Content Sharing and Distribution in Wireless Community Networks

by

Amr Alasaad

B.Sc., King Saud University, 2000
M.Sc., The University of Southern California, 2005

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate Studies

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

April 2013

Abstract

We consider the problem of content sharing and distribution in a wireless mesh community network (WMCN). Due to the community oriented nature of such networks, and with the evolution of advanced mobile computing devices; it is projected that the demand for content sharing and distribution in wireless community networks will dramatically increase in the coming years. A popular scheme for content sharing and distribution is through the use of Peer-to-Peer (P2P) technology. This dissertation studies the technical challenges involved while deploying P2P applications over WMCNs.

We advance the thesis that support from a number of infrastructure nodes in a wireless community network to P2P applications running on top of the wireless community network such as P2P content sharing and P2P media streaming, results in significant performance enhancement. Such support from infrastructure nodes benefits from awareness of the underlying network topology (i.e., information available at relay nodes about the true physical connections between nodes in the network). Cross-layer information exchange between the P2P system and the WMCN opens the door to developing efficient algorithms and schemes for P2P communications that account for the specific features of WMCNs (e.g., contention for wireless medium between neighbouring nodes and traffic interference). Moreover, such support benefits from the underutilized resources (e.g., storage and bandwidth resources) at a large number of infrastructure nodes in the WMCN. This creates the possibility of replicating into the caches of those under-utilized infrastructure nodes P2P contents that are desired by the community, and enabling those infrastructure nodes to participate in content distribution and play the role of helpers in P2P content sharing and distribution.

Our main contribution in this dissertation is P2P-with-helpers: Hybrid approaches for content sharing and distribution in a wireless community network that rely on support from infrastructure nodes to the P2P applications. We show that P2P-with-helpers approaches significantly enhance the performance of P2P content sharing and distribution in WMCNs. The performance enhancements reflect aspects such as average content download times, bandwidth and energy consumption in the network, Internet bandwidth cost, and traffic load imbalance in the network.

Preface

This thesis is based on the research work conducted at the Department of Electrical and Computer Engineering at UBC under supervision of Dr. Victor C.M. Leung and Dr. Sathish Gopalakrishnan. I am the principal author of all chapters in this dissertation.

Chapter 2 of this dissertation has been partially published in the proceedings of the IEEE Globecom 2010 Conference and IEEE PACRIM 2009 Conference, and in the journal of Wireless Communications and Mobile Computing (cf. the list of articles and papers below).

Chapter 3 has been partially published in proceedings of the IEEE PIMRC 2011 and IEEE Globecom 2010 Workshop on Ubiquitous Computing and Networks, and is to partially appear in the journal of Wireless Communications and Mobile Computing (cf. the list of articles and papers below).

Chapter 4 of this dissertation has been partially published in the proceedings of the IEEE ICC GCN 2012 Workshop, and in the journal of Peer-to-Peer Networking and Applications, special issue on Peer-to-Peer as infrastructure service (cf. the list of articles and papers below).

Chapter 5 of this dissertation is to partially appear in the journal of Telecommunication systems, special issue on innovations in emerging multimedia communication systems (cf. the list of articles and papers below).

Chapter 6 of this dissertation has been partially published in the proceedings of the IEEE PIMRC 2009, and in the journal of ACM/Springer Wireless networks (cf. the list of articles and papers below).

The research related to these publications was conducted by myself under supervision of Dr. Victor C.M. Leung and Dr. Sathish Gopalakrishnan. The research ideas, analytical evaluations, mathematical modelling, and simulation programs are all the results of my work. My supervisors helped with discussions, providing research guidance, defining the problems, checking the validity of my analytical and simulation results, and proofreading the respective conference papers and journal articles. One of the papers used in Chapter 6 was co-authored with Mr. Hasen Nicanfar, who is also a PhD candidate in the Department of Electrical and Computer Engineering at UBC. Mr. Nicanfar helped in setting up the simulation environment used to evaluate the ideas proposed in Chapter 6.

In the following, a list of articles and papers where the results in this thesis have been partially published/accepted is provided. In some cases, the conference papers contain materials overlapping with the journal papers.

Journal Papers Accepted/Published:

- (Part of Chapter 2) A. Alasaad, S. Gopalakrishnan, and V. C.M. Leung, "Extending P2PMesh: Topology-aware schemes for efficient Peer-to-Peer data sharing in wireless mesh networks," in *Wireless Communications and Mobile Computing*, John Wiley & Sons, Ltd., vol. 13, no. 5, pp. 483–499, 2013.

- (Part of Chapter 3) A. Alasaad, S. Gopalakrishna,n and V. C.M. Leung, "Replication schemes for Peer-to-Peer content in wireless mesh networks with infrastructure support," accepted for publication in *Wireless Communications and Mobile Computing*, John Wiley & Sons, Ltd.

- (Part of Chapter 4) A. Alasaad, S. Gopalakrishnan, and V. C.M. Leung, "Modelling and performance analysis of content sharing and distribution on community networks with infrastructure support," in *Peer-to-Peer Networking and Applications*, special issue on Peer-to-Peer as infrastructure service, Springer New York, published online in August 2012. DOI: 10.1007/s12083-012-0167-1.

- (Part of Chapter 5) A. Alasaad, S. Gopalakrishnan, and V. C.M. Leung, "A hybrid approach for cost-effective media streaming based on prediction of demand in community networks," accepted for publication in *Telecommunication Systems*, special issue on innovations in emerging multimedia communication systems, Springer New York.

- (Part of Chapter 6) A. Alasaad, H. Nicanfar, S. Gopalakrishnan, and V. C.M. Leung, "A ring-based multicast network topology with QoS support in wireless mesh networks," in *ACM/Springer Wireless Networks*, published online in March 2013. DOI:10.1007/s11276-013-0559-z.

Conference Papers Published:

- (Part of Chapter 2) A. Alasaad, S. Gopalakrishnan, and V. C.M. Leung, "Peer-to-Peer file sharing over wireless mesh networks," in *Proc. of the IEEE PACRIM Conference*, pp. 697–702, Victoria, BC, August 2009.

- (Part of Chapter 2) A. Alasaad, S. Gopalakrishnan, and V. C.M. Leung, "Mitigating load imbalance in wireless mesh networks with mixed application traffic types," in *Proc. of the IEEE Globecom Conference*, pp. 1–5, Miami, FL, December 2010.

- (Part of Chapter 3) A. Alasaad, S. Gopalakrishnan, and V. C.M. Leung, "Replication schemes for Peer-to-Peer content in wireless mesh networks with network support," in *Proc. of the IEEE PIMRC Conference*, pp. 1135–1139, Toronto, ON, September 2011.

- (Part of Chapter 3) A. Alasaad, S. Gopalakrishnan, and V. C.M. Leung, "Content caching and replication schemes for Peer-to-Peer file sharing in wireless mesh networks," in *Proc. of the IEEE Globecom Workshop on Ubiquitous Computing and Networks*, pp. 1707–1711, Miami, FL, December 2010.

- (Part of Chapter 4) A. Alasaad, S. Gopalakrishnan, and V. C.M. Leung, "Green content distribution in wireless mesh networks," in *Proc. of the IEEE ICC GCN Workshop*, pp. 5896–5900, Ottawa, ON, June 2012.

- (Part of Chapter 6) A. Alasaad, S. Gopalakrishnan, and V. C.M. Leung, "An architecture with QoS support for application layer multicasting over wireless mesh networks," in *Proc. of the IEEE PIMRC Conference*, pp. 1562–1566, Tokyo, Japan, September 2009.

Table of Contents

Appendices

List of Tables

List of Figures

List of Acronyms

P2P Peer-to-Peer

WMCN Wireless Mesh Community Network

WMN Wireless Mesh Network

MR Mesh Router

MC Mesh Client

ISP Internet Service Provide

CDN Content Distribution Network

TCP Transmission Control Protocol

IP Internet Protocol

MANET Mobile Ad-hoc Network

AODV Ad hoc On-Demand Distance Vector Routing

DSR Dynamic Source Routing

Acknowledgements

I would like to profoundly thank my supervisors Dr. Sathish Gopalakrishnan and Dr. Victor C.M. Leung for their continuous support, encouragement and guidance from the early stages of my research until the very end. Without their wisdom and valuable suggestions, this work would have not materialized. I am truly honoured to be supervised by pioneer researchers in the field of computer networks and wireless communications. I would like to also thank my supervisory committee members, Dr. Vincent Wong and Dr. Matei Ripeanu; and my university examiners, Dr. Mike Feeley and Dr. Cyril Leung; and my external examiner Dr. Hossam S. Hassanein from Queen's University for their constructive comments and feedback on the work of this dissertation.

My dear wife, Sarah, is singularly responsible for keeping my spirits up when nothing seemed to work. My parents gave me the gift of education and a sense of responsibility. Many thanks to my parents-in-law for their affection and the confidence they have placed in me.

I am also thankful to my fellow colleagues and friends at the Department of Electrical and Computer Engineering at The University of British Columbia who inspired me along the way with their constructive comments and feedback, especially Dr. Ali Alshidhani, Dr. Kaveh Shafiee, Mr. Hasen Nicanfar, Mr. Peyman Talebifard, and Mr. Geoffrey Lo. I also would like to extend my thanks and gratitude to all the friends I met in Vancouver. Each and every one of them supported me in one way or the other to complete this work.

This work was supported in part by the King Abdulaziz City for Science and Technology, Saudi Cultural Bureau in Canada, and NSERC under Discovery Grant RGPIN

342751-07.

Dedication

To my late Dad, words cannot describe how lucky I am to be your son

To my wife and my Mom, your unconditional love and support is what keeps me going

To my two lovely sons, the joy you brought to my life is second to none

Chapter 1

Introduction

Community initiated wireless mesh networks have emerged in many metropolitan areas worldwide, mainly driven by the low cost of IEEE 802.11-capable equipments, flexible structure, and their operation in license-free spectrum. A wireless mesh community network (WMCN) relies on wireless mesh network technologies to connect and deliver traffic between nodes in the network [1–3].

Users in a wireless mesh community network share their communication facilities (e.g., home networks and IEEE 802.11/Wi-Fi wireless access points) and form a Wireless Mesh Network (WMN) to be used by all community members (e.g., Microsoft project entitled Self Organizing Wireless Mesh Networks [4]).

1.1 Architecture of WMCNs

The dominant design approach for a wireless mesh community network deployment is a two-tier architecture, wherein an access tier connects mobile end-user computing terminals (called Mesh Clients, MCs) to stationary infrastructure nodes (called Mesh Routers, MRs), and the mesh routers form a mesh wireless backhaul tier that routes data packets between mesh clients within the WMCN and between mesh clients and gateways that are wired to the Internet [5–10] (Figure 1.1). Currently, mesh routers have been equipped with multiple radios which allow them to send and receive on multiple channels in parallel and consequently increase network capacity. The WMCN is usually connected to the Internet via a transit link to a higher level network service provider.

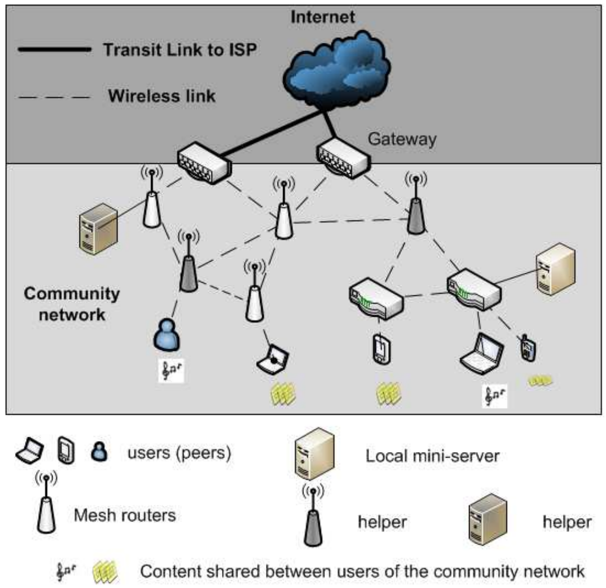


Figure 1.1: Wireless mesh community network architecture

There are many advantages of forming a wireless mesh community network. For example, when enough neighbours cooperatively use their wireless home networks to forward each other traffic and form a wireless mesh community network in a neighbourhood, neighbours do not need to individually install an Internet gateway, but instead can share a faster, cost-effective Internet access via few gateways that are distributed in their neighbourhood. WMN technologies have inspired many novel application scenarios in addition to the wireless mesh community and enterprise networking such as broadband home networking and area surveillance [11], temporary infrastructure in disaster and emergency situation [12], building and industrial automation [13], traffic control [14], sensor monitoring systems [15], and high-speed metropolitan area network [16].

1.2 Wireless Communities Around the World

Promising examples of a city-wide WMCN are Seattle Wireless [16], NYCwireless [17], CuWiN [2], OpenAirBoston [18], Houston WMN [10], FON community [19], and Athens wireless metropolitan network. In the case of Athens wireless, for example, the network comprises 1120 backbone nodes (mesh routers) (as of August, 2010) and more than 2900 client computers (mesh clients) connect to it. More than 9,000 people have stated their intention to join the network [20]. Wireless municipal networks are similar networks that are in the early stages of deployment (e.g., Fred-e Zone [21]).

1.3 Services and Applications in WMCNs

The advances in WMN technologies (e.g., routing algorithms, cognitive radios, nano-tube radios, multi-channel/multi-radio) enable WMCNs to provide innovative services for the end-users such as delay-sensitive services (e.g., Voice over Internet Protocol VoIP and media streaming) and delay-insensitive services (e.g., content sharing) [7, 22]. It has been shown that the demand for content sharing and video streaming is growing in the Internet [23, 24]. Due to the community oriented nature of such networks, users in a WMCN may like to deploy community related services such as content sharing (e.g., movies, music, software updates, popular web content) and video streaming. Hence, it is projected that the demand for content sharing and distribution in wireless community networks will increase in the coming years [25–27]. A. Pantelis *et al.* surveyed a large number of WMCNs around the world, and found that file sharing via Bittorrent tops the list of the most popular services among users of WMCNs. They also found that VoIP services, video streaming are offered as well [26].

1.4 Content Sharing in WMCNs

The proliferation of mobile computing devices, which enable users to produce digital content (especially multimedia content) anywhere and at any time, increases users appetite to generate and share content in the wireless mesh community network [26, 27]. Since WMCNs use the license-free spectrum, mobile users can exchange content over the free

WMCN rather than using a carrier frequency such as 3G and WiMax. The wireless mesh community network allows bits created locally to be used locally without having to go through a service provider and the Internet. This contains the cost of sharing and distributing content within the WMCN, and allows faster and easier dissemination of cached information that is relevant to the community. In urban areas, WMCNs may coexist with other networks (e.g., LTE). In this case, the LTE service provider may be happy with WMCN because it helps mitigating the traffic load in areas where the WMCNs are deployed. This consequently reduces resource consumptions at the LTE network

The common approaches to sharing content in a community network (e.g., college or office campus) is through the use of a centralized storage server (client-server scheme) or using services offered in the Internet (e.g., Google Drive, Dropbox, YouTube). A centralized storage server in a community network can be used for content sharing. Interested users can download content directly from the server using the client-server scheme. Since content demand grows quickly in a community network as every user contacts other users and make them interested in the content, the community network operator needs to invest in significant computing resources and networking bandwidth at the centralized storage server in order to cope with the viral evolution of content demand. However, a WMCN is self-organized and decentralized due to the nature of its distributed components designed to operate in dynamic network environments and hop-by-hop connection establishment. In networks that are owned and operated by users such as WMCNs, all users are seen as equal. This creates significant issues when deploying and financing a dedicated server-based content sharing system (caching infrastructure) with large computing, storage, and networking resources as the nature of such networks makes it difficult to apportion costs or retrieve monetary contributions. Moreover, traffic congestion and contention for the wireless medium at the centralized server deteriorate the performance of the centralized client-server approach. For these reasons, decentralized (distributed) designs are required for services offered by WMCNs.

A wireless community network is often connected to the Internet via a transit link to a higher level network service provider such as ISP. Thus, using a service offered in the Internet (e.g., Google Drive, Dropbox, YouTube, CoolStreaming) for sharing and distributing content in a community network needs to repeatedly fetch information from external sources (e.g. external servers or externally deployed peer-to-peer streaming services). This results in poor performance (e.g., channel zapping and large buffering times), high traffic congestion at the Internet gateways, and high monetary cost of Internet bandwidth (egress bandwidth) charged by the ISP for traffic delivered to users of the community network from the Internet (95th percentile billing is mainly used for Internet connections that are provided as "burstable" (variable rate) bandwidth) [28, 29]. Therefore, a scheme that contains the content sharing traffic within the community network is desirable. Although this scheme eliminates the capital investment required in the case of centralized storage server, we have operational cost required for renting a large storage capacity from the cloud provider when using a scheme like Google Drive or Dropbox (the monthly cost for upgrading Google Drive to 1TB is $49.99). This cost, although not exorbitant by the standards of developed economies, it creates a burden for low-income communities or for wireless community networks which run and operate by users.

A popular distributed scheme for content sharing and distribution is through the use of Peer-to-Peer (P2P) technology, by exploiting the upload capacity of peers who are interested in the same content (e.g., BitTorrent). Users of a community network are socially connected (e.g., neighbours or friends). Therefore, they can cooperatively deploy free distributed services desired by the community such as Peer-to-Peer (P2P) data backup (e.g., community file storage service - Wuala, Zoogmo, Cucku, CrashPlan), P2P community network resource sharing (e.g., CPU cycle sharing - BOINC, SETI, Community Grid [30]), P2P content sharing (e.g., BitTorrent), P2P distribution system for software package releases and updates (e.g., apt-p2p [31]), and P2P media streaming (e.g., PPLive, CoolStreaming, SopCast, Skype, EVE Community). These distributed schemes enable free content sharing and distribution between users within the community network.

Table 1.1 summarizes various performance aspects of different content sharing paradigms in WMCNs.

Table 1.1: Comparisons between different content sharing paradigms in WMCNs

Performance metric	Dedicated server	P2P	Internet services
Architecture	Centralized	Distributed	Cloud
Capital	Large	None	None
Operational cost	Large	None	Large
Internet bandwidth cost	None	None	Large
Traffic load imbalance	High	Low	High
Traffic congestion	High	Low	High
Scalability	Bad	Good	Good
ISP friendly	Good	Good	Bad

1.5 P2P Communication in WMCNs

The common characteristics shared by both WMCNs and P2P systems such as self-organization and decentralization strongly suggest that P2P communication is the most feasible scheme for content sharing and distribution in WMCNs. Moreover, the performance of P2P content distribution (content download time) scales well with number of interested users in the network because users exchange data with each others. Furthermore, P2P schemes for content sharing mitigate traffic load imbalance in the network since a downloading peer retrieves content from multiple content providers at different locations in the network (Table 1.1).

1.5.1 Fundamental Challenges of P2P Communication in WMCNs

Schemes used in conventional P2P systems are implemented at the application layer. Thus, these schemes are not cognizant of the physical network topology, and there is no consideration of the negative impacts on the underlying network [32–35]. For example, a downloading peer in a conventional P2P system randomly selects a content provider that may not be physically close to the downloading peer. Consequently, the established download path may consist of large number of hops which leads to large amount of bandwidth and energy consumption in the network.

There are extra challenges when enabling P2P content sharing service in a WMCN. Due to the broadcast nature of wireless transmissions, we have contention for the wireless medium between neighbouring mesh routers, and interference between traffic on adjacent wireless links [36, 37]. Moreover, establishing inefficient download paths that consist of large number of wireless hops degrades the performance of transport protocols such as TCP [35]. Therefore, traditional P2P algorithms are inefficient when used over a limited resource network such as the WMCN where network topology information is valuable. Therefore, topology-aware schemes for P2P communication allow for efficient establishment of download paths, content lookup, and content retrieval. Also, effective provision of internal content caching and replication are critical to the performance of content sharing and distribution in the WMCN. Increasing number of replicas for a content increases the likelihood that a downloading peer locates a replica of the required content at a content provider nearby. Thus, content caching and replication reduces both contention for the wireless medium between mesh routers and traffic interference. This consequently reduces the amount of bandwidth and energy consumption in the network, while increases throughput of the network.

1.6 Research Motivation

As we have discussed so far, P2P content sharing and distribution over a WMCN represent exciting possibilities, but at the same time several challenges need to be addressed to enable peers to consume the limited network resources efficiently. We advance the thesis that support for P2P applications at infrastructure nodes (e.g., mesh routers) results in significant performance improvements. The static nature of infrastructure nodes in a WMCN, which typically have low probability of leaving the network, and higher upload bandwidth as compared to end-users (peers) motivated us to investigate the role infrastructure nodes can play to support P2P applications. Such support from the infrastructure nodes benefits from awareness of the underlying network topology (i.e., information available at mesh routers about the true physical connections between nodes in the network). This opens the door to developing efficient algorithms and schemes for P2P communication in a wireless community network that account for the specific features of WMCNs (e.g., contention for wireless medium between neighbouring nodes and traffic interference) in order to enhance the performance of P2P communication. Moreover, such support exploits the under-utilized resources (e.g., storage and bandwidth resources) at a large number of infrastructure nodes in the WMCN. This creates the possibility of replicating into the caches of those under-utilized infrastructure nodes P2P contents that are desired by the community, and enabling those infrastructure nodes to participate in content distribution by uploading the cached content to interested users in the network. Given that the application-layer nodes (peers) are autonomous and may join or leave at will [38], participating infrastructure nodes can substantially enhance the performance of the overall system.

This motivates us to consider hybrid approaches for content sharing and media streaming in WMCNs. The hybrid approaches involve peers that cooperate together and use the P2P technologies (e.g., Bittorrent) to exchange content with each other, and a number of under-utilized infrastructure nodes in the WMCN. In this case, infrastructure nodes help peers in distributing contents by uploading contents they cache to interested users in the network using the client-server scheme. We refer to those infrastructure nodes that participate in content sharing and help peers in distributing content in the community network as *Helpers*. Hence, a hybrid approach can be viewed as two subsystems: Peer-to-Peer system and Helper-to-Peer system.

1.7 Main Research Contribution: Hybrid Approaches for Content Sharing and Distribution in WMCNs

Our main contribution in this dissertation is P2P-with-helpers: Hybrid approaches for content sharing and distribution in a wireless community network that rely on support from infrastructure nodes to the P2P applications. As we have discussed, an efficient hybrid approach for content sharing in a WMCN must enable efficient P2P communication in the WMCN, and must allocate optimal resources at helpers (e.g., upload bandwidth and storage) in order to ensure efficient use of the helpers. Hence, the P2P-with-helpers system consists of two main components: P2PMesh and HelperDesign Manager (Figure 1.2).

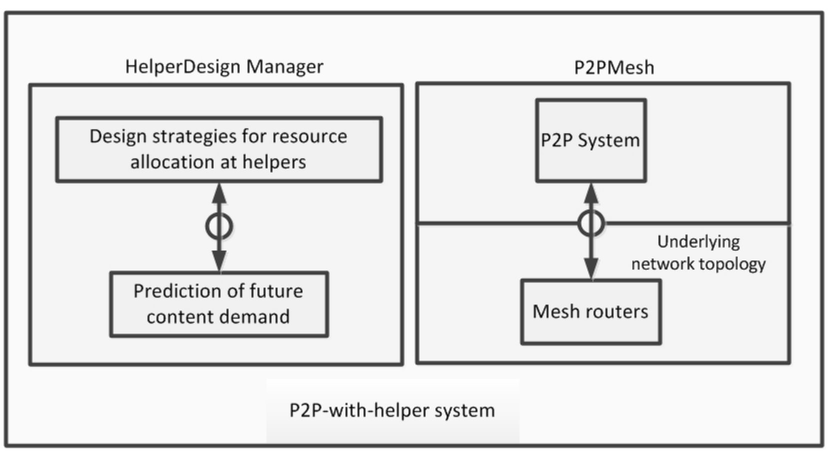


Figure 1.2: Design Components of P2P-with-helpers System

We propose an architecture and efficient schemes for P2P communication in a wireless community network that are aware of the underlying network topology (Chapter 2). We call the combination of proposed architecture and schemes P2PMesh (Figure 1.2). Our proposed schemes use the awareness of the underlying network topology (i.e., information about actual physical connections between nodes in the WMCN available at mesh routers) to account for the specific features of WMCNs (e.g., contention for wireless medium between neighbouring nodes and traffic interference) in order to enhance different aspects of P2P communication (e.g., download path establishment, selection of content providers, P2P content lookup, P2P content dissemination). Thus, P2PMesh supports P2P communication that are network friendly and enables peers to consume network resources efficiently.

We also develop design strategies that the HelperDesign Manager uses to allocate optimal resources at helpers in order to maximally utilize idle (under-utilized) resources at those helpers (Chapters 4 and 5). Unlike schemes that target enhancing the performance at the steady state, the design strategies are developed based on prediction of the future evolution of content demand in the community network. The proposed design strategies are general and can be dynamically used to allocate resources at helpers in a highly responsive manner to match the predicted content demand evolution in individual community networks.

We show that P2P-with-helpers significantly improves the performance of P2P applications. The performance enhancement reflects aspects such as P2P content download time (i.e., the average time required for a downloading peers to retrieve a desired content), P2P file access cost (i.e., the average number of wireless hops on download paths between downloading peers and content providers), traffic load imbalance in the network, energy consumption in the network, and monetary cost of Internet traffic. Our analytical and simulation results show that significant performance enhancement is realized along all dimensions indicated.

The proposed hybrid approaches for content sharing offer significant benefits for communities that do not possess the financial resources to deploy a dedicated cache node or a centralized storage server with over-provisioned resources for content sharing, or communities that require extremely dynamic management of their deployment such as wireless mesh community networks. P2P-with-helpers system is an alternative to dedicated video or content sharing cache designed for deployment in a WMCN. The uniqueness of P2P-with-helpers system comes in its ability to be dynamically instantiated, managed and scaled by community of users without administrative intervention, monetary investment, or dedicated resources.

1.8 Summary of Research Contributions

In this dissertation, we study how the support for P2P applications at infrastructure nodes (e.g., mesh routers) can be exploited to enhance the performance of P2P-with-helpers system that can be used for content sharing and media streaming in a wireless community network. To answer this research question, we sub-divide this dissertation into five technical chapters. In the following, we summarize our contributions in every chapter.

1. **P2PMesh: Topology-aware Schemes for Efficient Peer-to-Peer Content Sharing and Distribution in Wireless Mesh Community Networks**

   In Chapter 2, we propose P2PMesh: Topology-aware schemes for efficient P2P content sharing and distribution in wireless mesh community networks. P2PMesh benefits from the support to the P2P communication at mesh routers in a WMCN (Figure 1.2). We identified three major phases in any P2P system that supports content sharing. Each of these phases would need to be different for a WMCN from the techniques used in the wired networks (Internet). We explain the challenges and briefly summarize our contributions in each phase.

   - Content lookup:

     This is the process of identifying peers that store the required content. These peers are also referred to as content providers. In Chapter 2, we propose a P2P content lookup algorithm that mitigates traffic load imbalance in a wireless mesh community network.

- Selection of content providers and establishment of content download paths:

  This is the process of selecting a set of content providers from all potential content providers in the network, and establishing a download path between each selected content provider and the downloading peer. Due to broadcast nature of the wireless transmission in WMCNs, we have contention for the wireless medium between neighbouring mesh routers and interference between traffic on adjacent wireless links [36, 37]. P2P content sharing systems used in the Internet are implemented at the application layer and not cognizant of the underlying network topology. Therefore, algorithms used to select content providers and establish download paths are inefficient and consume high amount of bandwidth and energy in the WMCN. Moreover, traffic interference between adjacent parallel download paths deteriorates data throughput received at downloading peers. Thus, topology-aware innovative mechanisms are needed for WMCNs to enable P2P content sharing that allow peers to consume and distribute P2P content in efficient and resource-aware usage.

  We propose topology-aware schemes for P2P content sharing specifically designed for WMCNs that select ideal content providers for a downloading peer, and adjust download paths to take advantage of ongoing wireless transmissions at mesh routers in order to maximally utilize the network capacity.

- Data dissemination:

  We propose a "stateless" source multicast routing scheme for P2P content dissemination in a WMCN. The scheme exploits the characteristics of the WMCN such as the broadcast nature of radio transmissions and the static nature of mesh routers. Most P2P content sharing systems (e.g., BitTorrent) enable a downloading peer to download a desired content from multiple content providers in the same time. Our efforts are aimed at enabling efficient multicast traffic routing when multiple peers in the WMCN download the same content from the same provider at the same time.

2. **Extending P2PMesh: Replication Schemes for Peer-to-Peer Content in Wireless Mesh Community Networks with Infrastructure Support**

   In Chapter 3, we extend P2PMesh and propose an optimum strategy for replicating a set of P2P files at a number of mesh routers, which are to participate in content sharing in the WMCN, such that the average access cost of all P2P files in the network in minimized. As we have noted so far, minimizing the access cost of a P2P content i (i.e., the average number of wireless hops on the download paths between a downloading peer and providers of content i) is particularly useful in a WMCN. This consequently reduces both contention for the wireless medium between neighbouring mesh routers and traffic interference, while enhances throughput of the network.

   We show that the optimum content replication strategy is distinct from other replication strategies that replicate content proportional to the content popularity (e.g., Least Recent Used LRU, Least Frequently Used LFU). It is also distinct from the works in [39–43]. We propose an algorithm to implement our optimum replication strategy. However, our proposed algorithm requires information about the popularity of each P2P content in the system. Retaining a detailed record of requests that every P2P file receives in the network requires a centralized platform (e.g., file tracker), which may be hard to implement in a limited resource network such as a WMCN. Therefore, we also propose a distributed low cost (on-line) algorithm that does not require popularity information. We show that the performance of the distributed algorithm mimics the optimum strategy very well when the system is in steady state.

3. **Modelling, Performance Analysis, and Design Strategies for P2P-with-helpers Based on Prediction of Content Demand Evolution in the Community: The Case of Content Sharing**

   In Chapter 4, we evaluate the performance of P2P-with-helpers system for content sharing and distribution in a community network. We consider files that are stored in a digital format and that can be used only after downloading the entire file (e.g., P2P file sharing-like service). We use the fluid-flow approximation to model the evolution of both content demand and served peers in the community network over time.

   Many prior works used fluid model to analyze the performance of P2P content sharing in the Internet. The common assumption in most prior work is that evolution of demand for a content is constant (i.e., Poisson arrivals with constant rate) [44, 45]. However, we show in Chapter 4 that in many cases, the evolution of a content demand in a community network is different from Poisson arrivals, and the life-time of the content in the community network is limited. Thus, designs that target system optimization at the steady state (e.g., [44, 46–49]) are not useful in these cases.

   In Chapter 4, we consider the case, wherein a viral evolution of content demand in a community network is predicted. In particular, we consider a common setting in a community network, wherein a user (e.g., a student in a college campus) generates a content (e.g., campus newsletter, lecture/class note, experimental/scientific data, technical seminar video) or gets interested in a content available in the Internet, and spreads the interest in this content to other users (e.g., classmates) in the community. This scenario is becoming increasingly popular in community networks since users of a community network who share similar interests are often socially connected (e.g., classmates, neighbours, friends in a social network such as Facebook, Twitter, or email group). Therefore, content demand grows quickly in a community network as every interested user contacts others and makes them interested, but tapers off and diminishes when all potential interested users finish downloading the content. We can see that in such cases, the content life-time in the community network is limited.

We analytically characterize this viral evolution of content interest in such cases. We propose design strategies for the HelperDesign Manager in the P2P-with-helpers system that allocate optimal resources at helpers to match the predicted future evolution of content demand in the community network (Figure 1.2).

4. **Modelling, Performance Analysis, and Design Strategies for P2P-with-helpers Based on Prediction of Streaming Demand Evolution in the Community: The Case of Media Streaming**

   In Chapter 5, we propose hybrid approaches for media streaming in the Internet that are aimed at exploiting the redundancy and abundantly available network "micro-resources" in a community network to create an aggregate virtual "macro-resource". Specifically, an Internet Service Provider (ISP) can leverage its control over a large number of under-utilized infrastructure nodes (helpers) in a community network residing in the ISP network to replicate into caches of those micro-resources media files that are desired in that community. By allocating these idle infrastructure nodes to manage the demand in a community network on a per-need basis, a large size of the traffic load in the community network would be absorbed by the macro-resource, and the peak load at the media streaming edge-servers would be filtered out. Hence, much of the over-provisioning at edge-servers can be cut down. Moreover, the cost of media streaming can be contained within the community network, and bandwidth and energy consumption in the ISP network - that results from delivering media files from the edge-server to interested users in the community network over long download paths - can be significantly reduced.

Our main contribution here is design strategies for the HelperDesign Manager in the P2P-with-helpers system that allocate optimal resources at helpers to match the predicted future evolution of media streaming demand in the community network (Figure 1.2). Our derived results provide qualitative and quantitative performance analysis that increase our understanding of how idle resources at infrastructure nodes (helpers) in community networks can be utilized to mitigate the cost of media streaming on Content Delivery Networks (CDNs), ISPs, and community network operators.

5. **A Ring-based Multicast Routing Topology with QoS Support in Wireless Mesh Community Networks**

In Chapter 6, we consider the problem of group communication in a static wireless multi-hop network such as a WMCN. In particular, we consider a setting, wherein a relatively small number of users in a wireless mesh community network are involved in a group communication (e.g., group video conferencing or video gaming). In this setting, each group member sends multicast streaming traffic to every member in the group, and receives multicast streaming traffic from every member in the group.

Research motivation:

As we have discussed, the wireless mesh community network is mainly operated and owned by either volunteers (members of the community) or a non profit organization (e.g., municipalities and wireless-city initiatives). Since proliferation and sustainability of such networks rely mainly on users cooperation, it is expected that services desired by users of the wireless community network such as content sharing, VoIP, IPTV, and group communication (e.g., video gaming and conferencing) to be killer applications that would encourage members of the community to donate and cooperatively share their network resources. However, the problem of enabling such applications over the WMCNs in both efficient and resource-aware usage are not widely addressed in the literature. In addition to challenges of implementing group communication in wired networks, wireless mesh community networks possess extra challenges due to the broadcast nature of wireless medium and traffic interference.

The main problem of enabling multicasting in a wireless multi-hop network is that the 802.11 standard multicast MAC layer does not support packet recovery mechanism nor does it support collision avoidance mechanism. Therefore, multicasitng in a WMCN is considered unreliable [51–53]. We therefore, argue for a ring-based multicast routing topology for reliable group communication in a WMCN. The simple structure of the ring and the similarity between multicast traffic routing on a ring topology and traffic routing on a unicast path, allow us to extend the use of RTS/CTS and ACK mechanisms that are used in the unicast MAC layer to the case of traffic multicasting over the ring topology.

Research contributions:

We propose an analytical model to evaluate the performance of multicast streaming in a WMCN when a ring-based multicast routing topology is used for multicast traffic distribution. In particular, given the multicast routing topology, our model allows us to derive lower bounds on the end-to-end delay, energy consumption in the network, and upper bound on capacity of the multicast network (i.e., maximum group size that the constructed multicast routing topology can support with QoS guarantees). We demonstrate the effectiveness of our proposed analytical model using simulations. Our results show that despite the approximations used in our analytical modelling, the simulation results confirm that our bounds derived using the proposed model mimic the real values very well.

Another contribution in Chapter 6 is an efficient algorithm to enhance the IP multicast traffic routing on a ring-based multicast routing topology using simple network coding technique. We show that the end-to-end delay is reduced by a factor close to $\frac{2}{3}$ (33%) when our proposed algorithm is used. We further show that our proposed algorithm increases the capacity of a ring-based multicast routing topology by a factor $\frac{3}{2}$ (50%). The performance enhancement of our proposed algorithm is a result of better utilization of the available channel bandwidth at mesh routers on the ring routing topology.

Interestingly, we show that for a moderate multicast group size, a ring-based multicast routing topology coupled with our proposed algorithm for traffic routing outperforms a tree multicast routing topology in terms of both the end-to-end delay and capacity (Appendix B). However, this performance enhancement requires efficient construction of the ring-based routing topology. We, therefore, propose algorithm to construct efficient and interference-aware ring-based multicast routing topology for the group communication over a wireless mesh community network.

The technical work and contributions of this dissertation are presented in six chapters (Chapters 2- 6). Each chapter presents a short introduction to the problem discussed along with the related work, contributions, and detailed description of the preformed studies. Evaluation of the proposed schemes and algorithms in every chapter are outlined, followed by a discussion of the validity of the results and their limitations. The conclusion remarks of the dissertation and future work directions are outlined in Chapter 7.

Chapter 2

P2PMesh: Topology-aware Schemes for Efficient Peer-to-Peer Content Sharing and Distribution[1]

2.1 Introduction

A WMCN is self-organized and decentralized due to the nature of its distributed components designed to operate in dynamic network environments and hop-by-hop connection establishment. In networks that are owned and operated by users such as WMCNs, all users are seen as equal which creates significant issues when deploying and financing a centralized server-based content sharing system with large computing, storage, and networking resources as the nature of such networks makes it difficult to apportion costs or retrieve monetary contributions. For this reason, decentralized (distributed) designs

[1]This chapter is based in part on the following papers.

1. A. Alasaad, S. Gopalakrishnan, and V. C.M. Leung, "Extending P2PMesh: Topology-aware Schemes for Efficient Peer-to-Peer Data Sharing in Wireless Mesh Networks," in *Wireless Communications and Mobile Computing*, John Wiley & Sons, Ltd., vol. 13, no. 5, pp. 483–499, 2013.

2. A. Alasaad, S. Gopalakrishnan, and V. C.M. Leung, "Mitigating Load Imbalance in Wireless Mesh Networks with Mixed Application Traffic Types," in *Proc. of the IEEE Globecom Conference*, pp. 1707–1711, Miami, FL, December 2010.

3. A. Alasaad, S. Gopalakrishnan, and V. C.M. Leung, "Peer-to-Peer File Sharing over Wireless Mesh Networks," in *Proc. of the IEEE PACRIM Conference*, pp. 697–702, Victoria, BC, August 2009.

are required for services offered by WMCNs. A popular distributed scheme for content sharing and distribution is through the use of P2P technology, by exploiting the upload capacity of peers who are interested in the same file. The distributed characteristics shared by both WMCNs and P2P technologies strongly suggest that P2P content sharing is the most feasible scheme in WMCNs [25–27].

Structured P2P resource sharing systems typically rely on the maintenance of virtual (overlay) topology on top of the physical network topology, and distributed hash tables (DHTs) for locating content in the overlay network (e.g., Chord [54]). This lookup facility acts as a foundation for implementing many P2P resource sharing services such as P2P data sharing. Schemes used in conventional P2P content sharing systems are implemented at the application layer. Thus, these schemes are not cognizant of the physical network topology, and there is no consideration of the negative impacts on the underlying network. Therefore, traditional P2P resource sharing algorithms are inefficient when used over a WMCN where topology information is valuable. Few work address the techniques and challenges of developing underlay aware P2P systems in wireless mesh networks (e.g., [55, 56])

Traffic load imbalance is a major cause of performance degradation in WMCNs [57]. Most of IP network routing protocols in wireless mesh networks (e.g., DSR and AODV [60, 61]) use the minimum-hop (shortest path) routing metric. When an IP network routing protocol uses the minimum-hop metric, most of the data traffic between nodes in the WMCN is delivered over paths that traverse the center of the network deployment region and hence, creates traffic load imbalance [57]. Furthermore, P2P content lookup algorithms which use DHT strategies adds to the problem of traffic load imbalance at mesh routers [62]. This is due to the fact that many physical routes that are mapped from the overlay links between peers tend to cross the center of network deployment region through mesh routers which are often highly congested. We refer to the number of IP packets at each mesh router as traffic load. A P2P content lookup algorithm, which is both aware of the physical network topology and mitigates the traffic load imbalance (i.e., minimizes number of IP network-level query packets that traverse the congested areas in the network), is preferable for WMCNs to an algorithm which generates relatively low IP query traffic but introduces high traffic load imbalance into the underlying network [57].

In an effort to increase the received throughput and reduce content download times for peers, most P2P data sharing systems enable a downloading peer to retrieve required content from multiple providers, and rely on the underlying IP network routing protocol to establish a physical download path to each content provider. However, conventional IP network routing protocols, that are not specifically designed to take into consideration the characteristics of both the WMCNs and the P2P content sharing system (e.g., interference between traffic on neighbouring download paths and possibility that multiple peers in the vicinity download the same content at the same time) are not efficient for P2P content sharing in WMCNs [37]. Moreover, establishing inefficient download paths that consists of large number of wireless hops degrades the performance of transport protocols such as TCP [35].

We advance the thesis that support for P2P applications at infrastructure nodes (mesh routers) results in significant performance improvements. Such enhancements at the infrastructure nodes take advantage of the underlying network topology information available at mesh routers (i.e., the actual physical connections between nodes in the WMCN). This opens the door to developing efficient algorithms for P2P communication that account for the specific features of WMCNs such as the broadcast nature of wireless links and the static nature of mesh routers, in order to enhance the P2P communication and enable peers to consume the limited network resources (e.g., bandwidth) efficiently.

We define the underlying network topology awareness (topology-awareness), in the context of P2P communications, as usage of information collected from the underlying network (physical network) to enhance various performance aspects of P2P communications and mitigate negative impacts of P2P communications on the underlying network.

In this chapter, we propose P2PMesh: a combined architecture and efficient schemes for P2P content sharing in WMCNs. P2PMesh is a topology-aware system for content sharing that benefits from the support for the P2P applications at mesh routers, and achieves significant improvements in both the P2P content sharing performance and the overall WMCN throughput.

We identify three major phases in any P2P system that supports content sharing. As we have briefly noted so far, each of these phases would need to be different for a WMCN from the techniques used in the Internet. We briefly summarize our contributions in each phase.

I. Content lookup: This is the process of identifying peers that store the required content. These peers are referred to as content providers.

P2PMesh adopts a simple two-level content lookup approach that attempts to localize P2P traffic within the WMCN whenever the required content is available in the WMCN. We propose a P2P content lookup algorithm that mitigates traffic load imbalance at mesh routers. In particular, we refer to packet routing load at mesh routers (network-level traffic load) and not content query load at overlay nodes (other work describe techniques to handle load imbalance in the overlay network [59]). The proposed algorithm is motivated by the fact that most of the IP network routing protocols in multi-hop wireless networks (e.g., DSR and AODV [60]) use the minimum-hop (shortest path) routing metric to provide QoS for the delay-sensitive traffic [61]. As we have noted, IP routing protocols that use the minimum-hop routing metric increase the traffic load at mesh routers located in the center of the network deployment region and hence, create load imbalance (hot spots) [62]. Since the delay sensitive traffic coexists in the network with delay-insensitive traffic such as P2P data sharing, we believe that rather than modifying the IP network routing protocol (such as proposing a congestion-aware routing metric for the delay-insensitive traffic or solving a hard multi-metric routing problem at the IP network layer) to mitigate the load imbalance (e.g., LBAR [63] and IAR [64]), we can simply enhance the P2P content lookup algorithm in the overlay network. Although the P2P query traffic (lookup packets) represents a small portion of the total traffic in WMCNs, interference between packets at adjacent wireless links and contention for wireless channel between neighbouring mesh routers (i.e., stochastic back-off mechanism that is used in the MAC layer and broadcast transmission's blocking effect to neighbouring nodes) exacerbate the traffic load imbalance problem because it add to the traffic load at mesh routers in the congested area [58]. Since the P2P content lookup algorithm is implemented at the application layer, the proposed scheme is easy to implement and does not require any change to the IP routing protocol at mesh routers, which is implemented in kernel and hard to modify.

We propose a simple modification to existing content lookup methods in order to reduce the traffic load imbalance (again, we are concerned with IP routing load). Although our algorithm may route some of the IP query packets over longer paths in the underlying network so as to reduce number of query packets that traverse the congested area, the extra time required to lookup a content is tolerable for delay-insensitive traffic such as P2P content lookup (Section 2.3.1).

II. Selection of content providers and establishment of download paths: This is the process of selecting a set of content providers from all available potential content providers and establishing a download path between each selected content provider and the downloading peer. We propose an algorithm that selects ideal providers and adjusts download paths to take advantage of ongoing wireless transmissions at mesh routers in order to maximize the utilization of the network capacity (Section 2.3.2). We note here that the proposed download path establishment algorithm is specific to P2P traffic and that other traffic would use routes determined by the default IP network routing protocol in the WMCN.

