD-MPTOP, we formulate the network utility optimization as a MCF problem. Reusing the notations defined in 3.2.2.1, the SDN controller needs to instruct HetNet edge devices bandwidth allocation $C_{n,s}$ for user n and RAN system s. In SDN-SAT, we formulate the bandwidth allocation problem as the "Routing Assignment" problem 3.1, which minimizes the average packet delay according to the $M/M/1$ queueing model. Now we extend the original "Routing Assignment" setup by incorporating two extra terms (objectives): (a) link delay, as networks in HetNet have diverse link latency characteristics. (b) bandwidth allocation change, so as to minimize the interruption introduced by an update on bandwidth allocation.

The MCF formulation in SD-MPTOP is shown as follows:

$$\min_{C_{n,s}} T = \frac{1}{\gamma}\left\{\underbrace{\sum_{s\in\mathcal{S}} \frac{L\cdot\sum\limits_{n\in N_s} C_{n,s}}{BW_s - \sum\limits_{n\in N_s} C_{n,s}}}_{load\ balancing} + \underbrace{\sum_{s\in\mathcal{S}}\sum_{n\in N_s} C_{n,s}\cdot D_{n,s}}_{linkdelay}\right\} + \underbrace{\alpha\sum_{s\in\mathcal{S}}\sum_{n\in N_s}\left\|C_{n,s}-\tilde{C}_{n,s}\right\|)^2}_{interruption} \quad (4.1)$$

$$s.t.: \sum_{n\in\mathcal{N}_s} C_{n,s} \le BW_s, \forall s\in\mathcal{S}$$

$$\sum_{s\in\mathcal{S}_n} C_{n,s} = R_n, \forall n\in\mathcal{N},\ C_{n,s}\ge 0$$

where $\gamma = \sum\limits_{n\in\mathcal{N}} R_n$, $D_{n,s}$ is the link delay between mobile user n and its connected network s, and $\tilde{C}_{n,s}$ is the history bandwidth allocation recorded by the SDN controller. By adding the *interruption* term, we take into account the last bandwidth allocation to user n in each of its connected networks, which is currently affecting user n's transmission. Minimizing the objective T means that we jointly consider load balancing, link delay, and the difference between the new and last bandwidth allocation. The coefficient α is used to adjust the weight of term *interruption* with respect to the other terms. As both of the two additional terms are convex functions, we are able to solve 4.1 using the modified FDM algorithm shown in Algo. 4.1. Since the connection between user n and network s uniquely defines a link in the

network topology, denoted by $l_{n,s}$ in line 1, the flow assigned to that link $f^k_{l_{n,s}}$ represents the bandwidth allocation for user n from network s, i.e. $C_{n,s} = f^k_{l_{n,s}}$ as shown in the last line of Algo. 4.1.

Algorithm 4.1: Pseudocode of the modified **FDM** algorithm

Input: Graph $\mathcal{G}$ resulted from HetNet, $\mathbf{R}_n = \{R_n\}$.

Output: The optimal bandwidth allocation $\mathbf{C}_{n,s} = \{C_{n,s}\}$.

1 **Define:** f^k_l the traffic on link l in iteration k; $\mathcal{F}_k$ the traffic profile in iteration k whose elements are f^k_l; iBW^k_l the inflated capacity of link l in iteration k; d^k_l the length (cost) of link l in iteration k; D_l the link delay of link l; $l_{n,s}$ the link that corresponds to the connection between n, s.

2 **Initialize:** Link length $d^0_l = \frac{\partial T}{\partial f^0_l} = \frac{L}{\gamma}\frac{BW_l}{(BW_l - f^0_l)^2} + \frac{D_l}{\gamma}$, $f^0_l = 0$, and BW_l is the link capacity.

3 **while** $\mathcal{F}_k$ *does not converge* **do**

4 **for** *Link* l **do**

5 $f^{k+1}_l \leftarrow 0$;

6 **for** $n \in \mathcal{N}$ **do**

7 Find the shortest path π_n from n to destination;

8 **for** *Link* l *along* π_n **do**

9 $f^{k+1}_l \leftarrow R_n$;

10 ***Binary search*** $x^* \in [0, 1]$ such that $\mathcal{F}_{k+1} \leftarrow x^* \cdot \mathcal{F}_{k+1} + (1 - x^*) \cdot \mathcal{F}_k$ yields the optimal $\mathcal{F}_{k+1}$ that minimizes T;

11 $iBW^{k+1}_l \leftarrow \theta \cdot BW_l$, where $\theta = \max(1, \max_l \frac{f^{k+1}_l}{BW_l})$;

12 **for** *Link* l **do**

13 $d^{k+1}_l \leftarrow \frac{L}{\gamma}\frac{iBW^{k+1}_l}{(iBW^{k+1}_l - f^{k+1}_l)^2} + \frac{D_l}{\gamma} + 2\alpha(f^{k+1}_l - \tilde{f}_l)$;

14 **return** $\boldsymbol{C}_{n,s} = \{C_{n,s} \triangleq f^k_{l_{n,s}}\}$.

SD-MPTOP uses the above MCF formulation and runs the modified FDM algorithm to

efficiently steer the traffic across multiple networks in HetNet. It also respects the existing flows in the network by minimizing the bandwidth allocation difference between current and previous runs. There are several issues to be resolved in the modified FDM algorithm. In the rest of this section, we discuss how to choose the correct value of α to effectively make use of the *interruption* term. We also introduce a method to reduce the convergence time of the modified FDM algorithm when it is run repeatedly. When the bandwidth demand exceeds HetNet capacity, the FDM algorithm will not return valid bandwidth allocation. A complementary allocation algorithm is presented to handle this case.

4.2.1 Parameter Tuning

It is important to pick the correct value for parameter α for the modified FDM algorithm to work. Consider line 9 in Algo. 4.1. When binary searching the superposed traffic profile F_{k+1} that minimizes T, two parts of T are changed: the decrease in term *load balancing* and *link delay*, and the increase in *interruption*.[2] If α is too large, a slight change in F_{k+1} results in a big increase in T and leads to $x^* \ll 1$. Therefore the algorithm will falsely converge to a suboptimal solution. On the other hand, if α is too small, the *interruption* term will not have a impact on the allocation. Existing transmission will be greatly interrupted by the new bandwidth allocation.

In `SD-MPTOP`, we use the following formula to compute the value of α:

$$\varphi \cdot \alpha \cdot (\Delta \cdot \bar{R}_n)^2 \cdot n = \rho \cdot \frac{T_0 - \tilde{T}_{opt}}{\kappa}, \tag{4.2}$$

where $\Delta \cdot \bar{R}_n$ is the average maximum offset from the previous allocation, i.e., $\Delta \cdot \bar{R}_n = \max_{n,s} \left| C_{n,s} - \tilde{C}_{n,s} \right|$; $\tilde{T}_{opt}$ is the minimum value of T in the previous allocation; $\phi \geq 1$ is the coefficient that adapts the maximum offset to a reasonable average, $\kappa \in \mathbb{Z}$ is the expected number of iterations for the modified FDM to converge. The left-hand side of Eq. 4.2 represents the increase in *interruption*, while the right-hand side denotes the desired total decrease in T per iteration. As previously mentioned, to obtain the correct value of α, we

[2]Since term *interruption* is non-negative, if the current traffic profile F_{k+1} shifts from the previous allocation, this term will be positive, causing an increase in value T.

need to carefully adjust the increase of term *interruption*. Therefore we define parameter ρ as the ratio of the increase in term *interruption* in the total decrease of T per iteration. Once we pick a reasonable value for ρ, assuming that the traffic profile along the steepest descent direction has an average offset of $\varphi \cdot (\Delta \cdot \bar{R}_n)^2 \cdot n$, the increase in *interruption* is exactly ρ fraction of the total decrease. As long as ρ is non-negligible and not too large, we will obtain a good value for α.

4.2.2 FDM Quick-start

In HetNet, mobile users frequently change their network connections due to mobility and network quality fluctuations. Therefore, SD-MPTOP periodically run the modified FDM algorithm to update bandwidth allocation. In order to update the allocation in a timely manner, the modified FDM algorithm needs to quickly converge. We make a simple modification to Algo. 4.1 in order to quick-start FDM and reduce the convergence time to fewer iterations. In the initialization, set $f_l^0 = \tilde{f}_l$ instead of 0 for remaining links in the network.

4.2.3 Max-min Fairness

The modified FDM algorithm returns under two possible conditions: either it converges to a feasible or infeasible allocation. In the former case, in line 11 $\theta = 1$ when FDM returns, meaning that the flow does not exceed the capacity of any link. In the latter case, $\theta > 1$ when FDM returns, which violates the capacity constraint.

Recall that in SD-MPTOP, the user demand is estimated based on the traffic type. If the FDM allocation is feasible, then every user will get their desired bandwidth and the load is balanced across RANs. On the other hand, when FDM allocation is infeasible, HetNet will be saturated and at least one user will not be able to obtain its desired bandwidth. In this case, we need to decide a bandwidth allocation that satisfies as many users as possible while guaranteeing certain level of fairness among the users. In SD-MPTOP we choose max-min fairness when the demand is infeasible given the capacity constraint.

Definition 4.1. *In SD-MPTOP, an allocation is* ***max-min fair*** *if it is feasible under the*

Algorithm 4.2: Pseudocode of **`Max-min`** algorithm

Input: Graph $\mathcal{G}$, $\mathbf{R}_n$.

Output: Bandwidth allocation $\mathbf{C}_{n,s}$.

Define: Γ_{unsat} the set of unsaturated user; Γ_{sat} the set of saturated user, $\bar{\mathbf{R}}_n$ the reduced demand under current consideration.

Initialize: $\Gamma_{unsat} = \mathcal{N}$, $\Gamma_{sat} = \emptyset$, $\bar{\mathbf{R}}_n = \mathbf{R}_n$.

while $\Gamma_{sat} \neq \mathcal{N}$ **do**

  $\mathbf{C}_{n,s} = \texttt{MM}(\mathcal{G}, \bar{\mathbf{R}}_n)$

  Connectivity test on the residual graph

  for *n not connected* **do**

    $\Gamma_{potsat} = \Gamma_{potsat} \cup \{n\}$;

  for $i \in \Gamma_{potsat}$ **do**

    $\mathbf{R}'_n = \{R'_j = \sum_s C_{j,s}, \forall j \neq i, R'_i = R_i\}$

    $\mathbf{C}'_{n,s} = \texttt{MM}(\mathcal{G}, \mathbf{R}'_n)$

    if $\mathbf{C}'_{n,s} = \mathbf{C}_{n,s}$ **then**

      $\Gamma_{sat} = \Gamma_{sat} \cup \{i\}$

      $\bar{R}_i = \sum_s C_{i,s}$

return $\mathbf{C}_{n,s}$

capacity constraint, upper-bounded by the original demand, and an attempt to increase the allocation of any user necessarily results in the decrease in the allocation of some other user with equal or smaller allocation.

To achieve the max-min fair allocation when the original demand is infeasible, we run Algo. 4.2 to find the max-min bandwidth allocation as the traffic offloading solution. It uses Algo. 4.3 as a helper function to binary search the maximum demand that nearly saturates the capacity. For the correctness of Algo. 4.2, please refer to [AS08].

Algorithm 4.3: Pseudocode of **MM**

Input: Graph $\mathcal{G}$, $\mathbf{R}_n$.

Output: Bandwidth allocation $\mathbf{C}_{n,s}$.

Define: **FDM**($\mathcal{G}$,$\mathbf{R}_n$) as a function call to Algo. 4.1; $\mathcal{F}$ the traffic profile; R_{max} (R_{min}) the maximum (minimum) demand under current consideration.

Initialize: $R_{max} = \max\{R_n\}, R_{min} = 0, \mathbf{R}'_n = \min\{\mathbf{R}_n, (R_{max} + R_{min})/2\}$.

while $R_{max} - R_{min} > \delta$ **do**

  if $\boldsymbol{R}'_n$ *is feasible* **then**

    $\mathbf{C}_{n,s}$ =**FDM**($\mathcal{G}$,$\mathbf{R}'_n$);

    $R_{min} = (R_{max} + R_{min})/2$;

  else

    $R_{max} = (R_{max} + R_{min})/2$;

  $\mathbf{R}'_n = \min\{\mathbf{R}_n, (R_{max} + R_{min})/2\}$;

return $\mathbf{C}_{n,s}$

4.3 SD-MPTOP Implementation

We setup a test environment to evaluate SD-MPTOP using the Mininet-WiFi emulator [FAB15] in Linux Ubuntu 14.04 machine. Mininet-WiFi is a recent extension to the Mininet SDN network emulator. It adds virtualized WiFi interfaces on wireless devices to emulate WiFi mobile stations and access points. It also models hosts movement and wireless physical channel characteristics. Therefore, Mininet-WiFi is an ideal choice for the implementation and performance evaluation of software-defined wireless networks. In order to use MPTCP as the transport layer protocol, we install the Linux kernel implementation of MPTCP v0.92. Open vSwitch (OVS) is installed to emulate SDN edge devices which communicate with the SDN controller using the OpenFlow protocol. We implement SD-MPTOP control logic as an SDN controller, which closely interacts with Mininet-WiFi using python APIs to collect network and client information. The controller also implements Algo. 4.1 and Algo. 4.2 to

compute the optimal bandwidth allocation repeatedly. Upon bandwidth allocation updates, the SDN controller uses OpenFlow API to reconfigure flow table and queues on edge switches. An overview of the implementation is shown in Fig. 4.1. Next, we present the details of some main software modules developed in the testbed.

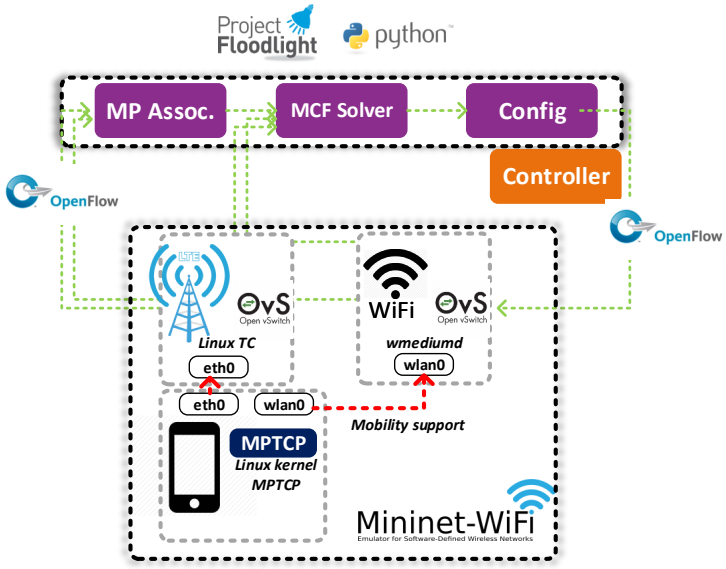


Figure 4.1: Implementation overview of SD-MPTOP.

4.3.1 Mininet Emulated Networks

We rely on Mininet-WiFi to create a HetNet environment, which contains LTE and WiFi networks, mobile users that are either static or moving, and a remote server from (to) which users download (upload) contents.

WiFi network: At the core of the emulated WiFi network is an OVS that acts as the access point (AP) accepting wireless connections from users. Mininet-WiFi adds virtualized wireless interfaces to SDN switch and configures wireless medium using the *mac*802_11*hwsim* kernel module. The switches are turned into access points through `hostapd` deamon, a user space software capable of turning normal wireless network interface cards in to access points and authentication servers. Mininet-WiFi also provides propagation and shadowing model

to simulate wireless channel characteristics such as path loss, interference, etc. In the emulation, each WiFi edge device is emulated by the virtualized AP, which contains one WLAN interface, and multiple Ethernet interfaces connecting to the backbone network. Since the virtualized AP is essentially an OVS switch, it has SDN capabilities and is configurable using OpenFlow APIs.

LTE network: We emulate LTE network using normal OVS switch which equips with only Ethernet interfaces. To setup realistic cellular physical channel characteristics such as bandwidth, link latency, loss, and jitter, we use Linux traffic control (tc) utilities upon the creation of LTE network in Mininet-WiFi. In principle, LTE networks have longer latency but larger bandwidth pool compared to WiFi network.

Mobile hosts: Mininet-WiFi extends Mininet hosts to be WLAN-capable similar to the way it extends WiFi switches. To simulate multi-homed devices that are equipped with both LTE and WiFi interfaces, the mobile hosts in Mininet-WiFi are created with both WLAN (WiFi) and Ethernet (LTE) interfaces. Mininet-WiFi also supports mobility of mobile hosts. It provides a set of default mobility models such as random walk, truncated Levy walk, random way point, etc., and also allows user to customize the movement. This feature is important for evaluating the mobility management performance of the proposed system in HetNet emulations. To flexibly manage the creation and use of interfaces in mobile devices, we rely on the python API `net.nameToNode` in Mininet to obtain the host object, which is essentially a kernel namespace, and invoke Linux command to configure its independent `ip route`. This is critical to the test of MPTCP because for MPTCP to concurrently use multiple interfaces, the routing table of hosts must be properly setup (e.g., the primary MPTCP interface is configured as the default route in the Linux routing table). Therefore the host object in Mininet serves as a handle to host operations. Note that although not explicitly implemented as part of `SD-MPTOP`, the handle may be used by the SDN controller as a direct channel to instruct mobile devices on interface creation, removal, and switching in HetNet environment.

4.3.2 SDN Controller

We develop an SDN controller which constantly monitors the network status, computes the optimal bandwidth allocation using our proposed MCF solver, and updates the SDN-enabled edge devices. The controller is written in Python. In order to collect the network information necessary for `SD-MPTOP`, the controller uses Mininet python API to query the LTE and WiFi networks as well as the mobile devices about connectivity, channel and link latency information. We note the information gathering in our testbed ideally shortens the signaling delay in the control panel, since the controller reads directly from edge devices. In real HetNet systems, the controller is usually away from the edge.

Information collection: `SD-MPTOP` feeds several information to the MCF solver including the connectivity between mobile devices and network systems, wireless channel measurements like link delay and capacity, etc. To gather connectivity information, we use Mininet python API `sta.params['associatedTo'], sta.intfList()` to obtain a list of networks mobile device `sta` is connected to. For wireless channel information, Mininet-WiFi adds a new python class `wirelessLink` which contains methods `setBW, setLatency, setRSSI` to retrieve link quality information. As for the bandwidth demand of each mobile user, although it can be inferred based on deep packet inspection, it is not implemented as part of the testbed. We assume a demand vector is initially assigned to mobile users and is made available to the controller.

MCF solver: Algo. 4.1 and 4.2 are implemented as a MCF solver in the python controller. The MCF solver eventually converges either to a feasible allocation given the demand vector, or to a max-min fair allocation by reducing some of the demands. The MCF solver implicitly chooses the path to use for each mobile device, i.e. the interface to network s should be shut off if $C_{n,s} = 0$. Since the controller periodically collect network status information, if there is a change in user connectivity due to mobility or network failure, or in user demand or channel quality, the MCF solver needs to be rerun to update the bandwidth allocation.

Bandwidth allocation enforcement: The SDN controller enforces the computed bandwidth allocation on edge devices by setting up OpenFlow flow table and queues. In the flow table, each flow entry contains matching fields and a set of actions. If a packet being processed by the edge switch matches the matching fields of a flow entry, the associated set of actions will be applied to this packet. Common actions include `Output`: specifying the output port this packet should be forwarded to, `Drop`: packet drop action, `Set-Queue`: associating packet with an OpenFlow queue at the output port, etc. In our testbed, we use the IP address of the every network interface as the matching field since we assign every user a separate IP domain, and every interface a unique IP address. The set of actions in each flow entry contains `Output` and `Set-Queue`. OpenFlow switch uses queuing mechanism for egress traffic rate control. Once the queue is created and bound to an output port, the controller can specify the queue id using `Set-Queue` action to inject the packets matched to this flow entry to a particular queue. This is how we implement bandwidth allocation enforcement in our OpenFlow switches. In addition, OVS provides `ovs-ofctl`, `ovs-vsctl` commands to set up OpenFlow tables.

4.4 Performance Evaluation

We evaluate the performance of `SD-MPTOP` and show its efficiency using the developed testbed. To thoroughly and fairly evaluate `SD-MPTOP`, we design three test scenarios and repeat the same experiments with single-path TCP (SPTCP) and MPTCP. We will demonstrate their performance in terms of throughput efficiency, fairness, load balancing, and mobility management.

4.4.1 Static Network Test Scenario

The first test scenario is designed to be a static HetNet with 1 LTE base station and 1 WiFi AP. The network topology is shown in Fig. 4.2. There are two types of mobile devices, type N_1 mobile devices who connect only to one network, and type N_2 mobile devices who are using multiple networks. This topology is mainly designed to highlight the advantage of

multipath transmission in MPTCP and SD-MPTOP. Intuitively, with MPTCP, the N_2 type users will use bandwidth from both networks, so that their throughput will be higher than the throughput of using SPTCP. The problem with MPTCP, however, is that it is unaware of the existence of N_1 users whose bandwidth is limited to a single network. Therefore, with only local view, MPTCP will fail to satisfy the goal of network load balancing or throughput fairness among all users. In the experiments, we use the following parameters: $BW_{WiFi} = 4Mbps, BW_{LTE} = 6Mbps$, $D_{WiFi} = 10ms, D_{LTE} = 50ms$, $\mathcal{S} = \{WiFi, LTE\}$; the total number of users $N = N_1 + N_2 = 10$, the number of single-connection users is $N_1 = 5$, and they are randomly connected to either WiFi or LTE network; $R_n = \lambda \cdot \frac{(BW_{WiFi}+BW_{LTE})}{(N_1+N2)}$, where λ varies from 0.5 to 2, simulating from non-saturated to saturated network conditions. For SPTCP, mobile device will by default use WiFi network if it is available. For MPTCP and SD-MPTOP, N_2 users concurrently use both networks. However, SD-MPTOP relies on the SDN controller to coordinate the bandwidth usage in the two networks.