III. Data dissemination:

We propose a "stateless" source multicast routing scheme for P2P content dissemination in the WMCN. The scheme exploits the characteristics of WMCNs such as the broadcast nature of radio transmissions, localized mobility of peers in WMCNs, and the static nature of mesh routers. Our efforts are aimed at enabling multicast routing for the P2P traffic when multiple peers in the WMCN download the same content from the same content provider at the same time. Our data dissemination scheme is implemented in middleware and, therefore, does not require any modification to the underlying IP network routing protocol (Section 2.3.3). We compared, using packet-level simulations, the proposed P2P content retrieval scheme against other conventional schemes used in multi-hop wireless networks such as random content providers selection strategy (e.g., Gnutella [65]) and closest content providers approach (e.g., SPAWN [66]). The simulation results show that the proposed schemes achieve higher throughput for the downloading peers and lower content download times (Section 2.4.3).

2.2 Related Work

Structured P2P communication systems typically rely on the maintenance of virtual overlay network topologies and DHTs for locating content on the overlay (e.g., Chord [54]). This lookup facility acts as a foundation for implementing many services such as P2P content and resource sharing. The hash function calculates both the key identifier by hashing the key (content meta-data), and the peer identifier by hashing the IP address of the peer. Chord is based on the idea of arranging both key and peer identifiers on a virtual ring called Chord ring; where a key is assigned to a peer whose identifier is the closest to identifier of the key. Chord specifies lookup operations for a key (k) that routes the inquiry on the virtual ring to the peer p which is responsible for that key in the virtual ring. Peer p, upon receiving the inquiry message, returns the IP address of the peer which is hosting the required content. To speed up the lookup operation in the virtual ring, every peer maintains a table of up to m distinct peers (fingers). Finger tables contain the finger peers' IDs and their IP addresses.

It is important for the overlay key lookup routing to exploit proximity in the underlying network. Recently, Canali *et al.* proposed a content lookup algorithm for P2P resource sharing over wireless mesh networks called MeshChord [25]. MeshChord exploits the stationary mesh routers to realize location-aware ID assignments. Each content is assigned an ID in interval [0,1] using a hashing function. Each mesh router with physical coordinates (x, y), that is mapped from its physical (geographical) location, is assigned an ID in the unit ring interval, such that any pair of mesh routers which are physically adjacent are assigned close IDs in the virtual unit ring. This mechanism achieves a close correspondence between the virtual overlay topology and the physical network topology. This specialized function mapping in MeshChord motivated us to adopt it in our proposed algorithm. However, the MeshChord method did not address traffic load imbalance at mesh routers. A recent study evaluated the network-level traffic load distribution at mesh routers when using Chord and MeshChord algorithms for P2P content lookup over a WMN, and concluded that higher degree of traffic load imbalance is observed in the WMN with increasing P2P overlay network size [62].

Many solutions have been proposed to tackle the overlay traffic load imbalance issue in DHT-based P2P systems [59]. Zhu *et al.* propose an overlay load balancing protocol which distribute load on overlay nodes proportional to their capacities [67]. However, those approaches only consider load balancing at overlay nodes without consideration to traffic load imbalance in the underlying network (i.e., traffic load imbalance at IP routers). To the best of our knowledge, our work is the first to propose a P2P content lookup algorithm to mitigate the IP network-level traffic load imbalance at mesh routers in WMCNs. Our proposed algorithm exploits the flexibility in selecting peer IDs that can be used to fill slots of the finger routing table in Chord algorithm at an overlay node. Since there is no constraint in selecting finger peer IDs in Chord, our algorithm, therefore, picks the peer in a physical location such that the physical route that is mapped from an overlay link between two peers on the overlay network does not cross the congested mesh routers. Although our lookup algorithm may route some query packets over longer paths in the underlying network, the extra time required to lookup a content is tolerable for delay-insensitive traffic such as P2P content lookup.

Many studies discussed the P2P traffic localization issue. The P4P project developed framework for traffic localization, that enables explicit communications between peers and network providers [68]. P4P requires each network provider to operate *itracker* server which serves as the portal for peers queries. A peer contacts *itracker* to retrieve information about neighbouring peers in the overlay network. This information is computed based on ISP's routing policies, inter-ISP cost agreement, etc. The IETF has formed a working group for application-layer traffic optimization (ALTO) with the goal of designing a query-response protocol for an ALTO server, which a peer may query for information about the underlying topology to achieve better traffic localization [69, 70]. The common design approach for the previously mentioned P2P traffic localization schemes is a server-based solution. However, due to the decentralized characteristics of the wireless ad-hoc networks/wireless mesh community networks, a server-less solution is preferred. Inspired by peers' multiple torrent behaviour, H. Wong and J. Liu developed a novel framework that traces and recovers the available contents at peers across multiple torrents, and thus effectively amplifies the possibilities of local sharing [71]

The advantages of our simple two-level content lookup scheme for P2P traffic localization over other existing schemes are two-fold. Firstly, it does not require support from any additional network entity (e.g., ALTO/*itracker* server). It avoids privacy and security issues associated with ALTO schemes since ISPs do not need to disclose underlying topology information or routing policy to peers; and peers, on the other hand, do not need to disclose their selection policy for peers to ISPs. Secondly, our protocol is implemented at mesh routers and gateways not at end-users. Excluding mobile peers from the overlay reduces the overhead introduced by the maintenance operations of the overlay routing table, and increases successful lookup rate.

The work of Castro *et al.* identifies Proximity Neighbor Selection (PNS) as a promising technique for selecting content providers [72]. PNS can be used to achieve low delay routes and low bandwidth usage. It selects content providers from among the closest nodes in the underlying network topology that satisfy certain constraint [73]. The proximity metric that is typically used in the definition of "closest" in most of PNS-based algorithms is the round trip delay. However, the delay metric is not the most important metric in wireless multi-hop networks. Due to limited resources and contention/interference between traffic in wireless multi-hop networks, the proximity metric that better suites the WMCNs is, therefore, the shortest physical path (i.e., minimum-hop distance in a download path between a downloading peer and a content provider).

Zhu *et al.* highlight the technical challenges of multi-path traffic streaming over a static wireless multi-hop network, and conclude that interference between traffic along parallel paths may cancel out the multi-path advantage [37]. Ruiz *et al.* showed that to lower the cost of multicast routing in WMNs, in terms of bandwidth consumption, the problem of minimizing the number of wireless transmissions (i.e., number of forwarding mesh routers per packet that are required to send a packet from a multicast source to all receivers in the multicast group) has to be solved [74]. Since finding a download path subject to multiple metrics (e.g., interference between multiple download paths, number of wireless transmissions, number of wireless hops on the path) is inherently difficult problem and is proved to be an NP-hard [74, 75], we suggest a heuristic algorithm that enables the downloading peers to establish efficient download paths to content providers in a distributed manner.

2.3 Design and Architecture of the P2PMesh

The basic set-up we consider is a WMCN consisting of many stationary mesh routers deployed in a two-dimensional squared region (grid-like topology), although our methods can be applied in other topologies as well. We focus on the grid-topology to simplify the explanation of our proposed design. Within the WMCN, data is communicated over wireless links and often over multiple hops. We assume that a WMCN is connected to the Internet via a gateway node that is operated by an Internet Service Provider (ISP), and the community network operator (Community Service Provider, CSP) is charged for the amount of traffic that is carried on the transit links between the community network and the Internet.

2.3.1 Content Lookup in P2PMesh

P2PMesh adopts a two-tier hierarchy of peers and a two-level content lookup approach to P2P data sharing running in WMCNs. The lower tier comprises of mobile mesh clients that participate in the P2P content sharing application, while the upper tier is composed of stationary mesh routers which construct the overlay network and implement the content lookup algorithm

In P2PMesh architecture, mesh clients do not directly participate in the content lookup service. Mesh clients share meta-data (file names, descriptions, etc.) with mesh routers, and it is only the mesh routers that support content lookup. The benefits of this architecture are significant in a WMCN because of the stability of mesh routers, which do not move or leave the network. This approach reduces the number of failed lookups and control messages (i.e., overhead on the overlay network), while also improving lookup latency.

The two-level approach for P2P content lookup constructs two decoupled overlays: a MeshP2P overlay that involves mesh routers, and a GlobalP2P overlay that involves gateways in the WMCN and peers in the Internet (Figure 2.1). This approach separates the MeshP2P overlay from the GlobalP2P overlay. In other words, peers located within the WMCN are transparent to peers on the Internet and vice versa.

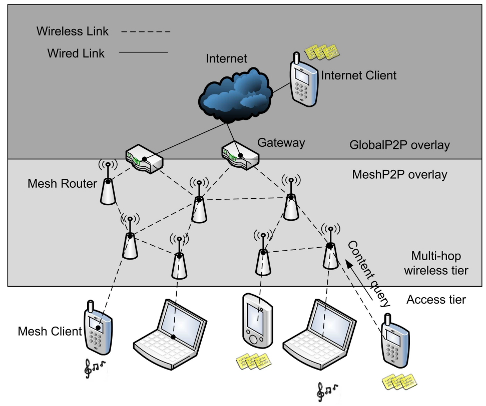
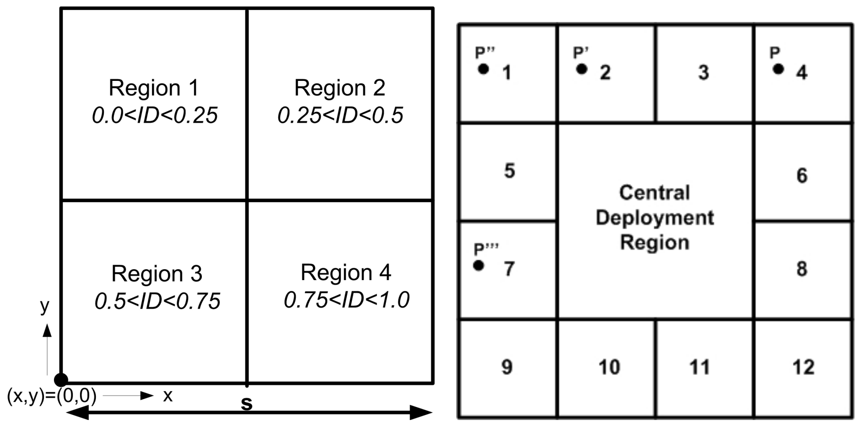


Figure 2.1: Wireless mesh community network architecture

Protocol Operation

When a peer (mobile mesh client) in a WMCN wants to share a content, the peer sends a message to the mesh router by which it is connected to the WMCN. The message contains the content's *key* obtained using a hash function (similar to Chord [54]). The mesh router upon receiving this message, registers the content's descriptor (i.e., *<key, IP address of the peer's home agent>*) in the appropriate mesh router based on the DHT scheme used.

When a peer requests a content, it applies a hash function on the content's meta-data to obtain the content's *key*. This *key* is sent via a request message to the mesh router by which it is connected to the WMCN. The mesh router upon receiving this request, forwards the *key* over the MeshP2P overlay to the mesh router responsible for that *key* according to the rules specified by our proposed content lookup algorithm, which we shall describe in details in the next section. If the requested content is not available within the WMCN, the *key* is sent to the gateway node. The gateway node upon receiving the lookup request, employs the GlobalP2P overlay to locate the required content at peers on the Internet. Gateways may monitor the P2P traffic on the transit link to the ISP. If delivering the required content from the Internet to peers in the WMCN imposes a financial cost to the CSP, the gateway can deny the lookup request (i.e., throttle P2P traffic). Thus, this method supports billing/pricing mechanism for P2P traffic which is of



(a) Peers' IDs assignment in P2PMesh.

(b) Location-aware finger peers in the P2PMesh.

Figure 2.2: WMCN deployment region

a great concern to the network operators [68].

Algorithm Design for Content Lookup in the MeshP2P overlay

As stated earlier, we assume that the WMCN is deployed in a squared region, and each side of the deployment region is of length s. This region is segmented into four smaller squared regions of equal area (Figure 2.2(a)). Each mesh router with physical location coordinate (x, y) located in Region i ($i \in \{1, 2, 3, 4\}$) is assigned a local identifier (ID_{local_i}) in the interval $[0, 1]$ based on MeshChord geographical mapping function [25] as follows.

$$ID_{local_1}(x,y) = \begin{cases} \dfrac{x \cdot \delta}{(\frac{s}{2})^2} + \lfloor \dfrac{y - \frac{s}{2}}{\delta} \rfloor \left(\dfrac{\delta}{\frac{s}{2}} \right), & \text{if } \lfloor \dfrac{y - \frac{s}{2}}{\delta} \rfloor \text{ is even} \\ \dfrac{(\frac{s}{2} - x) \cdot \delta}{(\frac{s}{2})^2} + \lfloor \dfrac{y - \frac{s}{2}}{\delta} \rfloor \left(\dfrac{\delta}{\frac{s}{2}} \right), & \text{if } \lfloor \dfrac{y - \frac{s}{2}}{\delta} \rfloor \text{ is odd} \end{cases}$$

$$ID_{local_2}(x,y) = \begin{cases} \dfrac{(x - \frac{s}{2}) \cdot \delta}{(\frac{s}{2})^2} + \lfloor \dfrac{y - \frac{s}{2}}{\delta} \rfloor \left(\dfrac{\delta}{\frac{s}{2}} \right), & \text{if } \lfloor \dfrac{y - \frac{s}{2}}{\delta} \rfloor \text{ is even} \\ \dfrac{(\frac{s}{2} - x) \cdot \delta}{(\frac{s}{2})^2} + \lfloor \dfrac{y - \frac{s}{2}}{\delta} \rfloor \left(\dfrac{\delta}{\frac{s}{2}} \right), & \text{if } \lfloor \dfrac{y - \frac{s}{2}}{\delta} \rfloor \text{ is odd} \end{cases}$$

$$ID_{local_3}(x,y) = \begin{cases} \dfrac{x \cdot \delta}{(\frac{s}{2})^2} + \lfloor \dfrac{y}{\delta} \rfloor \left(\dfrac{\delta}{\frac{s}{2}} \right), & \text{if } \lfloor \dfrac{y}{\delta} \rfloor \text{ is even} \\ \dfrac{(s-x) \cdot \delta}{(\frac{s}{2})^2} + \lfloor \dfrac{y}{\delta} \rfloor \left(\dfrac{\delta}{\frac{s}{2}} \right), & \text{if } \lfloor \dfrac{y}{\delta} \rfloor \text{ is odd} \end{cases}$$

$$ID_{local_4}(x,y) = \begin{cases} \dfrac{(x-\frac{s}{2}) \cdot \delta}{(\frac{s}{2})^2} + \lfloor \dfrac{y}{\delta} \rfloor \left(\dfrac{\delta}{\frac{s}{2}} \right), & \text{if } \lfloor \dfrac{y}{\delta} \rfloor \text{ is even} \\ \dfrac{(s-x) \cdot \delta}{(\frac{s}{2})^2} + \lfloor \dfrac{y}{\delta} \rfloor \left(\dfrac{\delta}{\frac{s}{2}} \right), & \text{if } \lfloor \dfrac{y}{\delta} \rfloor \text{ is odd} \end{cases}$$

where δ is a parameter which defines the granularity of location awareness [25]. Mesh routers are then assigned new (global) IDs according to their locations in the WMCN deployment region. In particular, a mesh router located in Region i with ID_{local_i} is assigned $ID = \frac{ID_{local_i}+(i-1)}{4}$. This new ID is then used to identify the mesh router in the MeshP2P overlay network. The terms mesh routers and peers are used interchangeably in this section.

The original squared region, over which the WMCN is deployed, is further subdivided into twelve subregions and a central deployment region as shown in Figure 2.2(b). Every mesh router located in subregion h, $h \in \{1, 2, \dots, 12\}$, maintains a table of two distinct finger peers. Finger tables mainly contain the finger peer IDs along with their IP addresses. Our proposed algorithm exploits flexibility in peer IDs that can be used to fill slots of the finger overlay routing table in the Chord algorithm. Since there is no constraints in selecting finger peer IDs in Chord, our algorithm therefore picks IDs, such that the physical route that is mapped from the overlay link between any two peers that are physically located in different regions does not cross the congested mesh routers in the WMCN. In particular, every mesh router in subregion h selects finger peer IDs to fill the finger routing table slots as shown in Table 2.1, where k is the *key* of a P2P file to be resolved, and ID is the identification of the mesh router on the P2PMesh (i.e., the next-hop node on the overlay network), which is to forward the lookup message to next node towards the destination.

Table 2.1: Fingers' IDs for peers in subregion h

Region (h)	$0.0 \leq k \leq 0.25$	$0.25 < k \leq 0.5$	$0.5 < k \leq 0.75$	$0.75 < k \leq 1.0$
1	–	$ID + 0.25$	$ID + 0.5$	$ID + 0.5$
2	–	$ID + 0.25$	$\frac{ID_{local_1}(x-\frac{s}{4},y)}{4}$	$ID + 0.25$
3	$ID - 0.25$	–	$ID - 0.25$	$\frac{ID_{local_2}(x+\frac{s}{4},y)+1}{4}$
4	$ID - 0.25$	–	$ID - 0.25$	$ID + 0.5$
5	–	$\frac{ID_{local_1}(x,y+\frac{s}{4})}{4}$	$ID + 0.5$	$ID + 0.5$
6	$\frac{ID_{local_2}(x,y+\frac{s}{4})+1}{4}$	–	$ID + 0.5$	$ID + 0.5$
7	$ID - 0.5$	$ID - 0.5$	–	$\frac{ID_{local_3}(x,y-\frac{s}{4})}{4} + \frac{1}{2}$
8	$ID - 0.5$	$ID - 0.5$	$\frac{ID_{local_4}(x,y-\frac{s}{4})+3}{4}$	–
9	$ID - 0.5$	$ID + 0.25$	–	$ID + 0.25$
10	$\frac{ID_{local_3}(x-\frac{s}{4},y)}{4} + \frac{1}{2}$	$ID + 0.25$	–	$ID + 0.25$
11	$ID - 0.25$	$\frac{ID_{local_4}(x+\frac{s}{4},y)+3}{4}$	$ID - 0.25$	–
12	$ID - 0.25$	$ID - 0.5$	$ID - 0.25$	–

Example: Suppose that a lookup operation for a key (k), $0.5 <= k <= 0.75$, is invoked at peer p with physical location coordinates (x, y) in subregion $h = 4$ (Figure 2.2(b)). The y^{th} finger for peer p ($y \in \{1, 2\}$), when $0.5 \leq k \leq 0.75$ is the peer which has the smallest ID larger than $ID - 0.25$ (Table 2.1), where ID is the identification of peer p. Thus, the lookup message is forwarded to peer p' located in subregion $h = 2$ (Figure 2.2(b)). Peer p' then forwards the lookup message to peer p'' which is the y^{th} finger for peer p' with ID : $\frac{ID_{local_1}(x-\frac{s}{4},y)}{4}$, where (x, y) is location coordinate of peer p'. Peer p'' then forwards the message to peer p''' with $ID : ID + 0.5$, where ID is the identification of peer p''. When the lookup message reaches peer p''', it finds that k belongs to a peer located in its region ($i = 3$) and, therefore, p''' forwards the message on the MeshP2P overlay to the peer which is responsible for that key according to the procedures specified by the MeshChord algorithm. Although our lookup algorithm route some query packets over longer paths in the underlying network, the extra time required to lookup a key is tolerable for a delay-insensitive application such as P2P data sharing. In fact, we show in the performance evaluation section that our algorithm reduces the content lookup latency when the WMCN carries high amount of traffic load (Section 2.4.2).

Our P2P content lookup algorithm is currently based on Chord/MeshChord algorithms. However, it can be slightly modified to work with any structured P2P content discovery algorithm that has flexibility in the choice of peer IDs to fill the routing table slots at an overlay node. In Kademlia, for example, any peer with ID that satisfies the required prefix can be used to fill a slot in the overlay routing table [76]. Our algorithm can, therefore, pick the peer from those whose IDs have the required prefix for Kademlia to satisfy our algorithm requirements (i.e., our algorithm can select peer IDs such that the physical path that is mapped from any overlay link between two peers do not traverse the congested mesh routers). However, the location-awareness characteristic in MeshChord motivated us to use it as a basis to our algorithm (see the related work section).

Our described algorithm is based in grid-topology assumption about the network deployment. However, the deviation from a grid-topology still results in performance improvement. However, we may need to choose sub-areas differently to account for any given topology.

2.3.2 Selection of Content Providers and Establishment of Download Paths

Conventional P2P content sharing algorithms are implemented at the application layer and, thus, not aware of the physical network topology. Therefore, they consume high network bandwidth due to mainly two problems that they cause: link stress and path stretch. The path stretch problem occurs when the P2P content is delivered from a content provider to a downloading peer over a long download path; while the link stress problem occurs when a P2P packet is transmitted over the same link multiple times. Thus, the purpose of our proposed topology-aware P2P content retrieval schemes in WMCNs is to reduce the network bandwidth required to deliver P2P traffic from content providers to downloading peers. We propose two schemes for P2P content retrieval in a WMCN: a scheme for establishing efficient download paths, and a scheme for efficiently disseminating (routing) P2P content over the established download paths.

The idea of the proposed scheme for establishing download paths is to utilize the ongoing transmissions of P2P content in the network. This occurs when a content provider y receives a request for P2P content j from peer k in the same time at which y is uploading content j to other peers in the WMCN. The idea is motivated by the fact that P2P files have large size and need long time to download. We shall see in Chapter 4 that the evolution of content demand grows quickly in community networks (i.e., we have viral evolution of demand in community network). Hence, it is likely that multiple peers in the WMCN download the same P2P content in the same time. The scheme aims to establish efficient multicast download paths between a content provider and peers that are downloading the same content from the same provider in the same time.

In order to both increase the received throughput (or equivalently reduce content download times for downloading peers), the proposed scheme enables each downloading peer to select multiple content providers, and establish a physical download path to each selected content provider such that the following metrics are minimized: (I) Interference between traffic on the multiple download paths from selected content providers to the downloading peer (Inter-Routes Interference IRI). Also known as route coupling; (II) Length of the download path in terms of number of wireless hops (HOP); (III) Number of disjoint mesh routers on the download path between a selected content provider and the downloading peer and download paths previously established between the selected content provider and other downloading peers that are downloading the same content (DISJ). DISJ metric minimizes number of wireless transmissions (number of forwarding mesh routers per packet) needed to send a multicast packet from a content provider to all peers that are downloading the same content. Since finding a route subject to multiple routing metrics is inherently difficult and is proved to be NP-hard problem [74, 75], we suggest a heuristic algorithm that enables peers to efficiently solve the problem in a distributed manner (Algorithm 1).

To speed up the process of establishing download paths, our scheme does not compute

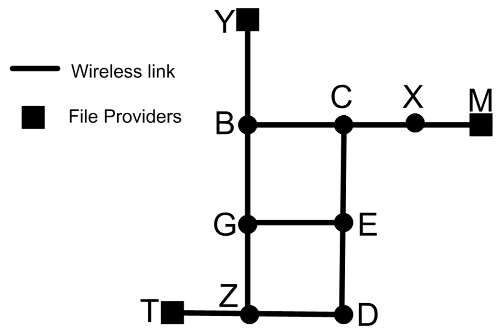


Figure 2.3: Selection of content providers and establishment of download paths.

Algorithm 1 Pseudocode for selecting content providers and establishing download paths from selected providers to the downloading peer.

Definitions:
A: set of potential content providers,
R_i: set of all candidate paths from a potential provider $(i) \in A$ to the requester ($i \in \{1,2,\ldots, |A|\}$),
R: set of all $R_i s$ ($R_i \subset R$),
B: set of selected providers,
p_i: set of all established download paths between potential provider (i) and other downloading peers, which are downloading the same required content from provider (i),
F: set of established paths between selected providers and the requester,
IFR: set of all mesh routers in set F and all mesh routers in the WMCN which interfere with mesh routers in set F,
α, β, χ: constants tunable parameters that can be used to assign different weight to each routing metric ($0 \leq \alpha, \beta, \chi \leq 1$),
ζ: number of providers required for each downloading peer.

```
Begin algorithm
for i = 1 : 1 : |A| do
   if p_i ≠ φ then
      Get p_i and R_i
   end if
end for
while |B| < ζ do
   for each R_i ∈ R do
      for each path j ∈ R_i do
         Compute: metrics IRI, DISJ, and HOP,
      end for
   end for
   for each j ∈ R do
      Compute: M = α.HOP + β.DISJ + χ.IRI,
   end for
   Select j with the minimum M,
   Select i associated with j,
   Add j to F,
   Add i to B,
   Update IFR
   Update R,
end while
End algorithm
```

new multicast download paths every time a downloading peer join/leave the network. Instead, the scheme enables peer k to add a download path to the already established multicast download paths between content provider y and other peers that are downloading content j, such that the aggregate received throughout at downloading peers is maximized, and number of mesh routers (wireless transmissions) that are needed to relay a P2P multicast packet from content provider y to all downloading peers is minimized.

Protocol Design:

When a peer gets interested in a P2P content, it employs the P2P content lookup service in P2PMesh. A list of potential content providers is returned to the P2P content requester. The requester then collects information about all available routes between itself and each potential content provider. This is achieved using the following protocol. The requester sends a download request message ($DREQ$) to each potential content provider. When a potential provider in the WMCN accepts the request, it broadcasts the download accept message ($DACC$) in the IP network (message flooding) after inserting into the $DACC$ message its IP address, sequence number, and the requester's IP address. Also, if the potential content provider is uploading the requested content to other peers, it inserts into the $DACC$ message information about all download paths which are already established between itself and other downloading peers (i.e., the list of IP addresses of all mesh routers (forwarders) on each download path from the provider to each destination). This message flooding scheme is similar to the route discovery scheme used in AODV. Most of the IP routing protocols in wireless multi-hop networks support IP broadcasting (e.g., AODV or DSR [77]). Each neighbouring mesh router which overhears the $DACC$ transmission, rebroadcasts the $DACC$ message after appending its IP address in the $DACC$ message. $DACC$ messages will reach the requester from multiple paths carrying information about all available routes between the requester and the potential provider, as well as information about the established download paths between the potential provider and other downloading peers. When the requester receives the $DACC$ messages from provider i ($i \in \{1,2,\ldots,$ number of potential providers$\}$), it adds all available routes between itself and provider i to set R_i. If provider i has already established download paths to other downloading peers, the requester adds those paths to set p_i. For all routes in set R ($R_i \subset R$), the requester runs Algorithm 1 to select ideal providers and establish efficient physical download paths to each selected provider.

Example: Consider the physical network topology given in Figure 2.3, where any two nodes that can communicate directly with each other are connected by an edge in the network graph. Suppose nodes Y, M, and T are potential providers for a P2P content required by peer D (A={Y,M,T} and i=3) (Figure 2.3). Suppose Y is uploading the content required by peer D to peer X on download path Y-B-C-X (p_1={Y-B-C-X}), and M is also uploading the content to peer X on path M-X (p_2 ={M-X}), while T is not uploading the content to any peer. Let TTL=4. Hence, candidate routes between Y and D are Y-B-G-E-D, Y-B-C-E-D, Y-B-G-Z-D (R_1 ={Y-B-G-E-D, Y-B-C-E-D, Y-B-G-Z-D}). Candidate route between M and D (R_2 = {M-X-C-E-D}, while candidate routes between T and D (R_3 = {T-Z-D, T-Z-G-E-D}. Hence, set R={Y-B-G-E-D, Y-B-C-E-D, Y-B-G-Z-D, T-Z-D, T-Z-G-E-D, M-X-C-E-D}. Peer D computes the routing cost: $\alpha.HOP + \beta.DISJ + \chi.IRI$, for all routes in set R as: $4+2+0=6$, $4+1+0=5$, $4+2+0=6$, $2+2+0=4$, $4+4+0=8$, and $4+3+0=7$ respectively (when $\alpha=\beta=\chi=1$). Peer D selects the route in set R which has the minimum cost. Thus, peers D selects route T-Z-D. Hence, B={T} and F={T-Z-D}. Suppose that peer D needs to download the required content from two providers. Thus, peer D needs to select another provider. Nodes G and E interfere with the route in set F (Interference={T,Z,D,G,E}). Peer D computes routing cost for all routes in set R after eliminating R_3 set, which is associated with the selected provider T, as: $4+2+3=9$, $4+1+2=7$, $4+2+3=9$, and $4+3+2=9$ respectively. Thus, Peer D selects route Y-B-C-E-D. Hence, Set F is now ={T-Z-D, Y-B-C-E-D} and set B={T,Y}.

2.3.3 Data Dissemination

We propose an efficient protocol for disseminating (routing) P2P traffic in a wireless multi-hop network with limited bandwidth such as WMCN. Specifically, we propose to add a sub-layer to the IP stack directly above the network layer to enable "stateless" multicast routing on top of the IP network. This sub-layer can be used for multiple purposes such as P2P traffic multicasting and P2P content caching and replication. In this section, we describe how the sub-layer is used for P2P content multicasting. We shall describe later in Chapter 3 how this sub-layer is also used for P2P content caching. We note here that our protocol for P2P data dissemination is implemented in middle-ware and, therefore, it does not require any modification to the underlying IP network routing protocol.

In our proposed scheme, the source (i.e., content provider) inserts in the sub-layer of every multicast packet the IP addresses of all forwarders and recipients located on each established download path from the provider to all downloading peers (Figure 2.4). The scheme benefits from the broadcast nature of radio transmissions in WMCNs. Since information about the entire download path from the provider to each destination (i.e., all forwarders' IP addresses on each download path) is available in the sub-layer, the provider needs to transmit a P2P multicast packet (broadcast in the IP network) through the wireless radio only once, and all mesh routers in the provider vicinity (one hop neighbours) will overhear the wireless transmission and process the sub-layer in the transmitted packet. However, only mesh routers which find their IP addresses in the sub-layer rebroadcast the packet; and so on until the multicast packet reaches all destinations.

To minimize the overhead in the sub-layer, each forwarder before rebroadcasting the packet, deletes any information in the sub-layer which becomes useless in the remaining paths. Due to the localized mobility of peers in WMCNs and the static nature of mesh routers, it is likely that an established download path remains the same for the entire content download session. Therefore, we assume that the protocol does not have to deal

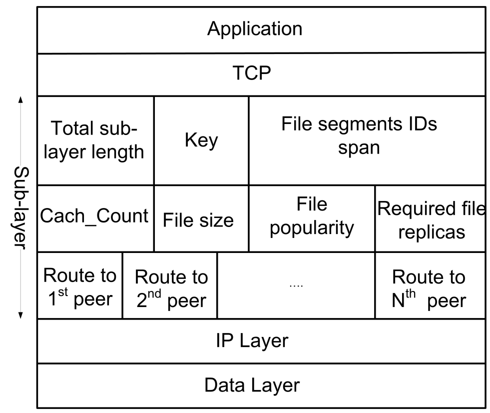


Figure 2.4: Sub-layer format

with fragmentation issues that may occur at the network layer since all fragments of the packet go through same nodes to destinations, and all segment packets have appropriate size.

Large content are fragmented into segments with appropriate size and each segment has an *identifier number*. After a downloading peer (receiver) selects its content providers and establishes a download path to each selected provider, the downloading peer assigns the provider with the shortest download path (smallest number of wireless hops) as the main source and other providers as secondary sources. When a receiver selects a provider as the main source, the receiver controls the segments that this source uploads to all receivers which download the same content from this source. In other words, the receiver that selects a provider as its main source is called the master, and can pull whatever segments it requires from that source at any time. Other downloading peers, which select that provider as a secondary source, are called slaves and collect segments which are pulled from that source by the master.

To avoid receiving duplicate segments from multiple providers, and in order to enable content streaming, a downloading peer first collects segments from its secondary sources, then it requests missing segments from its main source by appending the buffer-segments map identifiers (BM) in a request message. The main source upon receiving the BM

request uses our proposed stateless multicast data delivery scheme to transmit the required segments to the master and to all slaves. When a slave is no longer interested in receiving segments from a particular provider or if the collected segment from a provider was already downloaded from another provider, the slave sends a termination message to that provider. If uploading bandwidth at a provider is high, it can act as the main source for multiple receivers. To avoid excessive overhead and extra burden at content providers, the TCP protocol is only used for the data flows between main sources and masters, but not for flows between secondary sources and slaves. When a slave receives a corrupted P2P packet, it simply ignores that packet and requests it later from its main source.

It worth noting here that the IP multicast scheme is very inefficient when used for P2P content dissemination in this case. This is because the IP multicast scheme is designed to deliver a multicast packet from one source to all multicast group members. In IP multicast scheme, every node in the multicast routing topology has to maintain group membership (multicast state) for every multicast routing topology that it is involves in, so that it can rebroadcast (relay) a multicast packet that it overhears. Since every downloading peer in a multicast group may be interested in different segment of the content in any particular time instant, $n!$ IP multicast routing topologies must be established between the content provider (multicast source) and downloading peers in the multicast group, where n is the total number of peers that are retrieving the content from the multicast source. Since, large amount of overhead is required to maintain each multicast routing topology, IP multicast scheme is very inefficient in this case.

Comparison with DDM and AOM

We can classify our proposed scheme for P2P data dissemination as stateless P2P traffic multicast routing. Instead of requiring multicast forwarding state to be stored at all nodes on the multicast paths as in the case of IP multicast routing protocols, this approach enables stateless P2P traffic multicasting. In stateless multicasting, it is not necessary for nodes along the data forwarding paths to maintain multicast forwarding state. When an intermediate node receives a multicast data packet, it only needs to look at the header to decide how to forward the packet. This stateless approach avoids loading the network with pure signalling traffic required to maintain the forwarding table at every node on IP multicast routing topology. Due to mobility of peers in WMCN, maintaining multicast routing state is much more expensive operation in WMCNs than in wired networks. This is even worsen by the tight bandwidth constrains of WMCNs. Although packing routing information together with data traffic enlarge the size of data packets, it cuts down the transmission of large number of control packets in the networks.

DDM is one of the most popular source initiated stateless multicast protocol for small group size [78]. The multicast source in DDM lists all receivers, which share the next-hop forwarder, in the packet header. The source then transmits into the radio channel a unicast packet to each "next-hop" forwarder towards destinations. Each forwarding node in DDM employs the IP unicast network routing protocol to find next forwarder in the paths to destinations. Since the source in DDM builds a star overlay multicast network centred at the source, the degree of a node in the overlay multicast network is high. Therefore, DDM is not scalable and only suitable for small multicast group. AOM mitigates this problem by proposing a protocol that connects the source in the overlay multicast network with limited number of multicast nodes, and those nodes reach other members in the multicast group [79].

The main goal of the DDM and AOM is to establish a multicast overlay network that connects the source and group members. However, the connection between each parent and child in the overlay network is done using the underlying IP unicast routing protocol, which often employs the shortest route metric. Since the underlying IP network routing protocol is not aware of the data delivery on the overlay network, it cannot establish efficient physical multicast paths and, thus, interference between adjacent multicast paths can be high. On the contrary, our protocol is aware of the physical network topol-

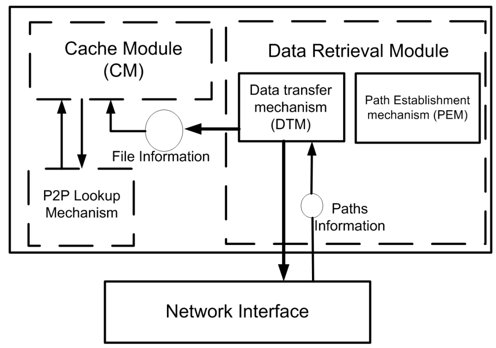


Figure 2.5: Middle-ware architecture

ogy. The physical download paths between the multicast source (content provider) and all downloading peers are established taking into consideration multiple routing metrics, with the goal of utilizing the network bandwidth and achieving better throughput for the downloading peers. Moreover, our scheme considers the setting where a downloading peer retrieves a P2P content from multiple content providers simultaneously. Furthermore, the degree of any forwarding node (i.e., number of wireless transmissions that any intermediate node on the multicast network topology needs to perform in order to relay the multicast packet to next-hop nodes toward destinations) is always one. Hence, our scheme is scalable.

Implementation tips

Figure 2.5 shows the design of the middle-ware used for P2P data dissemination. The data transfer mechanism (DTM) is the core component that provides operations such as checking passing by P2P packets, processing data in the sub-layer, and broadcasting P2P packets in the IP network. There is an interface between the DTM and the network routing daemon, from which the DTM accesses the P2P packets and obtains information about the download paths. The sub-layer in P2P packets contains information such as the total sub-layer length, identification of the delivered content's segment, popularity of the content, number of replicas for each content available in the WMCN, and list of all forwarders' IP on each download path to each destination (Figure 2.4). The information about popularity of the delivered content is used by the caching module (CM) for cache admission management (this is described in Chapter 3). This layered design decouples the data retrieval module that is used for P2P content dissemination from the default routing module implemented at the IP network layer.

2.4 Performance Evaluations

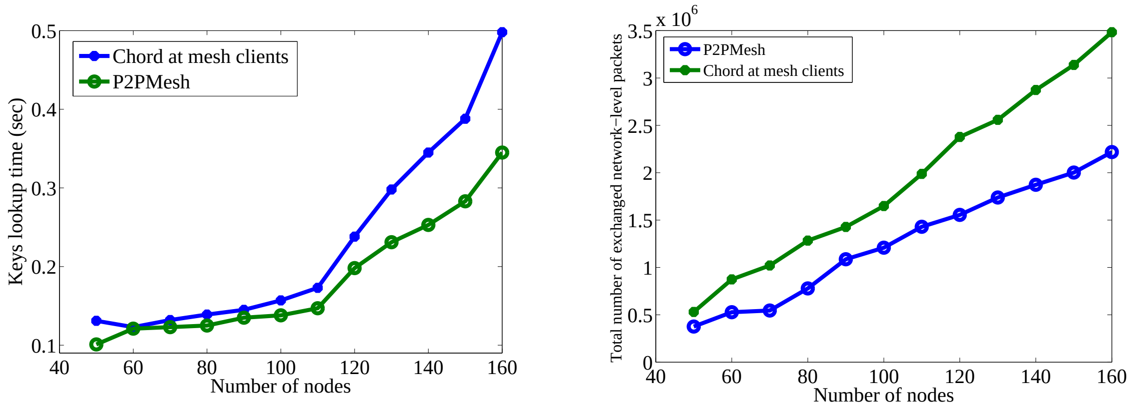
2.4.1 Simulation Settings

We simulated a planned static wireless multi-hop network deployed in a two-dimensional squared area, and consisting of n mobile mesh clients and m stationary mesh routers equally separated with distance 100 m (grid topology). Each mesh router is equipped with 802.11b radio with wireless link rate of 11Mbs and coverage range of $250m$. DSR [77], one of the most popular network routing protocol for wireless multi-hop networks, was employed. This choice of DSR is motivated by the fact that it performs the best among other various routing protocols when used in conjunction with Chord [80]. We evaluated the performance of our proposed schemes and algorithms used in P2PMesh against other existing schemes. The primary metrics for evaluation were: average number of network-level P2P query packets, lookup latency, Inter-domain P2P traffic; network-level traffic load at mesh routers, aggregate received throughput at peers, and average content download time.

2.4.2 Content Lookup Algorithm

We used the packet-level simulator *OverSim* to evaluate the proposed ideas [81]. We first demonstrate the benefits of the P2PMesh architecture. Namely, the two-tier hierarchy of peers, where mesh routers construct the overlay topology and participate in the P2P content lookup service; and the two-level content lookup approach to data sharing in WMCNs. We used a pre-existing implementation of Chord in *OverSim*. We carried out many simulations each for one hour to evaluate the performance in two cases. *Case 1:* Chord at mesh clients; where Chord algorithm was implemented at participating mobile mesh clients. In this case, there is no support from the mesh routers to P2P applications. Mesh clients were involved in an overlay network. Each mesh client was made to request a key uniformly at random at intervals exponentially distributed with mean $t_{req} = 60$ (second). Each mesh client stays connected to a mesh router for time interval exponentially distributed with mean 600 (second) [82]. To simulate mobility, mesh clients were made to leave and rejoin the network. *Case 2:* P2PMesh; where the proposed two-tier hierarchy approach of peers in the WMCN and the two-level content lookup approach were implemented at mesh routers. In this case, the underlying network (i.e., mesh routers) supports P2P applications. Mesh clients were excluded from the overlay and only mesh routers were involved in the MeshP2P overlay network. We set-up the stabilize interval (the interval between consecutive overlay maintenance operations in the MeshP2P overlay) to 5 seconds.

We make number of interesting observations from our simulations. Firstly, lower content lookup latency is observed in the P2PMesh case (Figure 2.6(a)), where lookup latency is defined as the time elapsed between the instant a lookup request for a *key* is issued by a peer, and the instant the *value* is retrieved. The performance improvement in P2PMesh case is mainly due to the location-aware ID assignment to mesh routers (geographical mapping between mesh router IDs in the MeshP2P overlay and physical locations of mesh routers). Secondly, in both cases, we see increasing trend in the number of network-level P2P query packets as extra number of participating peers are involved in the overlay network. However, Chord at mesh clients case generates about $1.75x$ number of packets compared to the P2PMesh case when number of participating nodes in the overlay is 160 (Figure 2.6(b)). This result is a consequence of the stability nature of mesh



(a) Average lookup time.

(b) Communication cost.

Figure 2.6: Performance of P2PMesh architecture

routers, which do not move or leave the network, thus less overlay topology maintenance operations is required.

To demonstrate the efficiency of our proposed two-level content lookup approach in reducing the Inter-domain P2P traffic, we simulated a P2P file sharing scenario where 1000 peers (mesh clients) were located within the WMCN and 5000 peers were located on the Internet. $10,000$ distinct files were distributed on the participating peers on the Internet such that each peer store two unique files. Each peer in the WMCN was made to store four files from the $10,000$ files uniformly at random. Each file is with 10 Kbyte size and each lookup operation returns one value only (i.e., the IP address of one file provider). We carried out simulations each for one hour and computed the average ratio of lookup requests received at peers on the Internet from peers in the WMCN to the total requests initiated by peers in the WMCN. We compared the results in the two cases. The results reveal that about 78% of file requests initiated by peers in the WMCN were located at file providers on the Internet in case 1, while about 57% in the P2PMesh case (Figure 2.7). Hence, P2PMesh reduces the average traffic downloaded from the Internet by ratio of about 26% in this particular scenario. This is because the overlay network in case 1 is unaware of peers physical locations and, therefore, the lookup for *keys* is not

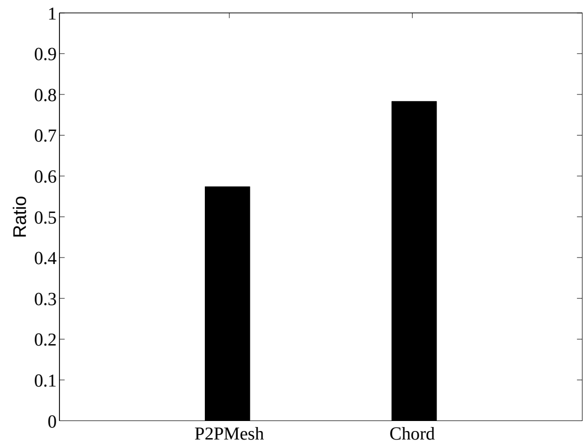


Figure 2.7: Average ratio of P2P content requests received at peers on the Internet from peers in the WMCN to total requests initiated by peers in the WMCN

efficiently resolved. On the other hand, the two-level content lookup approach used in P2PMesh enables locality awareness for P2P content lookups. In particular, the two-level approach used in P2PMesh for content lookup enables efficient content lookup resolution and returns IP addresses of file providers located within the WMCN whenever the required file is available in the WMCN.