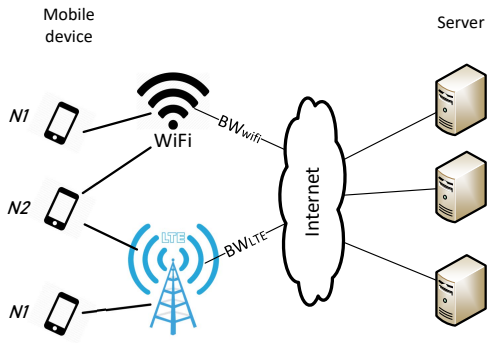


Figure 4.2: Emulation topology for the static test scenario.

The throughput results are shown in Fig. 4.3 where we boxplot the throughput of mobile users at different R_n when they use different protocols. Observing the average throughput of each protocol, it is clear that MPTCP and SD-MPTOP performs better than SPTCP. MPTCP however demonstrates greater variance than SD-MPTOP in the throughput, i.e., some users achieve much larger throughput than the others. This is not desired since we set the demand of mobile users equal, so the smaller the variance, the better the user fairness. In Fig. 4.4a

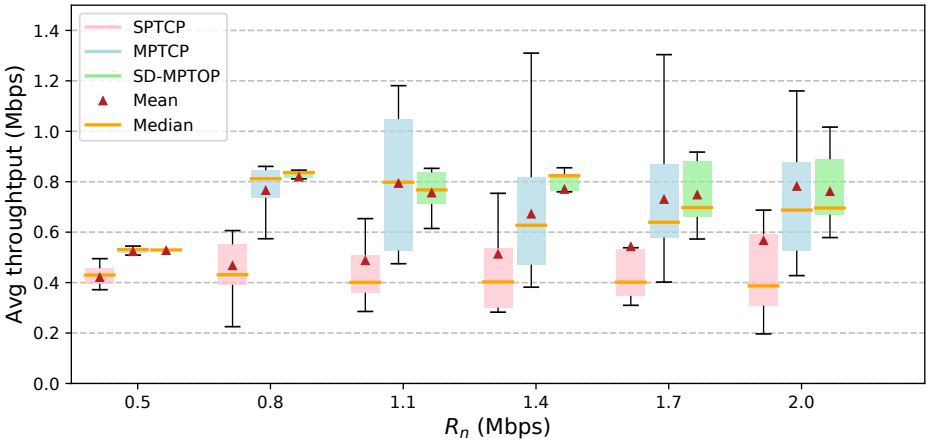
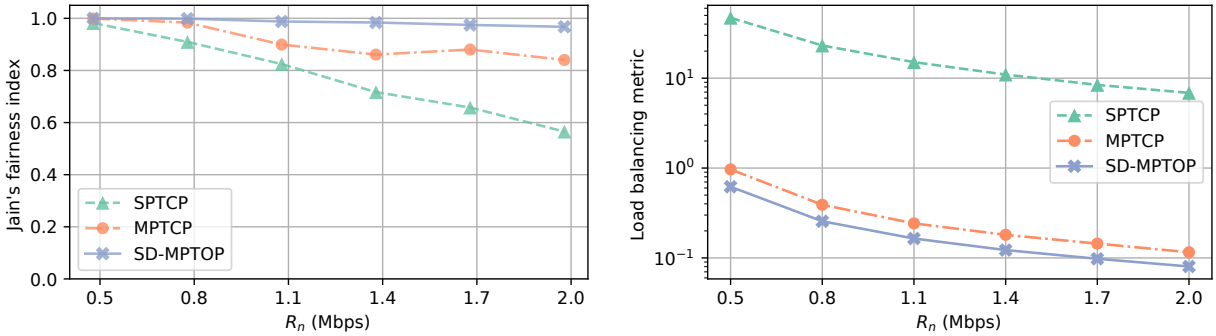


Figure 4.3: Throughput performance for the static test scenario.



(a) Jain's fairness metric.

(b) Load balancing metric.

Figure 4.4: Fairness and load balancing performance for the static test scenario.

we show the Jain's fairness index $\mathcal{J} = \frac{(\sum_n x_n)^2}{N \cdot \sum_n x_n^2}$ for each protocol versus R_n [3]. We confirm that due to the local view of SPTCP and MPTCP, there is higher level of bandwidth competition between $N1$ and $N2$ users. Consequently they fail to guarantee fairness in the bandwidth usage. `SD-MPTOP` on the other hand achieves $\mathcal{J} \approx 1$. Lastly, we measure the load at WiFi and LTE networks and compute the load balancing index defined based on our MCF

[3]Note that we denote $x_{n,s}$ as the actual throughput on link $l_{n,s}$, and $x_n = \sum_s x_{n,s}$

objective function:

$$\mathcal{LB} = \frac{1}{\sum_{n}\sum_{s\in\mathcal{S}} x_{n,s}} \left\{ \sum_{s\in\mathcal{S}} \frac{\sum_{n\in N_s} x_{n,s}}{BW_s - \sum_{n\in N_s} x_{n,s}} + \sum_{s\in\mathcal{S}}\sum_{n\in N_s} x_{n,s}\cdot D_s \right\}$$

The load balancing result is shown in Fig. 4.4b. We see that SD-MPTOP achieves the minimum $\mathcal{LB}$, i.e., the best load balancing provision in HetNet. SPTCP has the highest $\mathcal{LB}$ since mobile devices greedily congest WiFi networks.

4.4.2 Mobility Management Test Scenario

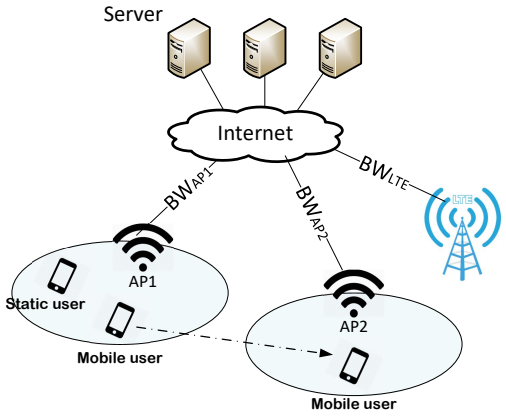


Figure 4.5: Emulation topology for the handover test scenario.

In this scenario, we add the movement of mobile devices and show the throughput performance in the presence of vertical handovers. The topology is shown in Fig. 4.5. Initially, both the static and mobile users are under the coverage of WiFi AP1. Then the mobile user starts to move towards AP2 during its transmission. The mobile user is equipped with both WiFi and LTE interfaces, while the static user uses only WiFi. The emulation parameters are: $BW_{AP1} = BW_{AP2} = 6Mbps, BW_{LTE} = 8Mbps$, $D_{LTE} = 50ms$, D_{AP1} and D_{AP2} change with respect to the location of the mobile user; $\mathcal{S} = \{AP1, AP2, LTE\}$; $R_n = 3Mbps$. The throughput results are shown in Fig. 4.6. We observe that the throughput of the static user is stable and is close to its demand throughout the three experiments. For the mobile user, using SPTCP means there are necessarily two interface switching: one from WiFi to LTE

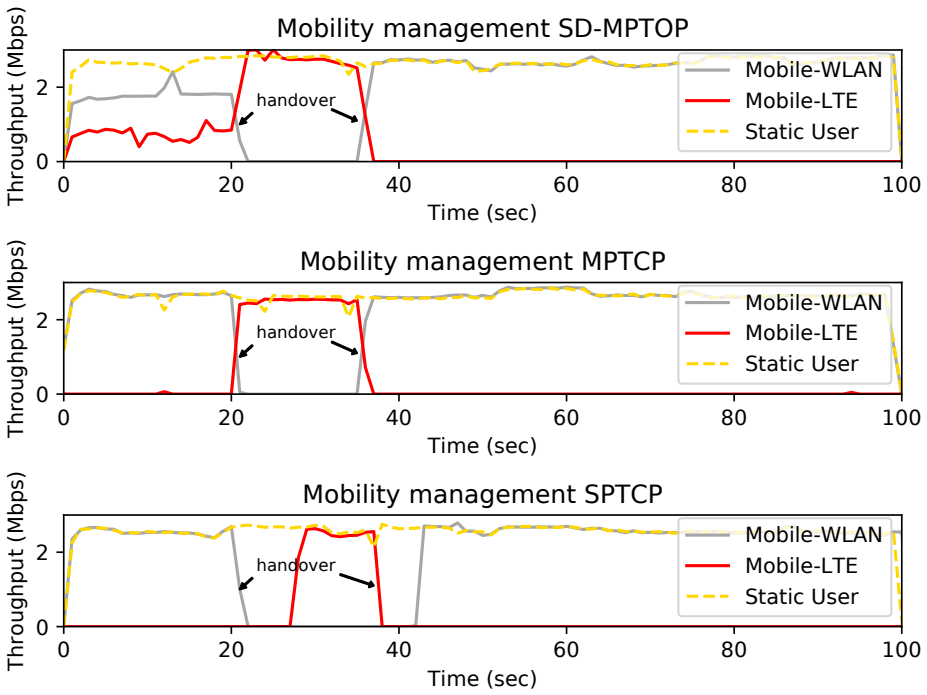


Figure 4.6: Mobility management performance for the handover test scenario.

(when leaving AP1) and the other from LTE to WiFi (when entering AP2). Therefore, the mobile user has to interrupt the ongoing TCP transmission and restarts using the alternative interface during its movement. MPTCP and SD-MPTOP however smoothly alternate the traffic between the two interfaces and do not interrupt the ongoing transmission. We note that the difference between MPTCP and SD-MPTOP is that, MPTCP schedules traffic based on the local view and it favors the interface with shorter RTT. As a consequence, before the first handover of the mobile user, most of its traffic is scheduled to go through the WiFi network, which saturates the WiFi network. In contrast, SD-MPTOP allocates only $2Mbps$ of bandwidth to the mobile user while the other $1Mbps$ goes to the LTE network, thanks to the FDM algorithm.

4.4.3 Dynamic Network Test Scenario

The last test scenario is a more sophisticated dynamic scenario. The topology is shown in Fig. 4.7. The HetNet contains 1 LTE base station and 2 WiFi APs. There are two types of mobile

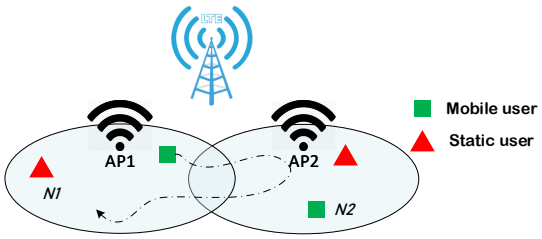


Figure 4.7: Emulation topology for the dynamic test scenario.

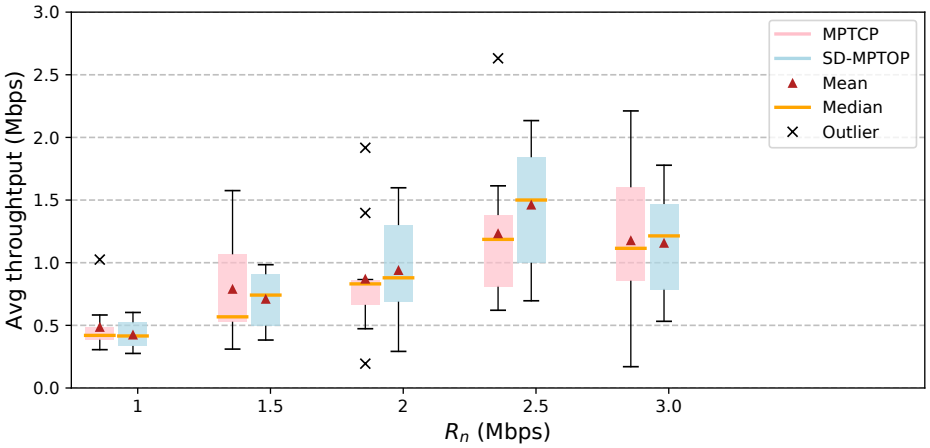


Figure 4.8: Throughput performance for the dynamic test scenario.

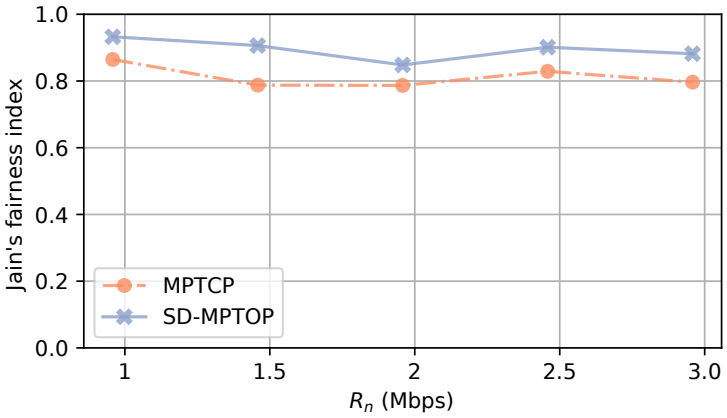


Figure 4.9: Fairness performance for the dynamic test scenario.

devices: type N_1 mobile devices who have one network interface and are statically connected to an AP, and type N_2 mobile devices who have dual connections to LTE and WiFi but are randomly moving between the two APs. Therefore, the network connectivity is frequently changing compared to the static test scenario. In the experiments, we use the following parameters: $BW_{WiFi} = 6Mbps, BW_{LTE} = 8Mbps, D_{LTE} = 50ms, \mathcal{S} = \{AP1, AP2, LTE\}$; $N_1 = N_2 = 5$. Similarly to the static scenario, R_n is set equal across the users and it gradually increases to saturate the network. Since in the handover scenario, we verify that SPTCP suffers from network disconnection during user movement leading to poor throughput performance in dynamic environments, we will focus on the comparison of MPTCP and `SD-MPTOP` in this third test scenario.

The throughput results are shown in Fig. 4.8. Compared to the static scenario, the variance in throughput distribution becomes larger. But overall, the average throughput of MPTCP and `SD-MPTOP` are comparable, and `SD-MPTOP` still has smaller variance. We also calculate the fairness metric using the Jain's fairness index. Results are shown in Fig. 4.9. Again, `SD-MPTOP` outperforms MPTCP by guaranteeing max-min fairness across the users.

CHAPTER 5

Opportunistic Traffic Offloading

In recent years, a lot of opportunistic traffic offloading approaches have been proposed. Some rely on DTN protocols to offload delay-tolerant data [LZ10, MW11], while others adopt the idea of collaborative content distribution [WCH14, PSS17, PBS14] to save the bandwidth of cellular systems.

In opportunistic traffic offloading, it has been commonly assumed that mobile users are cooperative and they comply with the offloading protocols. This assumption however does not always hold, especially considering the cost associated with the wireless transmission such as energy, bandwidth and storage consumption. Selfish users may choose not to contribute as it consumes their own resources. Due to selfishness, the number of contributors in the network is diminished and it results in the "tragedy of the commons" [Ost08].

In the past, credit or reputation-based systems have been developed [GA08][AML15] to stimulate cooperation. The idea is to reward cooperative actions by virtual payments or raising user's reputation. However, deploying these mechanisms incurs additional costs. In credit-based system, centralized management of currency and tamper-prof hardware need to be implemented, while in reputation-based system a network-wide reputation monitoring and update must take place. In addition to these cooperation enforcement schemes, a considerable amount of efforts have been devoted to game theoretic analysis of cooperation [LS12, Now06, PC99]. Among them, Evolutionary Game Theory (EGT) has proved to be an efficient approach to investigate cooperative behaviors [Wei97]. EGT is used to model an evolutionary (Darwinian) process where the more fit individuals pass on their strategies to more offspring and increase their representation in the population. The typical setup is the following: two strategies are concerned, Cooperator (C) or Defector (D); fitness is

assumed to be the payoff obtained from pairwise or group interactions defined in the form of social dilemma games; a reproduction (or strategy update) model that favors individuals with higher fitness; the predominant strategy at equilibrium state implies an evolutionarily stable strategy.

In this chapter, we first verify that opportunistic traffic offloading is positively correlated with the portion of cooperative users in the network, who are willing to contribute their resource either in term of relaying others' packets, or sharing contents. Having identified the importance of cooperation rate to opportunistic traffic offloading, we study the cooperation issue in Opportunistic Networks (OppNet[1]) using the EGT theory. It is revealed that the emergence of cooperation in OppNet highly depends on the structure of network. We also ask a series of research questions to understand the effect of user mobility patterns and offloading protocols on users' cooperative behaviors. Finally, we confirm through simulation that a high level of cooperation is likely to emerge in OppNet, which may significantly increase the offloading efficiency.

5.1 Correlation Between Offloading and Cooperation in OppNet

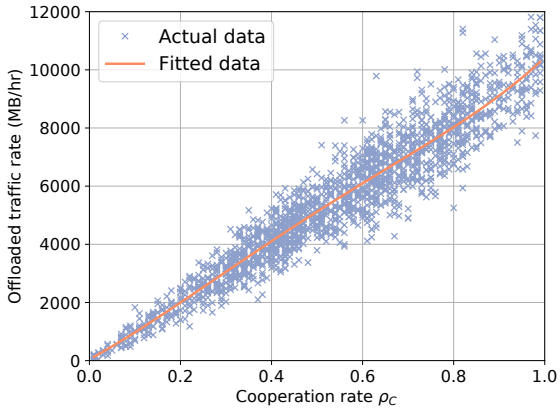


Figure 5.1: Traffic offloading and cooperation rate correlation in OppNet.

In OppNet, traffic offloading is carried out using D2D transmissions. Therefore, the

[1]In literature, OppNet is also called Mobile Social Networks (MSN). Therefore, throughout this chapter, we use OppNet and MSN interchangeably.

more users who participate in the opportunistic transmission, the more traffic OppNet can offload. To verify the correlation between traffic offloading and user participation in OppNet, we run a simulation experiment for the content distribution protocol [WCH14]. To simulate OppNet user movement, we use the SWIM [KMS10] mobility model. We vary the portion of cooperative users, who are willing to share their contents given the opportunity, from 0% to 100%, and measure the total size of contents that can be shared between mobile users in 1 hour. The result in Fig. 5.1 verifies the strong correlation between traffic offloading efficiency and the cooperation rate in OppNet.

In the following, we model DTN packet forwarding and content distribution offloading protocols as games, and then apply the EGT theory to study the cooperation rate in these games over OppNet.

5.2 Evolutionary Game Model for Opportunistic Traffic Offloading

We adopt the following standard EGT settings:

- The population is formed by the N mobile users.
- Every user chooses a pure strategy from the strategy set $\mathcal{S} = \{C, D\}$, where C stands for cooperation and D means to defect.
- Let p_C be the fraction of users taking strategy C. Since there are only two strategies in $\mathcal{S}$, p_C determines the strategy distribution across the population (as $p_D = 1 - p_C$). Therefore we use p_C to denote the current state of the population.
- The N users obtain payoff from the interactions with neighbors. The payoff depends on user's strategy, the strategy of their neighbors, as well as the specific offloading protocol. The payoff of a particular user i who interacts with a set of neighbors N_i, among which k_i take strategy C, is denoted by $f_i(s_i, N_i, k_i, \mathcal{G}) = f_{i|N_i,k_i,\mathcal{G}}(s_i)$, where $\mathcal{G}$ is the offloading protocol.
- Every player updates strategy based on the payoff obtained from the current interaction.

This update in turn changes the population state, which will be carried on to the next round of interaction. The evolution process repeats for many times until the state reaches equilibrium.

For different offloading protocol $\mathcal{G}$, the payoff function can be defined differently. For example, in content distribution protocol, the payoff is a function of the amount of contents of interests obtained from others, while in DTN packet forwarding, the payoff is a function of the number packets successfully delivered with the help of neighbors. It also depends on the individual strategy, as a cooperative user contributes its resource so there will be a positive cost associated with it, which decreases the payoff. In general, we can define the payoff of user i as follows:

$$f_{i|N_i,k_i,\mathcal{G}}(D) = g(N_i, k_i, \mathcal{G}),$$
$$f_{i|N_i,k_i,\mathcal{G}}(C) = f_{i|N_i,k_i,\mathcal{G}}(D) - C_i \cdot \varepsilon.$$

Here, function $g(\cdot)$ represents the utility with respect to the obtained benefit, C_i is the amount of consumed resource, and ε is the unit cost.

Next, we formally define the EGT game model for DTN packet forwarding and resource sharing based protocols.

5.2.1 DTN Packet Forwarding Game

We formulate the two-hop DTN packet forwarding protocol as an evolutionary multi-player game termed DMG. The payoff function is equal to the successful message delivery rate with the help of cooperative neighbors, minus the total cost to help neighbors. In the rest of this subsecion, we fix $\mathcal{G}$ as DTN packet forwarding and drop it from the payoff notation.

5.2.1.1 DTN network model

We consider a DTN network of N mobile users. The DTN network is sparse and users are disconnected most of the time in their movement. Mobile users communicate only when they are in proximity to each other. We refer to such opportunity of communication between two users as *contact*. The inter-contact times (ICT) between neighboring nodes are

Table 5.1: Payoff of DMG on K-regular Graph

Number of C neighbors k	0	...	K
$f_{K,k}(C)^2$	$1-e^{-\lambda\tau}-K\varepsilon$	...	$1-e^{-\lambda\tau}\left(\frac{1+\lambda\tau}{e^{\lambda\tau}}\right)^K-K\varepsilon$
$f_{K,k}(D)$	$1-e^{-\lambda\tau}$	...	$1-e^{-\lambda\tau}\left(\frac{1+\lambda\tau}{e^{\lambda\tau}}\right)^K$

independently and identically distributed (iid) exponential variables with parameter $\lambda > 0$. This assumption has been justified in previous work [HNA06].