In another set of simulations, we evaluate the performance of our content lookup algorithm used in the MeshP2P overlay network. We simulated a mesh network consisting of m stationary mesh routers. We carried out many simulations each for one hour and the results are then averaged over the simulation runs. We compared the performance of the proposed content lookup algorithm (P2PMesh) against Chord and MeshChord algorithms. The primary metrics for evaluation were: number of network-level P2P query packets, network-level traffic load at mesh routers, lookup latency, and overall network throughput. We implemented the geographical mapping scheme used in MeshChord algorithm but not the MAC cross-layering property. We refer to mesh routers as peers in the remaining section. We first computed the number of network-level P2P query packets transmitted by each peer when Chord algorithm is used for P2P content lookup and number of peers

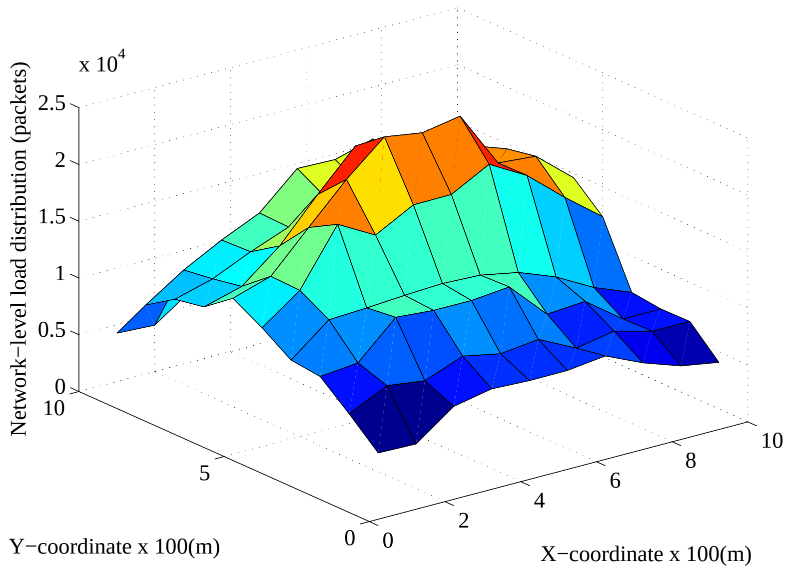
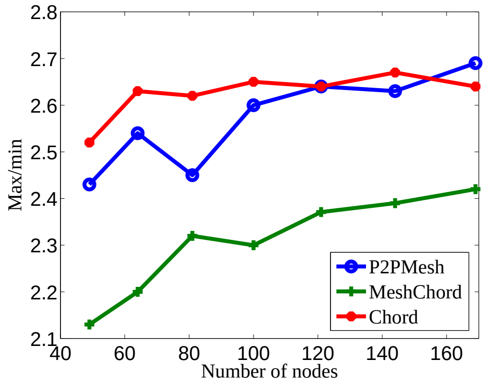


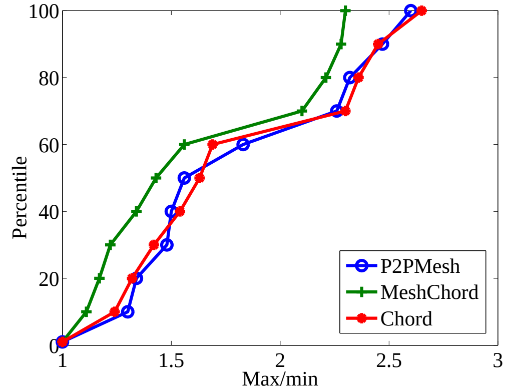
Figure 2.8: Network traffic load distribution using Chord (m = 100)

is ($m = 100$). The result is shown in Figure 2.8, where X and Y coordinates represent the physical locations of peers (mesh router) in the deployment area. We observe that mesh routers which are located in the center of the WMCN deployment region transmit higher number of network-level query packets compared to other mesh routers in the network. This is due to the fact that most of the network-level links that are mapped from the overlay links between peers on the overly network tends to cross this region creating hot-spots (traffic load imbalance). Similar trends were observed for MeshChord.

To compare the level of traffic load imbalance in the underlying network (routing load at mesh routers) that results when using P2P content lookup algorithms under investigation, we have used an intuitive index of load imbalance (max/min). (max/min) is the ratio between the load observed at the maximally loaded mesh router to the load observed at the minimally loaded mesh router in the network; where the traffic load at each mesh router is computed in terms of number of all packet transmissions (including data packets and P2P query packets). We first computed the (max/min) ratio for varied number of peers without coexistence of any other application in the network (no background traffic).



(a) Max/min ratio for network-level traffic.

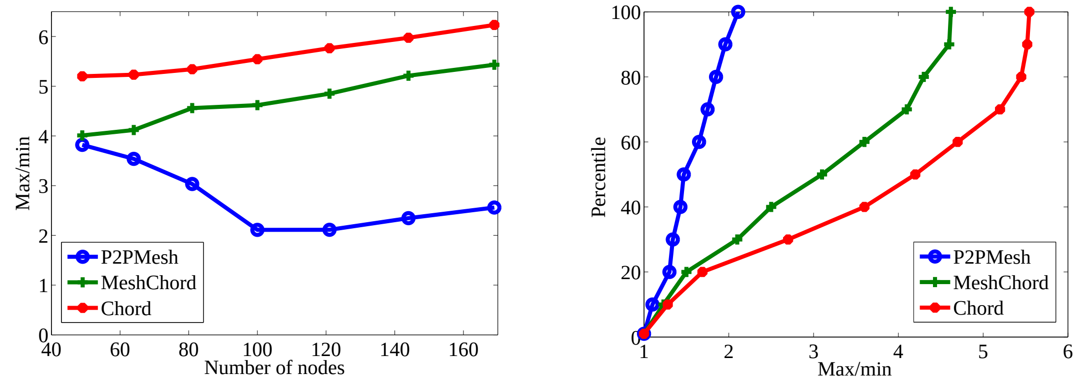


(b) Network-level load distribution ($m = 100$).

Figure 2.9: Network load imbalance without background traffic

We observe traffic load imbalance (max/min>2) in the network for all algorithms under investigation (Figure 2.9(a)). To compare traffic load distribution in the WMCN when the number of peers is ($m = 100$), we sorted the network-level traffic load observed at mesh routers in increasing order and computed the (max/min) percentile. The results show that about 30% of mesh routers carried higher than twice the traffic that is carried by the minimally loaded mesh router (Figure 2.9(b)).

The previous results conclude that high network-level traffic load imbalance is observed at the underlying network (mesh routers) when the investigated algorithms are used for content lookup in WMCNs. However, in the next set of simulations, we demonstrate how the content lookup algorithm proposed in the P2PMesh system mitigates the traffic load imbalance when the P2P data sharing traffic coexists in the network with other traffic such as the delay-sensitive traffic (Figure 2.10(a)). To simulate delay-sensitive traffic (background traffic), we selected $\sqrt{m}$ pairs of source/destination nodes uniformly at random and established constant bit rate (CBR) flows between them with rate 100 Kbps. While background traffic tends to cross the WMCN central deployment region, the content lookup algorithm in the P2PMesh is designed to detour the network-level P2P query traffic around this congested area via relatively less congested mesh routers (fingers



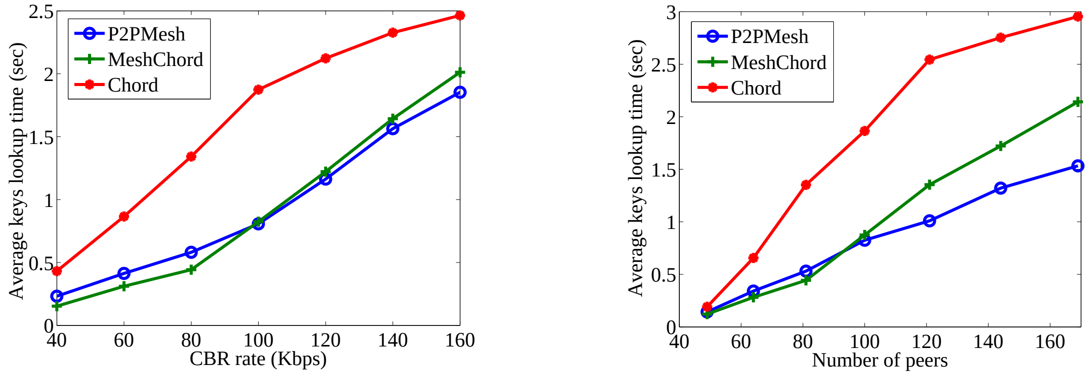
(a) Max/min ratio for network-level traffic.

(b) Network-level load distribution ($m = 100$).

Figure 2.10: Network load imbalance in presence of background traffic

peers). P2PMesh is shown to mitigate more than 50% of the network-level traffic load imbalance compared to MeshChord and Chord when number of peers in the network is > 100 (Figure 2.10(a)). About 70% of peers carried higher than two times the traffic that is carried by the minimally loaded peer in MeshChord, while 75% in Chord and about 5% in P2PMesh when number of peers is ($m = 100$) (Figure 2.10(b)).

In the following set of simulations, we investigate the impact of the proposed content lookup algorithm used in the P2PMesh on the performance of P2P data sharing, in terms of lookup latency and number of network-level lookup packets, when the P2P data sharing traffic coexists in the WMCN with other traffic (background traffic). We selected $\sqrt{m}$ pairs of source/destination peers uniformly at random and established constant bit rate (CBR) flows between them with varied rate. We present the lookup latency for all algorithms with increasing CBR rate (Figure 2.11(a)). We also computed the the lookup latency for varied number of peers, while the background traffic was fixed at (CBR rate = 100 Kbps) (Figure 2.11(b)). Furthermore, we computed number of network-level content lookup packet transmissions for varied number of peers when (CBR rate = 60Kbps) (Figure 2.12). We observe that Chord needs longer time to lookup required content due to lack of geographical mapping between the overlay topology and the true network



(a) Lookup latency when m=100.

(b) Lookup latency when CBR=100Kbps.

Figure 2.11: Lookup latency in presence of background traffic

topology. Another interesting observation is that when the network is lightly loaded (i.e., CBR rate is low or content query rate is low), MeshChord outperforms P2PMesh in terms of lookup latency and number of network-level lookup packet transmissions. This is due to the fact that network-level lookup packets in P2PMesh traverse longer physical path in terms of number of wireless hops. However, for moderate to high traffic load in the WMCN, mesh routers which are located in the central region become highly congested. Therefore, contention and interference between packets in that region result in higher lookup latency and higher number of packet re-transmissions in MeshChord compared to P2PMesh.

In the following set of simulations, we investigate the impact of using our proposed P2P content lookup algorithm on the performance of the overall WMCN, in terms of aggregate received throughput at all mesh clients in the WMCN, and compare the results with Chord and MeshChord algorithms. We simulated a wireless mesh community network consisting of 100 stationary mesh nodes. We selected 10 pairs of source/destination uniformly at random and established TCP flow between them (background traffic). TCP flows were used to deliver a file with size of 10 MBytes to destinations. TCP is used as a transport layer protocol for reliable data delivery. We observe that receivers (mesh clients) in

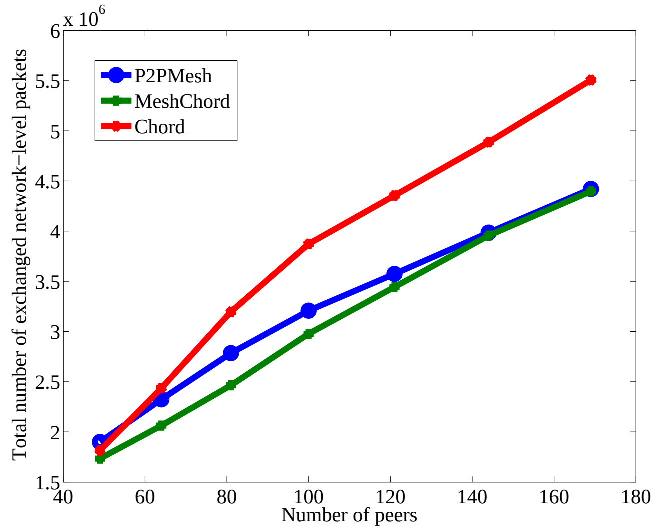
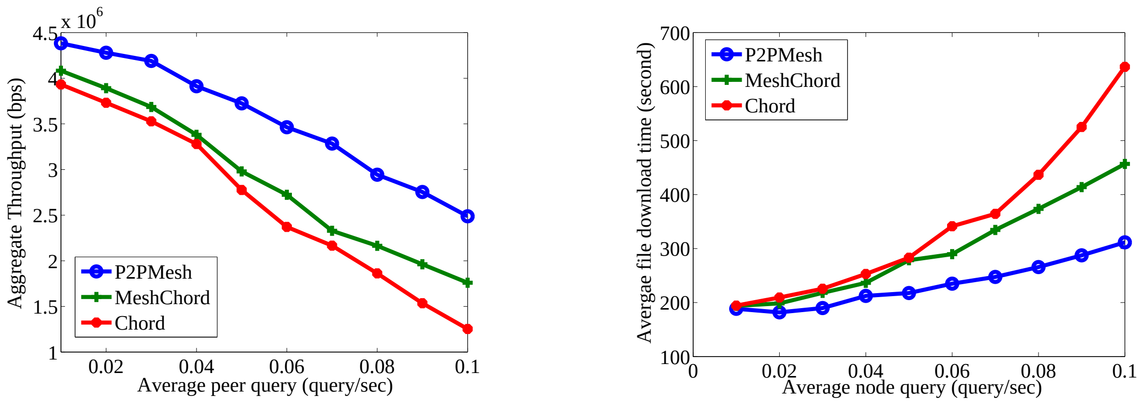


Figure 2.12: Average total number of network-level packet transmissions in presence of background traffic (CBR=60kbps)

WMCN achieve higher average aggregated received throughput (Figure 2.13(a)) and less average file download time (Figure 2.13(b)) when peers, which participate in P2P file sharing application, use the content lookup algorithm proposed in the P2PMesh system. The performance enhancement in P2PMesh is due to the fact that the P2P content lookup algorithm in P2PMesh mitigates the traffic load imbalance at mesh routers and, hence, improves the performance of the WMCN.

2.4.3 Selection of Content Providers, Establishment of Download Paths, and Data Dissemination Schemes

We compared the performance of our proposed algorithm for establishing download paths, and our proposed scheme for P2P content dissemination that are used in P2PMesh against two chosen benchmarks: (i) Random content providers strategy (Random peer), where the downloading peer selects content providers randomly. Such a strategy is adopted in some Gnutella implementations and achieves good performance in the Internet [84]. (ii) Shortest path strategy, where the downloading peer selects the closest content providers in



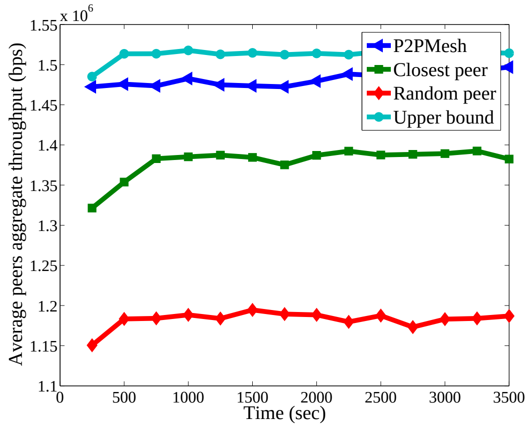
(a) Average total throughput received at destinations for all TCP flows (m=100).

(b) Average file download time for all TCP flows (m=100).

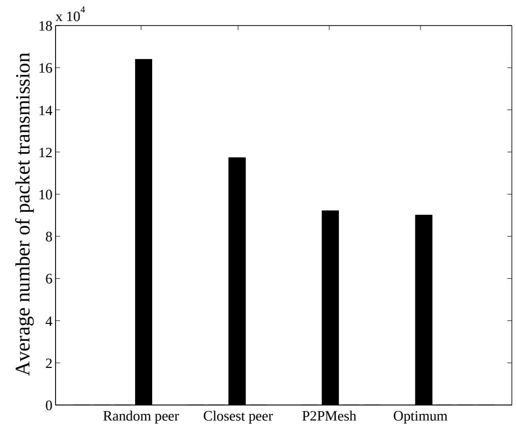
Figure 2.13: Impact of content lookup algorithms on the WMCN performance

terms of number of wireless hops distance (Closest peer). This strategy is implemented in SPAWS protocol [66]. The Random and the Closest content provider selection protocols use the default network routing protocol (DSR) to route and deliver P2P segments from providers to downloading peers. We also compared the performance of these schemes to an upper bound (Upper bound). Namely, we considered a single flow over a unicast path from every provider to a downloading peer and computed received throughput. We then added the aggregated throughput received by all downloading peers. In other words, we did not account for interference between traffic delivered on the established routes in the network in the case of optimum scheme. Although the results obtained using the "Optimum" scheme is not feasible, we consider it as an upper bound for the performance.

We simulated a network consisting of 50 mesh routers (each had a node degree of four on average), and 100 mesh clients. We used the packet-level simulator *OPNET* [83] to evaluate the proposed ideas. We compared the performance of the investigated protocols in terms of aggregate received throughput at peers and average total number of packet transmissions in the WMCN. Three mesh routers five hops-distances away from each other were chosen as potential providers for a P2P content. Three mesh clients were chosen as downloading peers and each peer establishes a download path to each potential



(a) Average peers aggregate throughput.



(b) Average number of transmitted packets in the network.

Figure 2.14: Performance evaluation of our schemes proposed for P2P content retrieval

provider to retrieve the content. In each simulation run, two other mesh clients were uniformly selected at random as downloading peers and each peer establishes download paths between itself and two of the potential content providers to retrieve the required content according to the investigated protocols. TCP flow was used as a transport layer protocol for reliable data delivery. All TCP data packets were with same fixed size (1460 bytes).

The simulation results reveal that downloading peers in the P2PMesh system have the highest received throughput (Figure 2.14(a)). The reason is that downloading peers in the P2PMesh system select providers and establish download path to each provider such that metrics like interference between established paths and hop-distance to providers are minimized. Random content providers (Random peer) protocol has the worst performance because download paths from providers to receiving peers can be very long compared to other protocols. P2PMesh outperforms the shortest path (Closest peer) protocol because interference between traffic transmitted along the parallel shortest paths cancels out the multi-providers/multi-paths advantages. Average number of packet transmissions which is needed to deliver a file with 10 Mbytes size to downloading peers was computed for all protocols under investigation (Figure 2.14(b)). Less number of packet transmissions is observed in the P2PMesh system because peers in the P2PMesh share wireless links when possible and the proposed multicast source routing schemes (sub-layer scheme) saves network bandwidth and reduces link stress.

2.5 Summary

This chapter introduced P2PMesh. We have demonstrated that support for P2P communication at mesh routers results in significant improvements for the performance of both P2P content sharing, and the overall wireless mesh community network throughput. Such support from the mesh routers benefits from the awareness of the topological structure of the underlying network available at mesh routers, and exploits the stationary characteristics of mesh routers in WMCNs. P2PMesh provides a combined architecture and efficient schemes for enabling efficient P2P content sharing over WMCNs. However, the proposed schemes advocated in this chapter need to consider practical implementation issues related to incorporating the proposed designs at mesh routers in order to enable them to participate in P2P activities. Also, the proposed schemes need to consider incentives for end users (peers) to adapt (implement) the P2PMesh. Thus, an interesting future research direction is to address possible implementation problems when incorporating the required features at mesh routers. Also, mechanisms that do not require end users to modify their peer-to-peer applications in order to benefit from the support at mesh routers need to be examined.

Another issue that needs attention is to study the impact of effective provision of internal content caching and replication on the efficiency of peer-to-peer content sharing. Content caching and replication are viable because of the low cost of storage units and the availability of multi-core processors to support high-speed implementation of both P2PMesh schemes and content caching schemes; that is what we study in the next chapter.

Chapter 3

Extending P2PMesh: Replication Strategies and Schemes for Peer-to-Peer Content[2]

3.1 Introduction

A popular distributed scheme for content sharing and distribution is through the use of P2P technology, by exploiting the upload capacity of peers who are interested in the same file. This scheme scales well with increasing number of users, and mitigates the load imbalance in the network. The distributed characteristics shared by both WMCN and P2P technologies strongly suggest that P2P content sharing is the most feasible scheme in a WMCN. Also, content sharing using P2P technology is not associated with any

[2]This chapter is based in part on the following papers.

1. A. Alasaad, S. Gopalakrishnan and V. C.M. Leung, "Replication Schemes for Peer-to-Peer Content in Wireless Mesh Networks with Infrastructure Support," accepted for publication in the journal of *Wireless Communications and Mobile Computing*, John Wiley & Sons, Ltd.

2. A. Alasaad, S. Gopalakrishnan, and V. C.M. Leung, "Replication Schemes for Peer-to-Peer Content in Wireless Mesh Networks with Network Support," in *Proc. of the IEEE PIMRC Conference*, pp. 1135–1139, Toronto, ON, September 2011.

3. A. Alasaad, S. Gopalakrishnan, and V. C.M. Leung, "Caching and Replication Schemes for Peer-to-Peer File Sharing in Wireless Mesh Networks," in *Proc. of the IEEE Globecom 2010 Workshop on Ubiquitous Computing and Networks*, pp. 1707–1711, Miami, FL, December 2010.

monetary costs, which makes it ideal for for content sharing over WMCNs.

The cost of accessing a P2P file i in a WMCN is defined as the average number of wireless hops on the download paths between a peer, which requests file i, and file providers from which the requester retrieves file i. Minimizing the access cost of a P2P file is particularly useful in a WMCN due to the characteristics of the wireless multi-hop networks. Since we have contention for the wireless medium between neighbouring mesh routers, and interference between traffic on adjacent wireless links in the WMCN, the data rate at a downloading peer is significantly reduced when data traverses a download path consisting of large number of wireless hops [36]. Moreover, the performance of transport protocols such as TCP degrades rapidly when it used to deliver data on a path that has many wireless hops [35]. Furthermore, since most of P2P files have large size, P2P content sharing consumes enormous amount of bandwidth and energy in the WMCN. Therefore, effective provision of internal content caching and replication is critical to the performance of content sharing and network throughput.

Content caching and replication increases number of replicas for every P2P content in the network. This increases the likelihood that a downloading peer can access a replica of the required content at a content provider nearby. Consequently, this reduces both contention for the wireless medium between neighbouring mesh routers and traffic collision/interference in the network, and enhances network throughput. Furthermore, the amount of bandwidth and energy consumption in the network can be significantly reduced.

The static nature of mesh routers in a WMCN, which typically have low probability of leaving the network, and higher upload capacity as compared to the end-users, motivated us to propose schemes for P2P content caching and replication at the mesh routers. We consider a P2P content sharing setting in a WMCN with support from mesh routers. In this setting, a number of mesh routers are equipped (over-provisioned) with additional storage capacity and P2P-aware devices (packet sniffers) that are programmed to cache and store P2P content that they forward (relay) in the WMCN. Those mesh routers participate in content sharing and exploit their idle bandwidth capacity to upload P2P content that they cache to interested peers. Caching P2P content at mesh routers increases throughput received by downloading peers because this scheme benefits from the large upload bandwidth at those participating mesh routers.

Many schemes and implementations are available for caching P2P traffic [85–92]. Many studies and implementations consider opportunistic caching at mesh routers in a wireless mesh networks [39–41, 93–98]. Many studies present design, implementation and evaluation of a proxy cache for P2P traffic [99–101]. Many techniques are used to identify P2P traffic in a community network and capture P2P traffic in the network based on port, IP address or application signatures [28, 102–107]. The cost of storage devices has significantly reduced over the years. Several terabyte+ drives have recently broken the \$0.10/gigabyte barrier, making the next milestone \$0.01/gigabyte, or \$10/terabyte. All of the above-mentioned technologies make the problem of caching P2P content at mesh routers feasible.

In the previous chapter, we introduced P2PMesh: Topology-aware schemes for constructing efficient download paths for P2P content sharing in WMCNs. In this chapter, we extend P2PMesh. Our main contribution in this chapter is a replication strategy for P2P files at the participating mesh routers. The strategy computes the optimal number of replicas for every P2P file in the network, such that the average access cost for all P2P files is minimized. Although our replication strategy may not guarantee a minimum time to transfer P2P files to downloading peers (ETX outperforms the hop count metric in this case), the overall performance of the WMCN (e.g., network throughput), which often carries many types of applications including P2P traffic and other delay sensitive applications, can be significantly improved. For all of the above-mentioned reasons, the performance metric (i.e., file access cost) - that we choose to optimize in our P2P content replication strategy - is the most important one in the case of P2P content sharing over WMCNs.

We analytically show that in order to minimize the average access cost of P2P files in a WMCN, the strategy should replicate P2P files at the participating mesh routers such that the number of replicas for each file is proportional to $\left(\frac{p_i}{1+k\cdot\rho\cdot p_i}\right)^{\frac{2}{3}}$, where p_i is the popularity of file i, ρ is ratio of the average upload rate at peers to the average upload rate at the participating mesh routers, and k is constant (Section 3.4). We first propose a centralized algorithm to implement our optimum P2P content replication strategy at the participating mesh routers (Section 3.5.1). One drawback in our replication strategy is that it requires information, a priori, about popularity of P2P files in the WMCN (p_i), which might be challenging (i.e. it requires a centralized entity such as a file tracker that can attain a log of requests for every P2P file in the system over time). We, therefore, propose a distributed and low cost (on-line) algorithm that does not require this information (Section 3.5.2). We show that the distributed algorithm mimics the optimum strategy very well when the system is at steady state.

3.2 Related Work

Many major operators have already considered the wireless mesh networks technology for their wireless city initiatives (e.g., Birmingham and Newcastle wireless city initiatives [108]). Promising examples of a WMCN deployment include Seattle Wireless [16], Houston WMN [10], MIT Roofnet [109], Berlin Freifunk [110]), FON community [19], and Athens wireless metropolitan network. In the case of Athens wireless, for example, the network comprises 1120 backbone nodes (mesh routers) (as of August, 2010) and more than 2900 client computers (mesh clients) connect to it. More than 9,000 people have stated their intention to join network in the near future [20]. Due to the community-oriented nature of WMCNs, it is likely that users of a WMCN would like to share content desired by the community such as movies, music, software updates, etc [23, 24].

Content caching schemes in wireless multi-hop networks are commonly used to mitigate the traffic load in the network. *Ditto* is a system that opportunistically caches overheard data at mesh routers to improve subsequent transfer throughput in wireless mesh networks [93]. Commonly used caching strategies work in a demand-driven fashion [94]-[95]. When a file is requested, it is pulled from the source of that file and any intermediate node on the path between the source and requester may store a replica of the file in its cache. Cao *et al.* proposed two schemes for content caching in Ad-hoc networks aiming at minimizing the file access cost and the lookup latency: CacheData which caches the data (files), and CachePath which caches information about the path to the required file at nodes on the route between source and destination [39]. Das *et al.* proposed a scheme to cache a web content at every mesh router along the path from the gateway to the requester [41]. The scheme uses a per-hop transport protocol for WMNs that breaks a single end-to-end transport connection (e.g., S-A-B-D) into multiple single hop transport connections along the route (e.g., S-A, A-B, B-D) and pipeline data over these sub-connections. These caching schemes result in replicating every file in the network proportional to its popularity. Hence, the number of replicas for file i is proportional to p_i, where p_i is the popularity of file i in the network.

Large number of content replication algorithms have been proposed to reduce the cost of locating content (content lookup) in P2P file sharing in the Internet. Performance metrics such as the cost of file query was considered (i.e., number of peers a requester should inquire to lookup the required file). Cohen *et al.* proposed a replication method (square-root rule) to minimize the file query cost in unstructured P2P file sharing networks [43]. The strategy replicates file i at end-users (peers) such that the density of a file in the network is proportional to the square root of the file's popularity ($p_i^{\frac{1}{2}}$). Kangasharju *et al.* proposed Top-K MFR replication algorithm to reduce the file query cost in structured P2P file sharing network [111]. However, these schemes replicate content at end-users (peers) not at infrastructure nodes. Moreover, these schemes were designed to reduce the cost of locating files (file lookup) in P2P file sharing on the Internet and, thus, cannot be directly used to reduce the cost of accessing P2P files in a WMCN.

Jin and Wang reveal a strategy for web-content and service replication tailored for the wireless mesh networks [42]. We follow very similar analysis, but for the case of P2P content. The common assumption in prior content replication strategies in wireless multi-hop networks is that all content providers (i.e., caches) have the same upload bandwidth, and a user downloads the entire content from the closest node that stores a replica of the desired content. Our derived results in this chapter show that all previously proposed strategies are far from optimum when used for replicating P2P files at mesh routers in a WMCN. This is because a downloading peer in a P2P content sharing system often simultaneously accesses segments of the desired file from replicas stored at multiple file providers including peers and mesh routers. Since a replica of a popular file (i) is likely to be stored at large number of peers in the network. A peer who downloads file i can access segments of file i from replicas stored at peers in the vicinity (peers nearby) at low cost. Hence, the optimum P2P content replication strategy should mitigate the bias towards popular files by allocating higher number of replicas at mesh routers for less popular files. To the best of our knowledge, we are the first to propose a replication strategy at infrastructure nodes (mesh routers) for P2P content in a wireless mesh community network. Our proposed strategy considers a practical content retrieval scheme implemented in most P2P content sharing systems that enables a downloading peer to retrieve the desired content from multiple providers simultaneously over parallel download paths (e.g., BitTorrent).

To lookup content providers from which a downloading peer can retrieve the desired content, a centralized file tracker can be used (e.g., BitTorrent), or a distributed DHT-based algorithms such as Chord [54] and MeshChord [25]. MeshChord, which is a modification to Chord, assigns peers in a WMN which are close in the physical network with close-by IDs in the Chord overlay ring. Since MeshChord provides a good mapping between the overlay topology and the physical network topology, it represents an ideal candidate for content lookup in WMCNs. In unstructured P2P file sharing systems, a query for a P2P file is flooded to neighbouring peers in the overlay network until a replica of the desired file is discovered. We do not consider the cost of locating replicas of a required file in the network (file query cost). The problem of locating content in P2P file sharing is out of scope of the work in this chapter. We rely on the default file lookup protocol to discover location of replicas.

Determining the optimal cache placement (i.e., the problem of selecting the best nodes in the network to cache specific file) in an arbitrary network topology has similarity to two problems in graph theory: the facility location problem and the k-median problem. It has been proved that the two problems are NP-Hard [112]. This problem is well investigated in the context of wireless multi-hop networks (e.g., POAH, ECHO) [40, 113, 114]. We do not consider the problem of content placement. We assume that the community network operator somehow selects the mesh routers that are to participate in P2P content sharing. Our effort is aimed at computing the optimum number of replicas that participating mesh routers should store in their caches for every P2P file in the WMCN such that the overall access cost for all P2P files is minimized.

In Chapter 2, we introduced P2PMesh: A topology-aware scheme for efficient P2P content sharing in WMCNs. P2PMesh benefits from the support of mesh routers to enable efficient P2P content loolup algorithm that mitigates the traffic load imbalance in the WMCM. Moreover, P2PMesh efficiently establishes and deliver traffic over download paths between a downloading peer and file providers. In this chapter, we compliment our prior work and enhance P2PMesh by proposing P2P content replication schemes that utilize the bandwidth and storage capacity at the participating mesh routers.

3.3 Problem Formulation

We consider a large wireless mesh community network consisting of many stationary mesh routers deployed in a 2-dimensional squared area (grid-like topology) as shown in Figure 3.1. Any two nodes that can communicate directly with each other are connected by an edge in the network graph. Within the WMCN, data is communicated over wireless links. Nodes may communicate directly over a wireless link or over multiple hops with intermediate nodes forwarding data. Although the assumption about grid topology may be restrictive, we use this assumption just for the purpose of formulating the problem and simplify the analysis. However, the model that we present and the results that we obtain are valid for any wireless mesh community network topology with large number of mesh routers.

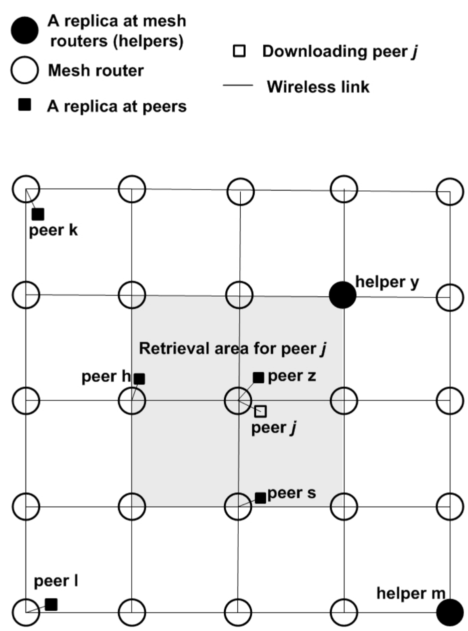


Figure 3.1: Network model

We consider P2P data sharing setting in a WMCN, wherein a number of mesh routers (K) in the network act as caches and participants in content sharing. Those nodes are equipped (over-provisioned) with additional storage and P2P-aware devices (packet sniffers) that are programmed to enable mesh routers to cache and store P2P content that they relay in the WMCN. We refer to those mesh routers as ***helpers***. During a period of time (T, where T is relatively large), peers in a WMCN share a set of N distinct P2P files. Request for P2P files are initiated at peers independently and uniformly at random with different rates (i.e., some files are more popular than others).

Suppose peer j in a WMCN is interested in P2P file i. Suppose that the nearest replica of file i at helpers is stored at mesh router y (Figure 3.1). Let the number of wireless hops between helper y and peer j be d ($d = 3$ in Figure 3.1). Suppose that peer j retrieves (downloads) segments of file i from the replica of file i stored at helper y, and from replicas of file i stored at peers with distance (in terms of wireless hops) not longer than than d (i.e., peers h, z, and s in Figure 3.1). We denote the physical area at which peer j uses to locate replicas of the desired file as ***retrieval-area*** for peer j (Figure 3.1). We can see that all replicas of file i at peers residing in this area are with distance not longer than d to peer j.

Every P2P file is fragmented into a number of segments with small size (e.g., BitTorrent). Let s denotes segment ID, and m be total number of segments of file i. Let c_i^s be the cost of accessing segment s of file i. Hence, we define the cost of accessing file i by peer j as

$$c_i = \sum_{s=1}^{m} c_i^s,$$

Let $P_{ret-area_i}$ be the number of file providing peers from which the downloading peer j retrieves segments of file i (i.e., number of peers in the retrieval-area of peer j that possess replicas of file i, $P_{ret-area_i} = 3$ in Figure 3.1).

Suppose peer j retrieves segments of file i from helper y at average rate of μ_{h_i} (in segments per unit time); while retrieves segments of file i from every file providing peer in the retrieval-area at average rate of μ_p (in segments per unit time). Although the assumption that a downloading peer receives traffic from all regular peers at the same data rate may be restrictive and impractical (specially when considering varied wireless channel condition and different number of wireless hops on download paths between a downloading peer and file provides), we use this assumption to simplify the analysis in this section. We relax this assumption and consider a general and more feasible case in Appendix A.

Since file providing peers and helpers upload segments of file i to the downloading peer j at different rates (i.e., $\mu_{h_i} \gg \mu_p$), the average fraction of of file i that downloading peer j retrieves from helper y can be computed approximately as follows.

$$\epsilon_i \approx \frac{\mu_{h_i}}{\mu_{h_i} + \mu_p \cdot P_{ret-area_i}},$$

$(0 < \epsilon_i < 1)$.

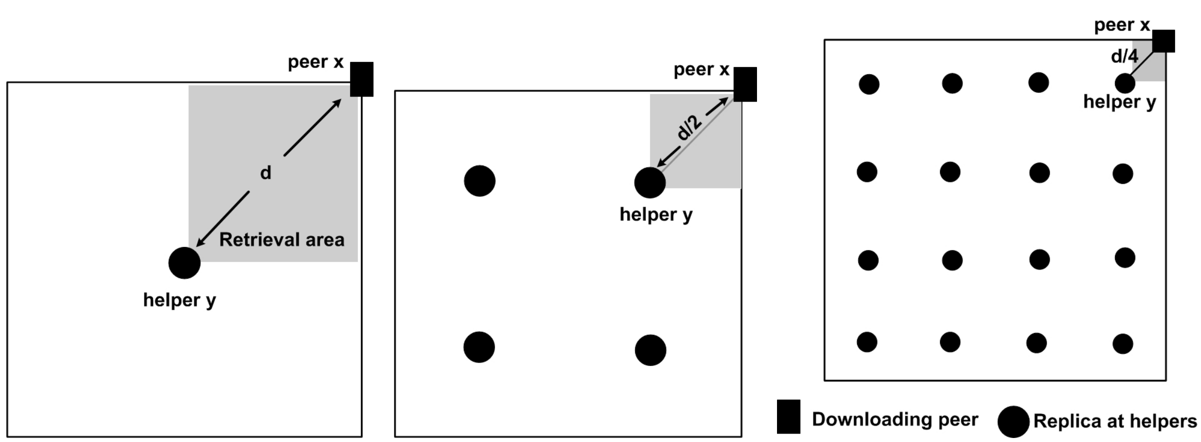
We define the cost of accessing a P2P file i in a WMCN as the average number of wireless hops between a peer, which requests file i, and file providers from which the requester retrieves file i. Since d is the cost of accessing a segment of file i from the replica stored at helper y, the average cost of accessing file i by peer j can be determined as:

$$c_i = \epsilon_i \cdot d + \frac{1}{2} \cdot (1 - \epsilon_i) \cdot d, \tag{3.1}$$

where $(\epsilon_i \cdot d)$ term accounts for the cost of accessing segments of file i from helper y; while $\frac{1}{2} \cdot (1 - \epsilon_i) \cdot d$ term accounts for the cost of accessing segments of file i from peers h, z, and s. It is easy to see that the average distance between the downloading peer j and content providing peers in the retrieval-area is about half the distance between peer j and helper y. Hence, we have the factor $\frac{1}{2} \cdot d$ in the second term of Eg. (3.1). Notations used in this chapter are listed in Table 3.1.

Table 3.1: Notations.

p_i	Popularity of file i in the WMCN.
μ_p	Average rate at which a peer uploads data to a downloading peer (segments per unit time).
μ_{h_i}	Average rate at which a helper uploads segments of a cached file i to a downloading peer (segments per unit time).
ρ	Ratio of the rate at which content providing peers upload data to a downloading peer to the rate at which a helper uploads data to a downloading peer ($\rho = \frac{\mu_p}{\mu_{h_i}}$).
ϵ_i	Ratio of the number of segments retrieved from a helper to the number of segments retrieved from peers.
d_i	The cost of accessing file i from a replica stored at helpers.
P_i	Average number of peers in the WMCN that possess replicas of file i in time period T.
A	Number of requests for all P2P files during time period T.
g_i	Number of requests for the i-th file during time period T.
$r_i^*(t)$	Optimum number of helpers that must cache file i at time t to achieve optimum file replication at helpers.
$r_i(t)$	Number of helpers that store a replica of file i at time instant t.
$D_i(t)$	Number of extra helpers (if any) that must store file i to achieve optimum P2P file replications at helpers at time instant t.
N	Number of distinct P2P files during time period T.
M	Average number of peers in the WMCN during time period T.
K	Number of mesh routers (helpers) participating in P2P contnet sharing in the WMCN.
B	Maximum number of files that a helper can store in the cache at any time.
k	Constant.
k_1	Constant.



(a) 1 replica (b) 4 replicas (c) 16 replicas

Figure 3.2: Average access cost computation

3.4 Optimum Replication Strategy

In this section, we determine the optimal number of replicas at helpers for a set of P2P files (N) in a WMCN during a period of time T, where T is large.

Suppose that file i has only one replica at helpers, and this replica is placed at helper y located in the center of network deployment area (Figure 3.2(a)). Denote the maximum cost of accessing the replica of file i at helper y by any peer in the WMCN as d. We can see that d is proportional to Euclidean distance between helper y and peer x located at the cornet of the network deployment region (Figure 3.2(a)). We note here that in a uniformly distributed WMCN, the Euclidean distance between two nodes is proportional to number of wireless hops (access cost). It is worth noting that a more accurate definition of the access cost relies on the routing algorithm. We assume that the routing algorithm employs a minimum hop routing metric such as AODV and DSR [61].

Let the average number of peers in the WMCN that possess replicas of file i during the time period T be P_i. Then, we can see that the average number of peers which possess replicas of file i and with distance not longer than d to peer x (i.e., peers residing in the retrieval-area of peer x, i.e., $P_{ret-area_i}$) is about $\frac{P_i}{4}$ (Figure 3.2(a)).

Suppose file i has 4 replicas at helpers, and those replicas are placed as shown in Figure 3.2(b). The cost of accessing a replica of file i, by peer x, from the nearest replica at helpers becomes $\frac{d}{2}$; while the average number of file providing peers residing in the retrieval-area of peer x becomes $P_{ret-area_i} = \frac{P_i}{16}$. Similarly, we can compute the access cost and $P_{ret-area_i}$ for peer x when the number of replicas at helpers is 16 (Figure 3.2(c)).

The computation of the cost of accessing file i, by peer x, from replicas at helpers, and the average number of file providing peers in the retrieval-area of peer x for varied number of replicas at helpers are shown in Table 3.2. We note here that our computations are based upon our assumption about the placement of replicas at helpers. Again, replica placement problem is out of the scope of this work.

Table 3.2: Summary of access cost computation for varied number of replicas.

Number of replicas at helpers	1	4	16	64
Access cost	d	$d/2$	$d/4$	$d/8$
Average number of file providers in the retrieval-area ($P_{ret-area_i}$)	$P_i/4$	$P_i/16$	$P_i/64$	$P_i/256$

Let r_i be the number of replicas of file i stored at helpers. We can see that when r_i is increased by a factor of 4, the average cost of accessing a replica of file i at helpers is decreased by a factor of $\frac{1}{2}$. Moreover, when r_i is increased by a factor of 4, the expected number of file providing peers residing in the retrieval-area of any downloading peer in the network (i.e., $P_{ret-area_i}$) is decreased by a factor of $\frac{1}{4}$ (Table 3.2).

Hence, the average cost of accessing file i by any peer in the WMCN can be written as function of r_i as follows

$$c_i \approx \epsilon_i \cdot \frac{d}{\sqrt{r_i}} + \frac{1}{2} \cdot (1 - \epsilon_i) \cdot \frac{d}{\sqrt{r_i}}, \tag{3.2}$$

where

$$\epsilon_i = \frac{\mu_{h_i}}{\mu_{h_i} + \mu_p \cdot \left(\frac{P_i}{4 \cdot r_i} \right)}.$$

The approximation in (3.2) is feasible especially if we consider a large-scale WMCN with many helpers.

The average total access cost of all P2P files in the WMCN during time period T can be computed as

$$c = \sum_{i=1}^{N} p_i \cdot c_i,$$

where p_i is the probability of requesting the i-th file by a peer (i.e. popularity of file i) during time period T. Thus,

$$c = \sum_{i=1}^{N} \frac{d \cdot p_i}{2 \cdot \sqrt{r_i}} \left(1 + \frac{\mu_{h_i}}{\mu_{h_i} + \frac{\mu_p \cdot P_i}{4 \cdot r_i}} \right). \tag{3.3}$$

P_i can be computed using a fluid-flow model such as in [45]. However, we know that P_i scales in p_i, M, and r_i, where M is the total number of peers in the network during time period T. Hence, we can write $P_i \approx k \cdot p_i \cdot M \cdot r_i$, where k is a constant. Hence, we can write (3.3) after omitting the negligible terms as

$$c \approx \sum_{i=1}^{N} \frac{p_i}{\sqrt{r_i}} \cdot \frac{d}{1 + k \cdot \rho \cdot p_i \cdot M},$$

where $\rho = \frac{\mu_p}{\mu_{h_i}}$.

To simplify the analysis, we assume equal-size files. We further assume that every helper can store a maximum of B replicas in its cache and $B \ll N$, where N is the total number of distinct P2P files in the network during time period T.

Hence, our objective is to minimize

$$\sum_{i=1}^{N} \frac{p_i}{\sqrt{r_i}} \cdot \frac{1}{1 + k \cdot \rho \cdot p_i \cdot M}.$$

The optimization is subject to

$$\sum_{i=1}^{N} r_i = K \cdot B,$$

$$1 \leq r_i \leq K,$$

and

$$i = 1, 2, \ldots, N.$$

Recall that K is the number of mesh routers that participate in P2P content sharing (helpers). The second constraint guarantees that every P2P file has at least one replica stored at helpers.

The above optimization problem is equivalent to the utility maximization problem, wherein a rational person with limited budget needs to spend his money to purchase a number of commodities such that his utility function is maximized. Let the utility function be $F_i(r_i) = -\frac{p_i \cdot \theta_i}{\sqrt{r_i}}$, where

$$\theta_i = \frac{1}{1 + k \cdot \rho \cdot p_i \cdot M}.$$

Let the marginal utility function be $\frac{dF_i(r_i)}{dr_i} = \frac{1}{2}(p_i\theta_i)r_i^{-\frac{3}{2}}$. Since $\frac{dF_i(r_i)}{dr_i}$ is a monotonically decreasing function in r_i, the optimum solution can be obtained by applying the law of equi-marginal utility, also known as the law of maximum satisfaction (an extension to the law of diminishing marginal utility that is widely used in economics). Thus, we can write the solution as

$$r_i \propto \left(\frac{p_i}{1 + k \cdot \rho \cdot p_i \cdot M}\right)^{\frac{2}{3}}. \tag{3.4}$$

We note here that for a limited number of helpers, the proposed strategy does not provide a practical number of replicas for each file (i.e., r_i may not be an integer number). However, it gives us an idea about how the optimal number of replicas should vary with respect to other parameters in the P2P data sharing system such as ρ and p_i. However, one way to overcome this problem is to cache fraction of the files that have $r^* < 1$ at the helpers proportional to the value of r^*.

3.5 Practical Implementation

3.5.1 Centralized Replication Algorithm

In this subsection, we assume that the P2P content sharing system in the WMCN employs a centralized unit for monitoring P2P content popularity (e.g., file tracker). The file tracker monitors number of requests that every P2P file receives as well as peers/helpers which possess replicas of every P2P file. Consider a cycle of period T, where T is large. At the beginning of every cycle, the file tracker computes the popularity of every P2P file in the network as $p_i(t) = \frac{g_i}{A}$, where A is the number of requests for all P2P files in the WMCN during the previous time period, and g_i is the number of requests for the i-th file during the previous time period. The file tracker computes the optimum number of replicas at helpers for every P2P file in the network (i.e., the optimum number of helpers in the WMCN that need to store a replica of file i) as

$$r_i^*(t) = \frac{K \cdot B \cdot \left(\frac{p_i}{1+k \cdot \rho \cdot p_i \cdot M}\right)^{\frac{2}{3}}}{\sum_i^N \left(\frac{p_i}{1+k \cdot \rho \cdot p_i \cdot M}\right)^{\frac{2}{3}}}, \tag{3.5}$$

Let $r_i(t)$ be the number of helpers that have file i stored in their caches at instant of time t; while $D_i(t)$ be the number of extra helpers (if any) that must store file i at instant of time t to achieve the optimum file replication strategy. Hence,

$$D_i(t) = r_i^*(t) - r_i(t). \tag{3.6}$$

At the beginning of every cycle, the file tracker sends a message to a number of helpers that possess files $i \in N$ such that $D_i \leq 0$ to evict those files from their caches. Due to the difficulties in incorporating caching functions into the routing module at the network layer [94], we propose to add a sub-layer to the set of layers defined by the IP protocol suite (Figure 3.3).

The proposed caching protocol works as follows: suppose peer j is downloading file i from file provider Y. File provider Y before transmitting the file, appends in the sub-layer of a data packet, information it receives from the file tracker about the transmitted file such as file's popularity p_i, r_i, and D_i. Any helper on the path between Y and j can overhear the transmission and cache file i. However, if multiple helpers are on the path between Y and j, all helpers will store the file. Thus, many replicas of the transmitted file will be concentrated in a small area. To avoid this problem, we add cache-counter

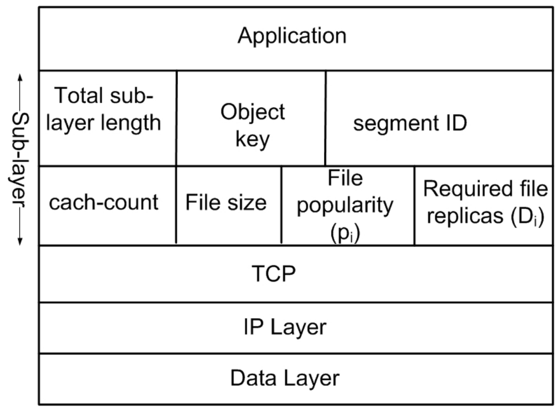


Figure 3.3: Sub-layer Format

field to the sub-layer (Figure 3.3). The cache-counter is increased by one every time a helper on the path to the destination caches the transmitted file.

When a helper in the path between Y and j (say helper x) overhears the transmission, it scans (sniff) the sub-layer in the data packet. If $D_i = 0$, the helper does not need to cache the file. However, if there is enough space in the cache for file i, the helper caches file i. We refer to caching in this case as *optional caching*. If $D_i > 0$, the helper checks if any other helper on the path between Y and x has cached the transmitted packet by scanning the cache-counter field in the packet's sub-layer. If cache-counter>0, this implies that another helper has cached the transmitted packet and it is not necessary for helper x to cache the file. However, if cache-counter=0, helper x caches file i. We refer to caching in this case as *network-demand caching*. If $D_i > 0$ and cache-counter=0, but the cache of helper x is full with network-demand files, the helper x evicts the file in the cache that has the highest popularity (say file L) and replaces it with the incoming file i. The motivation behind this replacement policy is that the popular file L is likely to be required by other peers in the network. Therefore, another helper in the network with under-utilized storage capacity will cache file L soon.

Our performance evaluation shows that the proposed caching policy enables the system to reach steady state fast (Section 3.6). The pseudocode for the caching policy at helpers

Algorithm 2 Pseudocode for the caching policy at helpers.

Define set *Option* as the set of optionally cached files at a helper.
Define set *Demand* as the set of network-demand cached files.
Define *Space* as the empty space in a helper cache at time t.
At a helper:
if $D_i = 0$ **then**
  { **optional caching:** }
  if $Space > 0$ **then**
    cache the file (add file i to set $Option$),
    register with the file tracker as content providing helper for file i,
    $\forall j \in Option$, get the value of c_j from the file tracker.
  else
    if $Option \neq \phi$ **then**
      find file $L \in Option$ such that $L = min\{c_j \forall j \in B\}$,
      if $c_i > L$, replace L with the incoming file i,
    end if
  end if
end if
if $D_i > 0$ **then**
  { **network-demand caching:** }
  if $Space > 0$ **then**
    cache the file (add file i to set $Demand$),
    register with the file tracker as content providing helper for file i.
  end if
  if $Space = 0$, and cache-counter$= 0$ **then**
    if $Option \neq \phi$ **then**
      find file $L \in Option$ such that $L = min\{c_j \forall j \in B\}$,
      replace L with the incoming file i,
      register with the file tracker as content providing helper for file i.
      if $Option = \phi$ **then**
        find file $L \in Demand$ such that $L = max\{p_j \forall j \in B\}$,
        replace L with the incoming file i.
      end if
    end if
  end if
  if $Space = 0$, and cache-counter> 0 **then**
    ignore file i.
  end if
end if

is shown in Algorithm 2. The average access cost for every content i is determined by the file tracker as follows. Every peer which downloads content i within the time period T, computes the file access cost of content i and sends this information to the file tracker. The file tracker upon receiving this information from all peers downloading content i, computes the average file access cost of content i (c_i).

The proposed caching scheme is particularly suitable for WMCNs because unlike MANETs, the route from a file provider to a downloading peer consists of stationary mesh routers. Therefore, the established route remains the same for the entire content download session unless the downloading peer changes its point of attachment. We assume that the proposed protocol does not have to deal with fragmentation issues that may occur at the network layer since all fragments of the data go through same nodes to destination and packets have appropriate size. We assume that a reliable end-to-end transport protocol such as TCP is used for the flow between the source and the destination.

3.5.2 Distributed Replication Algorithm

Our optimum replication strategy that we derive in Section 3.4 requires information about the popularity and number of replicas of every P2P file in the network. Tracking P2P files requires either exchanging large number of messages between peers, or using a centralized entity (e.g., file tracker). This may be costly and challenging in a limited resource and decentralized network such as a WMCN. In this section, we use a low cost distributed (on-line) algorithm for P2P content replication at helpers.

The policy that we consider is simple - easy to implement - because we make no assumptions about the ability of the system to retain a log of file requests and replicas for every P2P file in the network over time. This is not always a constraint, but this assumption is adequate for a WMCN. Our objective is to compare the performance of the centralized replication algorithm and the distributed replication algorithm.

The distributed algorithm works as follows. When a peer p requests file i, it locates replicas of the desired file at the nearest helper (say helper y) and at peers in the retrieval-area. Peer p computes the average cost of accessing file i (i.e., average number of hops on the download paths between peer p and providers of content i, c_i) and sends the value of c_i to helper y. Helper y appends the value of c_i in the sub-layer of a data packet and transmits the packet (segment) to peer p. When a helper (say helper x) on the path between y and p relays (sniffs) the data packet and its cache is full, it implements a replacement algorithm as follows: it finds the access cost value of i (c_i) stored in the sub-layer. Then, it compares (c_i) with $L = min\{c_j \ \forall j \in B\}$, where L is the minimum access cost of all files that are cached at helper x. If $c_i > L$, then x evicts file j which satisfies $c_j = L$ and replaces it with file i. When helper x stores file i, it increase the value of cache-counter by one. A helper on the path does not store the file if cache-counter > 0. Every helper updates $c_j \ \forall j \in B$ continuously.

Let $HelperCache_i(t)$ be the average number of replicas of file i that helpers cache during a unit of time; while $HelperEvict_i(t)$ be the average number of replicas of file i that helpers evict from the cache during a unit of time. We can see that $HelperCache_i(t)$ scales with both the rate at which file i is requested by peers in the WMCN, and the average access cost of file i at time t ($c_i(t)$). Hence, we can write

$$HelperCahce_i(t) = \gamma_2 \cdot c_i(t) \cdot \lambda \cdot p_i,$$

where λ is the rate at which requests for P2P files are initiated at peers in the WMCN, and γ_2 is constant. On the other hand, the $HelperEvict_i(t)$ scales with the number of replicas for file i stored at helpers at time t ($r_i(t)$). Hence, we can write

$$HelperEvict_i(t) = \gamma_1 \cdot r_i(t),$$

where γ_1 is the rate at which a replica of file i is evicted from the helper caches.

We can see that $r_i(t)$ is a random variable varies with time. Hence, we can use fluid-flow model to write

$$\frac{dr_i(t)}{dt} = HelperCache_i(t) - HelperEvict_i(t),$$

Hence, we have

$$\frac{dr_i(t)}{dt} = \gamma_2 \cdot c_i(t) \cdot \lambda \cdot p_i - \gamma_1 \cdot r_i(t),$$

By substituting the value of $c_i(t)$ (Eq. (3.2)) in the above formula, we have

$$\frac{dr_i(t)}{dt} = \gamma_2 \cdot \lambda \cdot p_i \left(\epsilon_i \frac{d}{\sqrt{r_i(t)}} + \frac{1}{2}(1-\epsilon_i)\frac{d}{\sqrt{r_i(t)}} \right) - \gamma_1 \cdot r_i(t).$$

At steady state ($\frac{dr_i(t)}{dt} = 0$), we can compute $\bar{r_i}$ (the equilibrium point) by solving the above equation for r_i as

$$\gamma_2 \cdot d \cdot \lambda \cdot p_i \cdot \left(1 + \frac{4\mu_{h_i}}{4\mu_{h_i} + \mu_p \cdot k \cdot p_i \times M} \right) - 2\gamma_1 \cdot \bar{r_i}^{\frac{3}{2}} = 0.$$

Hence, we can write $\bar{r_i}$ after omitting negligible terms as

$$\bar{r_i} = \left(\left(\frac{\gamma_2 \cdot \lambda \cdot d}{\gamma_1} \right) \cdot \frac{p_i}{1 + k_1 \cdot \rho \cdot p_i \cdot M} \right)^{\frac{2}{3}},$$

where k_1 is constant. We observe that the number of replicas of file i at steady state is consistent with the result in Eq. (3.4). We conclude that the performance of the distributed algorithm when the system is at steady state mimics the optimum replication strategy.

3.6 Performance Evaluation

As we have discussed earlier, P2P content replication problem has not been widely addressed in the wireless mesh networks such as WMCN. There are very few works that address this issue (see related work section). The most relevant replication strategy that can be used for evaluation comparison is the minimum access strategy [42]. Although the minimum access strategy was not designed for replicating P2P files in the WMCN, we believe that it is the best candidate for our performance comparison.

We have demonstrated that this differences in the way a P2P file is retrieved by a downloading user results in different replication strategy for P2P content at mesh routers. Our analytical results show that ratio of the optimal number of replicas of a popular file (i) to the optimal number of replicas of a less popular files (j) using our strategy is less than that of the minimum-access strategy. In other words,

$$\left(\frac{r_i^*}{r_j^*}\right)_{OurStrategy} < \left(\frac{r_i^*}{r_j^*}\right)_{MinAccess}, \forall p_i > p_j.$$

This implies that our replication strategy assigns higher number of replicas at helpers for less popular files as compared to the minimum-access strategy. This is because a downloading peer is likely to locate a large number of replicas of the popular file i at peers in the vicinity (peers nearby) and can, therefore, access segments of file i at those peers at low cost. Hence, the optimum strategy must mitigate the bias towards popular file by allocating higher ratio of the storage capacity at mesh routers to less popular files.

Another feature that distinguishes our replication strategy from the minimum-access is the parameter $\rho = \frac{\mu_p}{\mu_{h_i}}$. In our strategy, the optimum number of replicas for file i at helpers is function of ρ. When ρ increases, ratio of the optimal number of replicas of a popular file i to the optimal number of replicas of a less popular file j decreases. In other words,

$$\left(\frac{r_i^*}{r_j^*}\right)|\rho_1 > \left(\frac{r_i^*}{r_j^*}\right)|\rho_2, \forall\rho_1 < \rho_2, p_i > p_j.$$

This is because when ρ increases, number of segments of the popular file i that a downloading peer retrieves from peers in the vicinity increase. Therefore, the cost of accessing file i decreases. Hence, the optimum strategy should allocate more replicas at helpers for less popular files when ρ increases.

Referring to Eq.(3.4), we note that when $\rho \to 0$ (i.e., $\mu_p \to 0$), we have the minimum-access strategy. This is because when $\mu_p \to 0$, peers do not exchange data between themselves, and a downloading peer retrieves all segments of a required file from the nearest replica at helpers (similar to the minimum-access strategy).

We first used our analytical results to compare the performance of our replication strategy against the minimum-access strategy, when popularity of P2P files followed Zipf-like distribution (MZipf(1,200)) [101]. We computed the total normalized access cost using our replication strategy for varied number of P2P files when ρ takes the values $\rho_1 = 1/4$ and $\rho_2 = 1/3$. We observed that our replication strategy outperforms the minimum-access strategy, and the performance difference increases with increasing number of P2P files (Figure 3.4). Another interesting observation is that the performance difference increases with increasing ρ.

We were concerned about validating the accuracy of our analytical results derived for the P2P content replication in the WMCN. Therefore, we used the simulation results to ensure that the approximations used in our fluid-flow modelling do not impact the validity of our analytical results. To achieve this goal, we simulated a stochastic P2P system using MATLAB. We neither simulated the network layer nor the MAC layer as this is completely irrelevant in evaluating our content replication strategy. In other words, we are only interested in comparing the average number of wireless hops on the download paths between a downloading peer and file providers when different replication strategies

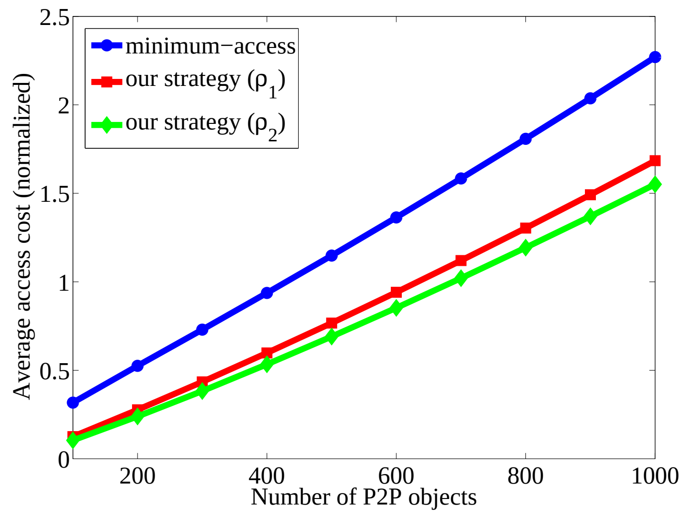


Figure 3.4: Analytical performance comparison between our strategy and the minimum-access strategy.

are used (not the effect of interference between traffic on parallel download paths).

We simulated a WMCN consisting of 1000 stationary mesh routers deployed in a grid topology. Every mesh router has four direct neighbours on average. Each mesh router was equipped with 802.11b radio. We used DSR [77] protocol to route network packets between nodes in the WMCN. 50 mesh routers were randomly selected to play the role of helpers and cache P2P files. 500 peers were uniformly distributed in the network. We did not consider peers mobility. Every peer was connected to the same mesh router for the entire simulation time. N distinct and equal-size P2P files were distributed at peers uniformly at random. Popularity of P2P files followed Zipf-like distribution (MZipf(1,200)). Request for files were generated at peers independently and uniformly at random with rate follows exponential distribution with average $\lambda = 1$ request/minute. Each helper could store 30 replicas in its storage disk; while each peer could store one replica of any file at any instant of time. Ratio of the average rate at which helpers upload data to a downloading peer to the average rate at which peers upload data to a downloading peer was 3:1 (i.e., $\rho = \frac{1}{3}$).

We computed the average access cost for all P2P files that were requested by peers during two hours of simulation time for varied number of P2P files in the network (N)

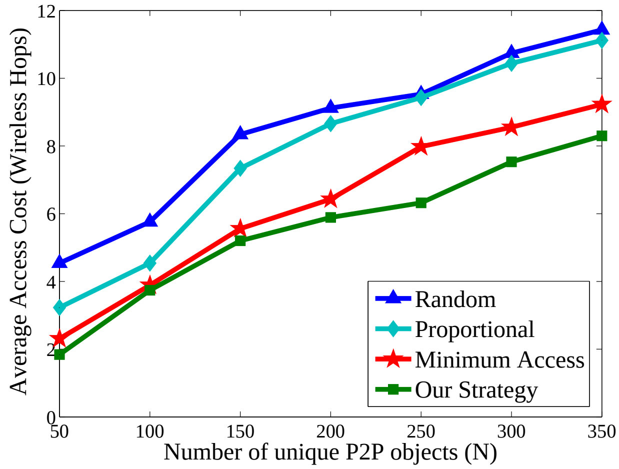


Figure 3.5: Average access cost for varied N in two hours of simulation time.

using the following content replication strategies at helpers:

- Our content replication strategy using the centralized algorithm introduced in Section 3.5.1 (our strategy).
- Minimum access strategy (minimum-access).
- Proportional replication strategy, which replicates each file proportional to its popularity such as LFU and LRU (Proportional).
- Random content replication, where files were randomly replicated at the helpers (Random).

We observed that the simulation results follow similar trends as the analytical results. We observed that our strategy outperforms all other strategies (Figure 3.5). We further observed that for all strategies considered, when the number of P2P files in the network increases, the average total access cost increases. This is because when more files are available in the network, number of replicas for each file decreases.

In another set of simulations, we computed the performance enhancement when using our strategy as compared to the minimum-access strategy during two hours of simulation

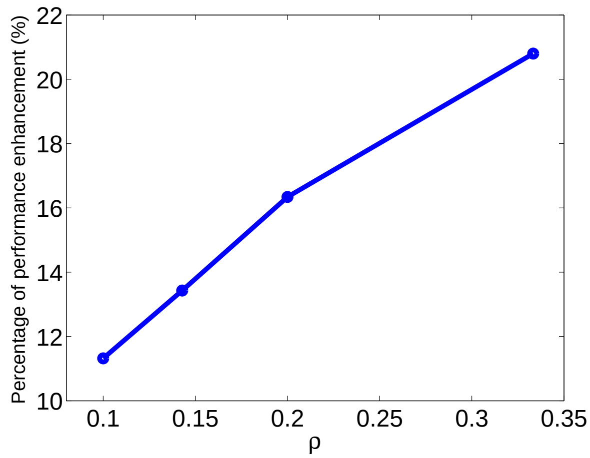


Figure 3.6: Performance comparison for varied ρ (N=250).

time for varied $\rho = \frac{\mu_p}{\mu_h}$ when $N = 250$. We define the performance enhancement as ratio of the average access cost for all P2P files using our strategy to the average access cost using the minimum-access strategy. We observed that the percentage of performance enhancement increases as ρ increases (Figure 3.6). This is because our strategy accounts for data that is exchanged between peers themselves. As we have discussed earlier, when ρ increases (i.e., μ_p increases), our strategy assigns a higher fraction of the storage capacity at helpers for less popular files. Thus, our strategy avoids the bias towards popular files and minimizes the overall access cost of all P2P files.

To better see this behaviour, in another set of simulations, we categorized P2P files into four groups; every group includes 250 file: Files in the first group were with high popularity. Each file in the first group was with popularity 0.0022. Files in the second group were with moderate popularity. Each file in the second group was with popularity 0.0012. Files in the third group were with low popularity. Each File in the third group was with popularity 0.0004. Files in the fourth group were with the least popularity. Each file in the fourth group was with popularity 0.0002. Figure 3.7 shows fraction of files belonging to every group stored at helpers at steady state (i.e., ratio of the number of files belonging to group $i \in \{1, 2, 3, 4\}$ stored at helpers to the number of all P2P files

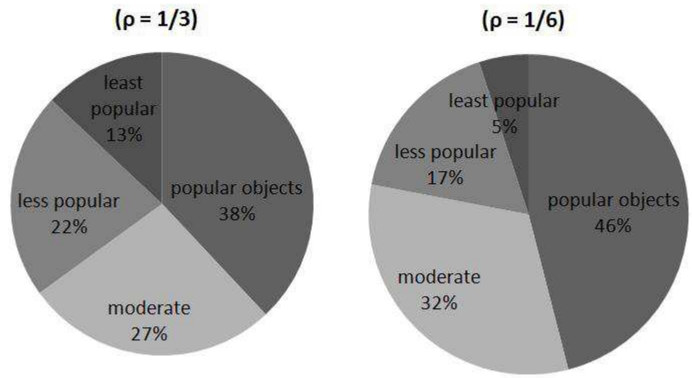


Figure 3.7: Percentage of files stored in the helper caches using our replication strategy.

stored at helpers) using our strategy when ρ takes the values $\rho = 1/3$ and $\rho = 1/6$. We can see that fraction of the popular files (files belonging to the first group) is less when $\rho = 1/3$ as compared to the case when $\rho = 1/6$; while the fraction of the least popular files (files belonging to the fourth group) is higher when $\rho = 1/3$ as compared to the case when $\rho = 1/6$.

In another set of simulations, we computed the average access cost for varied simulation time when using our centralized replication algorithm (Section 3.5.1) and when using our distributed (on-line) replication algorithm (Section 3.5.2) (Figure 3.8). We observed that the difference in the performance decreases with increasing simulation time. Hence, we can conclude that the on-line algorithm mimics the centralized algorithm very well when the system is at steady state. However, recent researches have shown that the popularity of P2P files vary quickly and the popularity of a popular file fades rapidly [99]. Thus, our centralized algorithm, although requires either higher overhead or centralized entity (file tracker), has the advantage over the distributed algorithm that it quickly adapts with the time varying popularity of P2P files and reaches the steady state faster.

In another set of simulations, we examined incentives for the WMCN operators to incorporate the caching features at mesh routers that are required to implement our proposed content replication strategy (i.e., the monetary cost that is required to equip

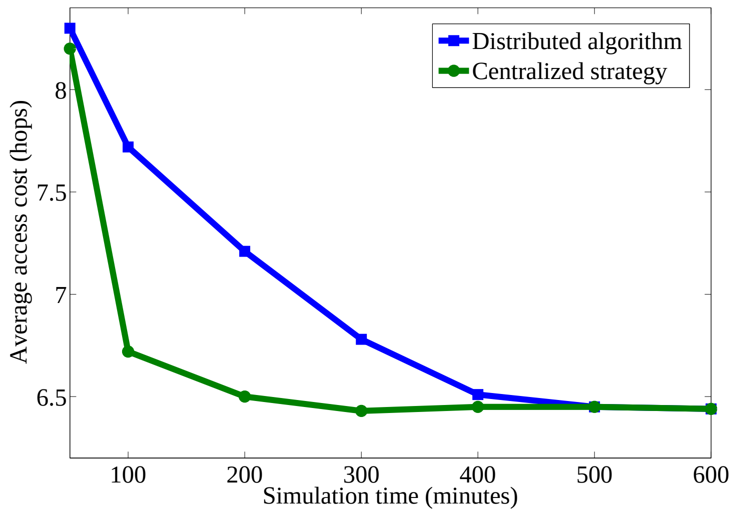


Figure 3.8: Average access cost with simulation time (N=250).

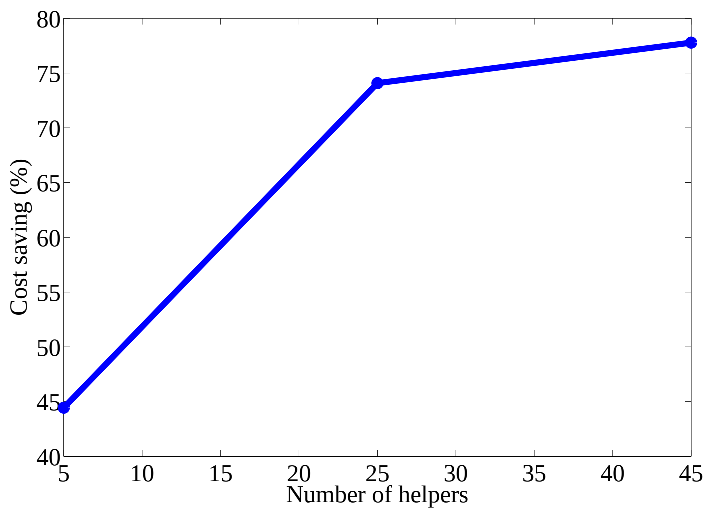


Figure 3.9: Cost saving for varied number of helpers ($N = 350$, $\rho = \frac{1}{3}$)

(over-provision) the mesh routers with storage capacity and P2P-aware devices to enable them to participate in P2P content sharing). In particular, we used simulations to compute the savings in bandwidth and energy that P2P traffic consumes in the WMCN for varied number of helpers as compared to pure P2P content sharing - during two hours of simulation time (Figure 3.9). We assume that bandwidth and energy consumption in a uniformly distributed nodes in a WMCN scale linearly with number of wireless transmissions in the network. Hence, we computed number of wireless transmissions using both P2P with helper system and pure P2P system. The results show that the percentage of saving increases with increasing number of helpers. This is because increasing number of helper increases the number of replicas for any P2P content in the WMCN and, consequently, a downloading peer retrieves large number of segments of a desired content from a nearby helper at low cost. It is worth noting that with only few mesh routers acting as caches and participants in P2P content sharing (5 helpers with limited bandwidth and storage capacity, $\mu_{h_i} = 3\mu_p$, and $B = 30$), the saving is about 45%. These results provide incentive for a WMCN operator to incorporate the required caching features at only small number of mesh routers in the network, and yet achieve high saving in both the cost of P2P communication and consumption of network resources (energy and bandwidth).

3.7 Summary

This chapter considered a P2P content sharing setting in a WMCN, wherein a number of mesh routers participate in the content sharing and cache P2P files. We have introduced our optimum P2P content replication strategy at those participating mesh routers; and we have shown that significant reduction in the cost of P2P content sharing in a WMCN can be realized. We have proposed a centralized content replication algorithm that enables the participating mesh routers to implement the optimum strategy. We have further proposed a distributed and low cost algorithm; and showed that its performance mimics that of the centralized algorithm very well when the system is in steady state.

Our simulation results show that large amount of energy and network bandwidth that P2P traffic consumes in the WMCN can be reduced when only few number of mesh routers participate and cache P2P content. This provides good incentive for a WMCN operator to incorporate the required caching features at the participating mesh routers; and justifies any investment required for equipping (over-provisioning) those mesh routers with additional storage capacity and P2P-aware devices. We note here that our proposed replication strategy can be extended to the case of wired networks. However, the problem is more difficult due to the complexity of the wired network topologies. One possible future research direction is to re-examine our proposed replication strategies when applying heuristics for the replica placement problem.

In this chapter, we considered the case, wherein the evolution of demand for P2P content is constant; and proposed replication strategies for P2P content that target optimizing the performance at the steady state. In the next chapter, we consider a popular setting in community networks, wherein a user who generates a content, spreads the interest in this content to other users in the community. In such cases, content demand grows quickly in a community network as every interested user contacts others and makes them interested, but tapers off and diminishes when all potential interested users in the community network finish downloading the content. In this case, life-time of the content is limited. Therefore, different design strategies are required in order to allocate optimal resources at the helpers that ensure effective usage of helpers; that is what we study in the next chapter.

Chapter 4

Modelling, Performance Analysis, and Design Strategies for P2P-with-helpers: The Case of Content Sharing[3]

4.1 Introduction

In this chapter, we consider a common setting in a community network, wherein a user in the community network (e.g., a student on a college campus) generates a content or gets interested in a content (e.g., campus newsletter, lecture/class note, experimental/scientific data, technical seminar video), and spreads the interest in this content to other users (e.g., classmates) in the community. This scenario is becoming increasingly popular in community networks since users of a community network who share similar interests are often socially connected (e.g., classmates, neighbours, friends in a social network such

[3]This chapter is based in part on the following papers.

1. A. Alasaad, S. Gopalakrishnan, and V. C.M. Leung, "Modelling and performance analysis of content sharing and distribution in community networks with infrastructure support," in *Peer-to-Peer Networking and Applications*, special issue on Peer-to-Peer as infrastructure service, Springer New York, published online in August 2012. DOI: 10.1007/s12083-012-0167-1.

2. A. Alasaad, S. Gopalakrishnan, and V. C.M. Leung, "Green content distribution in wireless mesh networks," in *Proc. of the IEEE ICC GCN workshop*, pp. 5896–5900, Ottawa, ON, June 2012.

as Facebook, Twitter, or email group). Therefore, content demand grows quickly in a community network as every interested user contacts others and makes them interested, but tapers off and diminished when all potential interested users in the community network finish downloading the content. We can see that in this case, the content life-time in the community network is limited and, thus, designs that target optimizing the performance at the steady state are not useful.

Our aim in this chapter is to develop an analytical model for analyzing the performance of the P2P-with-helpers system for content sharing and distribution in a community network. We investigate the role that infrastructure nodes in the community network can play to enhance the performance of content sharing in terms of content download times and energy consumption in the network.

By predicting the evolution of the content demand among users in the community network, we use our derived analytical model to determine the rate at which users are served (download the content) over time. This allows us to develop a demand-aware P2P-with-helpers system that effectively allocate resources at infrastructure nodes participating in content sharing such as the upload bandwidth that helpers allocate for every content and number of replicas for every content at helpers.

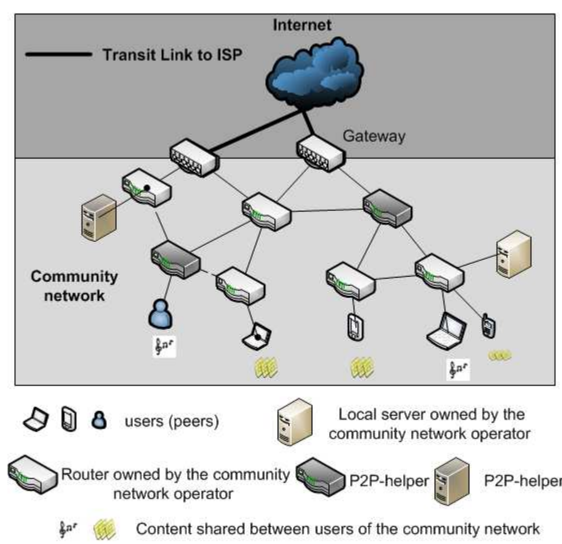


Figure 4.1: Community network architecture

In this chapter, we consider files that are stored in a digital format and that can be used only after downloading the entire file (e.g., P2P file sharing-like service). Although these content do not have a hard delay constraint, users do not like to experience a large waiting time (download time). Pure P2P content sharing may not be able to achieve an acceptable content download time and, hence, helpers are utilized to enhance the performance of content sharing and distribution.

We consider two cases for the hybrid approach for content sharing and distribution as follows. In the first case, few local "mini-servers" in the community network are selected to play the role of helpers. Those selected mini-servers are either owned by users of the wireless community network, or owned by the community network operator. Those mini-servers are typically abundantly available in the community network in large numbers for different purposes (Figure 4.1). Those mini-servers are not dedicated for content sharing, but use their idle resources to help in P2P content sharing and distribution. Examples of potential local mini-servers in a college campus community network that are capable of playing the role of helpers include servers used in laboratories to run simulations or perform experiments, and servers assigned to scientists for saving personal information, storing scientific data, or sharing data. Those local mini-servers are typically inexpensive servers, equipped with commodity hardware, and under-utilized most of the time. Although an individual mini-server is less powerful when compared to a centralized server (dedicated cache), the aggregate capacity of cluster of mini-servers participating in a content distribution absorb large amount of traffic demand in the community network and, consequently, has the potential to significantly enhance the performance of content sharing and distribution. Those mini-servers play the role of helpers in P2P content sharing and distribution. When a mini-server caches a content, it uploads the cached content to interested users in the community network using the client-server scheme (Section 4.4.1). We refer to those servers as *proactive helpers*.

In the second case, a number of relay nodes (e.g., mesh routers and access points APs) in the community network play the role of helpers (Figure 4.1). Those nodes are equipped with storage capacity and P2P-aware devices to enable them to act as caches and participants in the content distribution. Those relay nodes overhear (sniff) the transmission of a P2P content, which takes place on a path between two peers in the community network, and cache segments of the P2P content which they relay to next-hop nodes on the path towards the destination (Section 4.4.4). We refer to those routers as *demand driven helpers*. In this case, a relay node caches a pre-determined fraction of a content (say content i) (ρ_i) that it relays in the network. After downloading/caching the pre-determined fraction of content i, helpers upload the cached content to the interested peers in the network using the client-server scheme.

The proactive helpers do not join the swarm with preloaded content. They need to consume the P2P system bandwidth in order to download a content before being able to help. As we have discussed, in many cases, the content life-time in a community network is limited. Therefore, it is not clear how much bandwidth a local mini-server must contribute (return back) to the P2P system after downloading the content such that the overall bandwidth (capacity) of the P2P system is sustained. In the contrary, demand driven helpers (relay nodes) do not consume any P2P system bandwidth. However, when ρ_i is high, a helper needs a long time in order to cache the required fraction of content i before it can register itself with the file tracker as a content provider and starts uploading the content to interested users in the network. Thus, wasting its available (idle) upload bandwidth. On the other hand, when ρ_i is small, a helper quickly starts uploading the cached content to interested users, but the likelihood that an interested user in the community network is missing a segment of content i that is cached at the helper becomes less. Hence, the main challenge here is to find the optimum pre-determined fraction of content i (ρ_i^*) that a relay node must cache, such that the average download time for content i is minimized.

Since infrastructure nodes that participate in P2P content sharing and play the role of helpers have finite resources (i.e., scarce storage capacity and upload bandwidth), an innovative design is required to guide helpers to take decision on what and how much content to cache, and how to schedule the transmissions of the cached content in events where simultaneous multiple content distributions are required in the network.

Our main goal in this chapter is to provide qualitative and quantitative performance analysis that increases our understanding of how helper provisioning enhances the performance of P2P content sharing and distribution in a community network. We propose design strategies for the HelperDesign Manager in the P2P-with-helpers system that allocate optimal resources at helpers based on prediction of future evolution of content demand in the community network (Figure 1.2). Our derived results are general, and our proposed design strategies can be dynamically used to allocate resources at helpers in a highly responsive manner to match the predicted content demand evolution in individual communities.

We use simulations to evaluate the performance of our ideas and validate the developed analytical models (Section 4.6). The results show that significant reduction in both the cost of content sharing (e.g., energy consumption) and the average content download times can be realized when only few infrastructure nodes in the community network play the role of helpers and participate in P2P content sharing and distribution.

4.2 Related Work

There has been considerable research into understanding P2P communication [115–118]. Many studies have been performed on measurements of different P2P systems [119–122]. Other studies have been conducted to study the performance of Bit-torrent-like systems [45, 121–124]. The studies demonstrate the stability of the BitTorrent and the effectiveness of its incentives mechanism using experimental studies [125], and fluid models [45].

Most prior works used fluid models to analyze the performance of P2P data sharing in the Internet. Other works use probabilistic analysis to evaluate the performance of P2P systems using Markovian process [126, 127]. The fairly accurate qualitative and quantitative results obtained by fluid model at a low numerical complexity make it the de facto standard in analyzing P2P systems. The common assumption in most prior works is that either the evolution of demand for certain content is constant (i.e., Poisson arrivals with constant rate), or occurs simultaneously (i.e., flash crowd) [44, 45]. However, these assumptions have been shown to be unrealistic in an eight months measurement study [128]. The assumption about the Poisson arrivals with constant rate is acceptable in the Internet, where we can assume infinite pool of users who may potentially get interested in a content. However, these assumptions are not always feasible in community networks. We show that in many cases, we have viral evolution of content demand in a community, and limited potential number of interested users in the content in that community.

Many studies propose schemes to enhance the performance of P2P content distribution such as server-assisted P2P [129–133], and the use of helper peers [44, 46–49]. In the server-assisted P2P, a central server is used to boost the performance of P2P content distribution and guarantee service, while in the case of P2P with helpers, the system aims at exploiting the upload bandwidth of idle peers in the network in order to enhance the performance. In contrast with prior works, we carefully characterize the evolution of content demand in a community network (Section 4.3). We show how the awareness of the demand (future demand prediction) can be used to take optimum provisioning decisions in the P2P-with-helpers system for content sharing and distribution in a community network.

Shakkottai *et al.* evaluated the benefit of a hybrid system that combines P2P and a centralized server using a fluid model. The authors employed a word-of-mouth model to characterize the evolution of content interest on the Internet [133], and computed the average content download time. We follow a very similar approach to compute content download time, but for the case of P2P content distribution using helpers. The work by Shakkottai *et al.* assumes that the centralized server join the swarm with a preloaded content. The study suggests that peers use the server in the early phases of interest in a piece of content to boost the performance of the P2P system and switch to the P2P system when the number of individuals (peers) possessing the file increases [133]. In the contrary, we assume that the infrastructure nodes in a community network (the mini-servers and relay nodes) are helpers that need to possess a content before being able to help in distributing the content in the community network.

Wang *et al.* studied the performance of a BitTorrent-like system and investigate the role of helpers [44]. The study assumes that all peers and helpers implement the Tit-for-Tat mechanism that is used in the BitTorrent system. In this chapter, we do not assume symmetry in the traffic exchanged between peers and helpers. To simplify the analysis, we assume that helpers use a simple content retrieval scheme to deliver content to the peers (e.g., client-server scheme). Our aim is to develop a simple analytical model that allows us to quantify the gain that helpers provide to the P2P system, and evaluate conditions at which helpers are useful to the P2P system. Prior studies did not address the case of simultaneous multiple content distributions. We consider this case and we determine the optimum number of replicas that helpers must store in their cache for every content such that the overall average content download times is minimized.

Iamnitchi *et al.* reveal the existence of pattern across diverse data-sharing communities [134]. They introduced an interest sharing graph to capture the implicit relationships that form between users who are interested in the same files. The properties of the graph are used to design a new mechanisms in peer-to-peer content distribution that take into account, adapt to, and exploit user behaviour.

In structured P2P content sharing systems, protocols such as Chord [54] which are based on distributed hash tables are used to lookup P2P content. In unstructured P2P file sharing systems, a query for a P2P file is flooded to neighbouring peers in the overlay network until a replica of the desired file is discovered. We do not consider the cost of locating replicas of a required file in the network (file query cost). The problem of locating content in P2P file sharing is out of scope of the work in this chapter. We have already discussed this problem in Chapter 2.

The cost of storage devices has significantly reduced over the years. Several terabyte+ drives have recently broken the $0.10/gigabyte barrier, making the next milestone $0.01/gigabyte, or $10/terabyte. Many schemes and implementations are available for P2P content caching [39–41, 85–88, 93–96, 99, 100]. Many techniques are used to identify P2P traffic in a community network and capture P2P traffic in the network based on port, IP address or application signatures [28, 102, 105–107]. Balci explains based on his personal experience how to detect P2P activity using either port-based analysis, protocol-based analysis, client-based analysis, and behaviour analysis [102]. Karagiannis *et al.* developed a systematic methodology to identify P2P flows at the transport layer, i.e., based on connection patterns of P2P networks, and without relying on packet payload. Their approach was the first method for characterizing P2P traffic using only knowledge of network dynamics rather than any user payload [105]. Asorey-Cacheda *et al.* provided a brief description of the packet sniffing procedure to boost P2P satellite networks [106]. Cooperative approaches that are based on P2P content caching and P2P traffic monitoring (P2P packet detecting) have been widely used by ISPs to accelerate content delivered over peer-to-peer (P2P) networks, while reducing related bandwidth costs [28, 107]. A large number of P2P packet sniffer implementations are available (e.g., Sniffer Pro, Sinus Packet Sniffer, P2P Rocket 1.3.9, Gimme P2P 1.3.0.0, Zultrax P2P 4.34). All these technologies make caching and detecting P2P content at relay nodes feasible.

4.3 The Demand Model

What distinguishes the evolution of a content demand in a community network from the Internet is that users of the community network who share similar content interests are often socially connected. An example of socially connected users in a college campus community network includes classmates who are interested in sharing lecture notes, scientific data, video technical seminar, or campus newsletter. Those classmates are often socially connected over Facebook, Twitter, or email groups; and most of them are physically located within the college campus. Also, scientists in a college campus who work on a research may want to exchange scientific data to run their computations. Furthermore, employees in an office campus may share data of a common project or meeting minutes. Vehicles in a VANET may be interested in sharing maps or traffic data of a specific location. Thus, the interest in a content (demand) grows quickly in a community network as every interested user contacts others (friends) - using social networks or email broadcast - and makes them interested, but tapers off when a certain level of interest is reached. This is different from the common assumptions about the evolution of content demand in the Internet, that the demand arrives at constant rate (i.e., Poisson arrivals with constant rate), or arrives at once (i.e., flash crowd) [44, 45].

We consider a setting, wherein a user in a community network (e.g., a student on a college campus) generates a file (say file i) and wants to spread (share) it to his/her friends (classmates) in the community network. The object creator spreads the knowledge of the object existence to his friends through social networks (e.g., Facebook, Twitter). We assume that the number of friends to whom a user is connected in the social network (node's degree) at any time unit on average is N. A user who receives the notification about existence of the file i gets interested in the content with probability of p_i, and re-broadcasts the notification in turn to his friends on the social network. We assume that the social network graph is fully connected (i.e., a notification about existence of the file reaches all users in the community). We assume, without loss of generality, that uninterested users who receive multiple notifications for the same content do not rebroadcast the message; while every interested user contacts other users to whom he/she is connected in a social network (friends) and notifies them about the existence of content i every unit of time.

Define $\gamma(t)$ as the ratio of number of friends of an interested user j who receive a notification for the first time to number of friends of the user j whom they received multiple notifications for file i in any time instant (t) . Let $D_i(t)$ be the total number of users interested in content i at any time instant t. Hence, we can write $\gamma(t) := \frac{D_i(t)}{N_T}$, where N_T is the potential number of users in the network who will ultimately become interested in file i, $N_T = 100$ in Figure 4.2 (i.e., N_T is the maximum level of file i cumulative interest in the community network). Thus, we can thus use a fluid-flow model to characterize the the evolution of interest (demand) in content i as

$$\frac{dD_i(t)}{dt} = D_i(t) \cdot \Bigg(p_i(N - \gamma(t) \cdot N) \Bigg),$$

The above formula is a second order Bernoulli differential equation and can be solved as

$$D_i(t) = \frac{N_T \cdot D(0)}{D(0) + \Bigg(N_T - D(0) \Bigg) \cdot e^{(-p_i \cdot N) \cdot t}}. \tag{4.1}$$

The plot of the interest function $(D_i(t))$ is shown in Figure 4.2. We note that $D(t)$ has an S-shape similar to the demand function that was obtained using the word-of-mouth spreading by interested users (Bass model) [135]. It shows that the number of interested users increases quickly when the content becomes available and then gradually decreases.

As we can see from Eq. (4.1), the community network operator can predict the evolution of demand for file i in the community network if information, a priori, about the expected popularity of file i (p_i) and the maximum level of interest (N_T) are known. File popularity p_i is basically a measure of how often file i is requested by users in the community network. It can be computed as ratio of the number of requests generated by peers for file i to the total number of requests for P2P files.

We show in next sections that the awareness of the demand (i.e., future prediction of the content demand) can be used to take provisioning decisions by helpers in the P2P-with-helpers system for content sharing and distribution in a community network.

4.4 Modelling and Analysis of Content Distribution Using Helpers

Our goal in this section is to both analytically quantify the average delay that interested users experience when downloading an object in a community network using the P2P-with-helpers system, and design optimum system's parameters such that storage and bandwidth capacities of the helpers are maximally utilized. We note here that the proposed schemes are only useful for distributing viral files, which is the case in content sharing over community networks as we have seen in the previous section. In contrast to P2P content sharing in the Internet, where network service providers throttle the P2P traffic in their networks [68], we are concerned about traffic that is generated by the community users and shared between users located within the community network. In our proposed P2P-with-helpers systems, there is no traffic delivered to/from the Internet. Hence, using our proposed schemes for content sharing and distribution do not incur any bandwidth cost on the community network operator, and no reason for the community network operator to throttle the P2P traffic. We also note here that in community networks, there often exists a trust between users and the network operator. For example, users of an office campus community network are employees for the community network provider (the employer). Therefore, content illegality and violation of intellectual properties and copyrights are not issues in such content sharing scenarios. In a more general sense, illegal content sharing is a problem that would need solutions that are orthogonal to improving network performance and can be incorporated into our architecture.