In the network, every user is a source that constantly generates messages destined to some destination. Due to the intermittent connectivity and a lack of instantaneous end-to-end paths, messages are transmitted in a store-and-forward manner. Here, we assume that a two-hop routing scheme is employed. The source either directly delivers the message to the destination, or replicates the message to an intermediate relay. The relay will deliver the message when it meets the destination. Therefore every message reaches the destination in at most 2 hops. We assume that each message is relevant for a time interval of length at most τ. For timely delivery, the source seeks help from as many relays as possible. When the number of relays is equal to k, the delivery time is

$$T = \min\{T_{sd}, T_{sr_1} + T_{r_1 d}, \ldots, T_{sr_k} + T_{r_k d}\}$$

where T_{sd} is the time for direct delivery, and $T_{sr_k} + T_{r_k d}$ is the time for delivery via relay k. Based on the assumption of iid exponential ICT, the probability of timely delivery is

$$\begin{aligned} \mathbb{P}_k(T < \tau) &= 1 - \mathbb{P}(T_{sd} \geq \tau) \prod_{i=1}^{k} \mathbb{P}(T_{sr_i} + T_{r_i d} \geq \tau) \\ &= 1 - e^{-\lambda\tau}\left(\frac{1+\lambda\tau}{e^{\lambda\tau}}\right)^k \end{aligned} \tag{5.1}$$

where we use the fact that $T_{sr_i} + T_{r_i d}$ is $Erlang(2)$, and $\mathbb{P}(T_{sr_i} + T_{r_i d} \geq \tau) = \frac{1+\lambda\tau}{e^{\lambda\tau}}$.

5.2.1.2 DTN Multi-player Game (DMG) model

With the previous EGT setting, we can formulate the two-hop DTN packet forwarding problem as a multi-player game (DMG). The payoff of an individual user depends on the group

of users it interacts with, which we call its neighborhood. For user i whose neighborhood is N_i, among which k_i are cooperators, the payoff is defined as

$$\begin{aligned} f_{i|N_i,k_i}(D) &= \gamma_i \mathbb{P}_{k_i}(T < \tau) \\ f_{i|N_i,k_i}(C) &= f_{i|N_i,k_i}(D) - \sum_{j \in N_i} \gamma_j \varepsilon \end{aligned} \tag{5.2}$$

We assume that users playing C always relay packets for all neighbors. Here γ_i is the message rate at source i, and ε represents the cost (e.g. energy consumption, bandwidth, memory usage, etc.) for relaying one message. The defined payoff for cooperators equals to the benefit obtained from relays minus the contribution to others. We assume homogeneous packet rate in the population $\gamma_i = 1$. An implicit assumption made here is that the k_i cooperative neighbors can all reach the destination of user i. While this assumption is acceptable in many population structures, it might not be true when users are randomly connected. We will revisit this issue in 5.3.3.3.

5.2.1.3 DMG on K-regular graph

Now we focus on the evolution of DMG on K-regular graph. Each user has exactly K neighbors, and plays a single $(K+1)$-player game per round. The payoff obtained from the $(K+1)$-player game is shown in Table 5.1.

After users obtain their payoffs from $(K+1)$-player DMG, the population state is updated by a Moran death-birth (DB) process. Specifically, a random individual i is chosen to die and its K neighbors compete to take its place. The probability that a particular neighbor j wins the competition is proportional to its fitness, defined as $\mathcal{F}_j = 1 - \omega + \omega \times f_{K,k_j}(s_j)$ where $\omega \ll 1$. Therefore, the probability that a cooperator wins is $\sum_{j \in K_i^C} \mathcal{F}_j / \sum_{j \in K_i} \mathcal{F}_j$, where K_i is the set of neighbors of i and K_i^C is the set of cooperative neighbors. The Moran DB process stochastically updates the population state.

[2]As payoff on K-regular graph is symmetric, we drop the i index from the payoff notation $f_{i|N_i,k_i}$.

5.2.2 Opportunistic Resource Sharing Game

Resource sharing applications have received increasing attention in recent years because of their efficiency in challenged network environments. It generalizes a range of opportunistic traffic offloading protocols in Mobile Social Networks (MSN) such as content distribution [PSS17] or D2D communications [IGH17], as well as other sharing based applications [GDM12]. Here, we formulate opportunistic resource sharing in MSN as an evolutionary multi-player game termed OppShare. In the rest of this subsecion, we fix $\mathcal{G}$ as OppShare and drop it from the payoff notation.

We define the OppShare game model to imitate how resource sharing agreement can be reached between users in MSN. Initially, every user chooses a strategy that is either cooperation (C) or defection (D). Resource sharing only occurs when two users are in contact. To determine the resource sharing agreement between user i and j, we define the following two processes:

Matching process: Determine the need or supply of i and j for resource to see if they can be matched up for resource sharing. For example, if one user has an abundant residual cellular data plan and agrees to share some data with a nearby user who needs it, they can establish bandwidth sharing until one walks away from the other. More formally, we generate independent random variables $Need_i, Need_j$ from uniform distribution in $[0, 1]$. Suppose $Need_i < Need_j$, then if $|Need_i - Need_j| > \delta_N$, and the strategy of i is C, then resource sharing will be established between the i and j. Note that $\delta_N \in [0, 1]$ is a tunable constant.

Sharing process: Suppose i agrees to share resource with j, and its residual resource is $\mathcal{R}_i$. The amount of resource i shares is $\mathcal{S}_i = \min\{t_{i,j}, \mathcal{R}_i/\kappa\}$, where $t_{i,j}$ is the duration of this contact, and κ determines the maximum portion of residual resource i is willing to share. After this contact, i deducts $\mathcal{S}_i$ from $\mathcal{R}_i$ and adds $\mathcal{S}_i$ to its consumed resource $\mathcal{C}_i$, while j adds $\mathcal{S}_i$ to its obtained resource $\mathcal{B}_j$.

Allow users to move and interact for one round with time length $\mathcal{T} \gg 1$. The fitness of each user is calculated by

$$fitness(i) = 1 - \omega + \omega \cdot \underbrace{\{\log(1 + \mathcal{B}_i) - \mathcal{C}_i \cdot \varepsilon\}}_{f_{i|N_i,k_i}}, \tag{5.3}$$

where $\omega \in [0, 1]$, and ε denotes the cost of resource sharing. Here the payoff increases with $\mathcal{B}_i$ but has diminishing returns, and it decreases linearly with $\mathcal{C}_i$ and ε. Now users evolve their strategies according to the Moran death-birth process [DG17a]. We randomly pick one user whose strategy will be replaced by one of its neighbors. With probability

$$p = \sum_{j \in N_i^C} fitness\,(j) / \sum_{j \in N_i} fitness\,(j), \tag{5.4}$$

the strategy changes to C, where N_i denotes the set of neighbors of i and N_i^C is the subset of cooperative neighbors. The user zeros out $\mathcal{B}_i, \mathcal{C}_i, N_i$ and carries on the evolved strategy to the next round. The evolution continues until the population becomes either full cooperation or full defection.

5.3 Emergence of Cooperation in Structured Populations

In this section, we study the emergence of cooperation in MSN using the EGT framework. Previously, we formulate the opportunistic offloading as a multi-player game, and propose two game models for DTN packet forwarding game (DMG) and opportunistic resource sharing (OppShare). Here, we use DMG as an example to proceed with our study.

Two strategies can be adopted by a player: cooperation (C) or defection (D). In any mixed population, the average payoff of defectors is always higher than the cooperators, i.e. defection dominates cooperation. Therefore, individual has no incentive to cooperate. Indeed, we obtain that the ESS in DMG game is defection. This result, however, is restricted to networks of *infinite number of users* and *well-mixed interactions*. The two assumptions can be challenged by the following facts: *1)* some well-known mobility traces contain only tens to hundreds of nodes [KLV10, TLB09]; *2)* interactions in MSN are correlated rather than random as in the well-mixed case [HCY11]. Therefore, the result may not be able

to accurately approximate the outcome of real-world opportunistic offloading. A more authentic setting would be a evolutionary game over finite and structured populations, where interactions are confined within a group of similar entities.

There has been much interest in studying evolutionary games in structured populations [DHD14, AN14]. Recent developments in [OHL06, PWA16] revealed that cooperation can be promoted when the evolution takes place on lattice or random graphs. The condition for cooperators to dominate defectors is given by the "σ-rule", which depends not only on the payoffs but also on the underlying structures. An analytical approximation of the "σ-rule" is derived in [PWA16] for regular graph of arbitrary degree. Motivated by this line of research, we make the following contributions:

- Observing that the ESS of DMG in infinite and well-mixed population is defection, we turn our attention to the structured populations. A sufficient condition is derived under which cooperation is favored over defection in DMG on regular graphs with arbitrary degree K. Specifically, cooperation dominates when the cost to help a neighbor scales as $O(\frac{1}{K^2})$ for large K.
- The theoretical result is validated through numerical simulations on three types of graphs: ring, random regular graph, and community-based graph. Assumptions and implications of simulation results are discussed.

Our result substantially differs from previous work [UBG03, FHB06] by showing that under the assumption of natural selection and structured populations, cooperation can be favored by the system even without external incentive schemes.

5.3.1 ESS of DMG in Infinite and Well-mixed Populations

We first consider a well-mixed DTN network with infinite population, where everyone contacts $N-1$ random users with equal frequency λ in each round. The expected individual

payoff can be written as

$$
\begin{aligned}
f_i(D, p_C) &= \sum_{k=0}^{N-1} \mathbb{P}(k_i = k) f_{i|N_i,k_i}(D) \\
&= \sum_{k=0}^{N-1} \binom{N-1}{k} p_C^k (1-p_C)^{N-1-k} \mathbb{P}_k(T < \tau) \\
f_i(C, p_C) &= f_i(D, p_C) - (N-1)\varepsilon
\end{aligned}
$$

We are interested in the fraction of cooperators p_C when the system is at equilibrium state. The equilibrium point of EGT is given by the Evolutionarily Stable Strategy (ESS). More formally, we have the following definition:

Definition 5.1. *A population state p_C^* is an ESS if for any mutant strategy $mut \neq p_C^*$, there exists some $\epsilon_M \in (0,1)$, which may depend on mut, such that $\forall \epsilon \in (0, \epsilon_M)$ the following holds,*

$$
F(p_C^*, \epsilon \cdot mut + (1-\epsilon) p_C^*) > F(mut, \epsilon \cdot mut + (1-\epsilon) p_C^*) \tag{5.5}
$$

Due to the fact that $f_i(D, p_C) > f_i(C, p_C), \forall p_C$ when $\varepsilon > 0$, we immediately come to the conclusion that $p_C^* = 0$ is an ESS to DMG. This is not surprising as the game under study is the N-player Prisoner's Dilemma with non-linear payoff function, whose Nash Equilibrium is defection. This pessimistic result implies that incentive scheme has to be introduced by the network operator in order to stimulate cooperation. Previous papers have come to the same conclusion [FHB06, UBG03].

It is worth noting that the above ESS applies to a well-mixed population. In reality, the mobility patterns in DTN have strong location preference [GHB08, HSL11]. Therefore, the neighborhood of each user is restricted to a small set of $friend$ nodes. Interactions with these friends can be aggregated into a contact graph. The evolutionary dynamics on graphs has attracted lots of attentions. Recent progress in [PWA16] revealed the condition for cooperation to be favored on K-regular graph. Unlike in infinite and well-mixed populations, preference for cooperation or defection on graph structures is measured using the metric *fixation probability*. The fixation probability of cooperation, denoted by ρ_C, is the probability that a single cooperator out of $N-1$ defectors turns the whole population into cooperators,

i.e. the probability of convergence to complete cooperation. Similarly, the fixation probability of defection, denoted by ρ_D, is the probability of convergence to complete defection from a single defector. If $\rho_C > \rho_D$, then the long-term frequency of cooperators is higher than the frequency of defectors [DHD14], or equivalently, C is favored over D by nature selection. [PWA16] analytically showed the condition for $\rho_C > \rho_D$.

5.3.2 Evolution of DMG on K-regular Graph

From [PWA16], the condition for $\rho_C > \rho_D$ in DMG on K-regular graph is given by the following σ-rule:

$$\sum_{k=0}^{K} \sigma_k d_k > 0 \tag{5.6}$$

where $\sigma_0, ..., \sigma_K$ are the normalized structure coefficients characterizing the underlying population structure, and $d_k = f_{K,k}(C) - f_{K,K-k}(D), \forall k = 0, 1, ..., K$.

The structure coefficients of different graphs have been derived. Here we concentrate on two cases: $(N-1)$-regular graph, which corresponds to a well-mixed population of finite size N, and the general K-regular graph where $K \ll N$. We will see how the structure coefficients promote cooperation in K-regular graph compared to the well-mixed case. A sufficient condition under which cooperation is favored over defection in DMG on K-regular graph is then derived.

5.3.2.1 The σ-rule in well-mixed population

The structure coefficients in well-mixed population is given by the following lemma.

Lemma 5.1. *[GT10] For regular graph of degree $K = N - 1$, the structure coefficients are*

$$\sigma_k^{wm} = \begin{cases} \frac{1}{N-1}, & 0 \leq k < N-1 \\ 0, & k = N-1 \end{cases} \tag{5.7}$$

The above σ-rule has the same implication as ESS in well-mixed population, as reflected in the following theorem.

Theorem 5.1. *In a finite and well-mixed population of size N, where individual plays the N-player DMG with everyone else, $\rho_C < \rho_D$ always holds.*

Proof. Plugging equation (5.7) into condition (5.6) we have

$$\begin{aligned}\sum_{k=0}^{N-1}\sigma_k^{wm}d_k &= \frac{1}{N-1}(\sum_{k=0}^{N-2}f_{N-1,k}(C) - \sum_{k=1}^{N-1}f_{N-1,k}(D)) \\ &= \frac{1}{N-1}(f_{N-1,0}(D) - f_{N-1,N-1}(D)) - (N-1)\varepsilon \\ &= \frac{1}{(N-1)e^{\lambda\tau}}(\left(\frac{1+\lambda\tau}{e^{\lambda\tau}}\right)^{N-1} - 1) - (N-1)\varepsilon \\ &< 0.\end{aligned}$$

which indicates that $\rho_C < \rho_D$. □

Remark 5.1. *Compared to Theorem 5.1, the ESS of $p_C^* = 0$ we obtain in 5.3.1 applies to an infinite and well-mixed population where everyone is involved in DMG with N random neighbors per round. However, in both cases defection is favored over cooperation. This is because the well-mixed structure facilitates the invasion of defectors to cooperators [PGS13]. Cooperators are easily exploited when facing defectors.*

5.3.2.2 The σ-rule on K-regular graph

We have seen that the σ-rule does not favor cooperation in well-mixed population. In contrast, both [PWA16] and [OHL06] show that the σ-rule on K-regular graph favors cooperation when $K \ll N$. We introduce the following lemma from [PWA16].

Lemma 5.2. *[PWA16] For regular graph of degree $K \geq 3$ and $K \ll N$, the structure coefficients can be approximated by*

$$\sigma_k = \frac{(K-2)^{K-1-k}}{(K+2)(K+1)K^2}\sum_{l=0}^{K-1}(K-l)\{[K^2-(K-2)l]\upsilon_{l,k,K} + [2K+(K-2)l]\varsigma_{l,k,K}\} \quad (5.8)$$

$\forall 0 \leq k \leq K$, *where*

$$\upsilon_{l,k,K} = \binom{K-1-l}{K-1-k}\frac{1}{(K-1)^{K-1-l}} + \binom{l}{K-k}\frac{K-2}{(K-1)^l},$$

$$\varsigma_{l,k,K} = \binom{K-1-l}{K-k}\frac{K-2}{(K-1)^{K-1-l}} + \binom{l}{K-1-k}\frac{1}{(K-1)^l}.$$

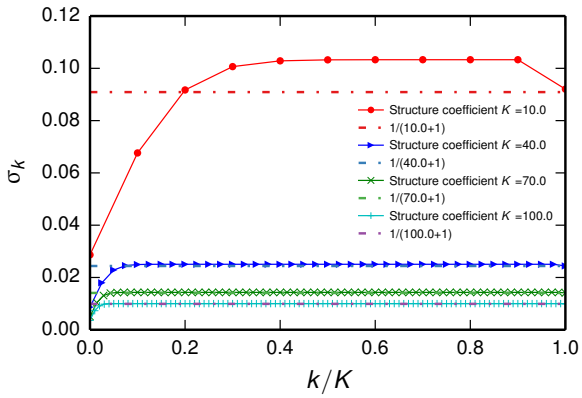


Figure 5.2: Structure coefficients on K-regular graph.

Due to the complex form of equation (5.8), we plot the structure coefficients of K-regular graph for different values of K. As shown in Fig. 5.2, the structure coefficients σ_k asymptotically approach $1/(K+1)$ when $K \gg 1$ and $k \gg 1$. In the meantime, σ_0 and σ_1 are always smaller than $1/(K+1)$. This quasi-increasing property clearly distinguishes K-regular graph from well-mixed graph, whose structure coefficient is given in equation (5.7)[3]. The concept that characterizes this difference is the *containment order* [PWT16]. Population structure $\mathcal{P}_1$ is greater than population structure $\mathcal{P}_2$ in containment order, if the set of games for which cooperation is favored under $\mathcal{P}_2$ is contained in the set of games for which cooperation is favored under $\mathcal{P}_1$. A sufficient condition for $\mathcal{P}_1$ to be greater than $\mathcal{P}_2$ in containment order is that the difference in structure coefficients under $\mathcal{P}_1$ and $\mathcal{P}_2$, $\sigma_k^1 - \sigma_k^2$, $0 \leq k \leq K$, has exactly one sign change from $-$ to $+$. Using this property, we can easily obtain that K-regular graph is greater than the well-mixed population in containment order. Therefore, cooperation evolves more easily in K-regular graph than in the well-mixed population.

5.3.2.3 Evolution of DMG on K-regular graph

Next, we study the condition under which cooperators are favored on K-regular graph. The only assumption we make on the structure coefficients is the quasi-increasing property:

[3]The general expression for structure coefficients of well-mixed populations with group size equal to $K+1$ can be found in [PWA16].

$\sigma_k - \sigma_{K-k} \geq 0$ when $k \geq K^* \triangleq \lfloor \frac{K}{2} \rfloor$, which is obvious from Fig. 5.2. This quasi-increasing property on K-regular graph results its higher containment order. Therefore, condition (5.6) has a chance to be satisfied when DMG evolves on a K-regular graph. In the following, we present the main result of this section: a sufficient condition for cooperation to be favored on K-regular graph. Specifically, based on Lemma 5.2, we show that $\rho_C - \rho_D > 0$ when the cost ε is sufficiently small.

Theorem 5.2. *Given parameters $\lambda\tau$ and K, there exists some constant $\epsilon_{\lambda\tau,K} \in (0,1)$ such that the cooperators are favored in DMG on K-regular graph if*

$$\begin{aligned} \varepsilon &< \epsilon_{\lambda\tau,K} \frac{1}{K^2} \sim O(\frac{1}{K^2}), \\ \epsilon_{\lambda\tau,K} &= e^{-\lambda\tau}\{(1-\frac{1}{e})[1-\left(\frac{1+\lambda\tau}{e^{\lambda\tau}}\right)^K] \\ &\quad + (1-\frac{2}{e})[\frac{1+\lambda\tau}{e^{\lambda\tau}} - \left(\frac{1+\lambda\tau}{e^{\lambda\tau}}\right)^{K-1}]\}, \end{aligned} \tag{5.9}$$

when $N \gg K \gg 1$.