We consider the following P2P-with-helpers content distribution schemes:

- Content distribution using proactive helpers (Section 4.4.1);
- Content distribution using demand driven helpers (Section 4.4.4).

In both schemes, we assume that the system has a file tracker that keeps records of peers and helpers in the system which possess every shared content. We further assume that any helper or peer can serve other peers only when it possesses the complete content, and after registering itself with the file tracker as a content provider. We understand that with the ability of a peer to upload data to others while downloading the file, this assumption is restrictive. However, this assumption allows us to obtain simple analytical results that both increase our understanding about the system, and provide us with useful insights about how helper capacities can be optimally utilized to provide the highest performance gain to the system.

We focus our attention on analyzing the performance of P2P-with-helpers system. We neglect the effect of wireless channels in the case where content sharing is taking place over a WMCN (e.g., variable link rates, traffic interference, congestion, shadowing).

4.4.1 Content Distribution Using Proactive Helpers

In this setting, few local mini-servers servers in the community network are selected to play the role of helpers. When a new content (say content i) is generated in the network, helpers download the content from peers who possess content i. In the early phase of content i distribution, only few peers possess the content. Therefore, in order for helpers not to overwhelm the P2P system and consume the limited upload bandwidth capacity of the P2P system, only fraction (ϵ) of the total upload bandwidth of served peers is assigned for serving the helpers. When a helper downloads (caches) the entire content, it participates in content i swarm and starts uploading the cached content to the interested peers.

For simplicity, we assume that the system serves the downloading peers according to their arrival time to the system (first in first served scheme (FIFO)). We show that sacrificing partial P2P system bandwidth (i.e., fraction of upload bandwidth of content providing peers), by uploading content i from peers to the helpers, can significantly decrease the average per-user content download time.

Let $X_{h_i}(t)$ be the number of helpers that download segments of content i at time t, $Y_{h_i}(t)$ be the number of helpers that possess (cache) the entire content i and upload the cached content to interested peers, and $S_i(t)$ be the number of peers that possess the entire content i, i.e., $S_i(t)$ is the number of peers that finish downloading content i. We refer to those peers as served peers. Let R_i be the number of local mini-server which are selected by the network operator to participate in content i swarm and cache replicas of content i, μ_p be the average uploading data rate of peers in file/sec, and μ_{h_i} be the average data uploading rate of helpers in file/sec. To simplify our analysis, we assume that all peers have the same upload bandwidth μ_p, and all helpers have the same upload bandwidth μ_{h_i}. Hence, we can use a fluid-flow model to write

$$\frac{\partial X_{h_i}(t)}{\partial t} = \begin{cases} -\epsilon \cdot \mu_p \cdot S_i(t), & \text{for } \ t \leq t_1 \\ 0, & \text{for } \ t > t_1 \end{cases}$$

$$\frac{\partial Y_{h_i}(t)}{\partial t} = \begin{cases} \epsilon \cdot \mu_p \cdot S_i(t), & \text{for } \ t \leq t_1 \\ 0, & \text{for } \ t > t_1 \end{cases}$$

Note that $X_{h_i}(t) + Y_{h_i}(t) = R_i$, and $X_{h_i}(t=0) = R_i$,

$$\frac{\partial S_i(t)}{\partial t} = \begin{cases} \Big((1-\epsilon)\cdot\mu_p\cdot S_i(t) + \mu_{h_i}\cdot Y_{h_i}(t) - w(t)\Big) - \theta\cdot S_i(t), \text{ for } t \le t_1. \\ \Big(\mu_p\cdot S_i(t) + \mu_{h_i}\cdot Y_{h_i}(t) - w(t)\Big) - \theta\cdot S_i(t), \text{ for } \ t > t_1, \end{cases}$$

where ϵ accounts for the fraction of upload bandwidth of peers that is assigned for serving helpers, t_1 is the instant of time at which $X_{h_i} = 0$ (i.e., the instant of time at which all helpers complete downloading file i), and $w(t)$ is the bandwidth that is wasted by peers who abort the system before downloading the entire file, and can be computed as

$w(t) = \theta\cdot\Big((1-\epsilon)\mu_p S_i(t) + \mu_{h_i} Y_{h_i}(t)\Big)$ for $t < t_1$, where θ is the rate (sec^{-1}) at which peers abort the system and $p_\theta(t) = \theta\cdot e^{-\theta t}$, $t > 0$ is the exponential pdf with mean $\frac{1}{\theta}$. The average wasted bandwidth can be computed as

$$\bar{w} = \frac{\int_0^{t_2} \theta\cdot p_\theta(t) t\Big((1-\epsilon)\mu_p \bar{S}_i + \mu_{h_i}\bar{Y_{h_i}}\Big)dt}{\int_0^{t_2} p_\theta(t)dt},$$

where t_2 is the instant of time at which all interested peers receive a complete copy of file i. It can be shown that the average wasted bandwidth ($\bar{w}$) $\to 0$ for small θ.

In this section, we assume that peers do not leave the system and, thus, $w(t) = 0$ as $\theta \to 0$. We relax this assumption and study the impact of peer departures in Section 4.4.2.

The above system of partially differential equations can be solved as

When $t \le t_1$;

$$X_{h_i}(t) = C_3 - C_1\cdot e^{a\cdot t} - C_2\cdot e^{b\cdot t}, \tag{4.5}$$

$$Y_{h_i}(t) = C_1\cdot e^{a\cdot t} + C_2\cdot e^{b\cdot t}, \tag{4.6}$$

$$S_i(t) = \frac{1}{\epsilon\cdot\mu_p}\cdot\Big(a\cdot C_1\cdot e^{a\cdot t} + b\cdot C_2\cdot e^{b\cdot t}\Big), \tag{4.7}$$

where

$$a = \frac{(1-\epsilon)\mu_p + \sqrt{(1-\epsilon)^2 \cdot \mu_p^2 + 4 \cdot \epsilon \cdot \mu_{h_i} \cdot \mu_p}}{2},$$

$$b = \frac{(1-\epsilon)\mu_p - \sqrt{(1-\epsilon)^2 \cdot \mu_p^2 + 4 \cdot \epsilon \cdot \mu_{h_i} \cdot \mu_p}}{2},$$

$C_1 = Y_{h_i}(0) - C_2$, $C_2 = \frac{\epsilon \cdot \mu_p \cdot S(0) - a \cdot Y_{h_i}(0)}{b-a}$, and $C_3 = X_{h_i}(0) + C_1 + C_2$

When $t > t_1$;

$$X_{h_i}(t) = 0, \tag{4.8}$$

$$Y_{h_i}(t) = R_i, \tag{4.9}$$

$$S_i(t) = C \cdot e^{\mu_p . t} - \frac{\mu_{h_i}}{\mu_p} \cdot R_i, \tag{4.10}$$

where

$$C = S_i(t_1) + \frac{\mu_{h_i}}{\mu_p} \cdot R_i.$$

We can find the total content download times experienced by all users who are interested in file i (T_i) by calculating the area between the curves of cumulative interested peers ($D_i(t)$) and cumulative served peers ($S_i(t)$) (Figure 4.2). Hence,

$$T_i = \int_0^{t_2} (D_i(t) - S_i(t))dt, \tag{4.11}$$

Substituting the values of $S_i(t)$ and $D_i(t)$ in Eq. (4.11), we obtain

$$T_i = \int_0^{t_2} D_i(t)dt - \int_0^{t_1} \frac{1}{\epsilon \cdot \mu_p}\left(a \cdot C_1 \cdot e^{a \cdot t} + b \cdot C_2 e^{b \cdot t}\right)dt - \int_{t_1}^{t_2} \left(C \cdot e^{\mu_p \cdot t} - \frac{\mu_{h_i}}{\mu_p} \cdot R_i\right)dt.$$

Hence,

$$
\begin{aligned}
T_i &= \frac{N_T}{p_i.N} \cdot log\bigg(\frac{D_i(0) \cdot e^{(p_i \cdot N) \cdot t_2} + N_T - D_i(0)}{N_T}\bigg) - \frac{1}{\epsilon \cdot \mu_p}\bigg(C_1 \cdot e^{a \cdot t_1} + C_2 \cdot e^{b \cdot t_1} - C_1 - C_2\bigg) \\
&+ \frac{\mu_{h_i} \cdot R_i}{\mu_p} \cdot (t_2 - t_1) - \frac{C}{\mu_p} \cdot \bigg(e^{\mu_p \cdot t_2} - e^{\mu_p \cdot t_1}\bigg).
\end{aligned}
$$

To compute t_1, we let $X_{h_i}(t) = 0$ in Eq. (4.5), and solve for t. Since $b < 0$ and t_1 is large, we can write

$$t_1 \approx \frac{1}{a} \cdot log\bigg(\frac{C_3}{C_1}\bigg).$$

To compute t_2, we let $S_i(t) = D_i(t)$ in Eq. (4.10) and solve for t. If we assume that the rate at which peers become interested in content i is much higher than the rate at which peers are served, we can compute t_2 by solving the following formula for t

$$C \cdot e^{\mu_p \cdot t_2} - \frac{\mu_{h_i}}{\mu_p} \cdot R_i = N_T.$$

Hence,

$$t_2 \approx t_1 + \frac{1}{\mu_p} \cdot log\bigg(\frac{N_T + \frac{\mu_{h_i}}{\mu_p} \cdot R_i}{C}\bigg).$$

Dividing T_i by the number of all potentially interested users in the network (N_T), the average per-user download time of file i is

$$T_{dl_i} = \frac{T_i}{N_T}.$$

We note that for large N_T, the average per-user content download time is approximately

$$T_{dl_i} = \bigg(\frac{1}{\mu_p} - \frac{1}{p_i N}\bigg) \cdot log(N_T) - (\frac{1}{\mu_p}) \cdot log\bigg(\frac{\mu_{h_i}}{\mu_p} R_i\bigg). \tag{4.12}$$

This implies that the average download time experienced by users when using this scheme

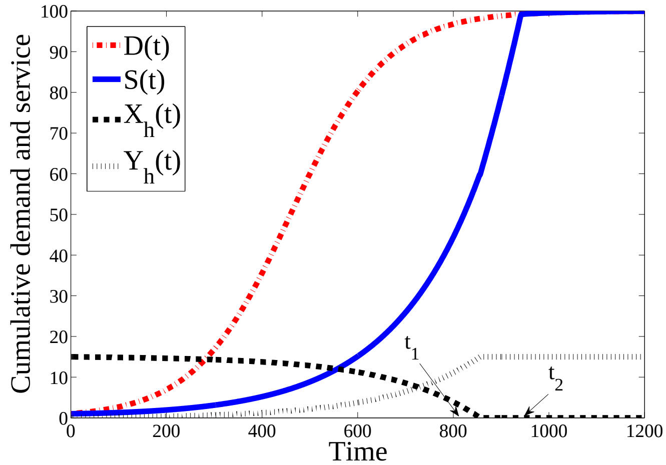


Figure 4.2: The evolution of the demand and served peers and helpers in the proactive helpers scheme when $p_i = 0.001$, $N = 10$, $T = 100000$, $D(0) = 1$, $R_i = 15$, $\mu_p = 0.003$, $\mu_h = 0.015$, $\epsilon = 0.45$.

is $\Theta\Big(log(N_T)\Big)$. We note that T_{dl_i} scales with the number of potentially interested users ($log(N_T)$); while scales inversely with both the ratio of the upload rate at helpers to the upload rate at peers, and the number of participating helpers ($log(\frac{\mu_{h_i}}{\mu_p}R_i)$). This generally implies that μ_{h_i} should be larger than μ_p for helpers to be useful to the P2P system, which motivates the use of infrastructure nodes as helpers since infrastructure nodes have much higher bandwidth compared to the peers.

Let

$$A_1 := \frac{1}{t_1}\int_0^{t_1} \epsilon \cdot \mu_p \cdot S_i(t)dt$$

be the average bandwidth (file/sec) that helpers consume form the P2P system, and let

$$A_2 := \frac{1}{t_2}\int_0^{t_2} \mu_{h_i} \cdot Y_{h_i}(t)dt$$

be the average bandwidth (file/sec) that helpers contribute (return back) to the P2P system. We note that for helpers to be useful to the P2P system (i.e., for helpers to sustain the P2P system bandwidth capacity), we need $A_1 \leq A_2$. We can approximate this condition as

$$\frac{R_i}{t_1} \leq \frac{1}{t_2}\left(\frac{C_1 \cdot \mu_{h_i}}{a} \cdot e^{a \cdot t_1} + \mu_{h_i} \cdot R_i(t_2 - t_1) - \frac{\mu_{h_i} \cdot C_1}{a} - \frac{\mu_{h_i} \cdot C_2}{b}\right).$$

Solving the above inequality for μ_{h_i} gives the minimum required upload data rate that helpers should assign for the distribution of content i (i.e., μ_{h_1}) in order to sustain the P2P system. However, in order to ensure that every helper is useful to the P2P system not only the overall cluster of helpers, the amount of traffic that the helper - who is the last to possess the content i - contributes (return back) to the P2P system must be higher than the amount it consumes from the P2P system. Thus, we have

$$\mu_{h_i} \cdot (t_2 - t_1) \geq L,$$

where L is the size of file i (normalized $L = 1$)). Hence, we have

$$\mu_{h_i} \geq \frac{1}{(t_2 - t_1)}.$$

Solving the above inequality for μ_{h_i} gives us the minimum required upload data rate at helpers (μ_{h_2}) such that the all helpers are useful. Hence,

$$\mu_{h_2} \geq \frac{\mu_p}{log\left(N_T + \frac{\mu_{h_i}}{\mu_p} \cdot R_i\right) - log(C)}.$$

Combining the above two conditions for μ_{h_i}, we can determine the minimum required upload rate at helpers that ensures effective use of helpers (i.e., efficient P2P-with-helpers system) as

$$\mu_{h_{min}} = argmax(\mu_{h_1}, \mu_{h_2}), \tag{4.13}$$

We note that for large N_T, $\mu_{h_{min}}$ is approximately $\frac{\mu_p}{log(N_T) - log(\mu_{h_i} \cdot R_i) + log(\mu_p)}$. We observe that $\mu_{h_{min}}$ scales with μ_p, but scales inversely with N_T.

If the upload data rate that helpers dedicate for content distributions is limited (i.e., μ_{h_i} is scarce), then one way to reduce content i download time is to increase number of replicas of content i that helpers store in their caches (R_i). However, increasing R_i increases the bandwidth that helpers consume from the P2P system in order to cache content i. The plot of function $T_{dl_i}(R_i)$ indicates that this function is indeed convex (Figure 4.7). Therefore, to find the optimum R_i (R_i^*), we need to solve $\frac{\partial T_{dl_i}}{\partial R_i} = 0$ for R_i. The problem is solved for $\frac{\mu_{h_i}}{\mu_p} \cdot R_i \ll N_T$ and large ϵ as

$$R_i^* \approx \frac{N_T^2 \cdot a}{log\left(\frac{C_3}{C_1}\right)}$$

Recall that N_T is the number of users in the network who will eventually become interested in content i ($N_T = 100$ in Figure 4.2). Note that N_T is proportional to the popularity of content i (p_i) in the community network. Hence, the above formula implies that R_i^* scales with p_i.

So far, we considered the case of single content distribution in a community network. However, in reality, multiple files can be distributed in the network simultaneously, and the evolution of interest (demand) in each file is different. We, therefore, study the performance of the P2P-with-helpers system when multiple files are distributed simultaneously in the community network, while finite resources (limited upload bandwidth and storage capacity) are employed at the helpers for content sharing and distribution.

Let F be the total number of files that are simultaneously distributed in the network in any time instant t. Our objective is to find both the optimum number of replicas for every file $i \in F$ stored at the helpers (R_i^*), and the optimum upload data rate that helper must assign (allocate) for every object i stored in their caches, such that the summation of the average download times of all F files is minimized. Let p_i be the popularity of each file $i \in F$, and the demand for each file be $D_i(t)$ as described in Section 4.3. Define T_{total}

as

$$T_{total} := \sum_{i=1}^{F} T_{dl_i}.$$

Our objective is to minimize T_{total}. This optimization is subject to

$$(i) \qquad \sum_{i=1}^{F} R_i = H \cdot B,$$

where H is the number of helpers that are selected to play the role of helpers and cache files, B is the maximum number of files that can be stored in any helper's cache (storage capacity at the helpers). We assume that all files have the same size, and that all helper have the same storage capacity. We further assume that helpers have limited storage capacity such that $H \cdot B \ll \sum_{i=1}^{F} R_i^*$.

$$(ii) \qquad 1 \leq R_i \leq H.$$

The second constraint guarantees that every file has at least one replica stored at the helpers, and that number of replicas for file i does not exceed the number of helpers. Let us define N_T as $N_T := \lambda_T \cdot p_i \cdot T$, where λ_T is the rate at which new files are generated in the network, and T is the population of the community network (number of users in the network). It can be shown that for $\mu_{h_i} \gg \mu_p$ and $t_2 \gg t_1$, the objective function above is equivalent to minimizing

$$\sum_{i=1}^{F} \frac{\lambda_T \cdot p_i \cdot T}{\mu_p} \cdot log\left(\frac{\lambda_T \cdot p_i \cdot T + \frac{\mu_{h_i}}{\mu_p} \cdot R_i}{\frac{\mu_{h_i}}{\mu_p} \cdot R_i}\right).$$

It can be shown that the solution of the above optimization problem is obtained when $R_i \propto p_i$. Similarly, it can be easily shown that the optimum upload bandwidth allocation at helpers for any file $i \in F$ is when $\mu_{h_i} \propto p_i$. This implies that despite the high cumulative bandwidth that the served peers contribute to the P2P system when uploading a popular file as compared to the case when uploading a less popular file, helpers should always favour popular files when it comes to selecting files to store in their limited size caches or when allocating upload bandwidth for transmitting the cached files.

4.4.2 Impact of Peer and Helper Departures

In this section, we consider the impact of peer departures on the average download time (i.e., $\theta > 0$). We further assume that helpers can join and leave the network any time at well. We assume that helpers join swarm i according to Poisson distribution with mean λ_h. let θ be the average peers departure rate, γ_h be the average departure rate of the served helpers, θ_h be the average departure rate of the downloading helpers. Hence, we can use the fluid model to write

$$
\begin{aligned}
\frac{\partial X_{h_i}(t)}{\partial t} &= \lambda_h - (1-\theta_h)\cdot\epsilon\cdot\mu_p\cdot S_i(t) - \theta_h\cdot X_{h_i}(t),\\
\frac{\partial Y_{h_i}(t)}{\partial t} &= (1-\theta_h)\cdot\epsilon\cdot\mu_p\cdot S_i(t) - \gamma_h\cdot Y_{h_i},\\
\frac{\partial S_i(t)}{\partial t} &= (1-\theta)\cdot\Big((1-\epsilon)\cdot\mu_p\cdot S_i(t) + \mu_{h_i}\cdot Y_{h_i}(t)\Big) - \theta\cdot S(t).
\end{aligned}
$$

Hence, we can solve the above system of partially differential equations as

$$
S_i(t) = \frac{1}{(1-\theta_h)\cdot\epsilon\cdot\mu_p}\cdot\Big(C_1\cdot a\cdot e^{a\cdot t} + C_2\cdot b\cdot e^{b\cdot t} + \gamma_h\cdot C_1\cdot e^{a\cdot t} + \gamma_h\cdot C_2\cdot e^{b\cdot t}\Big),
$$

and

$$
Y_{h_i}(t) = C_1\cdot e^{a\cdot t} + C_2\cdot e^{b\cdot t},
$$

where $C_1 = \frac{(1-\theta_h)\epsilon\mu_p}{a-b}$, $C_2 = -C_1$. $a = \frac{-\zeta_2+\sqrt{\zeta_2^2-4\zeta_1\zeta_3}}{2\zeta_1}$, and $b = \frac{-\zeta_2-\sqrt{\zeta_2^2-4\zeta_1\zeta_3}}{2\zeta_1}$, where $\zeta_1 = \frac{1}{(1-\theta_h)\epsilon\mu_p}$, $\zeta_2 = \frac{\gamma_h}{(1-\theta_h)\epsilon\mu_p} - \frac{(1-\epsilon)(1-\theta)}{(1-\theta_h)\epsilon} + \frac{\theta}{(1-\theta_h)\epsilon\mu_p}$, $\zeta_3 = \frac{-(1-\epsilon)\mu_p\gamma_h(1-\theta)}{(1-\theta_h)\epsilon\mu_p} - \mu_{h_i}(1-\theta) + \frac{\theta\gamma_h}{(1-\theta_h)\epsilon\mu_p}$, where we set $S(0) = 1$ and $Y_{h_i}(0) = 0$. Let t be the time at which all interested peers in the network are served. Thus, t can be computed as

$$t \approx \frac{1}{a} \cdot log\bigg(\frac{(1-\theta_h)\cdot \epsilon \cdot \mu_p \cdot N_T}{C_1 \cdot a + \gamma_h \cdot C_1}\bigg).$$

Hence, the total content download delay experienced by all users who are interested in file i can be computed as

$$\begin{aligned} T_i &= \int_0^t (D(t)-S(t))dt = N_T \cdot t - \frac{N_T}{N \cdot p_i} log(N_T) - \frac{1}{(1-\theta_h)\epsilon\mu_p} \cdot C_1(1+\frac{\gamma_h}{a}) \cdot \\ &\bigg(\frac{(1-\theta_h)\epsilon\mu_p N_T}{C_1 a + \gamma_h C_1} - 1\bigg) - \frac{1}{(1-\theta_h)\epsilon\mu_p} \cdot C_2(1+\frac{\gamma_h}{b}) \cdot \bigg(e^{\frac{b}{a} log(\frac{(1-\theta_h)\epsilon\mu_p N_T}{C_1 a + \gamma_h C_1})} - 1\bigg). \end{aligned}$$

Thus, for large N_T, the average per-user content download delay is approximately

$$T_{dl_i} = \frac{1}{a} \cdot log\bigg(\frac{(1-\theta_h)\cdot \epsilon \cdot \mu_p \cdot N_T}{C_1 \cdot a + \gamma_h \cdot C_1}\bigg) - \frac{1}{N \cdot p_i} \cdot log(N_T).$$

4.4.3 Sensitivity Analysis

In this section, we perform sensitivity analysis to identify system's parameters that are more important and have more impact on the P2P-with-helpers system's behaviour. The purpose of this study is to increase our understanding of the P2P-with-helpers system for content sharing in community networks. Since the average content download delay function (T_i) is well behaved (i.e., has a continuous derivative), we perform analytic sensitivity to analyze the P2P with proactive helper system in the case when no peer departures, and in the case of peer departures.

Without peer departures

To compare the effect of different system's parameters on the average content download time, we use relative sensitivity functions. The relative sensitivity of function T_i to the parameter μ_p evaluated at the normal operating point is given by

$$\bar{S}^{T_i}_{\mu_p} = \left(\frac{\partial T_i}{\partial \mu_p}\right)_{NOP} \left(\frac{\mu_{p_0}}{T_{i_0}}\right),$$

where NOP and the subscript 0 mean that all functions and parameters assume their normal operating point values. Normal operating point is defined as the point at which normal operation is expected and optimum efficiency of the system is desired. This is usually the point at which the vendor certifies that performance is within the tolerances stated in the standard. The relative sensitivity functions are ideal for comparing parameters because they are dimensionless and normalized functions. It also allows us to choose the easiest way (smallest percent change in an operating point parameter) to reduce the download time and enhance the performance.

We assume a normal operating point of the system to carry on our sensitivity analysis. We note here that different normal operating points can be used. However, we do not expect changes in the results of our sensitivity analysis that would lead to different conclusions. For a P2P-with-helpers system operating at its normal operating point, say: $\epsilon = 0.45$, $\mu_p = 0.003$, $\mu_{h_i} = 0.015$, $R_i = 15$, we can compute the relative sensitivity functions for file i with popularity $p_i = 0.001$, and number of potential interested users $N_T = 1000$ as

$$\begin{aligned}
\bar{S}^{T_i}_{\mu_p} &= \left(\frac{\partial T_i}{\partial \mu_p}\right)_{NOP} \left(\frac{\mu_{p_0}}{T_{i_0}}\right) = -1.2533, \\
\bar{S}^{T_i}_{\mu_{h_i}} &= \left(\frac{\partial T_i}{\partial \mu_{h_i}}\right)_{NOP} \left(\frac{\mu_{h_{i_0}}}{T_{i_0}}\right) = -0.7359, \\
\bar{S}^{T_i}_{\epsilon} &= \left(\frac{\partial T_i}{\partial \epsilon}\right)_{NOP} \left(\frac{\epsilon_0}{T_{i_0}}\right) = -0.4255, \\
\bar{S}^{T_i}_{R_i} &= \left(\frac{\partial T_i}{\partial R_i}\right)_{NOP} \left(\frac{\epsilon_0}{T_{i_0}}\right) = -0.0816.
\end{aligned}$$

We can see that the most sensitive parameter is μ_p. This implies that despite the high upload rate at helpers and the large number of participating helpers, peers need to upload data to each other in order to maintain an acceptable content download delay. Similar results were obtained in [45], when analyzing the performance of BitTorrent.

With peer departures

We now use relative sensitivity functions to compare the effect of different parameters on the average content download time in the case of peer and helper departures. For a P2P-with-helpers system operating at its normal operating point, say: $\epsilon = 0.45$, $\mu_p = 0.003$, $\theta = 0.01$, $\mu_{h_i} = 0.015$, $\gamma_h = 0.0015$, $\theta_h = 0.0015$, $R_i = 15$, we can compute the relative sensitivity functions for file i with $p_i = 0.001$ and $N_T = 1000$ as

$$\begin{aligned}
\bar{S}_{\mu_p}^{T_i} &= \left(\frac{\partial T_i}{\partial \mu_p}\right)_{NOP} \left(\frac{\mu_{p_0}}{T_{i_0}}\right) = -3.45, \\
\bar{S}_{\theta}^{T_i} &= \left(\frac{\partial T_i}{\partial \theta}\right)_{NOP} \left(\frac{\theta_0}{T_{i_0}}\right) = 2.5114, \\
\bar{S}_{\mu_{h_i}}^{T_i} &= \left(\frac{\partial T_i}{\partial \mu_{h_i}}\right)_{NOP} \left(\frac{\mu_{h_{i_0}}}{T_{i_0}}\right) = -3.147, \\
\bar{S}_{\gamma_h}^{T_i} &= \left(\frac{\partial T_i}{\partial \gamma_h}\right)_{NOP} \left(\frac{\gamma_{h_0}}{T_{i_0}}\right) = 1.4988, \\
\bar{S}_{\theta_h}^{T_i} &= \left(\frac{\partial T_i}{\partial \theta_h}\right)_{NOP} \left(\frac{\theta_{h_0}}{T_{i_0}}\right) = 0.0038, \\
\bar{S}_{\epsilon}^{T_i} &= \left(\frac{\partial T_i}{\partial \epsilon}\right)_{NOP} \left(\frac{\epsilon_0}{T_{i_0}}\right) = -2.3016.
\end{aligned}$$

The selection of parameters $\theta > \theta_h$ allow us to make the following observations. We can see that μ_{h_i} has a larger impact on the system as compared with the case of no peer departures. This implies that when peers depart the system at a high rate, the content providing peers alone are not sufficient to handle the demand, and support from helpers become necessary in order to maintain an acceptable system performance. The high value of $\bar{S}_{\epsilon}^{T_i}$ as compared to the case of no peer departures implies that high fraction of the upload bandwidth of the served peers should be assigned for serving helpers in order to utilize the high upload bandwidth at the helpers and substitute the bandwidth that is wasted due to the high rate of peer departures.

4.4.4 Content Distribution Using Demand Driven Helpers

In this P2P-with-helpers scheme, few relay nodes (e.g., routers) in the community network are selected to play the role of helpers. Those routers are equipped with packet sniffers to enable them to overhear (sniff) the transmission of a P2P content (say content i), which is taking place on a path between any two peers in the community network, and cache segments of the P2P content i which they relay to the next-hop node on the path towards destinations. Therefore, the rate at which helpers join the content i swarm and participate in the distribution of content i in the network is driven by the demand for content i generated by interested peers in the community network, so the name of this scheme. When a helper caches a pre-determined fraction of the content i (ρ), it registers itself with the file tracker as content provider, and starts uploading the cached content to interested peers in the network.

In contrast with the proactive helpers (mini-server scheme) discussed in section 4.4.1, peers do not upload P2P content to the helpers and, hence, helpers do not consume any P2P system bandwidth. However, when ρ is high, the helper needs a long time in order to cache ρ amount of content i before it can start uploading the cached content to interested peers. Thus, wasting its idle upload bandwidth capacity. On the other hand, when ρ is small, the helper quickly starts uploading the cached content to interested peers, but the likelihood that an interested peer in the network is missing a chunk of object i cached at the helper becomes less.

In our modelling, we assume that the system serves downloading peers according to their arrival time to the system (e.g., FIFO). We further assume, without loss of generality, that in every unit of time, a peer that is interested in content i requests missing segments from all helpers in the network. If a helper possesses a required segment, it uploads it to the peer.

Let P_{cache} be the probability that any participating router overhears (relays) a segment of content i that is transmitted on a path between any two peers, and can be defined as $P_{cache} := (\frac{H}{M}) \cdot HOPs$, where H is the total number routers that are selected to play the role of helpers, M is the total number of routers in the network, and $HOPs$ is the average number of hops in a route between any two peers in the network. Define $\eta_{h-p}(t)$ as the effectiveness of the helpers at time t and takes values $\in [0, 1]$ (i.e., $\eta_{h-p}(t)$ is the probability that in a unit of time t, at least one chunk of file i cached at a helper is missing by a downloading peer). We can see that $\eta_{h-p}(t) = 1 - (1-\rho)^{D_i(t)-S_i(t)}$. For simplification, let $\eta_{h-p}(t) := \rho$.

Hence, we can use a fluid model to write

$$\frac{\partial S_i(t)}{\partial t} = \begin{cases} \mu_p \cdot S_i(t), & \text{for } t \leq t_1 \\ \mu_p \cdot S_i(t) + \mu_{h_i} \cdot \eta_{h-p} \cdot H, & \text{for } t > t_1. \end{cases}$$

The above system of ordinary differential equations can be solved as

$$S_i(t) = \begin{cases} S_i(0) \cdot e^{\mu_p t}, & \text{for } t \leq t_1 \\ C \cdot e^{\mu_p t} - \dfrac{\mu_{h_i}}{\mu_p} \cdot \eta_{h-p} \cdot H, & \text{for } t > t_1, \end{cases}$$

where $S_i(0)$ is the initial number of interested peers, $C = S_i(t_1) + \frac{\mu_{h_i}}{\mu_p} \cdot \eta_{h-p} \cdot H$, and t_1 is the instant of time at which helpers cache the entire pre-determined fraction of file i (ρ) and start uploading the cached content to interested peers, and can be computed by solving the following formula for t.

$$\int_0^t \mu_p \cdot \left(\frac{P_{cache}}{H}\right) \cdot S(t)dt = \rho,$$

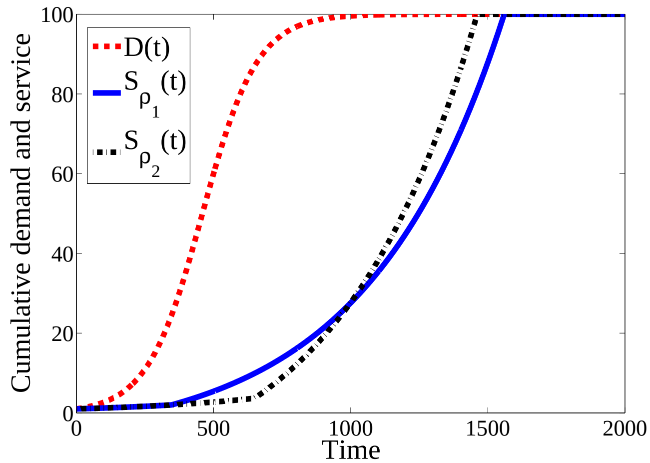


Figure 4.3: The evolution of the demand and served peers in the demand driven helpers scheme when $p_i = 0.003$, $\rho_1 = 2\%$, $\rho_2 = 4\%$, $M = 1000$, $HOPs = 5$, $N = 10$, $T = 100000$, $D(0) = 1$, $H = 10$.

where $\mu_p \cdot \left(\frac{P_{cache}}{H}\right) \cdot S(t)$ is the rate at which a helper caches (relays) segments of content i (we assume it to be same for all helpers). Hence,

$$t_1 = \frac{1}{\mu_p} \cdot log\left(\frac{H \cdot \rho + P_{cache} \cdot S_i(0)}{P_{cache} \cdot S_i(0)}\right). \tag{4.16}$$

Hence, we can find the total content download times experienced by all users who are interested in file i by calculating the area between the curves of the cumulative interested ($D_i(t)$) and the served peers ($S_i(t)$) (Figure 4.3) as

$$T_i = \int_0^{t_2} (D_i(t) - S_i(t))dt,$$

where t_2 is the time instant at which all peers are served. If we assume that the rate at which peers become interested is much higher than the rate at which peers are served, we can find t_2 by solving the following formula for t.

$$C \cdot e^{\mu_p \cdot t} - \frac{\mu_{h_i}}{\mu_p} \cdot \eta_{h-p} \cdot H = N_T.$$

Hence,

$$t_2 = t_1 + \frac{1}{\mu_p} \cdot log\bigg(\frac{N_T + \eta_{h-p} \cdot H \cdot (\frac{\mu_{h_i}}{\mu_p})}{C}\bigg).$$

Substituting the values of $D_i(t)$ and $S_i(t)$ in the T_i formula above, we obtain

$$\begin{aligned} T_i &= \frac{N_T}{p_i \cdot N} \cdot log\bigg(\frac{D_i(0) \cdot e^{p_i N \cdot t_2} + N_T - D_i(0)}{N_T}\bigg) - \frac{S_i(0)}{\mu_p} \cdot (e^{\mu_p \cdot t_1} - 1) \\ &+ \frac{\mu_{h_i}}{\mu_p} \cdot \eta_{h-p} \cdot H \cdot (t_2 - t_1) - \frac{C}{\mu_p}(e^{\mu_p \cdot t_2} - e^{\mu_p \cdot t_1}), \end{aligned}$$

Dividing T_i by the number of served users (N_T), the average per-user content download time of the file becomes $T_{dl_i} = \frac{T_i}{N_T}$. For large N_T, the average per-user content download time is approximately

$$T_{dl_i} = \bigg(\frac{1}{\mu_p} - \frac{1}{p_i \cdot N_T}\bigg) \cdot log(N_T) - \frac{1}{\mu_p} \cdot log\bigg(\frac{\mu_{h_i}}{\mu_p}\eta_{h-p} H\bigg). \tag{4.17}$$

We can see that the average content download time per-user is $\Theta\bigg(log(N_T)\bigg)$. We can also observe that the average content download time per-user scales inversely with $\frac{\mu_{h_i}}{\mu_p}$ and H.

As we have discussed so far, when ρ is large, the average time that helpers need in order to cache the required fraction of the file may be high, i.e., t_1 is large (Eq. (4.16)), which results in wasting helper idle upload bandwidth; while if ρ is small, the effectiveness of the helpers is low (i.e., η_{h-p} is small). Hence, to minimize T_{dl_i}, we have to find the optimum ρ (ρ^*). The plot of function $T_{dl_i}(\rho)$ shows that the function is indeed convex (Figure 4.8). Hence, we can solve $\frac{\partial T_{dl_i}}{\partial \rho} = 0$ for ρ to obtain ρ^*. Since $H \ll N_T$, $S_i(0) = D_i(0) = 1$, and $\mu_{h_i} \gg \mu_p$; and for small value of P_{cache}, we can solve the following formula for ρ to obtain ρ^*

$$\frac{\partial T_{dl_i}}{\partial \rho} \approx \frac{\mu_{h_i} \cdot H}{\mu_p^2} \cdot log\bigg(\frac{N_T \cdot \mu_p}{\mu_{h_i} \cdot \rho \cdot H}\bigg) - \frac{\mu_{h_i} \cdot H}{\mu_p^2} - \frac{H}{\mu_p \cdot P_{cache}} = 0.$$

Hence,

$$\rho^* \approx \left(\frac{N_T \cdot \mu_p}{\mu_{h_i} \cdot H}\right) \cdot e^{\left(-\frac{\mu_p + \mu_{h_i} \cdot P_{cache}}{\mu_{h_i} \cdot P_{cache}}\right)}.$$

Recall that P_{cache} is proportional to the number of participating helpers (H). Hence, we note that ρ^* scales with both H, and the ratio of upload rate at helpers to upload rate at peers ($\frac{\mu_{h_i}}{\mu_p}$). This is because when H and $\frac{\mu_{h_i}}{\mu_p}$ are large, the amount of bandwidth that helpers contribute to the P2P system is high and, thus, caching a large fraction of content i at helpers provides high performance gain. Another observation is that ρ^* scales with the potential number of interested users in the network (N_T).

Let us now consider the case of simultaneous multiple content distributions. Let F be the total number of files that are simultaneously distributed in the network in any time instant t. Our objective is to find the optimum fraction of every file i (ρ_i^*) that helpers must store in their caches, such that the summation of the overall average content download times of all objects is minimized. Define $T_{total} := \sum_{i=1}^{F} T_{dl_i}$. Our objective is to minimize T_{total}. The optimization is subject to

$$(i) \qquad \sum_{i=1}^{F} \rho_i \cdot L \leq C,$$

where C is the size of the helper cache given that $C \ll \sum_{i=1}^{F} \rho_i^* \cdot L$, and L is the size of file i (we assume all files have the same size); and

$$(ii) \qquad \rho_i > 0.$$

The second constraint guarantees that every file has a fraction of its content ρ_i cached at the helpers. It can be shown that the objective function above is equivalent to minimizing

$$\sum_{i=1}^{F} \frac{\lambda_T \cdot p_i \cdot T}{\mu_p} \cdot log\left(\frac{\lambda_T \cdot p_i \cdot T + \frac{\mu_{h_i}}{\mu_p} \cdot \rho_i \cdot H}{\frac{\mu_{h_i}}{\mu_p} \cdot \rho_i \cdot H}\right);$$

and the optimum solution is obtained when $\rho_i \propto p_i$. Similarly, it can be easily shown that the optimum upload bandwidth allocation at helpers for each file $i \in F$ is when $\mu_{h_i} \propto p_i$.

Similar to our observation in the proactive helpers scheme, we observe that despite the high cumulative bandwidth that served peers contribute to the P2P system when uploading a popular file as compared to the case when uploading a less popular file, helpers should always favour the popular files when it comes to selecting files to store in their limited size cache, or when allocating upload bandwidth for transferring the cached files to interested users in the network.

4.5 Energy Consumption in Content Sharing Using Proactive Helper Approach

In this section, we show that enabling a number of infrastructure nodes in the community network to participate in P2P content distribution results in significant reduction in power consumption in the network. We focus on the case of proactive helpers (mini-servers).

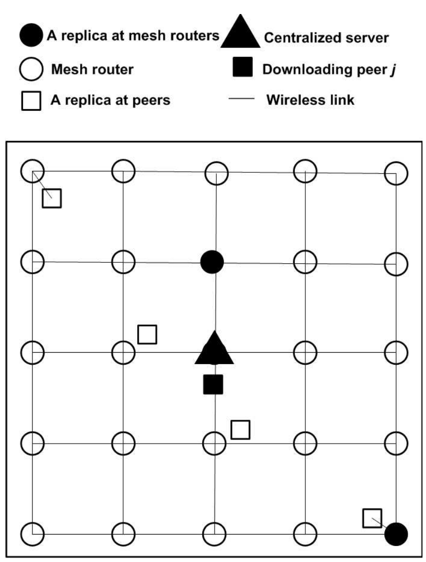


Figure 4.4: Network model

We assume, without loss of generality, a wireless community network consisting of wireless routers connected in the form of wireless mesh community network [7, 8]. We assume a large wireless mesh network consisting of many stationary mesh routers (M) uniformly deployed in a 2-dimensional squared area (grid-like topology) (Figure 4.4). Any two nodes that can communicate directly with each other are connected by an edge in the network graph. Within the mesh network, data is communicated over wireless links. Nodes may communicate directly over a wireless link or over multiple hops with intermediate nodes forwarding data. Mesh clients (peers) are usually with much less upload bandwidth compared to the mesh routers. We assume that the IP routing algorithm employs a minimum hop routing metric such as AODV and DSR.

Although the assumptions about community network being a WMCN with grid-like topology may be restrictive, we use these assumptions in our analysis for the sake of simplifying the problem formulation, analytically analyzing the model, and demonstrating the effectiveness of our approach in reducing the energy consumption. However, the results that we derive can be extended to any network topology no matter wired or wireless. However, the problem is more difficult in wired networks due to the complexity of the wired network topology.

Although energy is not a constraint in WMCNs as in the case of MANETs and sensor networks, the consumption of energy in communication systems is becoming a fundamental issue. In fact, it has been shown that the information and communication technology sector is responsible for 2 to 2.5% of the GHG annual emission [136, 137], and wireless access networks are largely responsible for the increase in energy consumption [138, 139]. It follows that being able to minimize the base station energy consumption in a wireless network represents an important green networking objective.

In this section, we compare the power consumption in two content distribution schemes in WMCN. The first scheme is content distribution using a centralized server (server-client scheme) that is physically located in the center of the network with upload bandwidth of μ_s. The second scheme is the P2P content distribution with proactive helpers, that we described in section 4.4.1.

Let the transmission power that is required at each mesh router in order to relay a data packet of content (say content i) to the next router towards the destination be P_{MR} (assuming all content packets have the same size). We consider the following content retrieval scheme in the P2P-with-helpers system. A peer j which downloads a content i retrieves segments of the content i from the nearest helper that caches a replica of i, and from all served peers in the network.

Let us suppose that content i has only one replica stored at the helpers at an instant of time t, and this replica is placed as shown in Figure 4.5(a). Let the maximum power that is consumed in the network as a result of accessing a replica of object i at that helper from any peer in the network at instant of time t be $P_h(t)$. We can see that P_h is the power that is consumed when peer x accesses the replica of content i at that helper (Figure 4.5(a)). We can further see that $P_h(t) = \sqrt{M} \cdot P_{MR}$, where M is the total number of mesh routers in the WMCN. Suppose that object i has 4 replicas stored at the helpers at an instant of time t, and those replicas are placed as shown in Figure 4.5(b). Then, the maximum power consumed in the network in order to access a replica of object i at the nearest helper becomes $P_h(t) = \sqrt{M} \cdot P_{MR}(\frac{1}{2})$. Similarly, we can compute the power consumption when the number of replicas stored at helpers is 16 (Figure 4.5(c)). The computations of power consumption when accessing a replica of object i at helpers for varied number of replicas stored at helpers are shown in Table 4.1. We note here that the computation of power consumption is based upon our assumption about the placement of replicas at the helpers.

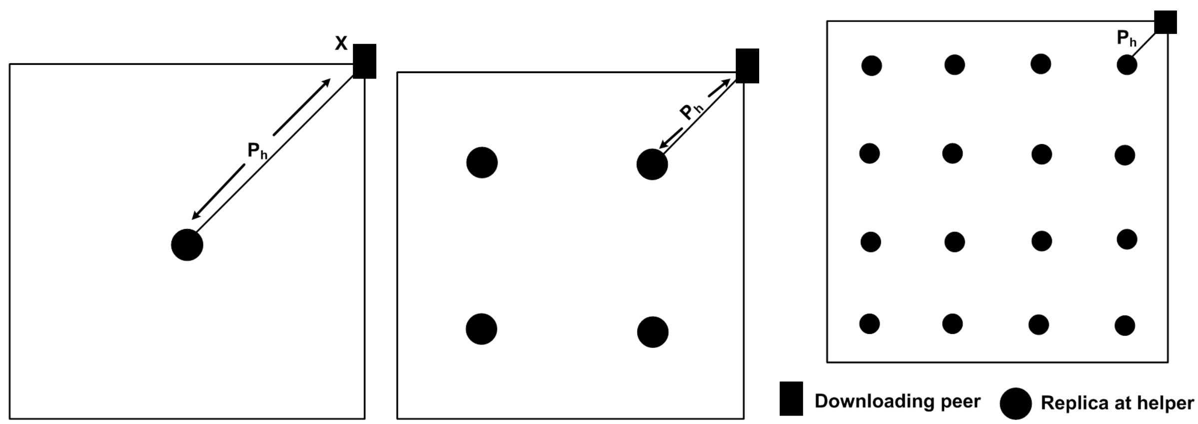
Table 4.1: Summary of network power consumption for varied number of replicas at the helpers.