Proof. First consider the following structure coefficients σ_k:

Case $k = 0$

$$\upsilon_{0,0,K} = \varsigma_{K-1,0,K} = \frac{1}{(K-1)^{K-1}},$$

and $\upsilon_{l,0,K} = \varsigma_{l,0,K} = 0$ elsewhere. Therefore

$$\begin{aligned} \sigma_0 &= \frac{(K-2)^{K-1}}{(K+2)(K+1)K^2} \frac{K^3 + [2K + (K-2)(K-1)]}{(K-1)^{K-1}} \\ &= \frac{(K-2)^{K-1}}{(K-1)^{K-1}}[\frac{K}{(K+2)(K+1)} + \frac{[2K+(K-2)(K-1)]}{(K+2)(K+1)K^2}] \\ &= (1-\frac{1}{K-1})^{K-1}[\frac{K}{(K+2)(K+1)} + \frac{[2K+(K-2)(K-1)]}{(K+2)(K+1)K^2}] \\ &\sim \frac{1}{e}\frac{1}{K} + O(\frac{1}{K^2}) \sim \frac{1}{e}\frac{1}{K} \end{aligned} \tag{5.10}$$

Case $k = 1$

$$\begin{aligned} &\upsilon_{0,1,K} = \upsilon_{1,1,K} = \varsigma_{K-2,1,K} = \varsigma_{K-1,1,K} = \tfrac{1}{(K-1)^{K-2}}, \\ &\upsilon_{K-1,1,K} = \varsigma_{0,1,K} = \tfrac{K-2}{(K-1)^{K-2}}, \end{aligned}$$

and $\upsilon_{l,1,K} = \varsigma_{l,1,K} = 0$ elsewhere. Therefore

$$\begin{aligned}\sigma_1 = \frac{(K-2)^{K-2}}{(K+2)(K+1)K^2}\{&\frac{K^3+(K-1)[K^2-(K-2)]}{(K-1)^{K-2}}\\ &+\frac{2[2K+(K-2)^2]+[2K+(K-2)(K-1)]}{(K-1)^{K-2}}\\ &+\frac{2K^2(K-2)+[K^2-(K-2)(K-1)](K-2)}{(K-1)^{K-1}}\}\\ &\sim \frac{2}{e}\frac{1}{K}+O(\frac{1}{K^2})\sim\frac{2}{e}\frac{1}{K}\end{aligned} \tag{5.11}$$

Case $k = K-1$

$$\begin{aligned}\upsilon_{l,K-1,K} &= \tfrac{1}{(K-1)^{K-1-l}} + l\tfrac{K-2}{(K-1)^l},\\ \varsigma_{l,K-1,K} &= (K-1-l)\tfrac{K-2}{(K-1)^{K-1-l}} + \tfrac{1}{(K-1)^l},\end{aligned}$$

$$\begin{aligned}\sigma_{K-1} = \frac{1}{(K+2)(K+1)K^2}\sum_{l=0}^{K-1}&(K-l)\{[K^2-(K-2)l]\upsilon_{l,K-1,K}\\ &+[2K+(K-2)l]\varsigma_{l,K-1,K}\}\\ \sim \frac{1}{\underline{K^4}}\{&\frac{K^3}{(K-1)^{K-1}}+\underline{(K-1)}[\underline{K^2}-(K-2)][\frac{1}{(K-1)^{K-2}}+\frac{K-2}{\underline{K-1}}]\\ &+O(K^2)\}+\frac{1}{K^4}\{2K^2[\frac{(K-1)(K-2)}{(K-1)^{K-1}}+1]+...\\ &+2[2K+(K-2)^2][\frac{K-2}{K-1}+\frac{1}{(K-1)^{K-2}}]\\ &+[2K+(K-2)(K-1)]\frac{1}{(K-1)^{K-1}}\}\sim\frac{1}{K}\end{aligned} \tag{5.12}$$

[4]

Case $k = K$

$$\begin{aligned}\upsilon_{l,K,K} &= \tfrac{K-2}{(K-1)^l},\\ \varsigma_{l,K,K} &= \tfrac{K-2}{(K-1)^{K-1-l}},\end{aligned}$$

$$\begin{aligned}\sigma_K = \frac{1}{(K+2)(K+1)K^2}\sum_{l=0}^{K-1}&(K-l)\{\frac{[K^2-(K-2)l]}{(K-1)^l}\\ &+\frac{[2K+(K-2)l]}{(K-1)^{K-1-l}}\}\sim\frac{1}{K}.\end{aligned} \tag{5.13}$$

[4]The underlined terms yield the limitation of $\frac{1}{K}$.

Now consider the condition (5.6)

$$\begin{aligned}
&\sum_{k=0}^{K}\sigma_k d_k = \sum_{k=0}^{K}\sigma_k[f_{K,k}(C) - f_{K,K-k}(D)] \\
&= \sum_{k=0}^{K}\sigma_k[f_{K,k}(D) - f_{K,K-k}(D)] - K\varepsilon \\
&\overset{(a)}{=} \sum_{k=K^*+1}^{K}\sigma_k[f_{K,k}(D) - f_{K,K-k}(D)] \\
&\qquad + \sum_{k=0}^{K^*}\sigma_k[f_{K,k}(D) - f_{K,K-k}(D)] - K\varepsilon \\
&\overset{(b)}{=} \sum_{k=K^*+1}^{K}(\sigma_k - \sigma_{K-k})[f_{K,k}(D) - f_{K,K-k}(D)] - K\varepsilon \\
&\overset{(c)}{\geq} (\sigma_K - \sigma_0)[f_{K,K}(D) - f_{K,0}(D)] \\
&\qquad + (\sigma_{K-1} - \sigma_1)[f_{K,K-1}(D) - f_{K,1}(D)] - K\varepsilon,
\end{aligned}$$

where in equation (a) we split the summation at $K^* = \lfloor \frac{K}{2} \rfloor$, (b) is due to the fact that $b_k = -b_{K-k}$ where $b_k = f_{K,k}(D) - f_{K,K-k}(D)$, and (c) is due to $b_k \geq 0$ and $\sigma_k - \sigma_{K-k} \geq 0$ when $K^* + 1 \leq k \leq K$.

Replacing the coefficients with values obtained from equation (5.10)-(5.13), we have

$$\begin{aligned}
\sum_{k=0}^{K}\sigma_k d_k \geq \frac{1}{K}e^{-\lambda\tau}\{(1-\frac{1}{e})[1-\left(\frac{1+\lambda\tau}{e^{\lambda\tau}}\right)^{K}] \\
+ (1-\frac{2}{e})[\frac{1+\lambda\tau}{e^{\lambda\tau}} - \left(\frac{1+\lambda\tau}{e^{\lambda\tau}}\right)^{K-1}]\} - K\varepsilon
\end{aligned} \tag{5.14}$$

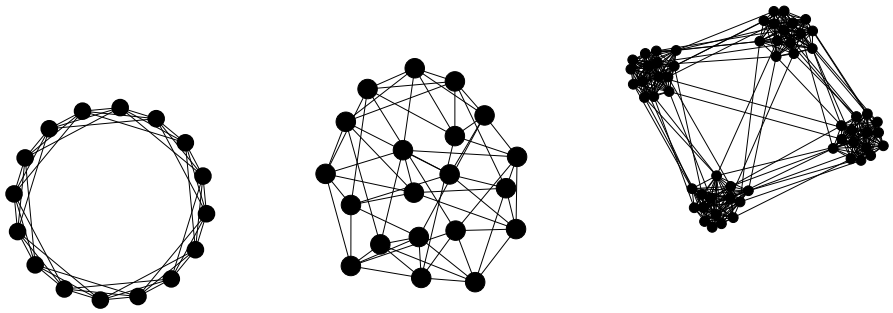
Therefore a sufficient condition for $\sum_{k=0}^{K}\sigma_k d_k > 0$ is

$$\begin{aligned}
\varepsilon < \frac{1}{K^2}e^{-\lambda\tau}\{(1-\frac{1}{e})[1-\left(\frac{1+\lambda\tau}{e^{\lambda\tau}}\right)^{K}] \\
+ (1-\frac{2}{e})[\frac{1+\lambda\tau}{e^{\lambda\tau}} - \left(\frac{1+\lambda\tau}{e^{\lambda\tau}}\right)^{K-1}]\} \sim O(\frac{1}{K^2}).
\end{aligned} \tag{5.15}$$

□

5.3.3 Numerical Simulations

In this subsection, we verify the theoretical finding with numerical simulations. First, we introduce the simulation setup. Three different population structures are considered: circular, random, and community-based graphs. Evolutionary dynamics of DMG is then simulated over the structured populations. We investigate how the difference in the fixation probability

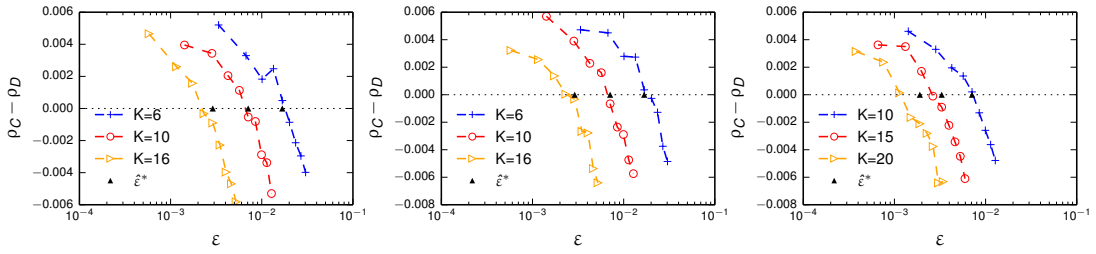


(a) Ring graph.

(b) Random regular graph.

(c) Community-based graph.

Figure 5.3: Population structures under study.



(a) Circular, $\lambda\tau = 0.1$, $\omega = 0.7$.

(b) Random, $\lambda\tau = 0.1$, $\omega = 0.7$.

(c) Community, $\lambda\tau = 0.1$, $\omega = 0.7$.

(d) Circular, $\lambda\tau = 1$, $\omega = 0.5$.

(e) Random, $\lambda\tau = 1$, $\omega = 0.5$.

(f) Community, $\lambda\tau = 1$, $\omega = 0.5$.

Figure 5.4: Simulation results for $\rho_C - \rho_D$.

$\rho_C - \rho_D$ changes with respect to relevant parameters. Assumptions, results and implications are discussed as well.

5.3.3.1 Simulation setup

To assess the theoretical approximation in Theorem 5.2, we implement the evolutionary algorithm of DMG on three different graphs[5]:

- **Ring:** Deterministic graph where nodes are placed in a circle. Every node is connected to $K/2$ nodes to the left, and $K/2$ nodes to the right.
- **Random Regular Graph:** Random graph where every node has the same degree K.
- **Community-based Graph:** Random graph that has high modularity coefficient and can be divided into communities/modules by community detection algorithms. Node has dense connections inside their own community and sparse connections to other communities.

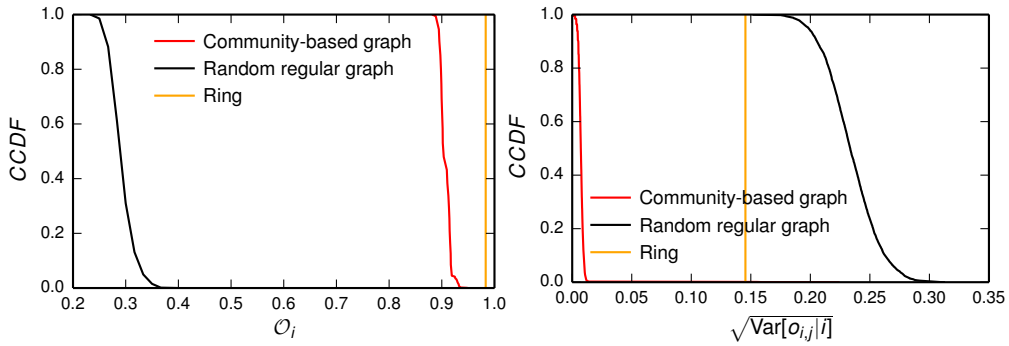
Examples of the graphs are shown in Fig. 5.3. In ring and random regular graph, every node is connected to exactly K neighbors. In community-based graph, as most edges are connecting nodes that belong to the same community (some nodes may have edges pointing to other random communities), the degree of each node is close to the community size. We keep the size of all communities equal, so that nodes have roughly the same degree $\overline{K} \simeq K$.

Unless otherwise specified, we use the following simulations parameters: $N = 100$, $6 \leq K \leq 20$, $\lambda\tau = 0.1$ (infrequent contact) or 1 (frequent contact). The fixation probability of cooperators ρ_C is calculated as the fraction of simulation runs in which a single cooperator turned the whole population to cooperators out of 10^6 runs. The fixation probability of defectors ρ_D is calculated similarly. Next, we present simulation results for $\rho_C - \rho_D$ under different structure and parameter conditions.

5.3.3.2 Simulation results

Simulation results in different graphs are shown in Fig. 5.4. Note that when computing the fitness $\mathcal{F}_i = 1 - \omega + \omega \times f_{K,k_i}(s_i)$, we do not follow the requirement $\omega \ll 1$ as commonly

[5]The graphs are generated using Matlab Tools for Network Analysis. Available: http://strategic.mit.edu/downloads.php?page=matlab_networks



(a) CCDF of $\mathcal{O}_i$. (b) CCDF of $\sqrt{\mathrm{Var}[o_{i,j}|i]}$.

Figure 5.5: Neighborhood similarity statistics in graph structures.

assumed in Moran process. The reason lies in the payoff function of DMG. In common multi-player games such as the N-player prisoner's dilemma or volunteer's dilemma [PWA16], the maximum individual payoff is $Kb - c \simeq K(K-1)c$ where $b \simeq Kc$ is the benefit and $c = 1$ is the cost for cooperators. The payoff becomes much larger than the baseline fitness $1 - \omega$ when $K \gg 1$. In our case, the maximum individual payoff $f_{K,K}(C) < f_{K,K}(D) < 1$ always holds. Therefore we adopt $\omega = 0.7$ when $\lambda\tau = 0.1$, and $\omega = 0.5$ when $\lambda\tau = 1$ to make the payoff and baseline fitness $1 - \omega$ comparable.

From Fig. 5.4, the difference in fixation probabilities decreases with cost ε. Moreover, this difference has a unique zero point at a critical cost ε^*, and $\rho_C > \rho_D$ holds when $\varepsilon < \varepsilon^*$. It means cooperation is favored in the structured populations as long as the cost to relay a message is sufficiently small. This finding on graph structures is in sharp contrast to the case of well-mixed population. Recall that defection is always favored in the latter case.

Comparing the simulation results with the critical value $\hat{\varepsilon}^* = \epsilon_{\lambda\tau,K} \cdot \frac{1}{K^2}$ predicted by equation (5.9), we see that $\hat{\varepsilon}^*$ agrees with the actual zero point ε^* of the simulated curves very well for all the three graph structures, although ε^* seems to deviate from $\hat{\varepsilon}^*$ (to the left) for large $\lambda\tau$, i.e. when mobile nodes encounter each other more frequently. Moreover, the theoretical $\hat{\varepsilon}^*$ slightly overestimates the actual zero point in community-based graph. This may be partially due to the fact that in community-based graph, nodes do not have uniform degree K.

5.3.3.3 Assumption on destination reachability

As mentioned in 5.2.1, the DMG game assumes that every source node as well as its cooperative neighbors are connected to the destination. The edge between source and destination allows direct message delivery, and the other edges allow the cooperative neighbors to contribute as relays. Obviously, this assumption is violated if the destination itself is a cooperative neighbor to the source. However, the effect can be negligible when the number of cooperative neighbors is large. A more critical constraint imposed on the graph structure is to make sure the cooperative neighbors, at least with high probability, are connected to the destination. A strong condition for this to be true is that every node together with its K neighbors form a $(K+1)$-clique. This can only happen when $K = N-1$, or the network can be partitioned into disconnected $(K+1)$-cliques. In either case, the network becomes a complete graph (or several complete subgraphs), i.e. finite well-mixed population structure. Therefore we relax the constraint. For every node i, consider the metric

$$\mathcal{O}_i = \max_{j \in K_i} o_{i,j},$$
$$o_{i,j} = \frac{|K_j \cap K_i|}{\overline{K}},$$

where K_i is the set of neighbors of node i, $|\cdot|$ the size of set, and $\overline{K}$ the average node degree. Choosing the neighbor $j \in K_i$ such that $\mathcal{O}_i = o_{i,j}$ as the destination, source node i maximizes its payoff provided all the neighbors are cooperative. Graph with $\mathcal{O}_i \simeq 1$ and small $\mathrm{Var}[o_{i,j}|i]$ complies more with the population structure assumed in DMG. In the following, we study the statistics of $\mathcal{O}_i$ and $o_{i,j}$ in different type of graphs.

In *ring graph*, nodes are connected in a circular manner and every node has the same degree K. Without loss of generality, we consider K is even. One can easily obtain $\mathcal{O}_i = 1 - \frac{2}{K}$ and $\frac{1}{2} - \frac{1}{K} \leq o_{i,j} \leq 1 - \frac{2}{K}$. When $K \gg 1$, $O_i \sim 1$ but the variance of $o_{i,j}$ is large. In *random regular graph* with degree K, nodes are connected randomly. Therefore $\mathcal{O}_i$ is small as there is little overlapping between adjacent nodes' neighborhood. Finally, we observe that *community-based graph* has high $\mathcal{O}_i$ and small $\mathrm{Var}[o_{i,j}|i]$. This is not surprising since community-based graph can be regarded as several quasi-cliques linked by sparse random connections. Therefore adjacent nodes tend to have very similar neighborhood.

In Fig. 5.5, we plot the Complementary Cumulative Distribution Function (CCDF) of $\mathcal{O}_i$ and the standard deviation $\sqrt{\text{Var}[o_{i,j}|i]}$ over the three population structures, and $N = 300, K = 60$. The distribution for random regular graph and community-based graph correspond to statistics collected from 10 instances. Random regular graph no doubt violates the assumption on destination reachability, as $\mathbb{P}(\mathcal{O}_i > 0.3) \simeq 0.2$, i.e. similarity in neighbors' neighborhood is too small. Therefore, the payoff function needs to be modified to reflect the relay-destination disconnection, and we expect to see a different result from Fig. 5.4. For ring graph, if the source-destination pairs are carefully chosen, then our assumption can be satisfied. Otherwise, due to the large standard deviation, this assumption does not hold. Lastly, community-based graph satisfies the destination reachability constraint very well, as $\mathbb{P}(\mathcal{O}_i > 0.9) \simeq 0.9$ and $\mathbb{P}(\sqrt{\text{Var}[o_{i,j}|i]} > 0.02) = 0$. Therefore Fig. 5.4 reflects convincing results.

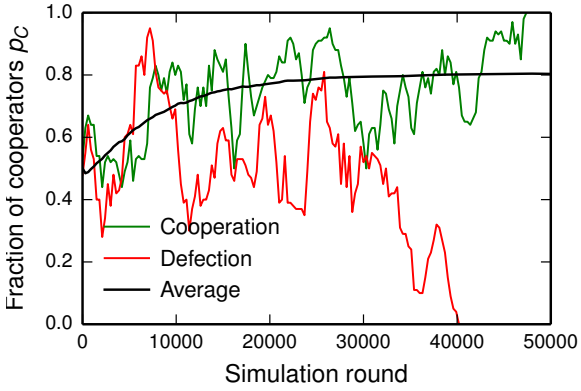


Figure 5.6: Evolution of cooperation rate on community-based graph with random initial state.

5.3.3.4 Evolution with random initial state

So far we have been focusing on the fixation probability of cooperators and defectors in different structured populations. In real mobile networks like DTN, the initial state is unlikely to be a single cooperator or defector, but a mix of the two strategies. Here, we repeat the simulation with a different initial condition. Instead of randomly picking one cooperator or defector, we initialize everyone's strategy randomly and independently, i.e. for each user,

with probability 0.5 it is a cooperator. To faithfully report the evolution of population state in DMG, we plot the fraction of cooperators in the population at different time stages (rounds) during the simulation. Fig. 5.6 shows the dynamics over 1000 repeated runs, and $\lambda\tau = 0.5, \omega = 0.8, N = 100, K = 10, \varepsilon = 0.2 * \hat{\varepsilon}^*$. Two instances (out of 1000) of evolution dynamics on community-based graph are plotted, one towards complete cooperation (in green) and the other towards complete defection (in red). We also calculate the average cooperation rate at each time stage, and plot the averaged evolution dynamics. We see that after 50000 rounds of evolutions, the average population state is $p_C = 0.8$. It means that simulations that reach complete cooperation are the majority. This again proves that structured populations promote the evolution of cooperation.

5.4 Promotion of Cooperation by Socialized Mobility Patterns

In the previous section, we studied the emergence of cooperation on top of a static network topology. But in OppNet, the network topology was deemed dynamic due to users' movement. The evolution is thus more complex as the time-varying network structure forces the users to interact with different groups of people. In [CMG12], the players move randomly with constant velocity and participate in group interactions. The correlation between cooperation and velocity is reported: an intermediate degree of mobility was found to be optimal for the evolution of cooperation. [ATB14, CK12, CGC11] employ similar network models but study pairwise interactions. They obtain consistent results as [CMG12].

The shift from static network to mobile network is an important step towards more realistic network conditions, yet the underlying mobility patterns have been regarded as homogeneous. In fact, all the above-mentioned works assume users to have constant moving speed and they can travel across the whole network. However, it has been well known that real-life *mobile social networks* are heterogeneous [WXH13, Pap, CML11], e.g. users frequently visit and stay around a few "home locations", and their moving ranges are confined by heterogeneous gyration radii. These properties are currently missing in this line of research.

In this section, we study the formation of cooperation in mobile networks following the EGT framework. Different from previous work, we propose a Socialized Speed-Restricted Mobility (SSRM) model to represent realistic human motions. Motivated by MSN, in SSRM, every user is initially located at a randomly picked "home position". Then a moving area A_i centered at the home location is drawn from some probability distribution to restrict its movement. SSRM is able to characterize both homogeneous and heterogeneous mobility patterns by adjusting the distribution of A_i. User randomly moves to a new position within the area per time step. Simultaneously to the movement, every user encounters instant neighbors if their distance is smaller than a certain communication range. We assume each user has a memory of τ time steps, and is able to accumulate the neighbors encountered during this period of time. This neighbor collection process can be seen as an exploration of the neighborhood surrounding the home position. After every τ time steps, users interact with the accumulated neighbors by playing the *Public Goods Game* (PGG), which is a common model for group interaction. Users reproduce by updating their strategies based the obtained payoff. This is seen as one round of evolution. We are interested in finding the predominant strategy (C or D) after a long term of evolutions. This section makes the following contributions:

– We show that the SSRM model generates similar structure as mobile social networks. By deriving an approximation to the degree distribution, we prove that two distributions that commonly appear in social networks: *exponential* and *power-law* distribution, can be reproduced from SSRM for large τ.

– An extensive simulation-based study on the evolution of cooperation is conducted:

  • We first focus on the simultaneous evolution of mobility and strategy by letting $\tau = 1$. This is the most popular setting in recent works [CMG12, ATB14, CK12, CGC11]. Under SSRM, we find similar result as previous works, i.e. cooperation is enhanced at infinitesimal mobility, but damaged when mobility is moderate.

  • Then τ is increased to separate the mobility and reproduction. This pictures a more realistic evolution as users are given more time to explore the neighborhood. We

show that cooperation is promoted in this case due to a reduction in the randomness of neighborhood. It is also found that degree heterogeneity further encourages cooperation compared to homogeneous movement.

We believe our findings are helpful to understand cooperative behaviors in MSN. It is also worth mentioning that although mobile interactions in this section are modeled as PGG game, our EGT framework could be used to study specific cooperation-based mobile applications once they are formulated in the PGG paradigm such as in [ZGL14, CWL10b].

5.4.1 System Model

The mobile network under consideration is composed of users that move according to the SSRM model. To study their cooperative behaviors, we consider that each user in its movement accumulates neighbors through a collection process, and engages them to participate in a group interaction modeled by PGG. Users update their game strategies afterwards to improve the fitness. Details are presented in the following.

5.4.1.1 Network geometry

We consider the network extension to be a 2-dimensional circular space $\mathcal{A}$ with area $A = \pi R^2$. A population of $N(A)$ mobile users are randomly distributed in the network according to Poisson Point Process (P.P.P.) of intensity ϕ, i.e. $\mathbb{P}[N(A) = k] = \frac{(A\phi)^k}{k!}e^{-A\phi}$. Without loss of generality, we assume $\phi = 1$ throughout this section, thus the density of mobile users in the network $\rho = N(A)/A \sim 1$[6] when A is sufficiently large.

5.4.1.2 Socialized Speed-Restricted Mobility (SSRM) model

Mobile users move under a speed-restricted model. In the initial stage (time step $t = 0$), all users are uniformly and randomly located at an initial position in the network. Then we assign each user a circular moving area $\mathcal{A}_i$ centered at the initial position, whose radius R_i

[6] $f(n) \sim g(n)$ means that $\lim_{n\to\infty} f(n)/g(n) = 1$.

restricts its moving speed. Once the network starts to operate, users will move randomly and independently within their own circular area. The location of each user, denoted as X_i^t, will be totally reshuffled from one time step t to $t+1$.

The size of $\mathcal{A}_i$ is denoted as $A_i = \pi R_i^2$. There are two ways to determine each user's A_i in SSRM model:

Homogeneous case: every user has the same moving range $R_i = R_{uni}$ and moving area $A_i = A_{uni} < A$.

Heterogeneous case: A_is are treated as independent and identically distributed (iid) random variables (rv) with probability density function (pdf) $f_{A_i}(a)$. In this section, we consider A_i to be either *Exponential* or *Pareto rv*, i.e.

$$f_{A_i}^{exp}(a) = \frac{\lambda}{\pi} e^{-\frac{\lambda}{\pi}a}, a \geq 0, \tag{5.16}$$

and

$$f_{A_i}^{pareto}(a) = \frac{\alpha(\pi\mu)^\alpha}{(\pi\mu + a)^{\alpha+1}}, a \geq 0, \tag{5.17}$$

where λ, μ and α are positive parameters that control the degree of heterogeneity in A_i. Both of the two cases imply that users are more likely to stay in a fixed area around some home location, and the heterogeneous model further guarantees the moving areas are different across the users. Therefore the SSRM model complies with moving patterns found in mobile social networks [WXH13, Pap, CML11].

5.4.1.3 Public goods game and strategy update

In addition to the initial position, each user is randomly assigned an initial strategy s_i^0 from C ($s_i^0 = 1$) and D ($s_i^0 = 0$). Mobile users have a constant communication range C_r, so that user i and j are mutual neighbors if $\left\|X_i^t - X_j^t\right\| \leq C_r$, where $\|\cdot\|$ denotes the Euclidean distance. At each time step t, user i moves to a new random location in $\mathcal{A}_i$ and encounters a set of neighbors $\mathcal{N}_i^t$.

Apart from the mobility model, we define a neighbor collection process $\mathcal{T}$ that incorporates τ time steps. Parameter τ acts as the memory of mobile users. During the collection

process, user i continues its random movement but keeps the history of the encountered neighbors. Therefore the set of neighbors user i accumulates in $\mathcal{T}$ is $\mathcal{N}_i^{\mathcal{T}}(\tau) = \bigcup_{t=t_0}^{t_0+\tau-1} \mathcal{N}_i^t$. Users play a round of PGG after every collection process as follows. First, each user i forms a group with its neighbors $j \in \mathcal{N}_i^{\mathcal{T}}(\tau)$. Suppose that d_i is the size of $\mathcal{N}_i^{\mathcal{T}}(\tau)$ and $d_i + 1$ is the group size. A single PGG game is played for every group in the network. Therefore every user i participates in exactly $d_i + 1$ games. We assume that each user has a fixed amount of resource c to contribute, and a Cooperator allocates $c_i = \frac{c}{d_i+1}$ to every group it is involved in, while Defector contributes nothing. In a single PGG game, the group accumulates the contributions of Cooperators and multiplies it by an enhancement factor r. The resulting public goods are equally distributed among all the participants. The payoff obtained by each user from a single PGG is equal to its share of the public goods minus its contribution. As every user is engaged in multiple PGG games, the total payoff obtained by user i after $\mathcal{T}$ is:

$$P_i^{\mathcal{T}}(\tau) = \sum_{j \in \mathcal{N}_i^{\mathcal{T}}(\tau) \cup \{i\}} \frac{\sum_{l \in \mathcal{N}_j^{\mathcal{T}}(\tau) \cup \{j\}} r c_l s_l^{t_0}}{d_j + 1} - (d_i + 1) c_i s_i^{t_0} \tag{5.18}$$

Next, individuals update their strategies based on the obtained payoff. Every user i chooses one of its neighbors j at random. The probability that i takes j's strategy in the next collection process is

$$\mathbb{P}[s_i(t_0 + \tau) = s_j(t_0)] = \frac{\max\{P_j^{\mathcal{T}}(\tau) - P_i^{\mathcal{T}}(\tau), 0\}}{\max\{P_k^{\mathcal{T}}(\tau) - P_l^{\mathcal{T}}(\tau)\}}, \forall k, l. \tag{5.19}$$

From equation (5.19), we see that users tend to adopt strategies of successful neighbors whose payoff is higher than their own. This is in line with [SSP08][CMG12]. Up to this point, one round of evolution is completed. The same process is repeated until the number of cooperators becomes stable.

5.4.2 Approximated Degree Distribution of SSRM Model

In this section, we investigate the degree distribution of SSRM model. The purpose is twofold: first, to confirm SSRM indeed generates similar network structure as mobile social network; second, to facilitate our analysis in 5.4.3. The latter is motivated by [PGS13, SSP08], which

have shown that degree distribution can greatly affect the level of cooperation in static networks.

At the end of a collection process, the network connectivity is given by $\mathcal{N}_i^{\mathcal{T}}(\tau)$s. The degree of user i is d_i. For the sake of tractability, given the moving area A_i, we approximate d_i with the number of users whose initial positions locate in $\mathcal{A}_i$, and denote it with $\hat{d}_i$. Since the population is generated by P.P.P process, conditioned on $A_i = a$, we have $\hat{d}_i = N(a) = Poisson(a)$, i.e. $p_{\hat{d}_i|A_i}(k|a) = \frac{a^k}{k!}e^{-a}$. This approximation is reasonable because users whose initial positions locate in $\mathcal{A}_i$ have higher probability to encounter i compared to others. Therefore they are more likely to appear in $\mathcal{N}_i^{\mathcal{T}}(\tau)$ when τ is large. This will be further verified in numerical simulations.

With respect to the notations in 5.4.1 and the assumption made above, we have the following theorem.

Theorem 5.3. *After every neighbor collection process $\mathcal{T}$, the approximated degree distribution of our considered mobile network is Mixed Poisson Distribution [KX05] with mixing distribution $f_{A_i}(a)$:*

$$\begin{aligned} p(k) &= \mathbb{P}(\hat{d}_i = k) \\ &= \int_0^\infty f_{A_i}(a)\frac{e^{-a}a^k}{k!}da, \qquad k = 0, 1, ... \end{aligned} \tag{5.20}$$

Proof. The probability mass function (pmf) of degree distribution, $p(k)$, can be interpreted as the fraction of users with $\hat{d}_i = k$ when $N(A)$ is sufficiently large. We denote this fraction as $f(k)$, and

$$f(k) = \frac{1}{N(A)}\sum_i \mathbf{1}_{\hat{d}_i=k} = \sum_{a\in\bigcup_i A_i} \underbrace{\frac{n_a}{N(A)}}_{\theta_a} \cdot \underbrace{\frac{\sum\limits_{A_i=a}\mathbf{1}_{\hat{d}_i=k}}{n_a}}_{\varphi_{k,a}},$$

where $\mathbf{1}_e$ is the indicator function and is equal to 1 if e is true and 0 otherwise; n_a (θ_a) is the number (fraction) of users with $A_i = a$; similarly, $\varphi_{k,a}$ is the fraction of users with $\hat{d}_i = k$ among the n_a users. By taking the limit of A, we obtain

$$
\begin{aligned}
p(k) &= \lim_{A\to\infty} f(k) = \lim_{A\to\infty} \sum_{a\in\bigcup_i A_i} \theta_a \cdot \varphi_{k,a} \\
&= \int_0^\infty f_{A_i}(a) \cdot p_{\hat{d}_i|A_i}(k|a)da \\
&= \int_0^\infty f_{A_i}(a) \frac{e^{-a}a^k}{k!} da, \qquad k = 0, 1, ... \qquad (5.21)
\end{aligned}
$$

□

In the rest of this section, we derive the approximated degree distribution for homogeneous and heterogeneous mobility model.

5.4.2.1 Homogeneous mobility model

Corollary 5.1. *In the homogeneous case, $\hat{d}_i$ is a Poisson* rv *with pmf*

$$
p_{uni}(k) = \frac{e^{-A_{uni}} {A_{uni}}^k}{k!}, \qquad k = 0, 1, ... \qquad (5.22)
$$

Proof. In this case, every user has the same moving area A_{uni}. Therefore $f_{A_i}(a) = \delta(a - A_{uni})$, where $\delta(\cdot)$ is the Dirac delta function. The distribution of $\hat{d}_i$ can be easily obtained from equation (5.21).

□

5.4.2.2 Heterogeneous mobility model

We consider the *Exponential* and *Pareto* density function as in equation (5.16) and (5.17).

Corollary 5.2. *For exponential moving area, the pmf of $\hat{d}_i$ is geometric($\frac{\lambda}{\lambda+\pi}$), thus has exponential tail.*

Proof. In this case, the degree distribution is

$$
\begin{aligned}
p_{exp}(k) &= \int_0^\infty f_{A_i}^{\exp}(a)\frac{e^{-a}a^k}{k!}da \\
&= \frac{1}{k!}\int_0^\infty a^k f_{A_i}^{\exp}(a)e^{-a}da \\
&= \frac{1}{k!}\int_0^\infty a^k \mathcal{L}_{\exp}(s)da\Big|_{s=1} \\
&= \frac{1}{k!}(-1)^k\frac{\partial^k \mathcal{L}_{\exp}(s)}{\partial s^k}\Big|_{s=1},
\end{aligned}
$$

where $\mathcal{L}_{exp}(s)$ is the Laplace transform of $f_{A_i}^{exp}(a)$. Replace $\mathcal{L}_{exp}(s)$ with $\frac{\lambda/\pi}{\lambda/\pi+s}$ we obtain

$$
p_{exp}(k) = \frac{\lambda}{\lambda+\pi}(\frac{\pi}{\lambda+\pi})^k, \qquad k = 0, 1, ...
$$

Obviously, the degree is geometric distribution with parameter $p \triangleq \frac{\lambda}{\lambda+\pi}$. The Complementary Cumulative Distribution Function (CCDF) is

$$
\bar{F}_{exp}(k) = (1-p)^k = e^{-\ln\frac{1}{1-p}k} = e^{-\Lambda k}.
$$

Therefore the degree distribution exhibits exponential tail. □

Corollary 5.3. *For Pareto moving area, the pmf of $\hat{d}_i$ is asymptotically equal to $f_{A_i}^{pareto}(k)$, thus has power-law tail at infinity.*

Proof. When the moving area is drawn from Pareto distribution, notice that

$$
f_{A_i}^{pareto}(a) = \alpha(\pi\mu)^\alpha\frac{a^{\alpha+1}}{(\pi\mu+a)^{\alpha+1}}\cdot a^{-(\alpha+1)} \triangleq C(a)\cdot a^\gamma,
$$

where $C(a) = \alpha(\pi\mu)^\alpha\frac{a^{\alpha+1}}{(\pi\mu+a)^{\alpha+1}}$ is a slowly varying function [BGT89] with respect to a since

$$
\lim_{a\to\infty}\frac{C(\xi a)}{C(a)} = \lim_{a\to\infty}\xi^{\alpha+1}\frac{(\pi\mu+a)^{\alpha+1}}{(\pi\mu+\xi a)^{\alpha+1}} = 1, \forall\xi > 0.
$$

In addition, $\gamma = -(\alpha+1) < -1$. We can apply Theorem 2.1 in [Wil90] to obtain the following tail distribution of $p_{pareto}(k)$:

$$p_{pareto}(k) \sim C(k)k^{\gamma} = \frac{\alpha(\pi\mu)^{\alpha}}{(\pi\mu+k)^{\alpha+1}}, \quad k \to \infty$$

The CCDF is

$$\bar{F}_{pareto}(k) \sim (\frac{\pi\mu}{\pi\mu+k})^{\alpha} \sim (\pi\mu)^{\alpha}k^{-\alpha},$$

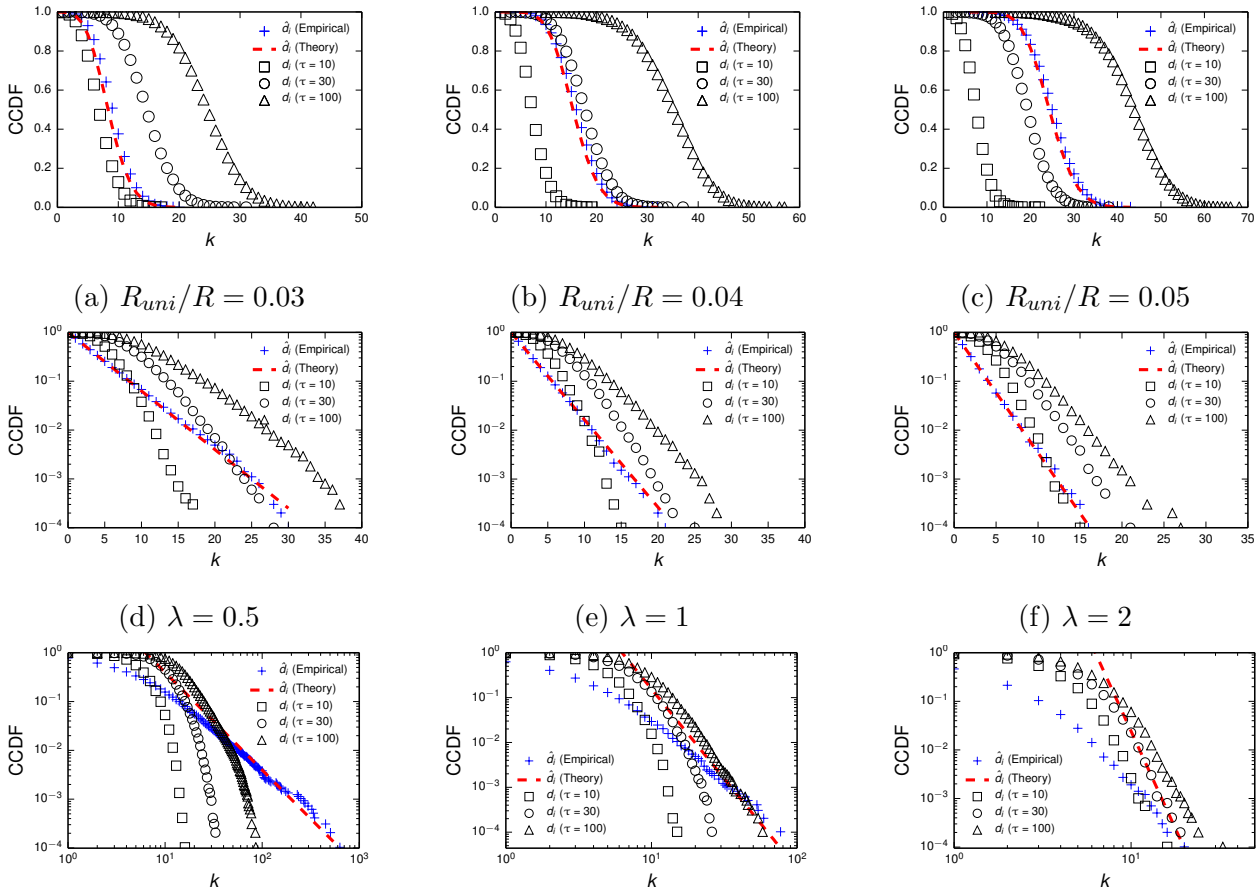
which exhibits power-law tail when k is large. □

5.4.2.3 Numerical validation of degree distribution

We verify our theoretical results in 5.4.2 through numerical simulations. First, we create a circular space with area A and generate $N(A)$ users from $Poisson(A)$. Then the users independently and randomly choose their initial positions within the space. This process is simulating P.P.P. of density 1 on the circular plane. Next, every user is assigned a moving area around its initial position according to our mobility model. We acquire the empirical distribution of degree approximation $\hat{d}_i$ by gathering $\hat{d}_i = \sum_j \mathbf{1}_{\|X_i^0 - X_j^0\| \leq R_i}, \forall i$. The actual degree distribution is obtained from $\mathcal{N}_i^{\mathcal{T}}(\tau)$s by simulating the collection process as in 5.4.1.3.

Our simulation uses the following configuration: $A = 10000$, communication range $C_r = 0.5$, $\tau = [10, 30, 100]$. For the homogeneous distribution, we consider R_{uni}/R to be in the range $0.03 \sim 0.05$. For exponential moving area, we consider λ from $0.5 \sim 2$. For Pareto moving area, we consider $\mu = 2$ and α from $2 \sim 8$. Every simulation is repeated 10 times to reflect an accurate empirical distribution.

Simulation results for the homogeneous case are shown in Fig. 5.7 (a)-(c). Looking at the theoretical and empirical CCDF of $\hat{d}_i$, we see that our result in Corollary 5.1 is accurate. Moreover, we compare the distribution of $\hat{d}_i$ and d_i at different τ. We are interested in the optimal value of τ, denoted as τ_{opt}, at which d_i is best approximated by $\hat{d}_i$. First, in each of the subfigures, we observe that when τ increases, the CCDF of d_i shifts to the right, causing a higher average degree. This is intuitive since larger τ means a longer collection process, thus the average number of encountered nodes is increased. Secondly, A_{uni} affects τ_{opt}. For example, in (a), $\hat{d}_i$ is well aligned with d_i at $\tau = 10$, but deviates from d_i when $\tau = 30$ or



(a) $R_{uni}/R = 0.03$ (b) $R_{uni}/R = 0.04$ (c) $R_{uni}/R = 0.05$

(d) $\lambda = 0.5$ (e) $\lambda = 1$ (f) $\lambda = 2$

(g) $\alpha = 2$ (h) $\alpha = 4$ (i) $\alpha = 8$

Figure 5.7: Simulation results for degree approximation. Homogeneous: (a)-(c). Exp: (d)-(f). Pareto: (g)-(i).

100. Increasing A_{uni}, the CCDF of $\hat{d}_i$ gets closer to d_i at $\tau = 30$. Notice that $\hat{d}_i$, the number of users covered by A_{uni}, will surely increase with A_{uni}. In the meantime d_i increases as well because $\mathcal{N}_i^{\mathcal{T}}(\tau)$ becomes larger as $X_i(t)$s are more spread out, and $\mathcal{N}_i^t$s are more diverse for a larger A_{uni}. However, $\hat{d}_i$ is more directly influenced by A_{uni} than d_i, which explains the phenomenon shown above.

When the moving areas are exponentially distributed, the semi-logarithmic plots in Fig. 5.7 (d)-(f) show that our theoretical result for $\hat{d}_i$ in Corollary 5.2 matches the empirical result very well. As for d_i, we observe similar phenomenon as in the homogeneous case: 1) larger τ leads to a longer tail in the degree distribution; 2) the larger the average moving area, the

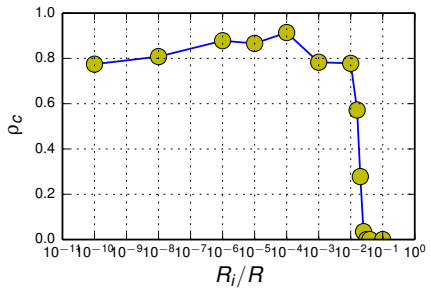


Figure 5.8: Effects of mobility on cooperation rate.