Number of replicas	1	4	16	64
Maximum power cost	$\sqrt{M} \cdot P_{MR}$	$\sqrt{M} \cdot \frac{P_{MR}}{2}$	$\sqrt{M} \cdot \frac{P_{MR}}{4}$	$\sqrt{M} \cdot \frac{P_{MR}}{8}$

Recall that $Y_{h_i}(t)$ is the number of replicas of object i that are stored at the helpers at any instant of time t. Hence, we can see from the above table that for a WMCN deployed in a grid-like topology, when Y_{h_i} is increased by a factor of 4, P_h is reduced by half. Hence, we can approximately write

$$P_h(t) \approx \sqrt{M} \cdot \frac{P_{MR}}{\sqrt{Y_{h_i}(t)}}.$$

This approximation is feasible especially when we assume a large-scale WMCN with many helpers.



(a) 1 replica (b) 4 replicas (c) 16 replicas

Figure 4.5: Average power consumption

The total power consumption of object i with size L (packets) using the P2P-with-helpers system ($P_{Helpers-System_i}$) can be written as $P_{Helpers-System_i} = P_{Peers-and-Helpers_i} + P_{Peers-to-Helpers_i}$, where $P_{Peers-and-Helpers_i}$ accounts for the power consumed when uploading content i from the served peers and helpers to interested peers during time $0 \leq t \leq t_2$, while $P_{Peers-to-Helpers_i}$ accounts for the power that is consumed when uploading content i from served peers to helpers during time $0 \leq t \leq t_1$. Hence, we can write

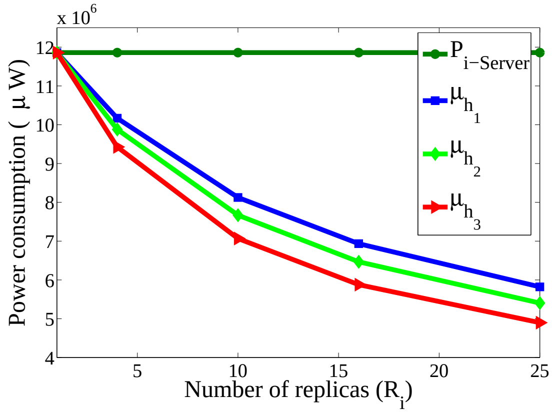
$$\begin{aligned} P_{Helpers-System_i} &= L \cdot \int_0^{t_2} \Big(\mu_p \cdot \eta_{P2H} \cdot S_i(t) \cdot E[P_p(t)] + \mu_{h_i} \cdot Y_{h_i}(t) \cdot P_h(t) \Big) dt \\ &+ L \cdot \int_0^{t_1} \epsilon \cdot \mu_p \cdot S_i(t) \cdot E[P_p(t)] dt, \end{aligned}$$

where $\eta_{P2H} = 1 - \epsilon$ when $t \leq t_1$, and 0 when $t > t_1$, and $E[P_p(t)]$ is the expected value of the maximum power that is consumed in the network as a result of accessing object i from the served peers. If we assume that the requests for content is generated at peers uniformly at random, it is easy to see that $E[P_p(t)] = \sqrt{M} \cdot P_{MR}$.

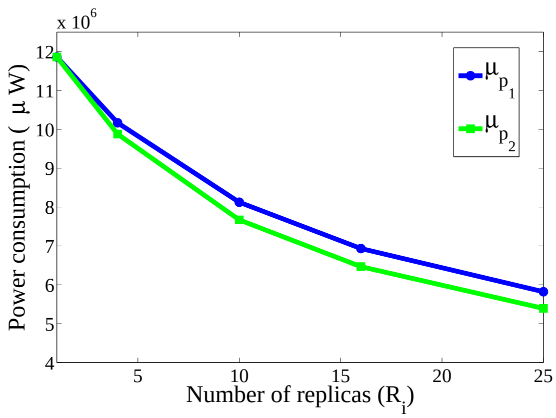
To compute the saving in network power consumption that is realized when using the P2P-with-helpers system as compared to the case of centralized server, we need to compute the total power consumption when using the centralized server (P_{Server_i}). P_{Server_i} can be computed as $P_{Server_i} = L \cdot N_T \cdot P_s$, where P_s is the power that is consumed in the network as a result of accessing object i of size L packets at the centralized server that is located in the center of the network, and $P_s = \sqrt{M} \cdot P_{MR}$. Thus, P_{Server_i} can be written as $P_{Server_i} = L \cdot N_T \cdot \sqrt{M} \cdot P_{MR}$, Hence, the saving in network power consumption using the P2P-with-helpers schemes (η_{save}) can be computed as $\eta_{save}\% = \frac{P_{Server_i} - P_{Helpers-System_i}}{P_{Server_i}}$.

We numerically evaluated the power consumption in the proactive helpers scheme using the derived fluid model for the cases when $\mu_p = 0.001$, $N = 10$, $p_i = 0.00075$, $M = 1000$, $P_{MR} = 1$ (μ W/packet), size of the file $L = 1000$ (packet), $N_T = 750$, and μ_{h_i} takes the values $\mu_{h_1} = 0.03$, $\mu_{h_2} = 0.04$, and $\mu_{h_3} = 0.06$. (Figure 4.6(a)). We observe that the saving in power increases with increasing the upload rate at the helpers. This is because increasing upload rate at the helpers implies that a higher fraction of content i is accessed from helpers nearby. Another observation is that the power saving increases with increasing number of replicas at helpers. This is because having more replicas at helpers reduce the average number of wireless hops in download paths between replicas and downloading peers.

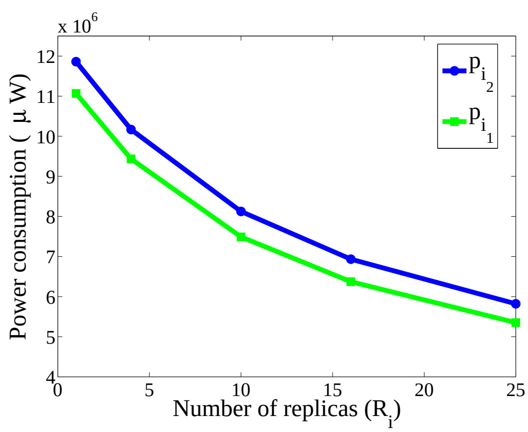
We also computed the power consumption in P2P-with-helpers system when the upload rate at the served peers takes the values $\mu_{p_1} = 0.001$ and $\mu_{p_2} = 0.00075$, while μ_{h_i} was kept constant at 0.03. We further computed the power consumption in the P2P-with-helpers system when $\mu_p = 0.001$, $\mu_{h_i} = 0.03$, and content popularity take the values $p_1 = 0.0007$ and $p_2 = 0.00075$. We observe that the saving in power consumption is less when the upload rate of peers is higher (Figure 4.6(b)). We also observe that the saving



(a) Power consumption for different μ_h and μ_s.



(b) Power consumption for different μ_p.



(c) Power consumption for different p_i.

Figure 4.6: Fluid model results.

reduces when the content is more popular (Figure. 4.6(c)). This is because the fraction of content i that is accessed from served peers that are physically far from downloading peers is higher. This results suggest some enhancement to the content retrieval scheme that we used in our problem formulation in this section. Specifically, the content retrieval scheme must enable interested peers to download the content from peers that are geographically nearby. In fact, we have considered the content retrieval problem in Chapter 2, and proposed efficient schemes for both selecting ideal content providers and establishing parallel download paths.

4.6 Simulation Results

We simulated the proposed P2P-with-helpers schemes as stochastic systems using Matlab. The purpose of the simulations was to confirm that the fluid approximation that we used does not impact the validity of our analytical results. To achieve this goal, we simulated a network that consists of $M = 10000$ routers. The total population of users (peers) in the networks was 100,000, each user was socially connected to ($N = 10$) other users in any time unit, and the average number of hops between peers was $HOP = 5$. We considered a single file generated at a random user with popularity of $p_i = 0.001$. The size of the file followed a normal distribution with mean $10Mbyte$. The upload rates at peers and helpers follow exponential distribution with means $\mu_p = 30kbps = 0.003$ (file/sec) and $\mu_{h_i} = 150kbps = 0.015$ (file/sec), respectively. We carried out the simulation many times for each statistical data and computed the average values with confidence intervals (95% confidence intervals).

In the first set of simulations, we computed the average file download delay for all served users (T_i) using the proactive helpers (mini-servers) scheme described in Section 4.4.1 for varied number of replicas at the helpers (R_i). We compared the simulation results with the analytical results obtained using our derived model (Figure 4.7). Al-

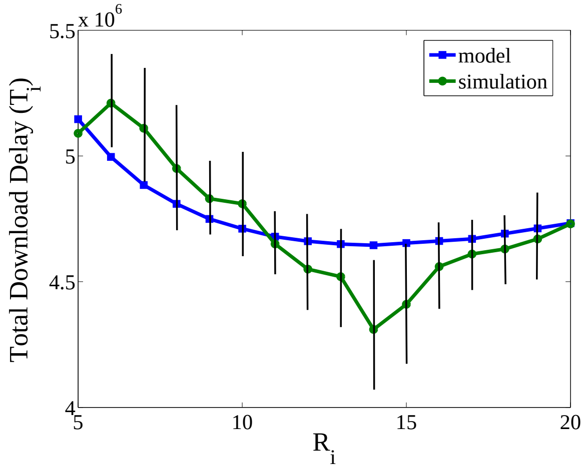


Figure 4.7: Total download delay for varied number of replicas at helpers in the proactive helpers case.

though the simulation results and the analytical results show similar trends, there is an obvious gap between them when the number of replicas is close to R_i^*. A possible reason for this observation is that the system is very sensitive when operating at this point as described in Section 4.4.3, and any small change around this operating point, which results from simulating a stochastic system, has a large impact on the system performance. In general, we can say that the proposed fluid model provides a good approximation of the performance of the real content distribution system; and the value of R_i^* that was obtained using simulation is close to the one that was obtained using the fluid model.

In the second set of simulations, we computed the average file download delay for all served users (T_i) using the demand driven helpers (relay nodes) scheme described in Section 4.4.4 for varied fraction of cached content at helpers (ρ). We compared the simulation results with the analytical results obtained using our derived model (Figure 4.8). Again, the simulation results show that the derived model provides a good approximation of the performance of the real content distribution system; and the value of ρ^* that was obtained using simulation is close to the one that was obtained using the fluid model. However, we observe that the gap between the analytical results and simulation results increases with increasing ρ. The reason for this observation is that the value of ρ used in the stochastic

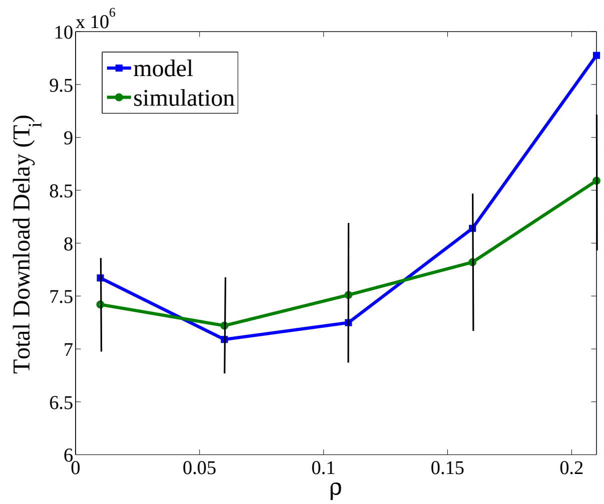


Figure 4.8: Total download delay for varied ρ at helpers in the demand driven case.

system was drawn from a normal distribution with variance $\sigma = 1/\mu$, where μ is the value of ρ used in the fluid model. Hence, as μ increases, σ increases proportionally causing high fluctuations around the average value of T_i. Therefore, we expect the simulation results to taper off and converge to the analytical results when the number of simulation runs is very high.

In the next set of simulations, we compared the two proposed methods: the proactive and the demand driven helpers. Specifically, we computed the download time for varied number of replicas at helpers (Figure 4.9), where we set the number of replicas (R_i) in the proactive helper case to be equal to the number of participating routers (H) in the demand driven case. We also set $\rho = \rho^*$ in the case of demand driven helpers. The simulation results show that the download delay in the reactive case is monotonically decreasing function with respect to number of participating helpers H; while the download delay in the case of proactive helpers is convex function with respect to R_i. This is because helpers in the demand driven case do not consume the P2P system resources. We also observe that the delay in the case of proactive helpers is generally less than that of the demand driven case. This is because participating helpers in the case of proactive helpers directly

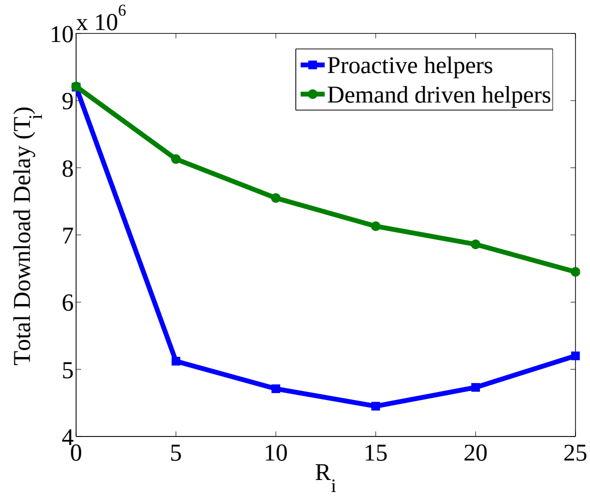


Figure 4.9: Comparison between proactive and demand driven cases.

download content from peers and do not wait to overhear (relay) transmitted content as in the case of demand driven. Hence, better utilization of helper uploads bandwidth.

In the next set of simulations, we simulated the P2P with helper scheme as a stochastic system using Matlab to evaluate the accuracy of our fluid-flow model derived in Section 4.5 to compute the average energy consumption in the network. We simulated a WMCN that consists of $M = 1,000$ mesh routers deployed in a grid topology. Total number of interested users who eventually download the file was $N_T = 750$, and each user was connected to ($N = 10$) other users (friend) in any time unit. We considered a single file i that is generated at a random user, and the popularity of file i was $p_i = 0.0075$. The generator of file i was selected uniformly at random, and requests for file i was generated at peers uniformly at random. The size of the file is 10 Mbyte; while the size of a packet follows a normal distribution with mean 1024 kbyte. Number of replicas of file i at helpers varied between 4 to 25. The upload rate of peers and helpers follow exponential distribution with means $\mu_p = 10kbps = 0.001$ (file/sec) and $\mu_{h_i} = 300kbps = 0.03$ (file/sec), respectively. The power at mesh routers was set fixed at $P_{MR} = 1$ ($\mu W/packet$). We carried out the simulation many times for each statistical data and computed the average total

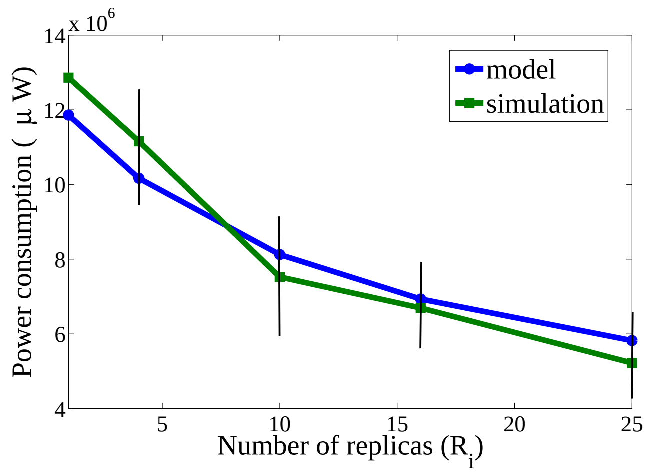


Figure 4.10: Average total power consumption.

power consumption in the network using the P2P-with-helpers scheme for varied number of replicas at helpers with confidence intervals (95% confidence intervals). We compared the simulation results with the numerical results that was obtained using our derived fluid model described in section 4.5. The results show that the proposed fluid model provides a good approximation of the real system, and the fluid approximation does not impact the validity of our analytical results (Figure 4.10).

4.7 Summary

This chapter evaluated the performance of P2P-with-helpers system for content sharing and distribution in a community network. We have investigated the role that infrastructure nodes in a community network can play to help in P2P content sharing and distribution in terms of content download times and energy consumption. The derived results increased our understanding of how helper provisioning affects the performance of content distribution in the community network. We have considered two cases for content sharing and distribution using helpers; namely, proactive helpers and demand driven helper. For each case, we have proposed design strategies for the HelperDesign

Manager in the P2P-with-helpers system that allocate optimal resources at helpers based on prediction of future evolution of content demand in the community network. We have further considered the case of simultaneous multiple content distributions, and computed the optimum number of replicas as well as the optimum upload bandwidth that helpers must allocate for each cached content.

In this chapter, we assumed that all peers and all helpers have same upload bandwidth. A possible future research direction is to generalize the problem by relaxing this assumption and considering a more sophisticated content distribution settings, where peers and helpers have different capacities. Measurements and experimental studies are also required to obtain more persuasive results, better analyze the demand evolution in community networks, validate our derived analytical models, and show the performance effectiveness and trends of our proposed schemes when used in real systems.

So far, we have studied the performance of content sharing and distribution of stored files that can only be used after downloading the entire file (e.g., P2P file sharing-like service). Another interesting application in community networks is media streaming. In the next chapter, we evaluate the performance of P2P-with-helpers system when used for the case of media streaming in a community network.

Chapter 5

Modelling, Performance Analysis, and Design Strategies for P2P-with-helpers: The Case of Media Streaming[4]

5.1 Introduction

Media streaming applications have recently attracted large number of users in the Internet. In 2006, the number of video streams served increased 38.8% to 24.92 billion as compared to 2005 [50, 140]. This large demand creates a burden on existing infrastructure such as centralized data centres and content distribution networks to sustain the QoS guarantees. The huge capacity resources and the large consumption of power and networking bandwidth are the main aspects that hinder the evolution of media streaming applications in the Internet [141]. This problem becomes more critical with the increasing demand for higher bit rates required for growing number of high-definition video quality

[4]This chapter is based in part on the following paper.

1. A. Alasaad, S. Gopalakrishnan, and V. C.M. Leung, "A Hybrid Approach for Cost-effective Media Streaming Based on Prediction of Demand in Community Networks," accepted for publication in the journal of *Telecommunication systems*, special issue on innovations in emerging multimedia communication systems, Springer New York.

desired by consumers. In this chapter, we explore new approaches in collaborative media streaming that mitigate the cost of media streaming.

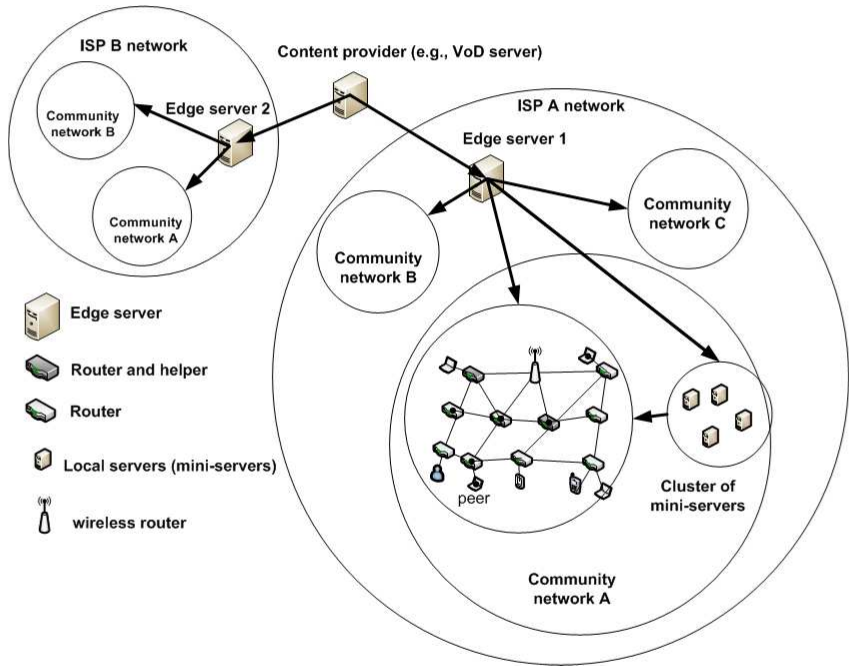


Figure 5.1: Hybrid content streaming architecture

Media content providers rely on Content Delivery Networks (CDNs) for content distribution in the Internet. The CDN replicates media files into caches of specialized edge-servers residing close to end-users. A desired media file is delivered to end-users directly from the edge-servers. The edge-servers are typically located inside the networks of the access connection providers such as Internet Service Providers (ISPs) (Figure 5.1). A Client interested in a media file available at the server of the content provider is connected to the edge-server that is physically closest to the client in order to reduce the delay and maintain QoS guarantees.

Media streaming traffic is delay sensitive. Therefore, the CDN must equip the edge-servers with computing resources and networking bandwidth such that the data rate of media streaming guarantees smooth video playback for all users at all times. Since it is possible to anticipate the size of usage peaks for streaming capacity in a weekly, monthly, and yearly basis, the CDN equips the edge-servers with networking bandwidth and computing capacities that target the expected usage peak (flash-crowd events). However, this causes economic efficiency problems. To ensure that servers can cope with peak demand, most servers in a typical data-center operate at only about 30% of their capacity [140, 143]. Hence, a huge amount of capacity at the servers will be idle most of the time, which is highly wasteful and inefficient.

Another scheme for media streaming is through the use of Peer-to-Peer (P2P) technology, by exploiting the upload capacity of peers who are interested in the same media content (e.g., PPStream and CoolStreaming). Although this approach scales well with number of interested users and mitigates load imbalance in the network, peers are volatile in nature making the QoS guarantees hard to sustain. Compared with the centralized approach, peers usually offer much lower out-bound data rate (upload bandwidth) and; hence, a large number of peers must jointly stream the media content to an interested user. Therefore, most existing P2P media streaming protocols also rely on a centralized platform to make up the difference in streaming rate when peers cannot by themselves distribute the media content [129–133, 144]. As a result, the media content provider is obliged to maintain over-provisioned server capacities at its data-center that target worst case scenarios, which is costly and inefficient.

Since ISPs typically attract new clients by charging low prices for large data usage (most of ISPs offer unlimited data plans to their clients), the large demand for media streaming by users in an ISP network does not generate additional revenue for the ISP, while consume enormous amount of bandwidth and power in the ISP network. Moreover, the community network is charged by the ISP for the Internet bandwidth delivered for its users from the edge-server. Hence, new designs for content distribution that mitigate the cost of media streaming is of interest to both ISPs, CDNs, and community network operators.

This motivates us to explore new approaches in collaborative media streaming in community networks (e.g., college and office campuses) residing in a larger ISP network (Figure 5.1). In this chapter, we propose hybrid approaches for cost-effective media streaming based on prediction of streaming demand in community networks. In particular, we investigate the role that infrastructure nodes in community networks can play to reduce the cost of media streaming. The proposed approaches are aimed at exploiting the redundancy and abundantly available network "micro-resources" in a community network to create an aggregate virtual "macro-resource". Specifically, the ISP can leverage its control over a large amount of under-utilized micro-resources in a community network to replicate into those micro-resources media files that are desired in that community. By allocating those idle micro-resources to manage the media streaming demand in a community network on a per-need basis, large size of the traffic load in the community network would be absorbed by the aggregated macro-resource, and the peak load at the edge-servers would be filtered out. Hence, much of the over-provisioning at the centralized architecture (edge-servers) can be cut down. Moreover, the cost of media streaming can be contained within the community network. Hence, the bandwidth and energy consumption in the ISP network, that results from delivering media files from the edge-server to interested users in the community network over long download paths, can be significantly reduced. Also, the cost of the Internet bandwidth on the community network operator is reduced.

We consider two instances of this philosophy. In the first instance, a number of local "mini-servers" in a community network are selected to participate in media streaming and assist the edge-server in sustaining the QoS guarantees (Section 5.5). We refer to those mini-servers as *helpers*. Those mini-servers already deployed in the community network in large numbers for different purposes such as traffic monitoring, authentication, etc. Examples of potential local mini-servers in a college campus community network that are capable of playing the role of helpers include servers used in laboratories to run simulations or perform experiments, and servers assigned for scientists to save personal information or storing scientific data. Those mini-servers are typically inexpensive servers, equipped with commodity hardware, and idle most of the time. Although an individual mini-server is much less powerful compared to the centralized server (edge-server), the aggregate capacity of cluster of mini-servers has the potential to absorb large size of traffic load in the community network.

Mini-servers, however, do not join the streaming session with preloaded media content. They need to consume the system resources, in order to download the media file, before being able help by providing (uploading) the cached media content to interested users in the community network. We can see that pushing too many replicas of a media content into caches of mini-servers without proper prediction for future streaming demand of that media content in the community network may harm the media streaming system (i.e, increase the traffic load at the edge-server). This is because mini-servers may consume (download) higher amount of traffic from the system than the amount of traffic they contribute (return back) the system. Hence, by optimizing on both the number of replicas pushed into caches of mini-servers for every media content desired in the community network and the upload bandwidth that mini-servers allocate for streaming every stored media content, mini-servers can minimize the cost of media streaming.

In another instance, we consider the setting, wherein a number of relay nodes (e.g., mesh routers) in a community network are over-provision with additional storage and P2P-aware devices (packet sniffers) that are programmed to enable those nodes to cache and store media content that they relay in the community network (Section 5.6). We refer to those infrastructure nodes as *helpers*. Examples of relay nodes that can play the role of helpers include routers, gateways, and wireless access points. When a helper caches a pre-determined fraction of a desired media content (ρ), it uploads the cached content to interested users. Hence, helpers in this setting need to optimize on what and how much content to cache in order to minimize the overall cost of media streaming.

In a community network, a group of users who share similar interests in media content are often socially connected (e.g., friends in a social network). An example of socially connected users in a college campus community network is classmates who are interested in streaming a video lecture or a video technical seminar. Those classmates are often socially connected over Facebook, or email groups. Thus, the interest in a media content (demand) grows quickly in the community network as interested users contact others (friends) and make them interested, but tapers off when a certain level of interest is reached. We have carefully characterized this viral evolution of content interest in community networks (Section 5.3). In this case, we observe that life-time of interest in a content in a community network is short, and number of potential users who may get interested in the content is limited. Therefore, schemes that target enhancing the performance of media streaming in the steady state such as [47, 48] are not feasible in community networks. In this chapter, we show that by using our demand-awareness model (i.e., future prediction of the streaming demand), helpers in a community network can be optimally utilized.

We propose design strategies for the HelperDesign Manager in the P2P-with-helpers system that allocate optimal resources at helpers to match the predicted future evolution of streaming demand in the community network (Figure 1.2). Our main contribution in this chapter is to provide qualitative and quantitative performance analysis that increases our understanding of how helpers in community networks can mitigate the cost of media streaming on both the CDN, ISPs, and community network operators.

We show analytically and using simulations that by optimally utilizing helpers in every community network residing in a larger ISP network (Figure 5.1), the CDN can alleviate the edge-server capacities. This reduce the cost in terms of both purchase and maintenance. Moreover, ISPs can reduce the bandwidth and power consumption in their networks by pushing media content into the caches of helpers in community networks that are closer to the interested users. Also, the community network operator cab reduce cost of the Internet bandwidth by localizing the streaming traffic with the community network.

5.2 Related Work

Video streaming applications have recently attracted large number of users over the Internet [50, 140, 141]. It is projected that Internet video including live streaming and video on demand will account for over 60% of all Internet traffic by 2013 [141]. In live media streaming, the media (video) content is available only at one particular time instant and interested users have synchronous playback times. On the other hand, media streaming of stored content (e.g., Video-on-Demand-like service (VoD)) allows users to watch any point of the video at any time. Media content in VoD-like service is often preloaded on the server and, hence, VoD offers more flexibility and convenience to users. In VoD service, although a large number of users may be watching the same video, they are asynchronous to each other and different users are watching different portions of the same video at any given moment. Moreover, some users (peers) may stay in the network after watching the content to serve other interested peers.

In P2P media streaming architecture, multiple peers share their resources among each other. The capacity of the overall system is proportional to the number of peers connected at any moment of time. Examples of P2P media streaming systems are PPStream and PPLive. A hybrid architecture for media streaming that involves CDN and peers are widely considered [50, 132]. Many studies propose schemes to enhance the performance of P2P content distribution such as P2P with server assistance [129–133]. NetTube is a novel peer-to-peer assisted delivering framework that explores the user interest correlation for short video sharing [142]. In such schemes, a central server is used to boost the performance of the P2P content distribution and guarantee service.

There are various works that introduce the notion of P2P content distribution using helpers [44, 47, 48]. The proposed systems aim at exploiting the upload bandwidth of idle peers in the network to enhance the performance. Wang *et al.* studied the performance of BitTorrent-like system and investigate the role of helpers [44]; while Zhang *et al.* proposed a P2P video-on-demand system using helpers [47]. Prior studies considered the case where helpers are regular peers that are not interested in the media content but their idle upload bandwidth is exploited to enhance the performance of content distribution. The lack of proper incentives for the helping peers, however, render these schemes impractical.

The prediction of CPU utilization and user access demand for web-based applications has been extensively studied in the literature. A prediction method has been proposed with respect to upcoming CPU utilization pattern demands based on neural networking and linear regression that is of interest in e-commerce applications [145]. Y. Lee *et al.* proposed a prediction method based on Radial Basis Function (RBF) networks to predict the user access demand request for web type of services in web-based applications [146].

Although the demand prediction for CPU utilization and web applications has been studied for a relatively long period of time, the prediction of demand for media streaming in the Internet has gained popularity more recently [147]-[150]. The access behaviour of users in Peer-to-Peer (P2P) streaming with time-series analysis techniques using non-stationary time-series models was predicted in [147]. The method of time-series prediction based on wavelet analysis was studied in [148]. In [149], principal component analysis is employed by the authors to extract the access pattern of streaming users. Although most of the above studies predict the average streaming capacity demands, few papers have also studied the volatility of the capacity demand, i.e., the demand variance at any future point in time, which yields more accurate risk factors [150].

What distinguishes the evolution of interest in a media content in a community networks (e.g., college or office campus) from the Internet is that users in a community network, who share similar interest, are often socially connected (e.g., friends in a social network). Thus, the demand grows quickly in the community network as interested users contact others (friends) and make them interested, but tapers off when a certain cumulative level of interest is reached. For example, consider a student, in a class of 100 students, spreads the knowledge about existence of a video file (e.g., technical seminar available on the campus server) to his classmates. If the popularity of this file is 0.2, the evolution of the demand increases quickly with time but tapers off when 20 students get interested in the file. When all the 20 students finish watching the video file, the life-time of that file in the community network expires (i.e., the popularity of that file diminished).

Large number of prior studies used fluid-flow models to analyze the performance of P2P data sharing and media streaming in the Internet. Most of prior works assume that requests (demand) for a P2P content either follow Poisson arrivals with constant rate (i.e., the common assumption in most prior works is that the evolution of interest in a P2P content over time is fixed), or occurs simultaneously (i.e., flash crowd) [44, 45]. However, these assumptions have been shown to be unrealistic in an eight-month measurement study [128]. Prior works also assume that there is infinite number of potential users who may get interested in any content, and thus, evaluate the performance of the media streaming system only at the steady state [47]. This assumption is acceptable in the Internet, where we can assume infinite pool of users. However, our analytical model that we derive in this chapter shows that this assumption is not feasible in the case of viral evolution of demand for media streaming in community networks.

Z. Wang *et al.* observed that the social videos, unlike regular videos, do not propagate among users randomly. Instead, they propagate along the social-network topology according to several rules determined by the social propagation. Exploiting the new design space enabled by this observation, they developed a propagation-based social-aware replication system PSAR, to effectively distribute social videos with superb QoE [151].

Our work differs from that of Zhang *et al.* [47] as follows. Zhang *et al.* proposed system for P2P video-on-demand on the Internet using helpers, and evaluated the optimal steady-state system's parameters that maximize the utilization of helpers' resources. As we have discussed, schemes that target enhancing the performance of media streaming in the steady-state are not practical in community networks. In the contrary, we design a hybrid approach for media streaming that benefit from the awareness of both the evolution of interest in media content, and the expected level of cumulative interest in the community network. The model that we derive for predicting future streaming demand of media content in the community network allows us to design a delicate hybrid media streaming system that involves the edge-server, peers, and helpers. We show how awareness of the demand can be used to optimally utilize helpers capacities. In contrast with the work in [47], helpers in our scheme are infrastructure nodes that are owned by the ISP. This eliminates the need of developing incentives for helpers, whereas incentives are crucial in the case where helpers are regular peers as in [47]. While most prior works aim at utilizing resources of helpers in order to reduce the media content download time, our schemes aim at utilizing resources of helpers in order to mitigate the overall cost of media streaming on the ISPs, CDNs, and community network operators.

Laoutaris *et al.* proposed (ECHOS): a media content distribution approach through what they call “nano” data-centres, which are essentially boxes deployed at the edge of the network (e.g., home gateways, set-top-boxes) that cooperate in a peer-to-peer manner [152]. The system shows very promising performance. Our work use similar “micro-resource” approach. However, we optimize the use micro-resources (helpers) to reduce the cost of media streaming. We note here that our proposed approach is designed for media streaming of stored content (e.g., VoD-like service). We do not consider the problem of streaming live content; that remains one of our future directions.

5.3 The Demand Model

In this section, we characterize the viral evolution of interest in a media content between users of a community network. In particular, we consider the setting, wherein a user in a community network (e.g., a student on a college campus) gets interested in media content i that is stored at the server of a media (VoD) content provider, and wants to spread the interest to his friends in the community network. The interested user spreads the knowledge about the content existence to his friends through social networks (e.g., Facebook, Twitter). We assume that the average number of friends to whom a user is connected in a social network (node's degree) in any time unit is N.

As we have shown in the previous chapter, we can model the evolution of interest (demand) in media content i as

$$\frac{dD_i(t)}{dt} = D_i(t)[p_i(N - \gamma(t) \cdot N)],$$

where $D_i(t)$ is the total number of interested users in content i at any instant of time t (cumulative demand), p_i is the popularity of content i between users in the community network, and $\gamma(t) := \frac{D_i(t)}{N_T}$, where N_T is the total number of potential users in the community network who will ultimately become interested in media content i. The above formula is a second order Bernoulli differential equation and can be solved as

$$D_i(t) = \frac{N_T \times D(0)}{D(0) + (N_T - D(0))e^{-p_i \times N \times t}}, \tag{5.1}$$

where $D(0)$ is the number of interested users at time instant $t = 0$. We note that $D(t)$ has an S-shape (Figure 5.2). It shows that the number of interested users increases quickly when the content becomes available in the community network, and then gradually decreases and tapers off once the level of interest reaches N_T. This is similar to the demand function that was obtained using word-of-mouth spread of information by interested users (Bass model) [135]. Similar demand evolution was also observed when measuring user interest in a video file on YouTube server [153], and when measuring user interest in popular video hosted on a university infrastructure (CoralCDN) [154].

As we can see from Eq. (5.1), the ISP/CDN can predict the evolution of demand for content i in a community network if information, a priori, about both the expected popularity of content i (p_i) and the maximum level of interest (N_T) is known. In the next

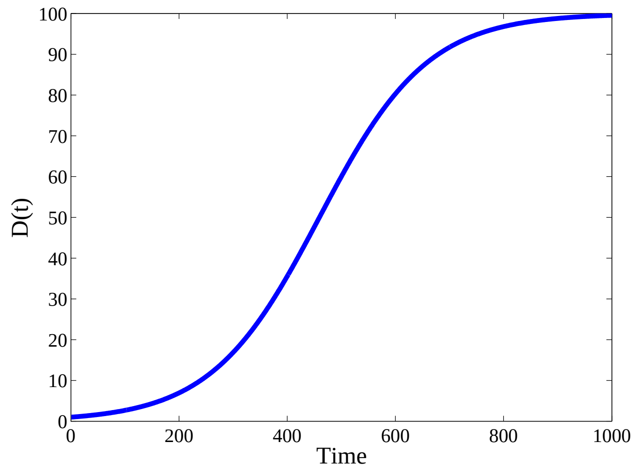


Figure 5.2: The evolution of the demand for file i when $p_i = 0.001$, $N = 10$, $T = 100000$, $D(0) = 1$.

sections (Sections 5.5 and 5.6), we assume that the ISP can accurately predict p_i from historical observations of interest in similar media files previously desired by users of the community network, while N_T can be predicted by knowing the total number of users in the community (i.e., total population of the community network). We use this awareness of the future demand for content i to determine optimum system parameters for media streaming.

5.4 Network Model

We consider a general architecture for media streaming that involves a media content provider (VoD provider), edge-servers in ISP networks, and peers in community networks. In particular, we consider a hybrid approach for media streaming in a community network located within a larger ISP network (Figure 5.1). A user in a community network may get interested in a media file stored at the edge-server located in the ISP network. Users in the community network, who are interested in the same file i exchange segments of the file with each other in a P2P manner. We call those users *peers*. Every peer attempts to retrieve segments of the desired file from other peers and helpers in the community network who possess the required segments. The edge-server needs to make up the difference in streaming rate to guarantee smooth video playback only when the dissemination of the content between the peers themselves and between helpers and peers cannot achieve the required QoS guarantees. We denote the required streaming rate (QoS guaranteed rate) by μ_Q.

We assume that the system employs a file tracker that keeps records of peers and helpers in the community network that possess a file (say file i) in any time. We assume that a helper/peer can serve other peers only when it possesses the complete media file i and after registering itself with the file tracker as provider of content i. We refer to peers who finish downloading the file i and become able to upload the cached file to other peers by *served peers* (S_i). We note here that with the ability of a peer to upload data to others while downloading the file, this assumption may be restrictive. However, this assumption allows us to obtain simple analytical results that increase our understanding of how capacity of helpers can be utilized to provide the highest gain to the media streaming system. We do not consider the problem of constructing the streaming overlay topology involving all peers and helpers; that beyond the scope of this chapter. In our analysis, we assume that peers can fully utilize their upload bandwidth when assisting other peers.

We denote the total amount of traffic that the edge-server uploads to users in the community network who are interested in file i by η_s. As we have discussed earlier, reducing η_s mitigates the traffic load at the edge-server and, consequently, reduces the cost required for equipping the edge-server with over-provisioned capacity. Moreover, by reducing η_s, the energy and bandwidth consumption in the ISP network is reduced and the cost of the Internet bandwidth on the community network operator is reduced. This is because reducing η_s implicitly implies that interested users in the ISP network retrieve desired media objects from nearby peers and helpers located within their community networks over shorter download paths. Hence, our goal is to minimize η_s. We achieve this goal by optimally utilizing the storage capacity and upload bandwidth at helpers in the community network, and by optimally replicating media objects at the helper caches.

For simplicity and convenience of discussion, we assume without loss of generality, that a peer typically behaves in the following manner. When a peer gets interested in file i, he/she starts downloading the file. After a pre-buffering period, the peer starts playback. When the peer is done downloading the media file, he/she stays in the network as seeder (file providing peer) and does not depart the system until all potentially interested users in the community network (N_T) finish downloading the file, even though the his/her playback may finish before that time. We assume that peers start playing the video from the beginning at a constant speed. Similar to other P2P-VoD algorithms like BiToS and popular centralised solutions like YouTube; we do not consider seeking or fast-forwarding.

Let us consider a simple demand function $D_i(t) = \lambda_i \times t$ as shown in Figure 5.3, where $\lambda_i = \lambda_T \times p_i$, and λ_T is the rate at which requests for media files are initiated at users in the community network, and p_i is the popularity of file i in the community network. Although this demand function differs from our demand function in Eq. (5.1) that we derived in Section 5.3. This simple demand evolution function allows us to obtain simple analytical results that provide useful insights about this complicated system. We assume that information, a priori, about popularity of file i (p_i) and the maximum number of users potentially interested in file i (i.e., N_T, $N_T = 85$ in Figure 5.3) is known to the ISP/CDN. Hence, our goal is to use this demand-awareness to design optimum parameters for this hybrid media streaming system.

We focus our attention on analyzing the performance of P2P-with-helpers system for the case of media streaming. We neglect the effect of wireless channels in the case where media streaming is taking place over a WMCN (e.g., variable link rates, traffic interference, congestion, shadowing).

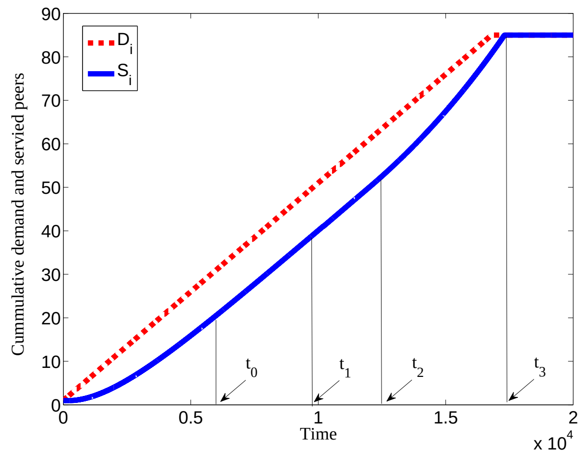


Figure 5.3: The evolution of the demand and served peers when $p_i = 0.05$, $\lambda_T = 0.1$, $R_i = 3$; $\mu_p = 0.0001$; $\mu_h = 0.0003$, and $\mu_Q = 0.0005$.

5.5 Media Streaming Approach Using Mini-servers in the Community Network as Helpers

We assume that the total number of mini-servers (helpers) in a community network that can be use for media streaming is H. In the early stage of streaming of media content i in the community network (i.e., when a media file available at the edge-server is requested by a user in the community network for the first time), the number of content providing peers (S_i) (i.e., number of peers who possess file i) in the community network is low. Thus, the cumulative supply bandwidth of content providing peers cannot meet the cumulative QoS demand of interested peers in the community network. Therefore, the edge-server makes up for the difference in the required cumulative QoS streaming rate. The edge-server also uploads (replicates) the media file i to a number of available mini-servers (helpers) in the community network (R_i, where $R_i \leq H$) to exploit the idle upload bandwidth and storage capacity of those mini-servers. When a helper finishes downloading the entire file, it uploads the cached content to the interested peers in the community network. We note here that mini-servers may download only a fraction of the media file and still provide high performance gain to the system; that however, is an extension of our analysis.

Figure 5.3 explains the approach that we advocate. Define t_0 as the instant of time at which all R_i helpers finish downloading entire file i from the edge-server. Hence, $t_0 = \frac{R_i}{X}$, where X is the rate at which the edge-server uploads segments of the file to helpers. Define t_1 as the instant of time at which the cumulative supply bandwidth of content providing peers and helpers meet the required cumulative QoS streaming demand of interested peers, and no extra bandwidth is required from the edge-server. Equivalently, t_1 is the instant of time at which the edge-server stops uploading segments of file i to downloading peers in the community network. Hence, At t_1, we can write

$$\mu_p \cdot S_i(t) + \mu_{h_i} \times R_i = \mu_Q \cdot \Big(D_i(t) - S_i(t)\Big), \tag{5.2}$$

where μ_p is the normalized average upload bandwidth of peers (in file per unit time) (assuming all peers have same upload bandwidth), μ_{h_i} is the normalized allocated upload bandwidth for file i at helpers (in file per unit time).

Define t_2 as the instant of time at which the cumulative supply bandwidth of content providing peers alone meet the QoS streaming demand of interested users and no need for helpers. Equivalently, t_2 is the time at which helpers stop uploading segments of file i to the downloading peers. Hence, At t_2, we can write

$$\mu_p \cdot S_i(t) = \mu_Q \cdot \Big(D_i(t) - S_i(t)\Big). \tag{5.3}$$

We assume that during the time period when $0 \leq t \leq t_1$, every helper uploads traffic to interested peers at rate μ_{h_i}, while during time period when $t_1 < t \leq t_2$, helpers only make up the difference in the required QoS streaming rate.

Define t_3 as the instant of time at which all users interested in file i in the community network finish downloading the entire file. We can see that at t_3, $D_i(t) = S_i(t)$. Hence, we can define t_3 as the life span (life-time) of file i in the community network.

Table 5.1: Notations.