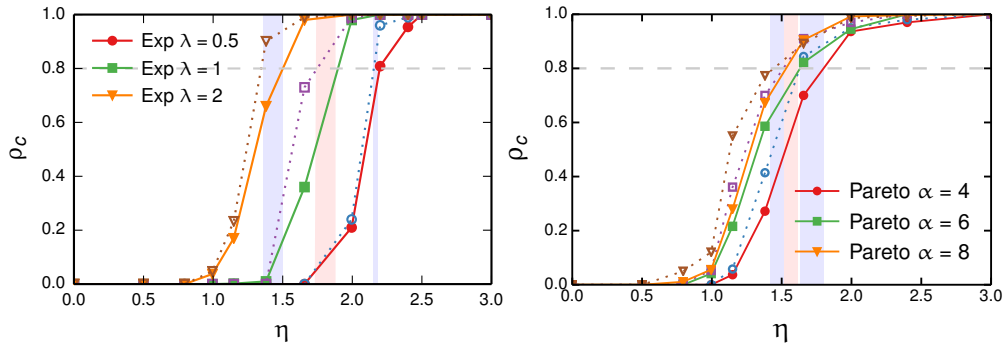
larger the value of τ_{opt}. In this case, the average moving area $\propto \frac{1}{\lambda}$. Therefore $\tau_{opt} \simeq 30$ when $\lambda = 0.5$, and $\tau_{opt} \simeq 10$ when $\lambda = 1$ or 2.

Lastly, we consider the Pareto moving area. According to Corollary 5.3, we are only able to capture the asymptotic behaviour of $\hat{d}_i$. Therefore we see in the log-log graph (Fig. 5.7 (g)) that when $\alpha = 2$ and the magnitude of the tail is 10^2, $\hat{d}_i$ is in good agreement with the power-law distribution. Otherwise $\hat{d}_i$ deviates from the power-law CCDF as the tail is short. Besides, the distribution of d_i at different τ and the value of τ_{opt} at different α have the same trend as before.

The numerical results suggest that although our theoretical distributions of $\hat{d}_i$ from 5.4.2 are correct, how well they can approximate the actual degree d_i depends on the average moving area as well as τ. In the following simulations we stick to $\tau = 30$, which provides relatively good approximation for the considered parameters. Even if $\hat{d}_i$ deviates from d_i, we see from Fig. 5.4.2.3 (d)-(i) that d_i still has exponential or pareto tail, i.e. SSRM is able to reproduce real-life degree distribution in mobile social networks.

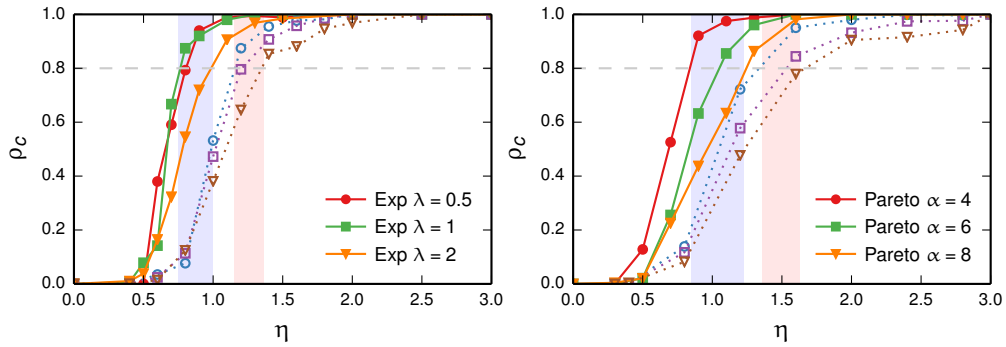
5.4.3 Numerical Simulations

In this section, we study the evolution of cooperation in the defined mobile network through simulations. The simulation is divided into epochs of τ time steps. Each epoch corresponds to a neighbor collection process. The movement and evolutions are simulated based on 5.4.1. Whether cooperation can survive is indicated by the fraction of cooperators ρ_c after a long term of evolutions, given some fixed enhancement factor r. If ρ_c is close to 1 at



(a) Exponential case. (b) Pareto case.

Figure 5.9: Cooperation rate ρ_c as a function of η in heterogeneous mobile networks when $\tau = 1$.



(a) Exponential case. (b) Pareto case.

Figure 5.10: Cooperation rate ρ_c as a function of η in heterogeneous mobile networks when $\tau = 30$.

relatively small r, we say cooperation survives more easily, or is promoted by the underlying network conditions. In practice, r is normalized with respect to the average group size $\langle d \rangle + 1$ [SSP08, CMG12, PGS13]. The resulting factor is denoted as $\eta = \frac{r}{\langle d \rangle + 1}$. It has been shown that ideally cooperation prevails at $\eta = 1$ in an infinite, well-mixed population. This serves as a benchmark in our evaluations. In the simulations, we compute ρ_c after 10000 epochs with respect to η. The network size $A = 1000$. Other configurations are kept the same as the previous subsection. Without loss of generality, we set $c = 1$.

We consider two different evolution processes by adjusting the value of τ. In the first case, $\tau = 1$ so that the movement and strategy update occur simultaneously. We find SSRM

has the same impact on cooperation as previous models in [CMG12][ATB14][CK12]. Then we turn to the evolution with larger τ, which leads to more realistic movement (as validated in 5.4.2.3). Moreover, the separation between movement and strategy update is more consistent with human behavior as movement and neighbor collection are microscopic processes that happen frequently, while strategy update (adaption) takes much longer time [GCF13]. Simulations reveal that heterogeneous mobility model significantly promotes cooperation compared to the homogeneous model. Further investigation is conducted by correlating such enhancement with the behavior of high degree users in heterogeneous mobile networks.

5.4.3.1 Simultaneous evolution of mobility and strategy

We first study the case where $\tau = 1$. Users move, play PGG, and update strategies simultaneously every time step. In homogeneous SSRM, we vary the value of R_{uni}/R to reflect different level of mobility. When R_{uni}/R is very small, the mobile network is reduced to a static one. Users interact with constant group of neighbors and the group size is $Poisson(\pi C_r^2 + 1)$. Increasing R_{uni} introduces more randomness in the evolution, as the set of neighbors changes every time step. It has been shown in [CMG12] and [CK12] that a slight increase in mobility improves users' ability to discover clusters of cooperators, while a higher level of mobility strongly reduces the formation of large cooperator clusters due to invasion of defectors. Fig. 5.8 shows the cooperation rate ρ_c as a function of mobility level. We set $C_r = 1.5$ to guarantee network connectivity, and $\eta = r/(\pi C_r^2 + 1) = 0.9$. Our result in a homogeneous mobile network is consistent with previous studies.

From Fig. 5.8, cooperation is maximally promoted at $R_{uni}/R = 10^{-4} \ll 1$. From this point on, ρ_c drastically decays. This result is rather pessimistic because in mobile social networks, the average moving range can be much larger, i.e. $\langle R_i \rangle = O(R)$. Those fast moving users are detrimental to the formation of cooperation as they may invade already-grouped cooperator clusters.

To verify this, we examine simultaneous evolution in heterogeneous mobile networks. According to our model, the moving area has heavy tail distribution. Therefore few users

travel in a large area while the majority moves in a very small range. In the mean time we keep the average moving range relatively large. We choose $\lambda = [0.5, 1, 2]$ for the exponential model, and $\mu = 2$, $\alpha = [4, 6, 8]$ for the pareto model. With these parameters, $\langle R_i \rangle / R$ is $(2.6 \sim 7) \cdot 10^{-2}$, which correspond to points of low cooperation in Fig. 5.8. We plot ρ_c with respect to η. In comparison, we also plot ρ_c of a homogeneous network with the same average moving range. Results are shown in Fig. 5.9[7].

Not surprisingly, the level of cooperation increases with η, as large value of r rewards contributors and thus encourages cooperation. However, to reach a high cooperation level $\rho_c = 0.8$, η must be at least 1.5 and sometimes even larger than 2, while in Fig. 5.8, cooperation prevails at $\eta = 0.9$ when moving range is much smaller. Moreover, we have the following observations in Fig. 5.9a and 5.9b:

1) Cooperation emerges more easily as we increase λ or α (or equivalently, decrease the average moving area).

2) Keeping their average moving ranges the same, homogeneous and heterogeneous mobility models have similar impact on the cooperation level as their critical η are almost the same.

Observation 1) is consistent with our findings in Fig. 5.8. To explain 2), recall that when $\tau = 1$, no matter in homogeneous or heterogeneous networks, the degree distribution is Poisson with the same mean $\pi C_r^2 + 1$. When the average moving range is large, users always encounter and interact with different groups of people characterized by the same distribution. Heterogeneity in moving area does not make a big difference in this scenario.

5.4.3.2 Increasing τ stimulates cooperation

Now we focus on the evolution of cooperation after a prolonged collection process. We set $\tau = 30$ and $C_r = 0.5$. As validated in 5.4.2.3, the chosen parameters generate degree d_i that

[7]The dotted curves correspond to homogeneous networks. Curves with the same markers have the same average moving range. The shaded area projects the critical range of η for which ρ_c reaches 0.8.

can be well approximated by exponential and pareto distribution. We repeat the simulation in Fig. 5.9. Results are shown in Fig. 5.10.

Comparing Fig. 5.9 and 5.10, we clearly see that Cooperators survive more easily when τ is increased, no matter in homogeneous or heterogeneous networks (although in Fig. 5.9, we bind the critical η for curves with the same marker in one shaded area, and in Fig. 5.10 for curves with the same line-style). Moreover, the cooperation level is improved even more in heterogeneous networks. In exponential degree network, cooperation prevails at $\eta \leq 1$ and in pareto degree network it prevails at $\eta \sim 1$. This is almost as good as the optimal case in Fig. 5.8.

First, to explain why cooperation is promoted once we increase τ, consider the randomness of neighbor set $\mathcal{N}_i^{\mathcal{T}}(\tau)$ in two different collection processes. When $\tau = 1$, $\mathcal{N}_i^{\mathcal{T}}(1) = \mathcal{N}_i^{t_0}$, which is a collection of users inside the range C_r around a random location. For a larger τ, this set becomes a union of multiple such $\mathcal{N}_i^{\mathcal{T}}(1)$s. Since every user has a restricted moving area $\mathcal{A}_i$, when τ increases, this union will not grow infinitely but is largely limited to the neighbors that frequently appear in the moving area, i.e. those whose initial positions are inside $\mathcal{A}_i$. Therefore the randomness is reduced when we increase τ. As a consequence, the network topology is more stable. To quantify the randomness, we define the following stability metric:

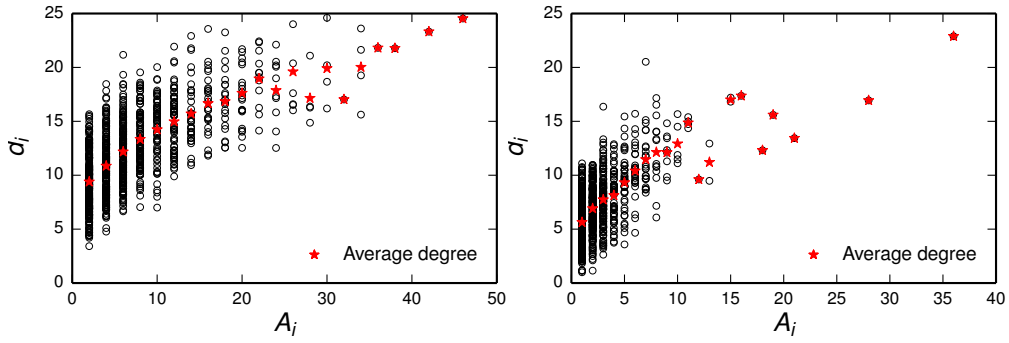
$$\mathcal{S} = \frac{\sum_{m=1}^{M} \left|\mathcal{N}_i^{\mathcal{T}_m}(\tau)\right|}{M \left|\bigcup_{m=1}^{M} \mathcal{N}_i^{\mathcal{T}_m}(\tau)\right|}.$$

$\mathcal{S}$ is equal to the average size of $\mathcal{N}_i^{\mathcal{T}}(\tau)$ in M epochs divided by the size of their union. One can easily prove that $0 < \mathcal{S} \leq 1$. $\mathcal{S} = 1$ when the $\mathcal{N}_i^{\mathcal{T}}(\tau)$ does not change. The more diverse (random) the $\mathcal{N}_i^{\mathcal{T}}(\tau)$, the smaller the $\mathcal{S}$. Comparison in term of $\mathcal{S}$ is shown in Table 5.2. We take $M = 100$ and present the average stability of 10% users with largest (top) and smallest (bot) moving areas. Obviously, stability in $\mathcal{N}_i^{\mathcal{T}}(\tau)$ is increased for larger τ.

With network randomness being controlled, the difference between the two mobility models lies in the degree heterogeneity. Recall $\mathcal{N}_i^{\mathcal{T}}(30)$ has exponential or power-law tail approx-

Table 5.2: Stability $\mathcal{S}$ comparison

Model	Top 10%		Bot 10%	
	$\tau = 1$	$\tau = 30$	$\tau = 1$	$\tau = 30$
Exp ($\lambda = 0.5$)	0.059	0.348	0.17	0.614
Exp ($\lambda = 1$)	0.092	0.47	0.263	0.706
Exp ($\lambda = 2$)	0.147	0.571	0.369	0.773
Pareto ($\alpha = 4$)	0.115	0.531	0.335	0.753
Pareto ($\alpha = 6$)	0.168	0.6	0.4216	0.815
Pareto ($\alpha = 8$)	0.2	0.638	0.48	0.8358



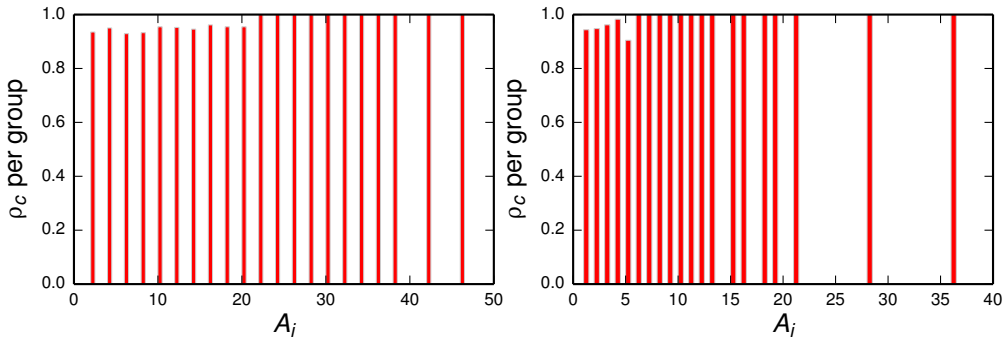
(a) Exponential case.

(b) Pareto case.

Figure 5.11: Degree and moving area correlation.

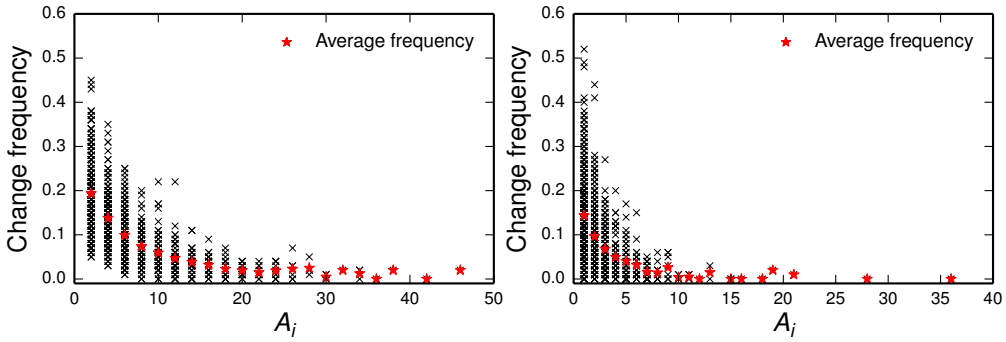
imation, which is heavier than poisson tail. Therefore there are more high degree users in the heterogeneous models than in the homogeneous one. Those users, together with their neighbors, form clusters that are analogous to social communities. According to previous studies [SSP08][RYW10], degree heterogeneity and clustering promote cooperation on a static network. Although we are concerning a mobile network, it has been shown that the restricted moving areas and a large τ limit the randomness induced by mobility. Consequently the benefit of degree heterogeneity appears and we observe the promotion of cooperation in heterogeneous mobile networks.

Next, we take a closer look at the evolution process in heterogeneous mobile networks.



(a) Exponential case. (b) Pareto case.

Figure 5.12: Cooperation rate in moving area groups.



(a) Exponential case. (b) Pareto case.

Figure 5.13: Strategy update frequency in moving area groups.

Of particular interests are the cooperative behaviors of high degree users.

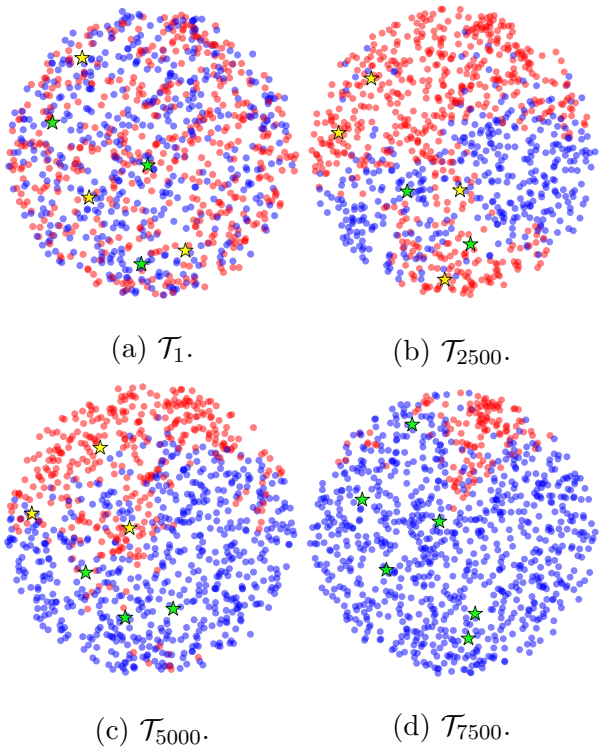
5.4.3.3 Cooperative behaviors of heterogeneous mobile users

We focus on two instances of heterogeneous mobile networks at $\tau = 30$:

1) Exponential with $\lambda = 0.5$.

2) Pareto with $\mu = 2$, $\alpha = 4$.

We first examine the correlation between degree d_i and moving area A_i. We group users based on their moving areas A_i, and plot their average degrees over 100 epochs. The average degree in every group is displayed as well. As shown in Fig. 5.11, user's degree is positively



(a) $\mathcal{T}_1$. (b) $\mathcal{T}_{2500}$.

(c) $\mathcal{T}_{5000}$. (d) $\mathcal{T}_{7500}$.

Figure 5.14: Snapshots in exponential mobile network. Cooperators (defectors) are labeled red (blue). Users marked as stars are cooperators (green) and defectors (yellow) at hubs.

correlated with its moving area. In the following, we refer to the users with large moving areas as "hub"s, as they are usually the centers of communities in mobile social networks.

Next, we look at the strategy distribution across the moving area groups, after the system enters equilibrium stage. Fig. 5.12 shows the average cooperation rate in different groups at $\eta = 1$. We see that at steady state, the hubs all become cooperators, and defectors only exist among non-hub users. In Fig. 5.13, we plot the average strategy update frequency. It is shown that the strategies of hubs rarely change, while non-hub users are more easily affected by neighbors.

Lastly, we present the snapshots (Fig. 5.14) of the evolution process in 4 different stages. Due to the limit of space, only exponential network is considered here. At $\mathcal{T}_1$, there are equal amount of C and D randomly spread in the space. The marked users are 3 hub Cooperators and 3 hub Defectors. After 7500 rounds of evolution, all of the hubs evolve to cooperators. In the meantime, we clearly see that the neighborhood around hubs is

dominated by cooperators, leaving defectors concentrated in regions far away from the cluster of cooperators.

5.5 Simulation Study of Opportunistic Traffic Offloading

In this section, we present our simulation results for the DTN (DMG) and resource sharing (OppShare) game in OppNet. We focus on the cooperation rate of mobile users in different game models and under different conditions of mobility pattern and cost. In addition, we demonstrate the opportunistic traffic offloading performance assuming an achievable cooperation rate.

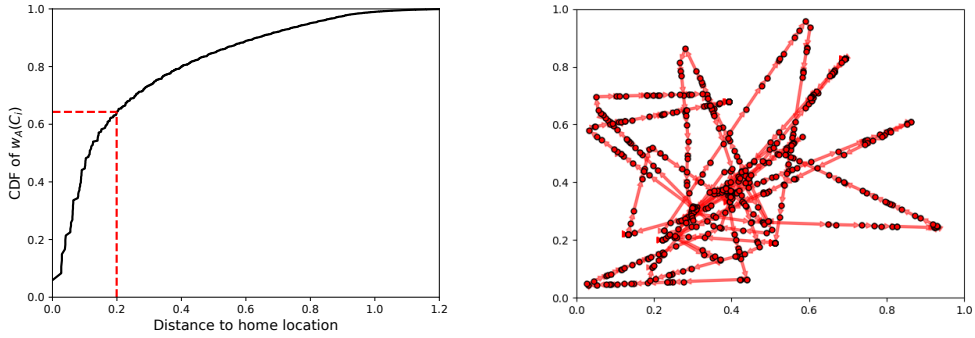
5.5.1 Mobility Model

In our simulation, user movements and contacts are extracted from their mobility traces in OppNet. To generate realistic mobility traces with the desired timespan for reaching the evolutionary equilibrium point, we adopt the SWIM synthetic mobility model [KMS10]. In OppNet, traffic offloading only occurs between users who are in contact[8]. Therefore, to understand user behavior, it is important to generate realistic contact events for OppNet users. The SWIM mobility model produces movement and contacts statistics that match real-world OppNet traces, e.g., it is able to reproduce the power-law distribution with an exponential tail in inter-contact time and contact duration data. The other feature SWIM implemented is that by adjusting the weight of each location, SWIM controls user's preference when choosing the destination of each movement. SWIM divides the map into cells, and the weight of cell C_i for user A is given by

$$w_A(C_i) = \alpha \cdot closeness(h_A, C_i) + (1 - \alpha) \cdot seen(A, C_i), \tag{5.23}$$

where h_A is the home location of user A, $closeness(h_A, C_i)$ measures the closeness between h_A and C_i, $seen(A, C_i)$ measures the fraction of nodes A saw in C_i the last time it visited,

[8]Two users are in contact if they are in the communication range of each other's device, i.e., they are physically close to each other.



(a) CDF distribution of moving distance from home location.

(b) Movement of sample node whose home location is $(0.3, 0.3)$.

Figure 5.15: Movement dispersion in the original SWIM model.

which denotes the popularity of cell C_i, and $\alpha \in [0, 1]$ adjusts the significance of the two terms. We refer readers to [KMS10] for the detailed definitions.

In our experiments, we noticed that even though SWIM favors cells close to the home locations or popular sites, the cell weights from equation (5.23) still cause dispersion in users' movements, i.e. there is a high probability that a user would visit some place that is far away from its home location and unpopular. We analyze the trace generated by the default SWIM model with $\alpha = 1$, i.e., with strong home locality. Fig. 5.15a shows the Cumulative Distribution Function (CDF) of traveling distance from the home location. It is observed that the probability of visiting a cell within distance 0.2[9] from the home location is only 0.6. There is still a relatively large chance for the node to visit far away places. The effect of such visits is shown in Fig. 5.15b, which illustrates the movement of a randomly sampled node for 1 hour simulation time. More dispersion in the movement means more randomness in the neighborhood, which disobeys the observations of regular movements and interaction patterns in real OppNet. To exclude such unusual movement, we propose the Extended SWIM (ExSWIM) model which adopts the following cut-off weight for cell C_i and user A:

[9]The network is a square area with side length equal to 1.

$$w'_A(C_i) = \begin{cases} w_A(C_i)/\max\limits_{C_j}\{w_A(C_j)\}, & \|h_A - C_i\| \leq R_A, \\ 0, & \|h_A - C_i\| > R_A, \end{cases} \tag{5.24}$$

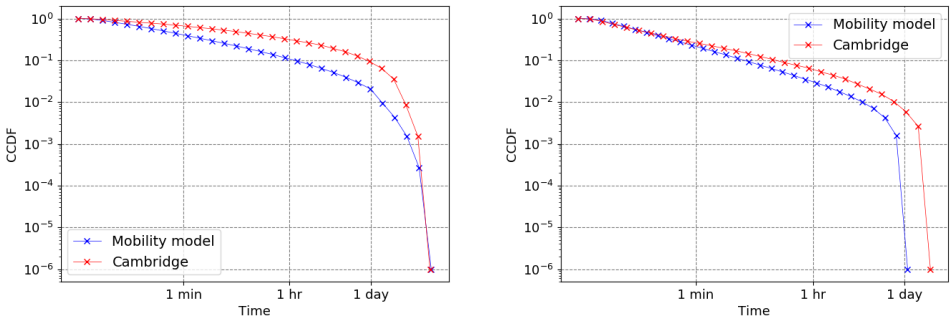
where $\|\cdot\|$ is the Euclidean distance, and R_A represents the moving range of A. In our simulation evaluation, R_A is either a constant (*homogeneous*) or a power-law random variable (*heterogeneous*) with probability density function

$$f_{R_A}(r) = \frac{\beta - 1}{r_{\min}} \left(\frac{r}{r_{\min}} \right)^{-\beta}, r \geq r_{\min}. \tag{5.25}$$

Note that the ExSWIM model and the previous SSRM proposed in 5.4.1.2 shares the same intuition. First, users' movements have the home locality characteristic. Second, users have different moving ranges. While the majority users move within a confined range, a few active users may travel a larger area and interact with more people. The difference between ExSWIM and SSRM however is that in SSRM, users randomly "jump" to different locations within their moving range from time to time, so the movement is discrete. In ExSWIM, in order to generate realistic inter-contact time (ICT) and contact duration (CD), users move continuously and also "pause" after each visit. Therefore the ExSWIM model inherits the locality and heterogeneity properties, but also respects the ICT and CD in OppNet. Also note that using homogeneous and heterogeneous R_A, we are able to reproduce similar mobility traces as the ones we used in 5.4. To show that our model generates realistic ICT and CD distribution, we compare our distributions with the ones generated from the Cambridge dataset, which is a collection of students' movements and contacts in the University of Cambridge that has been widely used to evaluate OppNet protocols. The results are shown in Fig. 5.16, and we see our distributions are very close to the real ones.

5.5.2 Numerical Simulations

We develop an EGT simulator which takes the mobility trace generated from the ExSWIM model as input, and evolves the strategy of users who play opportunistic traffic offloading games. The simulator works for both DMG and OppShare games. Users may as well define new opportunistic games and use our simulator to make predictions for user cooperation



(a) Inter-contact time distribution. (b) Contact duration distribution.

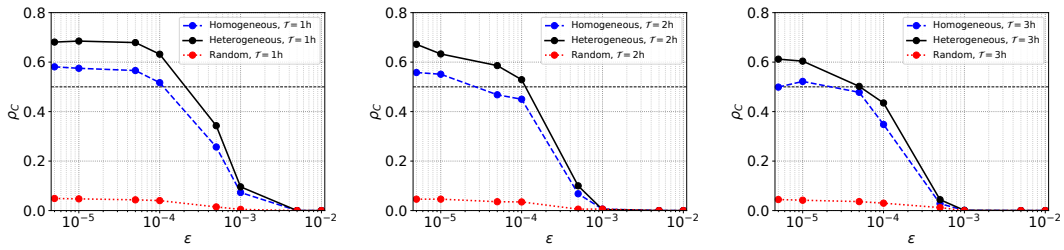
Figure 5.16: ICT and CD distribution of ExSWIM mobility model and Cambridge dataset.

Table 5.3: Simulation parameters

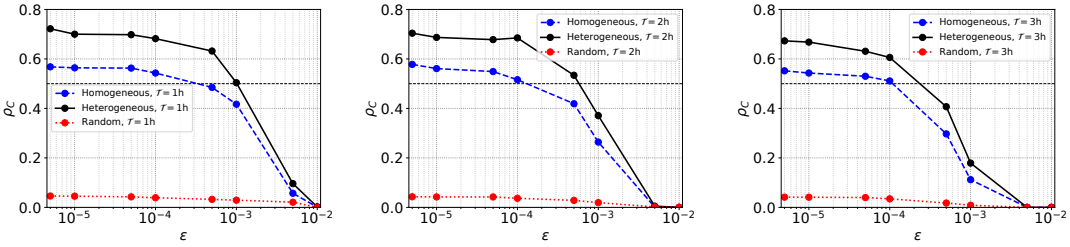
Parameter	Value
Communication range	0.04
Duration (hours)	1000
Devices number	100
Value of α	1
Waiting time slope	1.45
Waiting time bound (hours)	1
Power law β	3
Power law r_{min}	0.15

rate.

In our simulation, we adopt the ExSWIM parameters shown in Table 5.3, and $\lambda\tau = 0.5$ for the DTN game, and $\delta_N = 0.5, \kappa = 16$ for the OppShare game. In the fitness function, $\omega = 0.5$. Users move in a square space with area equal to 1 according to the ExSWIM mobility model. The round time $\mathcal{T}$ varies from 1 hour to 3 hours. Initially, users are randomly and independently assigned a strategy. Given the predetermined parameters as well as the cooperation cost ε, we are interested in the cooperation rate ρ_C when the network state converges, i.e., at $\rho_C = 1$ (full cooperation), or $\rho_C = 0$ (full defection). Each data point in our presented results is the average of $1,000$ simulation trials.



(a) DMG game, $\mathcal{T} = 1hour$. (b) DMG game, $\mathcal{T} = 2hour$. (c) DMG game, $\mathcal{T} = 3hour$.



(d) OppShare game, $\mathcal{T} = 1hour$.

(e) OppShare game, $\mathcal{T} = 2hour$.

(f) OppShare game, $\mathcal{T} = 3hour$.

Figure 5.17: Simulation results for opportunistic traffic offloading games.

Fig. 5.17 depicts user cooperation rate ρ_C with varying $\mathcal{T}$ and cost ε for DMG and OppShare games. It shows that for both games, cooperation emerges when ε is sufficiently small. We obtain the value of the critical ε, denoted by ε^* under different $\mathcal{T}$, $\mathcal{G}$ and mobility pattern for $\rho_C > 0.5$, i.e. when cooperation is favored over defection, and have the following observations:

1. Fixing $\mathcal{T}$ and $\mathcal{G}$, ρ_C increases as ε decreases.

2. Fixing $\mathcal{T}$ and $\mathcal{G}$, we evaluate the cooperation rate under three different mobility models: heterogeneous model (ExSWIM with heterogeneous R_A), homogeneous model (ExSWIM with homogeneous R_A), and random model (SWIM without pausing time). Consistent with our previous studies, cooperation emerges in structured populations (heterogeneous and homogeneous model) but not in random networks. In addition, heterogeneous mobility model promotes cooperation as the ε^* is larger for heterogeneous model, i.e. cooperation emerges more easily under heterogeneous model.

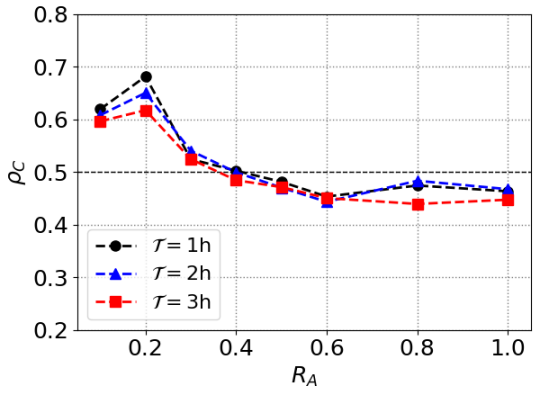


Figure 5.18: Effect of homogeneous moving range on ρ_C.

3. Fixing $\mathcal{G}$, the network exhibits higher ρ_C with shorter round time $\mathcal{T}$.

Fig. 5.18 shows how the maximum cooperation rate changes with the average moving range. R_A is ranging from 0.1 to 1, and cost $\varepsilon = 10^{-6}$. A limited moving range ($R_A = 0.2$) helps increase cooperation rate as the resultant contact graph resembles social network rather than a random network. However, too small moving range ($R_A = 0.1$) disconnects the graph and hence decreases cooperation rate.

5.5.3 Implications for Opportunistic Traffic Offloading

In Fig. 5.17 we see that the cooperation rate increases as ε decreases. When $\varepsilon < 10^{-4}$, ρ_C appears to be higher than 50%, which means the network favors cooperation. To have a sense of the actual value of $\varepsilon = 10^{-4}$, we convert it to a real-world cost of \$0.3 for 5 hours of wireless file transfer in the content distribution network. Considering that the electricity cost for charging a smart phone for a year is only \$0.5, which is equivalent to about \$0.0013 for 5 hours of wireless transfer, the cooperation level in a structured OppNet can be expected to be much higher than 50%, even close to the theoretical maximum.

Finally, we simulate the opportunistic traffic offloading protocols assuming a high cooperation rate to highlight their benefits.

For the DMG game, we define the fraction of offloaded traffic as

$$p_{off} = \frac{\sum_{i=1}^{N} \mathbb{P}_{i,k_i}(T < \tau)}{N},$$

where k_i is the number of cooperative neighbors of user i, and $\mathbb{P}_{i,k_i}(T < \tau)$ is the fraction of packets that can be delivered before the deadline via DTN packet forwarding. The rest packets will be transmitted via infrastructure for immediate delivery. We obtain that when the cooperation rate is $\rho_C = 70\%$, the fraction of traffic offloaded by DTN network is around $p_{off}^{actual} = 65\%$.

For the content distribution protocol, users first download content from the cellular network and then share with neighbors via opportunistic transmission. We define the per hour offloaded content for each user

$$d_{off} = \frac{D_{off} \cdot r}{N\mathcal{T}},$$

where D_{off} is the total offloading duration, r is the D2D transmission rate. From our result, we see that for cooperation rate $\rho_C = 70\%$, we obtain $D_{off} = 3500(sec)$. Assuming $r = 1MB/sec$, we get $d_{off}^{actual} = 35MB/hour$, or $280MB/day$ considering only the 8 hours daytime. This amount of contents is enough to stream YouTube videos for more than 2 hours according to a recent study in [PSS17], and it indicates a significant bandwidth saving for the cellular networks.

CHAPTER 6

Conclusion and Future Work

Facing the tremendous bandwidth demand of mobile users, the current mobile system is embracing a heterogenous structure. This dissertation discussed two options for traffic offloading in heterogeneous mobile networks. The first option is to use alternative infrastructure networks such as WiFi and satellites to offload cellular traffic. The second option is to enable opportunistic communications between mobile devices for data delivery and content sharing as a way to decongest cellular networks.

For the infrastructure-based traffic offloading, we proposed a software-defined multipath transmission system. Our system employs two existing technologies: MPTCP for multipath transmission using both cellular and other alternative network connections; SDN for centralized control and management. By formulating the traffic offloading as a multi-commodity flow network optimization problem, we are able to find the optimally balanced bandwidth usage in each infrastructure network to maximally offload cellular traffic. An SDN controller is developed which identifies MPTCP connections, solves the optimization problem, and centrally direct the routing and bandwidth usage of MPTCP subflows within the infrastructure networks. We implemented the system in both naval satellite communication (`SDN-SAT`) and LTE-WiFi (`SD-MPTOP`) scenarios to show its efficiency compared to existing solutions.

For the opportunistic traffic offloading, we first verified through simulation that a key factor to the offloading efficiency is user cooperation rate. The higher the cooperation rate, the more transmission opportunities arise in the network. Then we adopted the Evolutionary Game Theory to study the conditions for cooperation to emerge in the network. It is proved that for random networks where users contact everyone else with equal probability, users tend to be non-cooperative. Surprisingly, if the contact patterns are more structured, e.g.,

in a community-based network, users tend to be cooperative as long as the cost of opportunistic transmission is sufficiently small. To understand the actual offloading performance in a realistic opportunistic network, we developed a socialized mobility model to simulate user movements. Simulation results confirmed promising opportunistic traffic offloading performance.

For future work, we plan to extend the network optimization model in SD-MPTOP to better capture the effects of network characteristics (e.g., latency, loss, jitter, etc.) on the throughput of MPTCP. On the opportunistic network side, we will build a testbed with smart phones communicating using WiFi-direct. We will conduct real experiments to validate the offloading potentials of the opportunistic network.

REFERENCES

[ABD10] Eitan Altman, Tamer Başar, and Francesco De Pellegrini. "Optimal monotone forwarding policies in delay tolerant mobile ad-hoc networks." *Performance Evaluation*, **67**(4):299–317, 2010.

[ACO13] Hassan Ali-Ahmad, Claudio Cicconetti, Antonio de la Oliva, Vincenzo Mancuso, Malla Reddy Sama, Pierrick Seite, and Sivasothy Shanmugalingam. "An SDN-based network architecture for extremely dense wireless networks." In *Future Networks and Services (SDN4FNS), 2013 IEEE SDN for*, pp. 1–7. IEEE, 2013.

[AG00] M. Allman and J. Griner. "TCP behavior in networks with dynamic propagation delay." In *Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE*, volume 2, pp. 1103–1108 vol.2, 2000.

[AGC14] Behnaz Arzani, Alexander Gurney, Shuotian Cheng, Roch Guerin, and Boon Thau Loo. "Impact of path characteristics and scheduling policies on mptcp performance." In *Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on*, pp. 743–748. IEEE, 2014.

[AGG14] Sergey Andreev, Mikhail Gerasimenko, Olga Galinina, Yevgeni Koucheryavy, Nageen Himayat, Shu-Ping Yeh, and Shilpa Talwar. "Intelligent access network selection in converged multi-radio heterogeneous networks." *IEEE wireless communications*, **21**(6):86–96, 2014.

[AML15] Sohail Abbas, Madjid Merabti, and David Llewellyn-Jones. "On the evaluation of reputation and trust-based schemes in mobile ad hoc networks." *Security and Communication Networks*, **8**(18):4041–4052, 2015.

[AN14] Benjamin Allen and Martin A Nowak. "Games on graphs." *EMS Surveys in Mathematical Sciences*, **1**(1):113–151, 2014.

[AS08] Miriam Allalouf and Yuval Shavitt. "Centralized and distributed algorithms for routing and weighted max-min fair bandwidth allocation." *IEEE/ACM Transactions on networking*, **16**(5):1015–1024, 2008.

[ATB14] Alberto Antonioni, Marco Tomassini, and Pierre Buesser. "Random diffusion and cooperation in continuous two-dimensional space." *Journal of theoretical biology*, **344**:40–48, 2014.

[Bag11] Marcelo Bagnulo. "Threat analysis for TCP extensions for multipath operation with multiple addresses." 2011.

[BCV07] Andrea Baiocchi, Angelo P Castellani, and Francesco Vacirca. "YeAH-TCP: yet another highspeed TCP." In *Proc. PFLDnet*, volume 7, pp. 37–42, 2007.

[BGT89] Nicholas H Bingham, Charles M Goldie, and Jef L Teugels. *Regular variation*, volume 27. Cambridge university press, 1989.

[BLL13] Xuan Bao, Yin Lin, Uichin Lee, Ivica Rimac, and Romit Roy Choudhury. "Dataspotting: Exploiting naturally clustered mobile devices to offload cellular traffic." In *INFOCOM, 2013 Proceedings IEEE*, pp. 420–424. IEEE, 2013.

[BPB11] Sébastien Barré, Christoph Paasch, Olivier Bonaventure, et al. "MultiPath TCP-Guidelines for implementers." Technical report, UCL, 2011.

[BZ] Suzan Bayhan and Anatolij Zubow. "Enabling Flexibility of Traffic Split Function in LTE-WiFi Aggregation Networks Through SDN.".

[BZY14] Jinzhen Bao, Baokang Zhao, Wanrong Yu, Zhenqian Feng, Chunqing Wu, and Zhenghu Gong. "OpenSAN: a software-defined satellite network architecture." In *ACM SIGCOMM Computer Communication Review*, volume 44, pp. 347–348. ACM, 2014.

[CAI06] Pulak K Chowdhury, Mohammed Atiquzzaman, William D Ivancic, et al. "Handover schemes in satellite networks: State-of-the-art and future research directions." *IEEE Communications Surveys and Tutorials*, **8**(1-4):2–14, 2006.

[CDM13] Marco Conti, Franca Delmastro, Giovanni Minutiello, and Roberta Paris. "Experimenting opportunistic networks with WiFi Direct." In *Wireless Days (WD), 2013 IFIP*, pp. 1–6. IEEE, 2013.

[CF04] Carlo Caini and Rosario Firrincieli. "TCP Hybla: a TCP enhancement for heterogeneous networks." *International journal of satellite communications and networking*, **22**(5):547–566, 2004.

[CFL09] Carlo Caini, Rosario Firrincieli, and Daniele Lacamera. "Comparative performance evaluation of tcp variants on satellite environments." In *Communications, 2009. ICC'09. IEEE International Conference on*, pp. 1–5. IEEE, 2009.

[CGC11] Zhuo Chen, Jianxi Gao, Yunze Cai, and Xiaoming Xu. "Evolution of cooperation among mobile agents." *Physica A: Statistical Mechanics and its Applications*, **390**(9):1615–1622, 2011.

[CIH16] Kang Chen, Ryan Izard, Hongxin Hu, Kuang-Ching Wang, James Martin, and Juan Deng. "HetSDN: Exploiting SDN for intelligent network usage in heterogeneous wireless networks." In *Quality of Service (IWQoS), 2016 IEEE/ACM 24th International Symposium on*, pp. 1–6. IEEE, 2016.

[Cis17] Cisco. "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 20162021 White Paper.", 2017.

[CK12] Raymond Chiong and Michael Kirley. "Random mobility and the evolution of cooperation in spatial N-player iterated Prisoners Dilemma games." *Physica A: Statistical Mechanics and its Applications*, **391**(15):3915–3923, 2012.

[CLG13] Yung-Chih Chen, Yeon-sup Lim, Richard J Gibbens, Erich M Nahum, Ramin Khalili, and Don 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. ACM, 2013.

[CMG12] Alessio Cardillo, Sandro Meloni, Jesús Gómez-Gardeñes, and Yamir Moreno. "Velocity-enhanced cooperation of moving agents playing public goods games." *Physical Review E*, **85**(6):067101, 2012.

[CML11] Eunjoon Cho, Seth A Myers, and Jure Leskovec. "Friendship and mobility: user movement in location-based social networks." In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1082–1090. ACM, 2011.

[CS00] Y. Chotikapong and Z. Sun. "Evaluation of application performance for TCP/IP via satellite links." In *Satellite Services and the Internet (Ref. No. 2000/017), IEE Seminar on*, pp. 4/1–4/4, 2000.

[CWL10a] Yan Chen, Beibei Wang, W Sabrina Lin, Yongle Wu, and KJ Ray Liu. "Cooperative peer-to-peer streaming: An evolutionary game-theoretic approach." *IEEE Transactions on Circuits and Systems for Video Technology*, **20**(10):1346–1357, 2010.

[CWL10b] Yan Chen, Beibei Wang, W Sabrina Lin, Yongle Wu, and KJ Ray Liu. "Cooperative peer-to-peer streaming: An evolutionary game-theoretic approach." *IEEE Transactions on Circuits and Systems for Video Technology*, **20**(10):1346–1357, 2010.

[DG17a] Pengyuan Du and Mario Gerla. "An Evolutionary Multi-player Game Model for Two-hop Routing in Delay Tolerant Networks." In *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pp. 108–116. IEEE, 2017.