N_T	Potential number of users interested in media file i in the community network (i.e., the maximum level of cumulative interest in the file).
p_i	popularity of file i in the community network.
μ_p	Average upload bandwidth of peers (normalized and measured in file per unit time).
μ_{h_i}	Allocated upload bandwidth for file i at helpers (normalized and measured in file per unit time).
μ_Q	QoS streaming rate.
$D_i(t)$	Number of cumulative users interested in object i at time t.
$S_i(t)$	Number of cumulative served peers at time t (content providing peers of file i).
H	Total number of mini-servers (helpers) available in the community network.
R_i	Number of mini-servers that ISP selects to store replicas of file i ($R_i \leq H$).
R_i^*	Optimum number of replicas for file i at helpers.
$Y_{h_i}(t)$	Number of replicas of file i stored at helpers at time t.
X	Rate at which helpers download segments of file i from the edge-server.
B	Maximum number of files that every mini-server can store in the cache at any time.
F	Number of simultaneous media streaming files.
η_s	Total amount of traffic that the edge-server uploads to interested peers in a community network.
ρ_i	Fraction of file i stored at helpers.
ρ_i^*	Optimum fraction of file i that helpers must store to minimize the cost of file i streaming.

Notations used in our analysis are listed in Table 5.1. Let $\mu_{H_i}(t)$ be the cumulative supply bandwidth that helpers (cluster of helpers) upload to peers which are interested in file i during the time period $t_1 < t \leq t_2$, and can be computed as

$$\mu_{H_i}(t) = \mu_Q \cdot \Big(D_i(t) - S_i(t)\Big) - \mu_p \cdot S_i(t).$$

Let $\mu_{s_i}(t)$ be the rate at which the edge-server uploads data to the peers which are interested in file i during time period $0 \leq t \leq t_1$, and can be computed as

$$\mu_{s_i}(t) = \mu_Q \cdot \Big(D_i(t) - S_i(t)\Big) - \mu_p \cdot S_i(t) - \mu_{h_i} \cdot Y_{h_i}(t).$$

To illustrate the operations of the proposed approach, we can use fluid-flow approximation to write

$$\frac{\partial S_i(t)}{\partial t} = \begin{cases} \mu_p \cdot S_i(t) + \mu_{h_i} \cdot Y_{h_i}(t) + \mu_{s_i}(t), \text{ for } 0 \leq t \leq t_1. \\ \mu_p \cdot S_i(t) + \mu_{H_i}(t), \text{ for } t_1 < t \leq t_2. \\ \mu_p \cdot S_i(t), \text{ for } t > t_2. \end{cases}$$

where $Y_{h_i}(t)$ is the number of helpers which have a complete replica of file i stored in their caches at any instant of time t.

The above system of partial differential equations can be solved as follows.

$$S_i(t) = \begin{cases} \mu_Q \cdot \lambda_T \cdot p_i \left(\dfrac{t}{\mu_Q} - \dfrac{1}{\mu_Q^2} \right) + c_1 \cdot e^{-\mu_Q \cdot t}, \text{ for } 0 \leq t \leq t_1. \\ \mu_Q \cdot \lambda_T \cdot p_i \left(\dfrac{t}{\mu_Q} - \dfrac{1}{\mu_Q^2} \right), \text{ for } t_1 < t \leq t_2. \\ c_2 \cdot e^{\mu_p \cdot t}, \text{ for } t > t_2. \end{cases}$$

and

$$Y_{h_i}(t) = \begin{cases} X \cdot t, \text{ for } \quad 0 \le t \le t_0. \\ R_i, \quad \text{for} \quad t_0 < t \le t_2. \\ 0, \quad \text{for } \ t > t_2. \end{cases}$$

where $c_1 = S_i(0) + \frac{\lambda_T p_i}{\mu_Q}$, and $c_2 = S_i(t_2)$.

We can see that t_1 can be computed by solving Eq. (5.2) for t as

$$t_1 \approx \frac{1}{\mu_Q} + \frac{1}{\mu_p} - \frac{\mu_{h_i} \cdot R_i}{\mu_p \cdot \lambda_T \cdot p_i},$$

and t_2 can be computed by solving Eq. (5.3) for t as

$$t_2 = \frac{1}{\mu_Q} + \frac{1}{\mu_p}.$$

Recall that η_s is the total amount of traffic that the edge-server uploads to interested users in the community network during the life span of file i (i.e., during the time period $0 \le t \le t_3$). We can see that η_s = total traffic collected by all served peers at time t_1 − total traffic uploaded by peers and helpers in time period $0 \le t \le t_1$ + total traffic uploaded by the edge-server to the mini-servers (helpers). Hence, the amount of traffic uploaded by the edge-server (measured in file) can be computed as

$$\eta_s \quad = \quad S_i(t_1) - \int_0^{t_1} \mu_p \cdot S_i(t) dt - \int_0^{t_0} \mu_{h_i} \cdot Y_{h_i}(t) dt - \int_{t_0}^{t_1} \mu_{h_i} \cdot R_i dt + R_i.$$

Thus,

$$\begin{aligned} \eta_s \quad &= \quad S_i(t_1) - \mu_p \Bigg(\mu_Q \lambda_T p_i \bigg(\frac{t_1^2}{2\mu_Q} - \frac{t_1}{\mu_Q^2} \bigg) - \frac{c_1}{\mu_Q} (e^{-\mu_Q . t_1} - 1) \Bigg) - \frac{1}{2} \mu_{h_i} . X . t_0^2 - \mu_{h_i} . R_i (t_1 - t_0) \\ &+ \quad R_i. \end{aligned}$$

As we have discussed earlier, our goal is to minimize η_s. We plot η_s for varied number of replicas of file i stored at helpers (R_i) in Figure 5.6. We can see that $\eta_s(R_i)$ is indeed a convex function. We can, therefore, find the optimum R_i^* that yields the minimum η_s by solving $\frac{d\eta_s}{dR_i} = 0$ for R_i.

We can see that the optimum number of replicas at helpers can also be obtained as.

$$R^* = \left\lfloor X\left(t_1 - \frac{1}{\mu_{h_i}}\right)\right\rfloor.$$

The above formula is obtained by realizing that uploading the content from the edge-server to a helper reduces η_s as long as the helper uploads at least the same amount of content traffic to interested users before time t_1.

Hence, we can write R_i^* approximately as

$$R_i^* = \frac{\frac{1}{\mu_Q} + \frac{1}{\mu_p} - \frac{1}{\mu_{h_i}}}{\frac{1}{X} + \frac{\mu_{h_i}}{\mu_p \cdot \lambda_T \cdot p_i}}. \tag{5.7}$$

We can see that R_i^* scales with X, p_i, and μ_{h_i}, and scales inversely with μ_p and μ_Q. This implies that the optimum number of replicas for file i at helpers is proportional to the ratio of upload rate at helpers to upload rate at peers.

We can further see that for the cluster of mini-servers (helpers) to reduce the streaming cost of file i (i.e., reduce η_s), the total traffic that the cluster of helpers uploads to the system must be higher than the total traffic that the cluster of helpers downloads from the edge-server. Hence, we can write

$$\int_0^{t_0} \mu_{h_i} \cdot Y_{h_i}(t)dt + \mu_{h_i} \cdot R_i \cdot \left(t_1 - t_0\right) + \int_{t_1}^{t_2} \mu_{H_i}(t)dt \geq R_i,$$

where

$$\int_{t_1}^{t_2} \mu_{H_i}(t)dt = \left(S_i(t_2) - S_i(t_1)\right) - \int_{t_1}^{t_2} \mu_p \cdot S_i(t)dt.$$

Substituting the appropriate values in the formula above yields

$$\mu_{h_i} \cdot \tfrac{X}{2} \cdot \tfrac{R_i^2}{X^2} + \mu_{h_i} \cdot R_i \left(\tfrac{1}{\mu_Q} + \tfrac{1}{\mu_p} - \tfrac{\mu_{h_i} R_i}{\mu_p \lambda_T p_i} - \tfrac{R_i}{X} \right) +$$
$$\tfrac{\lambda_T p_i \mu_{h_i} R_i}{\mu_p \lambda_T p_i} - \tfrac{\lambda_T p_i \mu_p}{2} \cdot \left(\tfrac{\mu_{h_i} R_i}{\mu_p \lambda_T p_i} \right)^2 + \tfrac{\lambda_T p_i \mu_p}{\mu_Q} \cdot \tfrac{\mu_{h_i} R_i}{\mu_p \lambda_T p_i} \geq R_i$$

If we assume that the number of mini-servers in the cluster of helpers is given (i.e., R_i is given), then the minimum upload bandwidth ($\mu_{h_{i_{min}}}$) that every mini-server (helper) should allocate for file i (assuming it is the same for all helpers) can be computed by solving the above inequality for μ_{h_i} as

$$\mu_{h_{i_{min}}}^2 \cdot \left(\frac{-3R_i}{2\mu_p \lambda_T p_i} \right) + \mu_{h_{i_{min}}} \cdot \left(\frac{-R_i}{2X} + \frac{2}{\mu_p} + \frac{2}{\mu_Q} \right) - 1 = 0.$$

For the parameters that are used to plot Figure 5.3, and for file with size $10Mbyte$ and $R_i = 1$, we can see that $\mu_{h_{i_{min}}} \approx 35Kbps$ (we ignore the negative value). We can see that even a single mini-server (helper) with low upload bandwidth can provide performance gain to the system.

Let us now consider the case of simultaneous multiple media streaming (F media files) in the community network. The question that we would like to answer next is the following. Using our approach for media streaming, and given that we have limited idle upload bandwidth and storage capacity that can be used at helpers for media streaming, what is the optimum number of replicas (R_i) for every media file $i \in F$ that the edge-server should push into the available helper caches such that the total cost of all objects is minimized.

Define $C := \sum_{i=1}^{F} \eta_{s_i}$. Hence, our objective is to minimize C. The optimization problem is subject to

(i) $\sum_{i=1}^{F} R_i = H \cdot B$, and

(ii) $1 \leq R_i \leq H$, where B is the maximum number of media files that can be stored in any helper's cache (assuming equal-size media files and equal storage capacity at all helpers), The second constraint guarantees that every media file has at least one replica stored at the helpers. It is easy to show that the above objective function is equivalent to minimizing

$$\sum_{i=1}^{N} \frac{a \cdot R_i^2}{p_i} + b \cdot p_i \cdot e^{\frac{cR_i}{p_i}},$$

where $a = \frac{\mu_{h_i}^2}{2\mu_p \cdot \lambda_T}$, $b = \frac{\lambda_T \cdot e^{\frac{-\mu_Q}{t_2}}}{\mu_Q^2}$, and $c = \frac{\mu_Q \cdot \mu_{h_i}}{\mu_p \cdot \lambda_T}$. It can be shown that the solution is obtained when $R_i \propto p_i$.

This implies that despite the fact that a large number of peers in the community network may possess a popular file and the aggregate upload capacity of peers who store segments of the popular file is high, helpers should favour popular media content when it comes to selecting files to cache in order to minimize the overall cost of multiple simultaneous file streaming.

5.6 Media Streaming Approach Using Relay Nodes in the Community Network as Helpers

In this approach, a number of relay nodes in a community network (R_i) (e.g., mesh routers) are selected to play the role of helpers and participate in streaming of file i. Those routers are equipped (over-provisioned) with additional storage capacity and P2P-aware devices (packet sniffers) that are programmed to enable them to cache media content that they relay to next-hop nodes on paths toward destinations. Therefore, the rate at which helpers cache segments of media file i is driven by the demand for media file i initiated by users in the community network. When a helper caches a pre-determined fraction of the media file (ρ_i), the helper starts uploading the cached content to interested peers in the network.

In contrast with the mini-server scheme (Section 5.5), helpers in this setting do not consume any system resources. However, when ρ_i is high, a helper needs a long time to cache the required fraction of media file i before it registers itself with the file tracker as a content provider and starts providing (uploading) the cached content to interested peers. Thus, wasting its idle upload bandwidth. On the other hand, when ρ_i is small, a helper can quickly provides (uploads) the cached content to interested peers, but the probability that an interested peer in the community network is missing a chunk of file i that is cached at the helper becomes less.

Define $\eta_{h-p}(t)$ as the effectiveness of the helpers at any time instant t, and takes values $\in [0, 1]$ (i.e, $\eta_{h-p}(t)$ is the probability that in any unit of time, a chunk of file i cached at a helper is missing by at least one interested peer). We can see that $\eta_{h-p}(t) = 1 - (1 - \rho_i)^{D_i(t)-S_i(t)}$. Hence, η_{h-p} scales with the fraction of cached content (ρ_i) and the popularity of file i (p_i), but scales inversely with number of peers in the community network who are downloading file i. Hence, for simplification, let us define $\eta_{h-p} := c \cdot \frac{\rho_i}{\mu_Q}$, where c is constant.

Similar to the mini-servers case, we define t_0 as the instant of time at which a helper caches the required fraction of file i and becomes able to upload cached content to interested peers (same for all helpers). Define t_1 as the instant of time at which the cumulative supply bandwidth of content providing peers and helpers meet the required cumulative QoS streaming demand of interested peers, and no extra bandwidth is required from the edge-server. Define t_2 is the instant of time at which the cumulative supply bandwidth of content providing peers alone meet the QoS streaming demand of interested users and no need for helpers.

Let $\mu_{s_i}(t)$ be the rate at which the edge-server uploads data to the peers which are interested in file i during time period $0 \leq t \leq t_1$, Let $\mu_{H_i}(t)$ be the rate at which helpers upload data to peers which are interested in file i during time period $t_1 < t \leq t_2$,

To illustrate the operations of the proposed scheme, we can use the fluid-flow approximation to write

$$\frac{dS_i(t)}{dt} = \begin{cases} \mu_p \cdot S_i(t) + \mu_{s_i}(t), \text{ for } 0 \leq t \leq t_0. \\ \mu_p \cdot S_i(t) + \eta_{h-p} \cdot \mu_{h_i} \cdot R_i + \mu_{s_i}(t), \text{ for } t_0 < t \leq t_1. \\ \mu_p \cdot S_i(t) + \mu_{H_i}(t), \text{ for } t_1 < t < t_2. \\ \mu_p \cdot S_i(t), \text{ for } t > t_2, \end{cases}$$

We can compute t_0 by solving the following formula for t

$$\int_0^t \mu_Q \cdot \Big(D_i(t) - S_i(t)\Big) \cdot P_{cache} dt \approx \rho_i,$$

where P_{cache} is the probability that a participating helper overhears (relays) a segment of file i during any unit of time. The above equation is obtained under the assumption that the file is with large size such that the probability that a helpers overhears duplicate segments of the file during a unit of time is very small and can be ignored. We can see that $P_{cache} = \frac{1}{M} \cdot HOP$, where M is the total number of routers in the community network, and HOP is the average number of routers on a path between any two peers in the community network. Thus,

$$t_0 \approx \frac{\rho_i}{P_{cache} \cdot \lambda_T \cdot p_i}.$$

Similar to the mini-servers case, it can be easily shown that $t_1 = \frac{1}{\mu_Q} + \frac{1}{\mu_p} - \frac{\mu_{h_i} \cdot \eta_{h-p} \cdot R_i}{\lambda_T \cdot p_i \cdot \mu_p}$ and $t_2 = \frac{1}{\mu_Q} + \frac{1}{\mu_p}$.

Recall that η_s is the total amount of traffic uploaded by the edge-server to users interested in file i in the community network during time period $0 \leq t \leq t_2$. We can see that $\eta_s =$ total traffic collected by peers at time $t_1 -$ total traffic uploaded by peers and helpers in time period $0 \leq t \leq t_1$. Hence, the total traffic that the edge-server uploads to the community network (measured in file) can be computed as

$$\eta_s = S_i(t_1) - \int_0^{t_1} \mu_p \cdot S_i(t)dt - \mu_{h_i} \cdot \eta_{h-p} \cdot R_i \cdot (t_1 - t_0).$$

We can see that $\eta_s(\rho_i)$ is indeed a convex function (Figure 5.7). We can, therefore, find the optimum fraction of file i that helpers must cache (ρ_i^*) to yield the minimum cost (i.e., minimum η_s) by solving $\frac{d\eta_s}{d\rho_i} = 0$ for ρ_i.

Let Y be the amount of traffic that helpers upload to interested users in time period $t_0 < t < t_1$, and can be determined as

$$Y = (t_1 - t_0) \cdot \eta_{h-p} \cdot \mu_{h_i} \cdot R_i.$$

Hence, ρ_i^* can be also obtained by realizing that minimizing η_s is equivalent to maximizing Y. Thus, solving for $\frac{dY}{d\rho_i} = 0$ for ρ_i, we can determine optimum fraction of media file i cached at helpers as.

$$\rho_i^* \approx \frac{t_2 \cdot \lambda_T \cdot p_i}{\frac{c \cdot R_i \cdot \mu_{h_i}}{\mu_p \cdot \mu_Q} + \frac{1}{P_{cache}}}.$$

We can see that ρ_i^* scales with t_2 (the time instant at which peers alone are sufficient to handle the demand), the probability of caching segment of file i at helpers (P_{cache}), and μ_Q. This is because when t_2 and P_{cache} are large, there is enough time for helpers to cache large amount of data, yet be able to contribute (provide streaming bandwidth to the system) before the life-time of media file i expires. Another interesting observation is that ρ_i^* is proportional to the popularity of file i. This means that if helpers have limited storage capacity and a helper can only select few objects to cache in the case of simultaneous multiple media streaming, the helper must keep in the cache the most popular objects in order to minimize the overall streaming cost.

5.7 Energy Consumption in Media Streaming Using the Mini-servers Approach

Our objective in this section is to show that the proposed approaches for media streaming has the potential to reduce the power consumption in a community network, and consequently, in the larger ISP network. We only consider the proactive helpers (mini-servers) case in this section. However, similar results can be derived for the reactive helpers (relay nodes) case.

As we have discussed earlier, it is projected that Internet video traffic will account for over 60% off all Internet traffic by 2013 [141]. Consequently, the energy consumption in media streaming is becoming a fundamental issue. The problem becomes more critical especially with the increasing appetite for fatter bit rates for growing number of higher definition video quality desired by users.

To illustrate the capability of our proposed approaches in reducing the power consumption in the ISP networks, we consider a community network, residing in a larger ISP network (Figure 5.1). We assume, without loss of generality, a wireless community network consisting of wireless routers that are connected in the form of wireless mesh community network [7, 8].

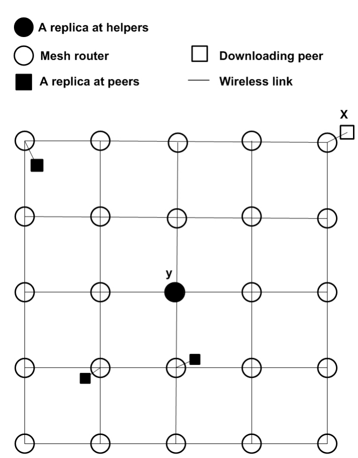


Figure 5.4: Network model

We assume a large WMCN consisting of many stationary mesh routers (M) uniformly deployed in a 2-dimensional squared area (grid-like topology) (Figure 5.4). Any two nodes that can communicate directly with each other are connected by an edge in the graph. Within the WMCN, data is communicated over wireless links. Nodes may communicate directly over a wireless link or over multiple hops with intermediate nodes forwarding data. Mesh clients (peers) are usually with much less upload bandwidth compared to the mesh routers. We assume that the routing algorithm employs a minimum hop routing metric such as AODV and DSR [61].

Although the assumptions about community network being a wireless mesh community network with grid-like topology may be restrictive, we use these assumptions in our analysis for the purpose of formulating the problem, explaining our model and verifying the effectiveness of our approach in reducing the energy consumption. The results that we obtain, however, can be extended to any community network topology no matter wired or wireless. However, the problem is more difficult due to the complexity of the wired network topology. Although energy is not a constraint in WMCNs, the consumption of energy in communication systems is becoming a fundamental issue. In fact, it has been shown that the information and communication technology sector is responsible for 2 to 2.5% of the GHG annual emission [136, 137], and the wireless access networks are largely responsible for the increase in energy consumption. It follows that being able to minimize the base station energy consumption in a wireless network represents an important green networking objective.

Let the normalized transmission power that any base station/access point (mesh router) in the community network consumes to relay a data packet of a media file to the next router on the path towards the destination be P_{MR} measured in (w/packet) (we assume equal-size packets). We assume that a peer j, which downloads file i, retrieves segments of file i from the nearest helper, which caches a replica of file i, and from all served peers in the network.

Let us suppose that file i has only one replica stored at helper at any time instant t, and this replica is stored at helper y located in the center of the network deployment area (Figure 5.4). Let the maximum power that is consumed in the network as a result of accessing a packet of file i at helper y by any peer in the network at any time instant t be $P_h(t)$. We can see that P_h is the power that is consumed when peer x located at the corner of network deployment area accesses the replica of file i at helper y (Figure 5.4) (i.e., $P_h(t)$ is the power that mesh routers consume to deliver a packet from helper y to peer x). We can further see that the average number of wireless hops between helper y and peer x (assuming a grid-like network topology) is approximately $\sqrt{M}$. Hence, we can write $P_h(t) \approx \sqrt{M} \cdot P_{MR}$, where M is the total number of mesh routers in the network.

We have seen in Chapter 4 that for a large scale wireless mesh network, when the number of replicas for file i stored at helpers is increased by a factor of 4, $P_h(t)$ is reduced by a factor of about half. Let $Y_{h_i}(t)$ be the number of replicas of file i that are stored at the helpers at time instant t. Hence, we can approximately write $P_h(t) \approx \sqrt{M} \cdot \frac{P_{MR}}{\sqrt{Y_{h_i}(t)}}$. This approximation is feasible, especially when we assume a large-scale WMCN with many helpers.

The total power consumed in the network when streaming file i with size L (packets) using our mini-servers approach ($P_{HelperSystem_i}$) can be written as

$$P_{HelperSystem_i} = P_{PeerHelper_i} + P_{ServerToPeer_i} + P_{ServerToHelper_i},$$

where $P_{PeerHelper_i}$ accounts for the power consumed when uploading file i from the served peers and helpers to interested peers, $P_{ServerToPeer_i}$ accounts for the power that is consumed when uploading file i from the edge-server to interested peers during time period $0 \leq t \leq t_1$, while $P_{ServerToHelper_i}$ accounts for the power that is consumed when uploading file i from the edge-server to helpers during time period $0 \leq t \leq t_0$.

Define P_s as the maximum power that is consumed in the network as a result of accessing a packet of file i from the edge-server by a peer in the community network. If we assume that the server is located in the center of the network deployment region, then $P_s = \sqrt{M} \cdot P_{MR}$. Define $E[P_p(t)]$ as the expected value of power consumed in the network as a result of accessing a packet of file i from the served peers. If we assume that peers are uniformly distributed in the network, and requests for objects are generated at peers independently, it is easy to see that $E[P_p(t)] = \sqrt{M} \cdot P_{MR}$. Hence, we can use the fluid-flow approximation to write

$$P_{HelperSystem_i} = L\Bigg(\int_0^{t_3} \mu_p \cdot S_i(t) \cdot E[P_p(t)]dt + \int_0^{t_0} \mu_{h_i} \cdot Y_{h_i}(t) \cdot P_h(t)dt + \sqrt{M} \cdot \frac{P_{MR}}{\sqrt{R_i}}(\mu_{h_i} \times R_i \cdot (t_1 - t_0) + \mu_{H_i}(t_2 - t_1)) + \int_0^{t_1} \mu_{s_i}(t) \cdot P_s dt + R_i \cdot P_s \Bigg).$$

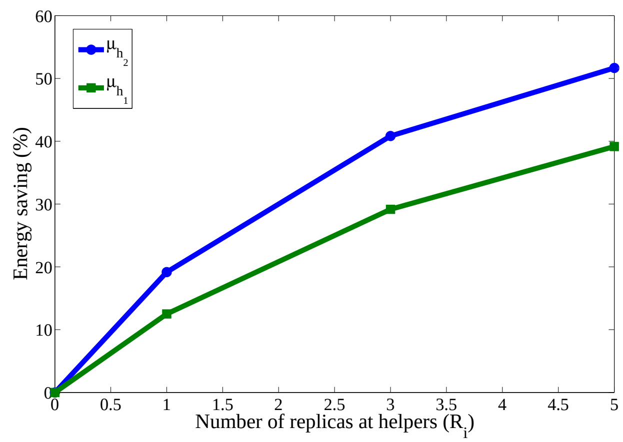


Figure 5.5: Energy saving for different μ_h.

We are interested in computing the saving in energy consumption that results from using our proposed approach as compared to a centralized media streaming approach (server-client). Let us now compute the power consumption in the network when using a centralized platform (i.e., when all interested peers download file i from the edge-server only) (P_{Server_i}). We can see that $P_{Server_i} = L \cdot N_T \cdot P_s$. Hence, P_{Server_i} can be written as

$P_{Server_i} = L \cdot N_T \cdot \sqrt{M} \cdot P_{MR}$. Thus, the saving in power consumption when using our proposed approach for media streaming as compared to using the edge-server only (η_{save}) can be computed as $\eta_{save}\% = \frac{P_{Server_i} - P_{HelperSystem_i}}{P_{Server_i}}$.

We used our analytical results to evaluate the percentage of energy saving ($\eta_{save}\%$). We considered the cases when $N_T = 500$, $\mu_p = 0.00005$, size of the file $L = 1000$ (packet), and μ_{h_i} takes two values $\mu_{h_1} = 0.0003$, and $\mu_{h_2} = 0.0005$ (Figure 5.5). We observed that the saving increases with increasing number of replicas at the helpers. This is because having more replicas stored at the helpers reduce average number of hops between replicas and downloading peers, and consequently, the power consumption. Another observation is that the saving in power increases with increasing upload rate at the helpers. This is because increasing upload rate at helpers implies that a higher fraction of file i is accessed from nearby helpers.

5.8 Simulation Results

We simulated the proposed approaches for media streaming as stochastic systems using a stochastic discrete time packet level simulation in Matlab. The first purpose of the simulations is to confirm that the fluid-flow approximation that we used does not impact the validity of our analytical results. To achieve this goal, we simulated streaming of media file i with popularity of $p_i = 0.05$. The number of potentially interested users in file i in the community network was $N_T = 85$. The size of the media file was 10 Mbyte, while the size of a packet follows a normal distribution with mean 1024 bytes (equal-size packets). The upload rate at peers and helpers followed normal distribution with means $\mu_p = 0.0001$ (file/sec) and $\mu_{h_i} = 0.0003$ (file/sec), respectively. The streaming download rate that is needed to meet a required QoS guarantee was $\mu_Q = 0.0005$ (file/sec). For each collected statistical data, we carried out many simulations and computed the average value with confidence intervals (95% confidence intervals).

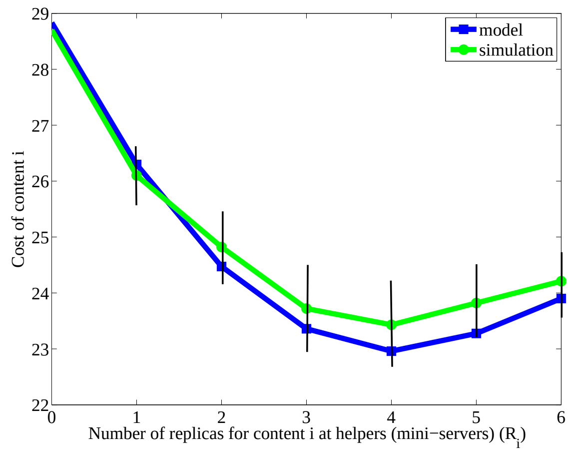


Figure 5.6: Cost of media streaming for varied number of replicas of content i stored at helpers (mini-server case).

We simulated the media streaming approach using mini-serves as helpers. We computed the normalized traffic (measured in file) that the edge-server uploads to peers in the community network for varied number of replicas for media file i stored at the helpers (R_i) (Figure 5.6). We observed that the simulation and the analytical results show similar trends.

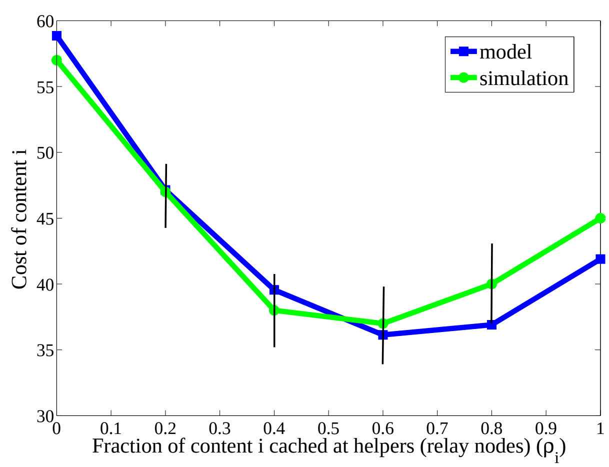


Figure 5.7: Cost of media streaming for varied fraction of content i cached at helpers (relay nodes case).

The results show that the proposed mini-servers approach reduces the amount of traffic

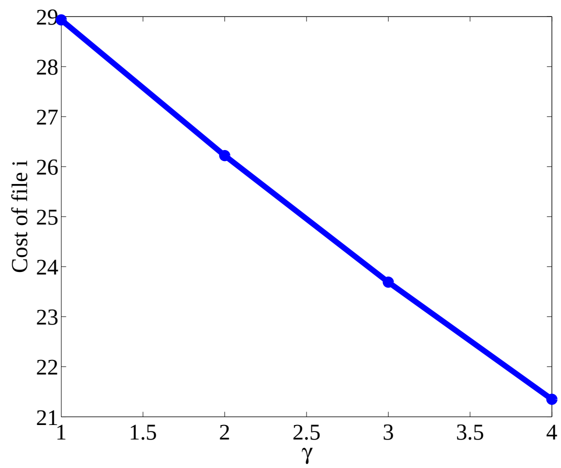


Figure 5.8: Cost of media streaming for varied $\gamma = \frac{\mu_{h_i}}{\mu_p}$ when $R_i = 3$ (mini-server case).

that the edge-server uploads to users in the community network about 20% when only 4 replicas are stored at mini-servers with very low upload bandwidth ($\mu_{h_i} = 3\mu_p = 24$ kbps). This implies that a small cluster of mini-servers (helpers), each with small upload bandwidth compared to the edge-server, has the potential to absorb large size of the traffic load in the community network and, thus, mitigate the load at the edge-server. This filters out the peak load at the edge-server and eliminates the need for over-provisioning capacities at the edge-server that target the peak expected load. This reduces the cost on the CDN in terms of purchase and maintenance.

In another set of simulations, we simulated the case of media streaming using relay nodes (e.g., routers) as helpers. We computed the normalized traffic (measured in file) that the edge-server uploads to peers in the community network for varied fraction of file i cached at the helpers (ρ_i) (Figure 5.7). We observed that the simulation and the analytical results show similar trends. This implies that the fluid-flow approximation in our modelling does not impact the validity of our analytical results.

We also used our analytical results in the case of mini-servers to compute the normalized traffic that the edge-server uploads to peers in the community network for varied γ, where γ is defined as ratio of the upload rate at helpers to the upload rate at peers (Figure 5.8). We can see that the cost decreases with increasing γ. We observe that even when γ is as low as 4 (i.e., low upload bandwidth at helpers), high cost saving is achieved. This result motivates our approach that utilizes mini-servers in community networks which are under-utilized most of the time.

Algorithm 3 Pseudocode for the on-line algorithm for determining number of replicas of object i stored at helpers (mini-servers).

Define:
h as the number of mini-servers which store a replica of file i in the community network.
t_h as the instant of time at which the edge-server finishes uploading file i to the hth mini-server in the community network.
$\eta_s(t_h)$ as the traffic load at the edge-server at the instant of time t_h.
At the edge-server:
Add a helper to file i streaming swarm,
Upload file i to the new helper,
Compute $\eta_s(t_h)$.
while $\eta_s(t_h) > \eta_s(t_{h-1})$, **do**
  Add another helper to the system.
end while

As we have discussed so far, the media streaming approaches that we advocate in this chapter determine both the optimum number of helpers (R_i^*) in the case of using mini-servers as helpers, and the optimum fraction of media file i to cache (ρ_i^*) in the case of using relay nodes as helpers based on prediction for the future streaming demand of file i in the community network (as derived in Eq. (5.1)). However, this demand function requires information, a priori, about the popularity of file i (p_i) and the maximum expected level of cumulative interested users (N_T). This can be obtained by comparing content i to historical recorded observations of interest in similar media files previously desired by users in the community network. Acquiring information about the popularity of a media content in the community network requires a centralized platform (e.g., file tracker) that can gather information about number of requests initiated at peers for the file. Although this may not be hard to achieve, we like to consider an approach for media streaming that uses mini-servers as helpers, but does not require information about popularity of content i in the community network.

In the following set of simulations, we considered a simple approach (on-line), wherein the edge-server adds helpers to the system according to the current streaming demand it receives for file i in the community network. Specifically, when media file i is generated in the community network, the edge-server uploads file i to one of the available mini-servers in the community network. Then, the edge-server monitors the traffic load (i.e., the rate at which the server uploads data to interested peers in the community network). If the the traffic load is increasing, the edge-server adds another helper to the system and uploads the file to the new helper. The process is repeated until the traffic load at the edge-server begins to decrease. When the traffic load at the edge server begins to decrease, the edge-server does not upload the content to more helpers. The pseudocode for this simple algorithm is described in Algorithm 3.

We define the parameter ($Ratio$) as percentage of cost saving when using the algorithm that acquires information about p_i (centralized) as compared to the algorithm that does not need this information (i.e., the on-line scheme). $Ratio = 100 \times (\frac{\eta_s^{on-line} - \eta_s^{centralized}}{\eta_s^{on-line}})$, where $\eta_s^{on-line}$ is the total traffic uploaded by the edge-server to interested users in the community network during the life span of file i using the on-line scheme; while $\eta_s^{centralized}$ is the total traffic uploaded by the edge-server to users using the centralized algorithm (i.e., our proposed mini-server scheme described in Section 5.5). We observed that the saving in cost scales with the upload rate at helpers (Figure 5.9). This is because in the on-line scheme, the edge-server stops uploading file i to helpers once the traffic load at the server begins to decrease. However, our simulations show that even when the traffic load at the server decreases, there is still enough time for extra helpers to join the system and provide performance gain. We observed that $Ratio$ is close to 35% when ratio of the upload rate at helper to the upload rate at peers is 7.

In the next set of simulations, we aim at validating our analytical results (power consumption in media streaming) that we derived in Section 5.7, and show that the approximations used in the fluid-flow modelling do not impact the accuracy of the results. Specifically, we simulated a WMCN that consists of 1,000 mesh routers ($M = 1000$)

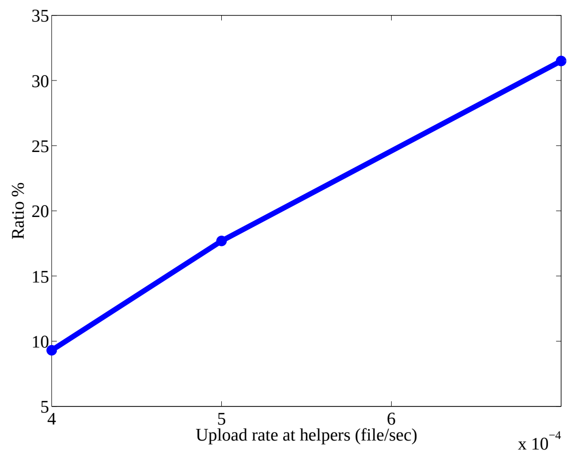


Figure 5.9: Centralized and on-line schemes comparison.

deployed in a grid-like topology. The power that a mesh router consumes to transfer a packet of data one hop towards the destination was set fixed at $P_{MR} = 1(\mu W/packet)$, $N_T = 500$, $\mu_p = 0.00005$, and $\mu_{h_i} = 0.0003$.

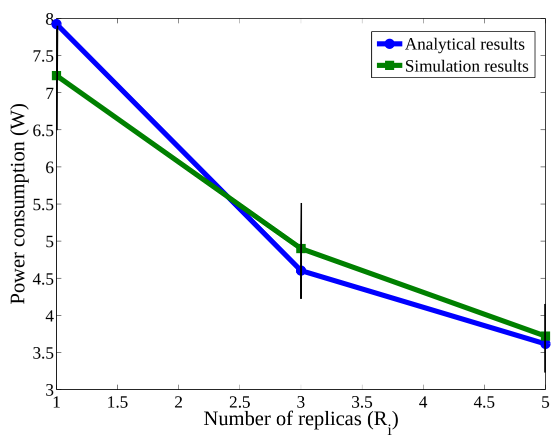


Figure 5.10: Average total power consumption

We ran many simulations and computed the average total power consumption in the network for varied number of replicas at helpers. The placement of replicas at helpers were done uniformly at random. We compared the simulation results with the analytical results that we obtained using the fluid-flow model (Figure 5.10). The results shows that our model mimics the real system very well.

5.9 Summary

This chapter introduces innovative hybrid approaches for cost-effective media streaming of stored content (e.g., VoD-like service) in the Internet. The proposed approaches exploit the abundantly available and under-utilized micro-resources in community networks to create an aggregate virtual macro-resource. In order to maximize the utilization of those micro-resources, we have developed a model that allows the ISP to anticipate the evolution of interest in a media content (future demand for media streaming) in community networks. We have shown that by allocating those micro-resources to manage the predicted future streaming demand in every community network, much of over-provisioning in the centralized architecture (i.e., the excessive computing and bandwidth capacities at the edge-servers) can be cut down, and the system scales gracefully and economically. Moreover, the bandwidth and energy consumption in the ISP networks can be significantly reduced, and the Internet bandwidth on the community operator is also reduced.

One possible future research direction is to investigate the impact of peer and helper departures on the system performance. Streaming of live media content is also an interesting problem for future work. Measurements and experimental study are required to verify our analytical demand function derived for predicting the future demand of media streaming capacity in community networks.

Chapter 6

A Ring-based Multicast Routing Topology with QoS Support in Wireless Mesh Community Networks[6]

6.1 Introduction

In a wireless mesh community network, stationary infrastructure nodes (often called mesh routers) form an access tier that connects end-users computing terminals to the network [7]. The proliferation of mobile computing devices that are equipped with video cameras and ad-hoc communication mode creates the possibility of exchanging streaming data between a group of mobile users over the WMCN. WMCNs can support wide range of application scenarios that involve group communication (e.g., video conferencing or video gaming [155, 156]). Furthermore, the self-configuring and easy deployment features of

[6] This chapter is based in part on the following papers.

A. Alasaad, H. Nicanfar, S. Gopalakrishnan, and V. C.M. Leung, "A Ring-based Multicast Network Topology with QoS Support in Wireless Mesh Networks," in *ACM/Springer Wireless networks*, published online in February 2013. DOI:10.1007/s11276-013-0559-z.

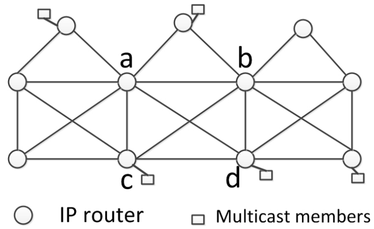
2. A. Alasaad, S. Gopalakrishnan, and V. C.M. Leung, "An Architecture with QoS Support for Application Layer Multicasting over Wireless Mesh Networks," in *Proc. of the IEEE PIMRC Conference*, pp. 1562–1566, Tokyo, Japan, September 2009.

a wireless mesh network makes it a promising alternative to the shortcomings of current communication technology used in disaster recovery applications [157, 158]. Sharing video images (streaming multicast traffic) from the incident site between the emergency responders is extremely valuable and would allow more efficient decision making. We note here that in this chapter, we use the terms wireless mesh network (WMN) and wireless mesh community network (WMCN) interchangeably.

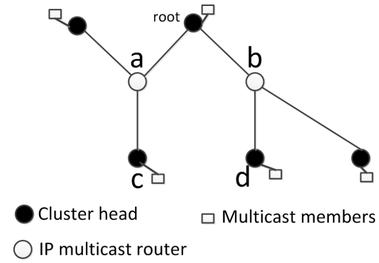
As we have discussed earlier, the proliferation and sustainability of wireless mesh community networks rely mainly on users cooperation and willingness to share their network resources in order to forward other users traffic. Therefore, services desired by users of the community networks such as group communications (e.g., video conferencing or video gaming) can be the killer applications that would encourage members of the community to donate resources to the community network.

In this chapter, we consider a setting, wherein a relatively small number of users in a WMCN are involved in a group communication. Each group member generates multicast streaming traffic and simultaneously receives multicast streaming traffic from each member in the group. These applications put further stress on the WMCN resources to maintain the QoS end-to-end delay. Moreover, these applications require a considerable amount of bandwidth and storage. Hence, innovative mechanisms are needed in limited resource networks such as WMCNs to support a reliable group communication and enable group members to consume and distribute multicast traffic in efficient and resource-aware usage.

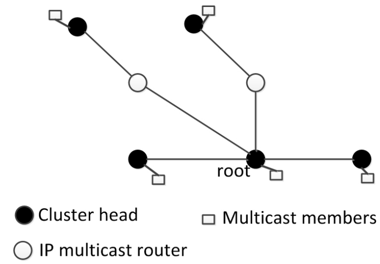
The IP multicast scheme enables efficient one-to-many and many-to-many real-time communication over an IP infrastructure in a network. Instead of sending a unicast packet to every multicast group member, IP multicasting is a method of sending a packet to a group of one-hop receivers in a single transmission. Each multicast group is assigned a unified IP address. Routers in the IP multicast network need to maintain the multicast membership information. Only IP multicast routers that are part of the multicast network



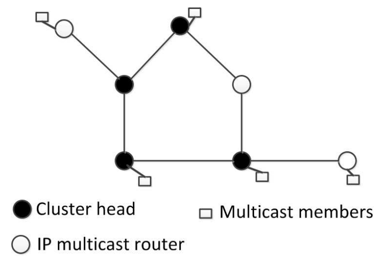
(a) Physical network topology



(b) Tree multicast routing topology



(c) Star multicast routing topology



(d) Ring-based multicast routing topology

Figure 6.1: Possible multicast routing topologies on top of the underlying physical network topology

re-broadcast (forward) the multicast packets.

Large number of IP multicast routing protocols are designed specifically for wireless multi-hop networks [159, 160]. The IP multicast routing protocol constructs a multicast network topology to connect the IP multicast routers and group members. We distinguish between the multicast routing topology and the physical network topology. The physical network topology is the connections between nodes in the wireless multi-hop network which may be grid-like or mesh (Figure 6.1(a)); while the multicast routing topology is the connections between the IP multicast routers that are selected by the IP routing protocol to forward the multicast traffic (Figures 6.1(b), 6.1(c), 6.1(d)). Some common multicast routing topologies are tree [161], ring-based [162–164], and hybrid (mesh) [160].

The performance of multicast streaming on a multicast routing topology relies heavily on a reliable and efficient MAC multicast layer. The IEEE 802.11s MAC layer standard has been developed to allow interoperability between heterogeneous mesh network devices.

802.11 unicast MAC layer, which is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), employs RTS/CTS mechanism and ACK messages to avoid collision (hidden node problem) and enable reliable communication. However, 802.11 multicast MAC layer does not involve any RTS/CTS mechanism nor does it support ACKs due to the high probability of collision between messages in this case. Consequently, the increasing probability of lost multicast frames deteriorate the quality of multicast services offered at upper layers. Therefore, multicasting in wireless mulit-hop networks is considered unreliable [51–53].

Recent studies have proposed protocols to employ RTS, CTS and ACK in the multicast MAC layer for a tree multicast routing topology to enable reliable multicasting and reduce multicast packet collisions [52, 165–169]. These protocols are designed for the case of single source multicast, and cannot be extended to the case of many source multicasting (group communication) due to design complexity and large overheads (i.e., large number of RTS/CTS messages) introduced to the network. However, the simple structure and special characteristics of a ring-based multicast routing topology when used for disseminating group communication (multicast) traffic (i.e., similarity between multicast traffic routing on a ring topology and the unicast routing) allows for simple exchange of CTS/RTS and ACK messages between neighbouring nodes on the ring (i.e., less likelihood of collisions between exchanged CTS/RTS and ACK messages). We, therefore, argue for a ring-based IP multicast routing topology for many-to-many (group) communication over a WMCN.

This chapter introduces a simple analytical model to evaluate the performance of multicast streaming on a ring-based multicast routing topology over a wireless mesh network, when the RTS/CTS mechanism is used at the 802.11 multicast MAC layer to enable reliable communication. Given the ring multicast routing topology constructed by the IP multicast routing protocol, our model allows us to derive lower bounds on the end-to-end delay and energy consumption in the network; and upper bound on capacity of the multicast network (i.e., maximum group size that the constructed ring-based multicast routing topology can support with QoS guarantees). Our results that we obtain using a model, we call it node colouring, increase our understanding of how 802.11 MAC layer with RTS/CTS mechanism affects the performance of group (multicast) communication on a ring-based routing topology over a WMCN.