[DG17b] Pengyuan Du and Mario Gerla. "Promotion of cooperation in public goods game by Socialized Speed-Restricted movement." In *Wireless On-demand Network Systems and Services (WONS), 2017 13th Annual Conference on*, pp. 41–48. IEEE, 2017.

[DHD14] Florence Débarre, C Hauert, and M Doebeli. "Social evolution in structured populations." *Nature Communications*, **5**, 2014.

[DLL15] Pengyuan Du, Xiao Li, You Lu, and Mario Gerla. "Multipath TCP over LEO satellite networks." In *Wireless Communications and Mobile Computing Conference (IWCMC), 2015 International*, pp. 1–6. IEEE, 2015.

[DMC10] Francesco De Pellegrini, Daniele Miorandi, and Iacopo Carreras. "Optimal two-hop routing in delay-tolerant networks." In *Wireless Conference (EW), 2010 European*, pp. 881–888. IEEE, 2010.

[DSB14] Shuo Deng, Anirudh Sivaraman, and Hari Balakrishnan. "All your network are belong to us: A transport framework for mobile network selection." In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, p. 19. ACM, 2014.

[EDK10] Rachid El-Azouzi, Francesco De Pellegrini, and Vijay Kamble. "Evolutionary forwarding games in delay tolerant networks." In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2010 Proceedings of the 8th International Symposium on*, pp. 76–84. IEEE, 2010.

[EIS75] Shimon Even, Alon Itai, and Adi Shamir. "On the complexity of time table and multi-commodity flow problems." In *Foundations of Computer Science, 1975., 16th Annual Symposium on*, pp. 184–193. IEEE, 1975.

[EOS15] Barry Evans, Oluwakayode Onireti, Theodoros Spathopoulos, and Muhammad Ali Imran. "The role of satellites in 5G." In *Signal Processing Conference (EUSIPCO), 2015 23rd European*, pp. 2756–2760. IEEE, 2015.

[FAB15] Ramon R Fontes, Samira Afzal, Samuel HB Brito, Mateus AS Santos, and Christian Esteve Rothenberg. "Mininet-WiFi: Emulating software-defined wireless networks." In *Network and Service Management (CNSM), 2015 11th International Conference on*, pp. 384–389. IEEE, 2015.

[FGK73] Luigi Fratta, Mario Gerla, and Leonard Kleinrock. "The flow deviation method: An approach to store-and-forward communication network design." *Networks*, **3**(2):97–133, 1973.

[FHB06] Mark Felegyhazi, J-P Hubaux, and Levente Buttyan. "Nash equilibria of packet forwarding strategies in wireless ad hoc networks." *IEEE Transactions on Mobile computing*, **5**(5):463–476, 2006.

[FRH] A Ford, C Raiciu, M Handley, and O Bonaventure. "TCP extensions for multipath operation with multiple addresses. RFC6824, 2014.".

[FRH11a] Alan Ford, Costin Raiciu, Mark Handley, Sebastien Barre, Janardhan Iyengar, et al. "Architectural guidelines for multipath TCP development." *IETF, Informational RFC*, **6182**:2070–1721, 2011.

[FRH11b] Alan Ford, Costin Raiciu, Mark Handley, Olivier Bonaventure, et al. "TCP extensions for multipath operation with multiple addresses." *IETF MPTCP proposal-http://tools. ietf. org/id/draft-ford-mptcp-multiaddressed-03. txt*, 2011.

[FYG15] Ruolin Fan, Yu-Ting Yu, and Mario Gerla. "RobustGeo: A disruption-tolerant geo-routing protocol." In *Computer Communication and Networks (ICCCN), 2015 24th International Conference on*, pp. 1–8. IEEE, 2015.

[GA08] Anargyros Garyfalos and Kevin C Almeroth. "Coupons: A multilevel incentive scheme for information dissemination in mobile networks." *IEEE Transactions on Mobile Computing*, **7**(6):792–804, 2008.

[GCF13] Anatolij Gelimson, Jonas Cremer, and Erwin Frey. "Mobility, fitness collection, and the breakdown of cooperation." *Physical Review E*, **87**(4):042711, 2013.

[GDM12] Negin Golrezaei, Alexandros G Dimakis, and Andreas F Molisch. "Device-to-device collaboration through distributed storage." In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pp. 2397–2402. IEEE, 2012.

[GHB08] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. "Understanding individual human mobility patterns." *Nature*, **453**(7196):779–782, 2008.

[GT10] Chaitanya S Gokhale and Arne Traulsen. "Evolutionary games in the multiverse." *Proceedings of the National Academy of Sciences*, **107**(12):5500–5504, 2010.

[HCY11] Pan Hui, Jon Crowcroft, and Eiko Yoneki. "Bubble rap: Social-based forwarding in delay-tolerant networks." *IEEE Transactions on Mobile Computing*, **10**(11):1576–1589, 2011.

[HHK12] Bo Han, Pan Hui, VS Anil Kumar, Madhav V Marathe, Jianhua Shao, and Aravind Srinivasan. "Mobile data offloading through opportunistic communications and social participation." *IEEE Transactions on Mobile Computing*, **11**(5):821–834, 2012.

[HKM13] Chi-Yao Hong, Srikanth Kandula, Ratul Mahajan, Ming Zhang, Vijay Gill, Mohan Nanduri, and Roger Wattenhofer. "Achieving high utilization with software-driven WAN." In *ACM SIGCOMM Computer Communication Review*, volume 43, pp. 15–26. ACM, 2013.

[HNA06] Ahmad Al Hanbali, Philippe Nain, and Eitan Altman. "Performance of ad hoc networks with two-hop relay routing and limited packet lifetime." In *Proceedings of the 1st international conference on Performance evaluation methodolgies and tools*, p. 49. ACM, 2006.

[Hop00] Christian E Hopps. "Analysis of an equal-cost multi-path algorithm." 2000.

[HRB09] M Handley, C Raiciu, and M Bagnulo. "Outgoing packet routing with MP-TCP." *draft-handley-mptcp-routing-00 (work in progress)*, 2009.

[HRX08] Sangtae Ha, Injong Rhee, and Lisong Xu. "CUBIC: a new TCP-friendly high-speed TCP variant." *ACM SIGOPS Operating Systems Review*, **42**(5):64–74, 2008.

[HS03] Josef Hofbauer and Karl Sigmund. "Evolutionary game dynamics." *Bulletin of the American Mathematical Society*, **40**(4):479–519, 2003.

[HSL11] Theus Hossmann, Thrasyvoulos Spyropoulos, and Franck Legendre. "A complex network analysis of human mobility." In *Computer communications workshops (INFOCOM WKSHPS), 2011 IEEE conference on*, pp. 876–881. IEEE, 2011.

[IGH17] George Iosifidis, Lin Gao, Jianwei Huang, and Leandros Tassiulas. "Efficient and fair collaborative mobile internet access." *IEEE/ACM Transactions on Networking (TON)*, **25**(3):1386–1400, 2017.

[Jac88] Van Jacobson. "Congestion avoidance and control." In *ACM SIGCOMM Computer Communication Review*, volume 18, pp. 314–329. ACM, 1988.

[JCL14] Chunxiao Jiang, Yan Chen, and KJ Ray Liu. "Evolutionary dynamics of information diffusion over social networks." *IEEE Transactions on Signal Processing*, **62**(17):4573–4586, 2014.

[Kel03] Tom Kelly. "Scalable TCP: Improving performance in highspeed wide area networks." *ACM SIGCOMM Computer Communication Review*, **33**(2):83–91, 2003.

[Ker93] Aaron Kershenbaum. *Telecommunications network design algorithms*. McGraw-Hill, Inc., 1993.

[KF13] Hyojoon Kim and Nick Feamster. "Improving network management with software defined networking." *IEEE Communications Magazine*, **51**(2):114–119, 2013.

[KGP12] Ramin Khalili, Nicolas Gast, Miroslav Popovic, Utkarsh Upadhyay, and Jean-Yves Le Boudec. "MPTCP is not pareto-optimal: performance issues and a possible solution." In *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, pp. 1–12. ACM, 2012.

[KLV10] Thomas Karagiannis, Jean-Yves Le Boudec, and Milan Vojnovic. "Power law and exponential decay of intercontact times between mobile devices." *IEEE Transactions on Mobile Computing*, **9**(10):1377–1390, 2010.

[KMS10] Sokol Kosta, Alessandro Mei, and Julinda Stefa. "Small world in motion (SWIM): Modeling communities in ad-hoc mobile networking." In *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on*, pp. 1–9. IEEE, 2010.

[KRV15] Diego Kreutz, Fernando MV Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. "Software-defined networking: A comprehensive survey." *Proceedings of the IEEE*, **103**(1):14–76, 2015.

[KS17] Udit Narayana Kar and Debarshi Kumar Sanyal. "An overview of device-to-device communication in cellular networks." *ICT Express*, 2017.

[KX05] Dimitris Karlis and Evdokia Xekalaki. "Mixed poisson distributions." *International Statistical Review*, **73**(1):35–58, 2005.

[LE10] Chul-Ho Lee and Do Young Eun. "Exploiting heterogeneity in mobile opportunistic networks: An analytic approach." In *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on*, pp. 1–9. IEEE, 2010.

[LGC14] Yi-Shing Liou, Rung-Hung Gau, and Chung-Ju Chang. "A bargaining game based access network selection scheme for HetNet." In *Communications (ICC), 2014 IEEE International Conference on*, pp. 4888–4893. IEEE, 2014.

[LLR18] Daniela Laselva, David Lopez-Perez, Mika Rinne, and Tero Henttonen. "3GPP LTE-WLAN Aggregation Technologies: Functionalities and Performance Comparison." *IEEE Communications Magazine*, **56**(3):195–203, 2018.

[LS12] Ze Li and Haiying Shen. "Game-theoretic analysis of cooperation incentive strategies in mobile ad hoc networks." *IEEE Transactions on Mobile Computing*, **11**(8):1287–1303, 2012.

[LZ10] Hao Liang and Weihua Zhuang. "DTCoop: delay tolerant cooperative communications in DTN/WLAN integrated networks." In *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, pp. 1–5. IEEE, 2010.

[LZG16] Yuqing Li, Jinbei Zhang, Xiaoying Gan, Luoyi Fu, Hui Yu, and Xinbing Wang. "A contract-based incentive mechanism for delayed traffic offloading in cellular networks." *IEEE Transactions on Wireless Communications*, **15**(8):5314–5327, 2016.

[MAB08] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. "OpenFlow: enabling innovation in campus networks." *ACM SIGCOMM Computer Communication Review*, **38**(2):69–74, 2008.

[MCG01] Saverio Mascolo, Claudio Casetti, Mario Gerla, Medy Y Sanadidi, and Ren Wang. "TCP Westwood: Bandwidth estimation for enhanced transport over wireless links." In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pp. 287–297. ACM, 2001.

[MVS14] Rajesh Mahindra, Hari Viswanathan, Karthik Sundaresan, Mustafa Y Arslan, and Sampath Rangarajan. "A practical traffic management system for integrated LTE-WiFi networks." In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pp. 189–200. ACM, 2014.

[MW11] Christoph P Mayer and Oliver P Waldhorst. "Offloading infrastructure using delay tolerant networks and assurance of delivery." In *Wireless Days (WD), 2011 IFIP*, pp. 1–7. IEEE, 2011.

[NCD98] Paolo Narvaez, Antonio Clerget, Walid Dabbous, et al. "Internet routing over LEO satellite constellations." In *Third ACM/IEEE International Workshop on Satellite-Based Information Services (WOSBIS98)*. Citeseer, 1998.

[Now06] Martin A Nowak. "Five rules for the evolution of cooperation." *science*, **314**(5805):1560–1563, 2006.

[NS04] Martin A Nowak and Karl Sigmund. "Evolutionary dynamics of biological games." *science*, **303**(5659):793–799, 2004.

[NSD13] Shobanraj Navaratnarajah, Arsalan Saeed, Mehrdad Dianati, and Muhammad Ali Imran. "Energy efficiency in heterogeneous wireless access networks." *IEEE wireless communications*, **20**(5):37–43, 2013.

[NTA10] Martin A Nowak, Corina E Tarnita, and Tibor Antal. "Evolutionary dynamics in structured populations." *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, **365**(1537):19–30, 2010.

[Nug16] Pavan Nuggehalli. "LTE-WLAN aggregation [Industry Perspectives]." *IEEE Wireless Communications*, **23**(4):4–6, 2016.

[OHL06] Hisashi Ohtsuki, Christoph Hauert, Erez Lieberman, and Martin A Nowak. "A simple rule for the evolution of cooperation on graphs and social networks." *Nature*, **441**(7092):502–505, 2006.

[OSS14] Rogério Leão Santos de Oliveira, Ailton Akira Shinoda, Christiane Marie Schweitzer, and Ligia Rodrigues Prete. "Using mininet for emulation and prototyping software-defined networks." In *Communications and Computing (COLCOM), 2014 IEEE Colombian Conference on*, pp. 1–6. IEEE, 2014.

[Ost08] Elinor Ostrom. "Tragedy of the Commons." *The New Palgrave Dictionary of Economics*, pp. 3573–3576, 2008.

[Pap] Luca Pappalardo. "The origin of heterogeneity in human mobility ranges.".

[PBS14] Francesco Pantisano, Mehdi Bennis, Walid Saad, and Mérouane Debbah. "In-network caching and content placement in cooperative small cell networks." In *5G for Ubiquitous Connectivity (5GU), 2014 1st International Conference on*, pp. 128–133. IEEE, 2014.

[PC99] Madan M Pillutla and Xiao-Ping Chen. "Social norms and cooperation in social dilemmas: The effects of context and feedback." *Organizational behavior and human decision processes*, **78**(2):81–103, 1999.

[PDD12] Christoph Paasch, Gregory Detal, Fabien Duchene, Costin Raiciu, and Olivier Bonaventure. "Exploring mobile/WiFi handover with multipath TCP." In *Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design*, pp. 31–36. ACM, 2012.

[PFA14] Christoph Paasch, Simone Ferlin, Ozgu Alay, and Olivier Bonaventure. "Experimental evaluation of multipath TCP schedulers." In *Proceedings of the 2014 ACM SIGCOMM workshop on Capacity sharing workshop*, pp. 27–32. ACM, 2014.

[PGS13] Matjaž Perc, Jesús Gómez-Gardeñes, Attila Szolnoki, Luis M Floría, and Yamir Moreno. "Evolutionary dynamics of group interactions on structured populations: a review." *Journal of The Royal Society Interface*, **10**(80):20120997, 2013.

[PKP07] Evangelos Papapetrou, Stylianos Karapantazis, and F-N Pavlidou. "Distributed on-demand routing for LEO satellite systems." *Computer Networks*, **51**(15):4356–4376, 2007.

[PSS17] Ioannis Psaras, Vasilis Sourlas, Denis Shtefan, Sergi Rene, Mayutan Arumaithurai, Dirk Kutscher, and George Pavlou. "On the feasibility of a user-operated mobile content distribution network." In *A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2017 IEEE 18th International Symposium on*, pp. 1–9. IEEE, 2017.

[PWA16] Jorge Peña, Bin Wu, Jordi Arranz, and Arne Traulsen. "Evolutionary games of multiplayer cooperation on graphs." *PLoS Comput Biol*, **12**(8):e1005059, 2016.

[PWL13] Qiuyu Peng, Anwar Walid, and Steven H Low. "Multipath TCP: Analysis and Design." *CoRR*, 2013.

[PWT16] Jorge Peña, Bin Wu, and Arne Traulsen. "Ordering structured populations in multiplayer cooperation games." *Journal of the Royal Society Interface*, **13**(114):20150881, 2016.

[RC15] Vitor G Rolla and Marilia Curado. "A Simple Survey of Incentive Mechanisms for User-Provided Networks." *Wireless Personal Communications*, **83**(4):2579–2591, 2015.

[RHW11] C Raiciu, M Handley, and D Wischik. "Coupled congestion control for multipath transport protocols." *draft-ietf-mptcp-congestion-01 (work in progress)*, 2011.

[RYW10] Zhihai Rong, Han-Xin Yang, Wen-Xu Wang, et al. "Feedback reciprocity mechanism promotes the cooperation of highly clustered scale-free networks." *Physical Review E*, **82**(4):047101, 2010.

[Sax14] Piyush Saxena. "Satellite Communication Advancement, Issues, Challenges and Applications." *International Journal of Research*, **1**(10):1172–1176, 2014.

[SJ15] Smitha Shivshankar and Abbas Jamalipour. "An Evolutionary Game Theory-Based Approach to Cooperation in VANETs Under Different Network Conditions." *IEEE Transactions on Vehicular Technology*, **64**(5), 2015.

[SKT13] Kohei Sugiyama, Takeshi Kubo, Atsushi Tagami, and Abhay Parekh. "Incentive mechanism for DTN-based message delivery services." In *Global Communications Conference (GLOBECOM), 2013 IEEE*, pp. 3108–3113. IEEE, 2013.

[SPR05] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S Raghavendra. "Spray and wait: an efficient routing scheme for intermittently connected mobile networks." In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pp. 252–259. ACM, 2005.

[SSP08] Francisco C Santos, Marta D Santos, and Jorge M Pacheco. "Social diversity promotes the emergence of cooperation in public goods games." *Nature*, **454**(7201):213–216, 2008.

[SXK13] Amanpreet Singh, Mei Xiang, A Konsgen, Carmelita Goerg, and Yasir Zaki. "Enhancing fairness and congestion control in multipath TCP." In *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP*, pp. 1–8. IEEE, 2013.

[TCK13] Alvin Ting, David Chieng, Kae Hsiang Kwong, Ivan Andonovic, and KD Wong. "Dynamic backhaul sensitive network selection scheme in LTE-WiFi wireless HetNet." In *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, pp. 3061–3065. IEEE, 2013.

[TKD17] Sacha Trifunovic, Sylvia T Kouyoumdjieva, Bernhard Distl, Ljubica Pajevic, Gunnar Karlsson, and Bernhard Plattner. "A decade of research in opportunistic networks: challenges, relevance, and future directions." *IEEE Communications Magazine*, **55**(1):168–173, 2017.

[TLB09] P-U Tournoux, Jérémie Leguay, Farid Benbadis, Vania Conan, M Dias De Amorim, and John Whitbeck. "The accordion phenomenon: Analysis, characterization, and impact on dtn routing." In *INFOCOM 2009, IEEE*, pp. 1116–1124. IEEE, 2009.

[UBG03] A Urpi, M Bonuccelli, and Silvia Giordano. "Modelling cooperation in mobile ad hoc networks: a formal description of selfishness." In *WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pp. 10–pages, 2003.

[VB00] Amin Vahdat, David Becker, et al. "Epidemic routing for partially connected ad hoc networks." 2000.

[WCH14] Xiaofei Wang, Min Chen, Zhu Han, Dapeng Oliver Wu, and Ted Taekyoung Kwon. "TOSS: Traffic offloading by social network service-based opportunistic sharing in mobile social networks." In *INFOCOM, 2014 Proceedings IEEE*, pp. 2346–2354. IEEE, 2014.

[Wei97] Jörgen W Weibull. *Evolutionary game theory*. MIT press, 1997.

[Wil90] Gordon E Willmot. "Asymptotic tail behaviour of Poisson mixtures with applications." *Advances in Applied Probability*, pp. 147–159, 1990.

[WJL95] Markus Werner, Axel Jahn, Erich Lutz, and Axel Bottcher. "Analysis of system parameters for LEO/ICO-satellite communication networks." *Selected Areas in Communications, IEEE Journal on*, **13**(2):371–381, 1995.

[WRG11] Damon Wischik, Costin Raiciu, Adam Greenhalgh, and Mark Handley. "Design, Implementation and Evaluation of Congestion Control for Multipath TCP." In *NSDI*, volume 11, pp. 8–8, 2011.

[WXH13] Jie Wu, Mingjun Xiao, and Liusheng Huang. "Homing spread: Community home-based multi-copy routing in mobile social networks." In *INFOCOM, 2013 Proceedings IEEE*, pp. 2319–2327. IEEE, 2013.

[ZGC11] Xuejun Zhuo, Wei Gao, Guohong Cao, and Yiqi Dai. "Win-Coupon: An incentive framework for 3G traffic offloading." In *Network Protocols (ICNP), 2011 19th IEEE International Conference on*, pp. 206–215. IEEE, 2011.

[ZGL14] Jun Zhang, Vincent Gauthier, Houda Labiod, Abhik Banerjee, and Hossam Afifi. "Information dissemination in vehicular networks via evolutionary game theory." In *2014 IEEE International Conference on Communications (ICC)*, pp. 124–129. IEEE, 2014.

[ZS14] Dizhi Zhou and Wei Song. *Multipath TCP for User Cooperation in Wireless Networks*. Springer, 2014.

[ZSM10] Dan Zhang, Ryoichi Shinkuma, and Narayan B Mandayam. "Bandwidth exchange: An energy conserving incentive mechanism for cooperation." *IEEE Transactions on Wireless Communications*, **9**(6), 2010.

[ZWW14] Huanyang Zheng, Yunsheng Wang, and Jie Wu. "Optimizing multi-copy two-hop routing in mobile social networks." In *Sensing, Communication, and Networking (SECON), 2014 Eleventh Annual IEEE International Conference on*, pp. 573–581. IEEE, 2014.

[ZZG10] Xiaolan Zhang, Honggang Zhang, and Yu Gu. "Impact of source counter on DTN routing control under resource constraints." In *Proceedings of the Second International Workshop on Mobile Opportunistic Networking*, pp. 41–50. ACM, 2010.