Another contribution in this chapter is an efficient algorithm for enhancing traffic routing on a ring-based multicast routing topology using simple network coding mechanism. The proposed algorithm is simple, easy to implement, and does not require high processing power at mesh routers. We show that the end-to-end delay is reduced by a factor close to $\frac{2}{3}$ (33%) when our proposed algorithm is used (Section 6.4.1). We further show that our proposed algorithm increases the capacity of a ring-based multicast routing topology by a factor of $\frac{3}{2}$ (50%). The performance enhancement is a results of better utilization of available channel bandwidth at the IP multicast mesh routers on the ring topology.

Interestingly, we have shown that for a moderate multicast group size, a ring-based routing topology may outperform the tree routing topology in terms of the end-to end delay and capacity (detailed performance comparisons can be found in Appendix B). However, the performance enhancement of a ring-based multicast routing topology requires efficient construction of the ring. We, therefore, use our analytical results that we obtain for delay and capacity to develop heuristic algorithms for constructing efficient interference-aware ring-based multicast routing topology with QoS guarantees (Section 6.5). The proposed algorithms benefit from information about the physical network connections (i.e., topology of the wireless mesh network) that is available at mesh routers. Moreover, the proposed algorithms take into account all possible multicast traffic interference between wireless links in the WMCN when constructing the ring. Thus, the constructed ring topology provides QoS guarantees for the multicast traffic with high confidence in probabilistic sense, and minimizes the cost of group communications in the WMCN.

We demonstrate the effectiveness of our proposed model and algorithms using simulations (Section 6.6). Despite the approximations used in our modelling, the simulation results confirm that the bounds on performance metrics that we derive using our proposed analytical model mimic the real values very good.

6.2 Related Work

Most of earlier protocols for IP multicast routing were designed for wired networks [170]. Recently, many studies addressed multicast routing in multi-hop wireless networks (e.g., MAODV, CAMP, and ODMRP [159, 160, 171–173]). Traffic multicast requires constructing a multicast routing topology that involves all members of the multicast group. Route construction process generates large overhead to manage membership at the IP multicast routers. Therefore, a shared routing topology is constructed and shared by all group members in the case of group communication regardless of the source. Some common multicast routing topologies are tree [161], ring-based [162–164], and hybrid (mesh) [160].

Most proposed protocols for traffic multicasting are aimed at constructing an efficient (optimum) multicast routing topology (e.g., best tree routing topology) to improve specific performance metric (e.g., throughput and delay) [51, 174]. There are many studies that consider interference issues in the 802.11 MAC [175–178]. Many studies propose models to estimate the channel interference [176, 177]. Many previous studies consider a protocol model (one wireless link is interfered by another link if they are within the carrier sense range of each other) to describe the binary interference between any two links in the network; and the capacity is evaluated using graph theory [179–181]. Many studies propose interference-aware IP routing protocols [182, 183]. Other studies attempt to reduce interference between multicast traffic using directional antennas [175, 178] or multi-channel/multi-radio [184, 185]. In [186], authors propose a protocol for minimizing interference between IP multicast traffic in a tree multicast routing topology to improve network throughput. In prior work, we proposed an efficient overlay multicast routing protocol for wireless mesh networks that guarantees QoS using cross layer approach [187].

The IEEE 802.11s MAC layer standard has been developed to allow interoperability between heterogeneous network devices. The 802.11 MAC unicast layer, which is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), employs Request-To-Send (RTS)/Clear-To-Send (CTS) mechanism to avoid hidden node problem. RTS/CTS is a mechanism used to reduce frame collisions introduced by the hidden node problem. A node wishing to send data initiates the process by sending a Request to Send frame (RTS). The destination node replies with a Clear To Send frame (CTS). Any other node receiving the RTS or CTS frame should refrain from sending data for a given time (solving the hidden node problem). The amount of time the node should wait before trying to get access to the medium is included in both the RTS and the CTS frame. This protocol was designed under the assumption that all nodes have the same transmission ranges [188].

The performance of the IP multicast routing protocol relies heavily on a reliable and efficient MAC layer. However, the 802.11 multicast MAC layer standard does not use the RTS/CTS mechanism nor does it offer any MAC layer recovery (acknowledgement of received packets, ACK). Consequently, the increasing probability of lost multicast frames deteriorates the quality of multicast services offered at upper layers. Therefore, multicasting in wireless multi-hop networks is considered unreliable [52, 53].

Recent studies have proposed protocols to extend RTS, CTS and ACK to the MAC multicast layer for a tree multicast routing topology to enhance the multicast reliability and reduce multicast packet collisions [52, 165–169]. However, these protocols are designed for the case of single-source multicast, and cannot be directly extended to the case of many-source multicasting (group communication) due to their high design complexity and large overhead (i.e. large number of RTS/CTS messages) they introduce to the network.

The special structure of a ring-based multicast routing topology used for disseminating multicast traffic of a group communication (i.e., similarity between multicast traffic routing over a ring topology and the unicast routing) allows for efficient use of RTS/CTS and ACK mechanisms used in the 802.11 unicast MAC layer. Since every node on the ring has only one descendent in the multicast routing topology, every node on the ring can simply exchange RTS/CTS message with its predecessor node on the ring. To the best of our knowledge, we are the first to introduce an analytical model for analyzing the effect of 802.11 MAC layer with RTS/CTS mechanism on the performance of multicast streaming in a ring-based multicast routing topology over a wireless mesh network. We also the first to propose algorithms to construct an efficient interference-aware ring-based multicast routing topology in a wireless mesh network that guarantees QoS for the streaming traffic with high confidence, in probabilistic sense.

Examples of QoS multicast routing protocol in wireless multi-hop networks include Lantern-trees [161], QAMNet [189], OMRPCAH, and ODQMN [190, 191]. QoS multicast schemes aim at identifying a set of required components: QoS routing, resource reservation, and QoS capable MAC layer [51]. QoS schemes need to identify potential segments (links and routers) in the network that may have sufficient resources to meet the required QoS routing. Those segments will be used as candidates for selecting the best topology that can admit the session. If there is at least one potential topology, an admission control is used to determine whether source, relaying node or receiver can be connected to form the distribution topology. A per source or per multicast group resource reservation scheme is then used to secure the required resources on the distribution (multicast) topology.

The lantern-tree protocol uses a CDMA-over-TDMA [161]. Available bandwidth is measured in terms of the amount of free slots at the MAC layer. The QAMNet approach extends ODMRP routing by introducing traffic prioritization, and admission control mechanisms to provide QoS multicasting [189]. The available bandwidth is estimated as the deference between the threshold rate of real-time traffic and the current rate of real-time traffic (similar to SWAN method). However, it is difficult to realize such schemes in a dynamic and contention based wireless environment such as IEEE 802.1 (the widely used standard at the MAC layer in the wireless multi-hop networks).

The last necessary element in a QoS scheme is pre-emption (QoS policing). The stochastic changes in the conditions of the radio channel and the mobility of users in wireless multi-hop networks change the network topology and affect the reserved resources. Earlier admission decisions may no longer be valid. Therefore, new QoS routes must be selected, while previously selected links and routers on the outdated routes must be rejected and their reserved resources must be released. In this chapter, we ignore the mobility of the group members. We focus our effort at developing the algorithm that can construct an efficient and interference-aware ring-based topology. We note here that the proposed algorithm can be extended to account for mobility of group members. However, large overhead is introduced in the network.

Biradar *et al* propose an agent-based multicast routing scheme in MANET that builds a backbone in the form of a reliable ring [162]. Multi-ring Construction Algorithm (MCA) constructs a ring-based two-level hierarchical structure in a wired network, and give a mathematical formulation of the delay-constrained multi-ring construction problem [163]. However, the results cannot be extended for multicast ring in a wireless multi-hop network. VRing and SelfS protocols are based on virtual ring topology (overlay multicast) involving all multicast group members [192, 193]. In [187], we proposed an efficient overlay multicast routing protocol in a wireless multi-hop network based on ring topology.

A reliable QoS scheme relies on a good estimation of both available and required resources. The complexity of the estimation depends on the underlying MAC and physical layers. It is difficult to estimate the available bandwidth at the IP routers when a contention-based scheme such as IEEE 802.11 is used at the underlying MAC layer. Therefore, in this chapter, we use our analytically derived bounds on the delay as inputs to our proposed algorithms for constructing interference-aware ring. In other words, based on the derived analytical results, we develop heuristic algorithms for constructing a ring topology that both guarantees QoS for the multicast traffic in terms of end-to-end delay and capacity, and minimizes cost of multicasting in terms of power and bandwidth consumption. The constructed ring can be modified to account for any traffic interference that has been ignored in the analytically derived bounds.

6.3 Model Description and Assumptions

We consider a wireless mesh network such as WMCN, wherein infrastructure stationary nodes, often called mesh routers, form an access tier that connects end-users computing terminals to the network (Figure 6.1(a)). Any two nodes that can communicate directly with each other are connected with an edge in the network graph. Within the network, data is communicated over wireless links. Nodes may communicate directly over a wireless link or over multiple wireless hops with intermediate nodes forwarding data.

We summarize the assumptions used in our analysis as follows.

- All IP multicast routers in the network use MAC 802.11s radios with single frequency channel for both receiving and transmitting, and use omnidirectional antenna (360 degree beam). The Mac layer uses CSMA/CA with the RTS/CTS mechanism.
- All nodes in the network are with same transmission power and coverage area, and data traffic is much more than control traffic.
- All multicast packets have the same size.
- All nodes in the network have the same channel bandwidth (B in frame/second).
- Interference and contention for the wireless medium occurs only between the multicast traffic. In other words, we assume that a dedicated frequency channel is allocated for the multicast service at the IP multicast routers. We note here that with multi-channel MAC radios becoming more common, this assumption is feasible.
- Noise free channels. Therefore, every multicast router can fully use its channel bandwidth (B) when it gains access to the wireless medium (if no interference or

traffic collision occur).

- Every mesh router on the ring topology transmits all cached multicast traffic in a single transmission. We also assume that mesh routers on the ring attempt to access the wireless medium to relay traffic directly after receiving multicast traffic from predecessor nodes.

6.4 Performance Analysis of Multicast Streaming Using a Ring-based Routing Topology

In this section, we analytically evaluate the performance of multicast streaming running on a ring-based multicast routing topology over a wireless mesh network such as WMCN. It is not always possible to construct a ring multicast routing topology involving all group members on top of the underlying physical network that is connected in a mesh topology. However, a ring-based routing topology is always possible (Figure 6.1(d)) and in some cases lead to better performance as compared to the tree topology (see Appendix B).

The reason for so called "***ring-based***" routing topology is that not all group members are directly connected to the ring. We may have number of routers in the ring topology (we call those nodes cluster heads) that connect multicast members to the ring through few multiple wireless hops. An example of a ring-based routing topology is shown in Figure 6.1(d). Multicast members which are physically close to each other are grouped in a cluster. We assume n cluster heads, and each cluster includes m multicast members on average (i.e., the multicast group is with size $N = m \times n$). The IP multicast routing protocol establishes a routing network that connects all cluster heads using a ring topology. We call this network: ***multicast network***. Every cluster head establishes another network that connects the cluster head with members in its cluster. We refer to the network that connects a cluster head and members in its cluster as ***cluster network***

(Figure 6.2). Hence, every cluster head is involved in two separate networks: one that involves all cluster heads (multicast network), and one that involves the cluster head and members in its cluster (cluster network).

We use the following notations in our analysis:

- We denote the end-to-end delay as D. We define the end-to-end delay as the maximum time that a multicast packet spends in the multicast network to reach all group members.

- We denote the time that is required to deliver a chunk of m multicast frames (i.e., multicast traffic of a cluster) one hop on the multicast routing topology as T, where $T = m \times \frac{1}{B}$, and B is the channel bandwidth measured in frame/second.

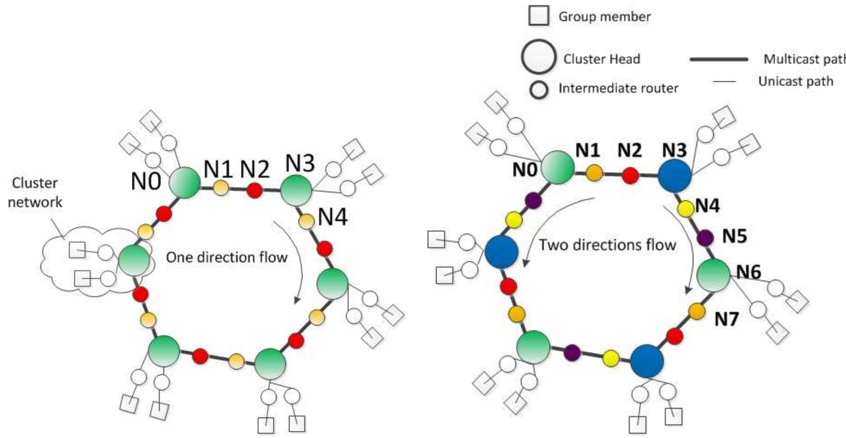
- We denote the rate at which multicast traffic is generated at each group member as b (frame/second). We assume that a group member transmits a multicast packet and waits until it receives a multicast packet from every group member before it transmits a new multicast packet. Since we assume that all multicast packets have the same size, we have $b = \frac{1}{D}$.

- Due to limited buffer size at the nodes, we assume that multicast routers can cache a maximum of N multicast packets.

- We denote the power that is consumed in the network in order to relay a multicast packet to next node (one hop) on the multicast routing topology as $P_{topology}$.

To illustrate our node colouring model, we first consider a shared multicast ring routing topology, wherein the multicast traffic is routed in one direction of the ring (Figure 6.2(a)). Every node on the ring forwards the traffic it receives from the predecessor node on the ring to the successor node.

Since the MAC multicast layer at mesh routers on the ring uses 802.11 with RTS/CTS mechanism, we can label (colour) nodes on the ring according to their MAC operating



(a) Unidirectional ring (b) Bidirectional ring

Figure 6.2: Ring multicast network topology

states as follows:

- Green (G): when the node is transmitting traffic.
- Orange (O): when the node is receiving traffic.
- Red (R): when the node is neither receiving nor transmitting in order not to cause traffic collision with the neighbouring node (hidden node problem).

When a node on the ring senses the media and finds it free, it exchanges RTS/CTS messages with its successor node on the ring. However, we assume that the delay of RTS, CTS, or ACK are very small compared to delay of multicast traffic and, thus, can be ignored.

Referring to Figure 6.2(a), suppose that at any instant of time t, node N0 is in transmission mode (Green colour) and node N1 is in receiving mode (Orange colour). Due to the RTS/CTS mechanism that is used at the MAC layer, node N2 receives the CTS message from node N1 and thus, is in hold mode (Red colour) to avoid collision at node N1. Since node N3 are not in the coverage area of node N0, it can simultaneously transmit traffic to node N4. We can see that all nodes that are three hops away on the ring can be with the same colour in the same instant of time (Figure 6.2(a)). We can also see that for successful multicast traffic routing, every node has to alternate between three colours (G,O,R) (i.e., $C = 3$, where C is number of colours). All nodes with green colour can simultaneously switch to the next colour in the sequence G-R-O. Similarly, nodes with Orange or Red colours can simultaneously switch to the next colour in the same order, and so on (i.e., all nodes with colour G can simultaneously switch to colour R, all nodes with colour O switch to colour G, and all nodes with colour R switch to colour O). Only nodes with Green colour can relay (transmit) multicast traffic at any instant of time.

We denote the time that is required to deliver a chunk of m MAC frames (a cluster multicast traffic) one hop on the multicast network as T, where $T = m \times \frac{1}{B}$ and B is measured in $frame/second$. At steady state, every mesh router with colour G needs at least $n \times T$ time period to transmit N packets ($N = n \times m$ packet) to its successor on the ring. Every mesh router on the ring with colour O needs $n \times T$ time period to receive traffic from its predecessor node on the ring; while every mesh router with colour R has to remain in hold mode for at least $n \times T$ time period to avoid collision at the neighbouring node on the ring. Since only those routers with the G colour can transmit traffic, the minimum time that is required to relay the multicast traffic between two adjacent nodes on the ring is $C \times n \times T$. We note here that due to asynchronous MAC operation at the nodes and contention based mechanism at MAC layer (RTS/CTS), the real time is larger than $C \times n \times T$.

Every cluster head needs T time period to receive a multicast packet from every member in its cluster. Every multicast packet needs to traverse all cluster heads on the ring to reach all group members. Let I be the maximum number of intermediate nodes between any two cluster heads that the multicast traffic traverse on the ring ($I = 6$ in Figure 6.2(a)). Thus, a lower bound on the end-to-end delay can be computed as

$$\begin{aligned} D_{Ring-1D_{LB}} &= (n + I) \times (3 \times n \times T) + n \times T \\ &= T(3 \times n^2 + 3 \times I \times n + n). \end{aligned}$$

We note here that in the above computation, we did not account for the traffic interference that may occur between nodes that are not direct neighbours on the ring topology. However, we will consider this interference in Section 6.5.2 when we propose schemes to construct an interference-aware ring topology. We also note that since the delay on the multicast network is much higher than the delay in the cluster network, we ignored the cluster network delay.

To achieve the end-to-end QoS guarantees for the multicast traffic (η), D must not exceed η (i.e., $D \leq \eta$). Moreover, the multicast routing topology must be able to support multicast traffic with rate of at least $N \times b$, where N is the number of group members, and b is the rate at which multicast traffic is generated at each member (frame/second). We can see that due to bandwidth constraint, contention for the wireless medium, and interference between multicast traffic, the rate (in frame/second) at which any cluster head can relay (forward) multicast traffic is limited. Referring to $D_{Ring-1D}$ computation, we can see that the maximum channel bandwidth that any cluster head can use to relay multicast traffic to the successor node on the ring is

$$\left(\frac{n \times T}{3 \times n \times T + T}\right) \times B.$$

Hence, in order for the one directional ring-based topology to support the multicast traffic, we have

$$\left(\frac{n \times T}{3 \times n \times T + T}\right) \times B \geq N \times b.$$

We know that b must exceed $\frac{1}{\eta}$ (i.e., $b \geq \frac{1}{\eta}$). Combining the two constraints, a feasible upper bound on the number of group members that a one directional ring-based topology can support with QoS guarantees (for large n) is

$$N_{Ring-1d_{UB}} = \frac{B \times \eta}{3}.$$

We can also see that the power consumed in the network in order to deliver a multicast packet to all group members can be approximately computed as $P_{Ring-1D} = P_{topology} \times (n + I)$.

Let us now consider a more practical case. In particular, a shared ring-based multicast routing topology, wherein the multicast traffic is routed in two directions over the ring (Figure 6.3). Every mesh router on the bidirectional ring-based topology relays half of the multicast traffic (i.e., $\frac{n}{2}$ cluster heads traffic) to its successor router in every direction of the ring. Therefore, we have two flows with equal amount of multicast traffic running on the bidirectional ring. When a node on the ring receives traffic from the predecessor

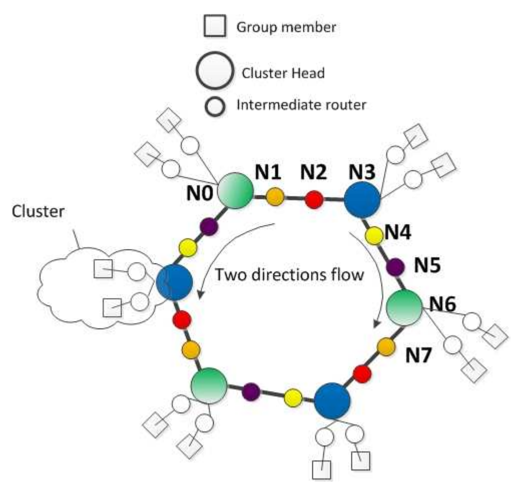


Figure 6.3: Bidirectional ring-based multicast routing topology

node on the ring, it exchanges RTS/CTS messages with its successor node in order to avoid the hidden node problem.

Proposition: The end-to-end delay of a ring-based IP multicast routing topology with two-direction flows (bidirectional ring) when RTS/CTS mechanism is used at the MAC multicast layer is

$$D_{Ring-2D} \geq T\left(\frac{3 \times n^2}{2} + 3 \times I \times n + \frac{n}{2}\right).$$

Proof. Consider the ring-based topology that is shown in Figure 6.3. If at any instant of time t node N0 is in transmission mode, node N1 is in receiving mode. Due to the RTS/CTS mechanism that is used at the MAC layer, node N2 refrains from accessing the wireless media upon receiving the CTS frame from N1 to avoid traffic collision at node N1 (hidden node problem). Also, N2 cannot receive traffic from N3, because N2 cannot exchange RTS/CTS messages with N3. Thus, N2 is in hold mode. However, N3 can exchange RTS/CTS messages with node N4 and, thus, can simultaneously receive traffic from N4. N5 cannot simultaneously transmit traffic because it is located in the coverage area of a transmitting node (N4) (exposed node problem). Also, N5 cannot simultaneously receive traffic from N6 successfully due to traffic collision at node N5. Thus, N5 is in hold mode. However, N6 is not in the coverage area of N4 and, thus, can simultaneously access the free medium and transmit traffic to node N7.

We label nodes on the ring according to their MAC operating states as follows.

- Green (G): when the node is transmitting (relaying) multicast traffic to the successor node in the direction of clockwise traffic flow.

- Orange (O): when the node is receiving traffic from the predecessor node in the direction of clockwise traffic flow.

- Red (R): when the node is neither receiving nor transmitting in order not to cause collision with the multicast traffic in the clockwise direction (hidden nod problem).

- Yellow (Y): when the node is transmitting multicast traffic to the successor node in the direction of anti-clockwise traffic flow.

- Blue (B): when the node is receiving traffic from the predecessor node in the direction of anti-clockwise traffic flow.

- Purple (P): when the node is neither receiving nor transmitting because neighbouring nodes are relaying traffic to other nodes (exposed node problem).

For successful packets delivery on the ring, every node must be in one of the above MAC operating modes at any instant of time. Since nodes in the silent (hold mode) waste (not utilizing) their channel bandwidth, we can see that every node utilizes only $\frac{4}{6}$ of its bandwidth B in the best MAC layer operation scenario. Note that due to the stochastic contention based CSMA/CA mechanism used at the MAC layer and asynchronous MAC operations at the nodes, the bandwidth utilization of the nodes is less than $\frac{4}{6}$.

We can see that in the best MAC layer operation scenario, nodes which are six hops away on the ring are in the same MAC operating state (same colour) in the same time (Figure 6.3). Suppose that at time instant t, nodes N0, N1, N2, N3, N4, N5, N6, N7 were coloured as Green (G), Orange (O), Red (R), Blue (B), Yellow (Y), Purple (P), Green, Orange, respectively. We can see that nodes with Green colour can simultaneously switch to the next colour in the order G-P-Y-B-R-O (i.e. node in clour G switch to colour P). Other nodes can simultaneously switch to the next colour in the same order (i.e. nodes in colour P switch to Y, nodes in clour Y switch to B, and so on) (Table 6.1).

Table 6.1: Node colours on the ring-based topology

Node	Time Period (t)						
N0	G	P	Y	B	R	O	G
N1	O	G	P	Y	B	R	O
N2	R	O	G	P	Y	B	R
N3	B	R	O	G	P	Y	B
N4	Y	B	R	O	G	P	Y
N5	P	Y	B	R	O	G	P
N6	G	P	Y	B	R	O	G
N7	O	G	P	Y	B	R	O

We can see that only nodes with the Green colour can transmit (relay) multicast traffic to successor node in the clockwise direction of the ring; while only nodes that are in Yellow colour can transmit (relay) multicast traffic to successor node in the anti-clockwise direction of the ring. We can see that we have six colours ($C = 6$) for the nodes on the ring (G, O, R, B, Y, P) (Figure 6.3). Hence, each packet needs at least 6 time periods in order to transfer one hop in any direction of the ring.

In order to compute the end-to-end delay, we consider the following delays. Cluster heads on the ring can simultaneously receive a multicast packet from every member in their clusters during at least T time period. Since IP multicast is employed at the network layer, when a cluster head transmits traffic, the traffic can be simultaneously detected by neighbouring nodes on the ring and by next routers on the paths towards group members in its cluster. Since, the multicast traffic is routed in two directions of the ring, each node on the bidirectional ring relays half of the cluster heads traffic in every direction. Hence, each node on the ring needs at least $C \times (\frac{n}{2}) \times T$ time period to relay half of the group multicast traffic to the successor node in every direction. Each multicast packet needs to traverse half of the cluster heads on the ring in every direction to reach all group members. Thus, the maximum number of nodes on the ring that a multicast packet must traverse on the ring is $\lfloor \frac{n}{2} \rfloor + I$. Hence, a feasible lower bound on the end-to-end delay can be computed as

$$\begin{aligned} D_{Ring-2D_{LB}} &= (\frac{n}{2} + I) \times (6 \times (\frac{n}{2}) \times T) + \frac{n}{2} \times T \\ &= T\left(\frac{3 \times n^2}{2} + 3 \times I \times n + \frac{n}{2}\right) \end{aligned}$$

We can see that due to bandwidth constraint at nodes on the ring topology and contention (interference) between traffic, the rate (in frame/second) at which any cluster head can relay (forward) multicast traffic is limited. Referring to $D_{Ring-2D}$ computation, we can see that the maximum channel bandwidth that any cluster head can use to relay multicast traffic is

$$\left(\frac{\frac{n}{2} \times T}{6 \times (\frac{n}{2}) \times T + T}\right) \times B.$$

Hence, in order for the bidirectional ring-based topology to support the multicast traffic, we have

$$\left(\frac{\frac{n}{2} \times T}{6 \times (\frac{n}{2}) \times T + T}\right) \times B \geq \frac{N \times b}{2}.$$

We know that $b \geq \frac{1}{\eta}$. Thus, a feasible upper bound on the number of group members that a bidirectional ring-based routing topology can support with QoS guarantees (for large n) is

$$N_{Ring-2d_{UB}} = \frac{B \times \eta}{3}.$$

We can also see that the power consumed in the network in order to deliver a multicast packet to all group members can be approximately computed as $P_{Ring-2D} = P_{topology} \times (n + I)$.

6.4.1 An Enhanced IP Multicast Routing Algorithm for the Ring-based Topology

We have seen in the previous section that in the case of a bidirectional ring-based routing topology, each node on the ring is at hold states (R and P colours) in $\frac{2}{6} \times 100$ percent of the time. This results in wasting (not utilizing) of channel bandwidth. To improve this situation and for better utilization of the channel bandwidth at nodes on the ring, we propose to use a simple network coding algorithm on top of the IP routing layer. Specifically, each node on the ring uses the XOR ($\bigoplus$) function to encode the traffic that it receives from predecessor nodes in every direction of the ring.

The proposed algorithm exploits the structure of a ring topology, where the amount of traffic that each node on the ring relays in every direction of the ring is the same. The network coding algorithm that we use is linear because the encoding and decoding functions are linear. Since the proposed algorithm is implemented on top of the IP layer, it does not require any modification to schemes implemented at the IP multicast layer. Therefore, our algorithm is merely an extension to the IP multicast scheme. The pseudocode for operations of the proposed algorithm that is implemented at each node on the ring is shown in Algorithm 4.

Proposition: The end-to-end delay of a bidirectional ring-based IP multicast routing

Algorithm 4 Pseudocode for the network coding operations at nodes on a ring-based topology.

Define:
RL_j as the traffic j that a node on a ring-based topology receives from the predecessor node in the direction of clockwise traffic flow,
RR_j as the traffic j that a node receives from the the predecessor node in the direction of ant-clockwise traffic flow,
DL_j as the clockwise decrypted traffic j ,
DR_j as the ant-clockwise decrypted traffic j ,
E_j as the encrypted traffic j (the output of the XOR function).
In time period i:
Receive traffic RL_j,
Decrypt traffic RL_j as $DL_j = RL_j \bigoplus DR_{j-1}$.
In time period $i+1$:
Receive traffic RR_j,
Decrypt traffic RR_j as $DR_j = RR_j \bigoplus DL_{j-1}$,
Encrypt traffic E_j as $E_j = DR_j \bigoplus DL_j$.
In time period $i+2$:
Every cluster head uses IP multicasting to transmit the encrypted traffic E_j to its neighbouring nodes on both directions of the ring and to its cluster members,
In time period $i+3$:
Every cluster head uses IP multicasting to transmit either traffic DL_j or DR_j to its neighbouring node on both directions of the ring and to its cluster members.

topology using our proposed network coding algorithm is

$$D_{Ring-NC} \geq T\left(n^2 + 3 \times I \times \frac{n}{2} + \frac{n}{2}\right). \tag{6.1}$$

Proof. According to Algorithm 4, we label (colour) nodes on the ring as follows

- Green: when a node is transmitting the encoded traffic (E traffic).
- Orange: when a node is receiving traffic from the predecessor node in the the direction of clockwise traffic flow (RL traffic).
- Red: when a node is receiving traffic from the predecessor node in the direction of ant-clockwise traffic flow (RR traffic).

Consider the following consecutive nodes on a ring-based topology ($R0$, $R1$, $R2$, $R3$, $R4$, $R5$) (Figure 6.4). Suppose that at any instant of time, nodes $R1$ and $R4$ are simultaneously transmitting encoded traffic using the IP multicast scheme (IP multicast

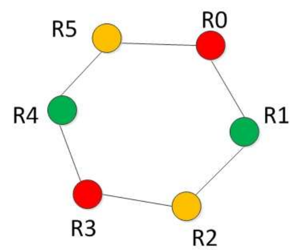


Figure 6.4: Enhanced IP multicast routing on the bidirectional ring-based topology

address). Hence, $R1$ and $R4$ are with Green colour. Nodes $R0$ and $R2$ are simultaneously receiving traffic from $R1$. Hence, $R2$ is with Orange colour, and $R0$ is with Red colour. Similarly, nodes $R3$ and $R5$ are simultaneously receiving traffic from $R4$. Hence, $R3$ is with Red colour and $R5$ is with Orange colour. Let us assume the best MAC layer operation scenarios in order to compute bounds on the performance metrics. Thus, in the next time period, nodes with Green colour simultaneously become Red, nodes with Orange colour become Green, while nodes with Red colour become Orange. We can see that every three consecutive nodes on the ring can have the same MAC operating mode at the same instant of time.

We can observe that all nodes on the ring are busy all the time (i.e., all nodes on the ring are either in receiving or transmitting mode) and there is no hold times. This is because each encoded traffic contains information that is useful to neighbouring nodes on both directions of the ring. Note that each member in a cluster needs to receive the encoded traffic E and either traffic DL or DR (Algorithm 4) in order to decode the multicast traffic.

To enable reliable multicasting, we propose the following simple operations at the MAC layer. Referring to Figure 6.4, when a relay node (say node R1) attempts to access the wireless medium to relay traffic, it exchanges RTS/CTS messages only with it successor node on the clockwise direction of the ring (i.e., node R2). However, it requires ACK message only from the predecessor node (i.e., node R0). Since node R2 attempts to access the channel directly after receiving traffic from node R1 according to our proposed algorithm for traffic routing on the ring (Algorithm 4), the transmission of node R2 is by itself an indication of successful reception of the traffic at node R2. If the node R2 does not relay the traffic within specific time period, R1 retransmits the traffic. Every multicast wireless transmission is associated with a sequence number in order to distinguish retransmitted traffic. Traffic previously received by a node R0 is ignored (dropped from the cache).

In order to compute the end-to-end delay, we consider the following delays. Cluster heads can simultaneously receive multicast traffic from group members in their clusters in at least T time period, and can simultaneously transmit traffic DL or DR to members in their clusters in at least $\frac{n}{2} \times T$ time period. Each node on the ring needs at least $C \times (\frac{n}{2} \times T)$ time period to transmit the encoded multicast traffic to next nodes in both directions of the ring ($C = 3$).

Since, the multicast traffic is routed in two directions of the ring, each multicast packet needs to traverse half the number of cluster heads on the ring to reach all group members. Thus, the maximum number of nodes on the ring that a multicast packet traverse to reach all group members is $\lfloor \frac{n}{2} \rfloor + I)$. Hence, a feasible lower bound on the end-to-end delay can be computed as

$$\begin{aligned} D_{Ring-NC_{LB}} &= (\frac{n}{2} + I)(3 \times \frac{n}{2} \times T) + \frac{n}{2} \times (\frac{n}{2} \times T + T) \\ &= T\left(n^2 + 3 \times I \times \frac{n}{2} + \frac{n}{2}\right). \end{aligned} \tag{6.2}$$

We can see that due to the bandwidth constraint at cluster heads and interference between traffic (traffic congestion at cluster heads), the rate (in frame/second) at which any cluster head can relay (forward) multicast traffic is limited. Referring to $D_{Ring-NC}$ computation, we can see that the maximum channel bandwidth that any cluster head on the ring can use to relay multicast traffic is

$$\left(\frac{\frac{n\times T}{2}}{3\times\frac{n}{2}\times T+\frac{n}{2}\times T+T}\right)\times B.$$

Hence, in order for the ring-based topology with network coding scheme to support the multicast traffic, we have

$$\left(\frac{\frac{n\times T}{2}}{3\times\frac{n}{2}\times T+\frac{n}{2}\times T+T}\right)\times B\geq\frac{N\times b}{2}.$$

We also know that $b \geq \frac{1}{\eta}$. Thus, a feasible upper bound on the number of group members that a ring-based topology using the proposed scheme can support with QoS guarantees (for large n) is

$$N_{Ring-NC_{UB}}=\frac{B\times\eta}{2}.$$

Let us now compute the power consumption in a ring-based topology using our proposed scheme. Each encoded multicast packet contains implicit information about two different multicast packets, and each cluster head needs to transmit either traffic DL or DR to members in its cluster. Thus, the power consumption in the multicast network is

$$P_{Ring-NC}=P_{topology}\times\frac{1}{2}((n+I)+n).$$

It is important to note here that the proposed algorithm for enhancing the IP multicast traffic routing is only useful for a ring-based routing topology. This is because of the symmetry in the amount of traffic that each node on a bidirectional ring relays to successor nodes in every direction of the ring. In the case of tree routing topology, for example, a node that is closer to the leaf relays high amount of multicast traffic to its descendant nodes towards the leaf, while relays very little traffic to next node towards the root. This significantly reduces the benefit of the proposed algorithm when used over a tree or star routing topologies.

Table 6.2: Performance comparison of multicast routing on a ring-based topology

Topology	End-to-End Delay	Maximum group members
One directional ring	$T(3 \times n^2 + 3 \times I \times n + n)$	$\frac{B.\eta}{3}$
Bidirectional ring	$T(\frac{3 \times n^2}{2} + 3 \times I \times n + \frac{n}{2})$	$\frac{B.\eta}{3}$
Bidirectional ring with network coding scheme	$T(n^2 + 3 \times I \times \frac{n}{2} + \frac{n}{2})$	$\frac{B.\eta}{2}$

6.5 Constructing a Ring-based Multicast Routing Topology

Our goal in this section is to construct an efficient and interference-aware ring-based routing topology that provides QoS guarantees for the multicast traffic. We observe the following dilemma: how can we select the optimum numbers of cluster heads on the ring (n^*) and the optimum number of members per-cluster (m^*) such that the end-to-end delay ($D_{Ring-NC}$ in Eq. (6.2)) is minimized? Moreover, what is the set of cluster heads (i.e., mesh routers that are to play the role of cluster heads) such that the average distance - in terms of hop-count on the path - between any cluster head and members in its cluster is minimized. The second minimization objective allows every group member to receive streaming traffic from every other member in the group within the tolerable delay, while minimizes the communication cost (i.e., bandwidth and power consumption in the network).

We first compute the optimum number of cluster heads (n^*), and the optimum number of members per-cluster (m^*) for a ring-based IP multicast topology when our enhanced routing algorithm - proposed in the previous section (Section 6.4.1) - is used. Consider a wireless multi-hop network with number of mesh routers equals M. Let N be the number of multicast members (group size). Let the end-to-end delay that is required to achieve the QoS guarantees for the streaming traffic be η. Recall that the channel bandwidth at mesh routers is B (B measured in frame per second). Let the maximum distance (in terms of number of hops on a path) between any member in a cluster and its cluster head be x. Thus, the multicast packet generated from a member in a cluster needs at least $x \times T_0$ time period to traverse the path between the member and its cluster head, where $T_0 = \frac{1}{B}$. According to our end-to-end computation ($D_{Ring-NC}$ in Eq. (6.2)), each cluster head on the ring needs at least $2 \times \frac{n}{2} \times T$ time period to receive multicast traffic from neighbouring nodes on every direction of the ring (where $T = \frac{m}{B}$). Hence, a multicast packet transmitted from a member in a cluster to its cluster head must take place during this time period in order for the cluster head to be able to collect the cluster traffic and relay it to other nodes on the ring in a timely manner without incurring extra delay. Thus, we need $x \times T_0 \leq 2 \times \frac{n}{2} \times T$. Hence,

$$x \leq \frac{n \times T}{T_0}.$$

Also, the group multicast traffic needs at least $x \times n \times T$ time period to traverse a path from a cluster head to all members in the cluster. We observe that the largest delay between any group members in the multicast group (D) can be computed as

$$D = D_{Ring-NC} + x \times T_0 + x \times n \times T.$$

To maintain QoS for the streaming traffic, we need

$$\eta \geq D_{Ring-NC} + x \times T_0 + x \times n \times T,$$

where η is the maximum tolerable delay. From the above formula, we have

$$x \leq \frac{\eta - D_{Ring-NC}}{n \times T + T_0}.$$

Combining the two conditions for x, we have

$$x \leq argmin\left(\frac{n \times T}{T_0}, \frac{\eta - D_{Ring-NC}}{n \times T + T_0}\right).$$

We assume that in a uniformly distributed nodes in a wireless multi-hop network, the average number of intermediate nodes between cluster heads on the ring (I) is directly proportional to n and, thus, I can be given as a function of n. Thus, we let $I := f(n)$. We have also shown in Section 6.4.1 that the maximum group size that a ring topology can support can be written as $N \leq \frac{B \times \eta}{2}$.

Hence, our objective is to minimize

$$D_{Ring-NC} = T(n^2 + 3 \times I \times \frac{n}{2} + \frac{n}{2}).$$

The optimization is subject to

$$\begin{cases} (i)\ N \leq n \times m \leq \dfrac{B \times \eta}{2}, \\ (ii)\ I = f(n), \\ (iii)\ x \leq argmin\left(\dfrac{n \times T}{T_0}, \dfrac{\eta - D_{Ring-NC}}{n \times T + T_0}\right). \end{cases}$$

The results of the above optimization problem is the optimum number of cluster heads (n^*), and the optimum number of members in a cluster (m^*).

We now need to select the optimum set of mesh routers in the network that are to play the role of cluster heads such that both the total cost for members to join the multicast network is minimized (i.e., number of hops on paths between members and cluster heads), while the maximum number of hops on a path between any member and its cluster head (x) does not exceed $argmin\left(\frac{n \times T}{T_0}, \frac{\eta - D_{Ring-NC}}{n \times T + T_0}\right)$. Let a_i^j be the cost for member i to join the multicast network using mesh router j as cluster head (i.e., a_i^j is the number of hops between member i and mesh router j). Let $d_j \in \{0, 1\}\ \ \forall j = 1, \ldots, M$, where 1 indicates that mesh router j is nominated as a cluster head, while 0 is otherwise. Let $b_i^j \in \{0, 1\}$, where 1 indicates that member i is using mesh router j as a cluster head, while 0 is otherwise. Define C as the total cost (i.e., total number of hops on the paths between every member and its cluster head).

Hence, our objective is to minimize

$$C = \sum_{j=1}^{M} \sum_{i=1}^{N} d_j \times a_i^j \times b_i^j.$$

The optimization problem is subject to

$$\begin{cases} (1)\ \displaystyle\sum_{j=1}^{M} d_j = n^*, \\ (2)\ d_j \in \{0,1\}\ \ \forall\, j = 1, \ldots, M, \\ (3)\ \displaystyle\sum_{j=1}^{M} b_i^j = 1,\ \ \forall\, i = 1, 2, \ldots, N, \\ (4)\ b_i^j \in \{0,1\}\ \ \forall\, i = 1, 2, \ldots, N, \text{ and } \forall\, j = 1, \ldots, M, \\ (5)\ \displaystyle\sum_{i=1}^{N} d_j \times b_i^j = m^*,\ \ \forall\, j = 1, 2, \ldots, M, \\ (6)\ a_i^j \leq argmin\left(\dfrac{n \times T}{T_0}, \dfrac{\eta - D_{Ring-NC}}{n \times T + T_0}\right),\ \ \forall\, i = 1, 2, \ldots, N, \text{ and } \ \forall\, j = 1, 2, \ldots, M. \end{cases}$$

Note that minimizing C reduces the number of wireless transmissions that are required to relay multicast traffic to all members of the group, which consequently reduces the bandwidth and energy consumption in the network. Hence, minimizing C is particularly useful in wireless multi-hop networks such as the WMCN. The result of the above optimization problem is vector D with size M, and matrix B with size $N \times M$, where $D = \{d_j | j = 1, ..., M\}$ consists of entries $k \in \{0, 1\}$, where 1 indicates that mesh router j is a cluster head; while $B = \{b_i^j | i = 1, \ldots, N; j = 1, \ldots, M\}$ consists of entries $k \in \{0, 1\}$, where 1 indicates that member i is using mesh router j as cluster head. The constraint in (6) ensures that the distance from every member and its cluster head does not exceed

$argmin\left(\frac{n\times T}{T_0}, \frac{\eta - D_{Ring-NC}}{n\times T+T_0}\right)$ (i.e., all members in the multicast group collects streaming traffic from every other member within the tolerable delay $D \leq \eta$).

The total number of possibilities (P) that we need to examine in order to find the optimum set of mesh routers that are to play the role of cluster heads is

$$\begin{aligned} P &= \begin{pmatrix} M \\ n^* \end{pmatrix} \times \begin{pmatrix} N \\ m^* \end{pmatrix} \times \begin{pmatrix} N-1 \\ m^* \end{pmatrix} \times \begin{pmatrix} N-m^* \\ m^* \end{pmatrix} \times \cdots \times \begin{pmatrix} N-m^*(n^*-1) \\ m^* \end{pmatrix} \\ &= \begin{pmatrix} M \\ n^* \end{pmatrix} \Pi_{i=0}^{n^*-1} \begin{pmatrix} N-i\cdot m^* \\ m^* \end{pmatrix}. \end{aligned}$$

It can be shown that the above problem is NP-Hard [74]. Hence, we propose a heuristic scheme to select mesh routers that are to play the role of cluster heads as follows. We assume that the group multicast session is initiated by member h. We denote member h as the group leader. Every multicast member i exchanges control messages with every other member j in the group to compute the distances (number of hops $d_{i,j}$) between itself and other members. Define vector S_i as ($S_i = [d_{i,j},\ \forall i, j \in N]$). Every member i sends the vector S_i to the multicast leader. We note that this process can be eliminated if the location of the members is known (e.g., using GPS). Upon receiving all vectors at the multicast leader h, the leader computes the average distance from every member and other members in the group, and finds member Z that has the shortest average distance to all other members. Note that member Z is located in the center of the geographical area over which the group multicasting is taking place.

The leader then sends a message to member Y, which is the furthest to Z in terms of hop-count, with a request to form a cluster. When Y receives the request, it sends a message to all $m^* - 1$ group members that are physically closest to Y with a request to form a cluster. This process ensures that no group member is isolated, and all members are close to the ring (i.e., x is short). When members receive the request, they nominate a cluster head as follows: Members compute the average distance between each others, and find member K that has the minimum average distance to other $m^* - 1$ members in the cluster. K sends a message to the mesh router that directly connects K to the WMCN with a request that the mesh router play the role of cluster head.

After forming the cluster, the multicast leader h eliminates members of the formed cluster from the set of group members. Then, the leader sends another request to a member in the remaining set of group member, which is the furthest to Z, with another request to form another cluster. The process is repeated until every group member becomes a member of a cluster. The pseudocode for this algorithm for ring construction is shown in Algorithm 5. After selecting the set of mesh routers that are to work as cluster heads, we need to connect those selected cluster heads such that number of nodes on the ring is minimized, so as to reduce the end-to-end delay on the ring (recall that $D_{Ring-NC}$ scales with number of nodes on the ring Eq. (6.2)). We, therefore, suggest using the adaptive ring heuristic salesman approach proposed in [194]. We refer to the constructed ring at this stage as the ***minimum-cost ring***. If any member in the cluster is with distance to the nominated cluster head that exceeds $argmin\left(\frac{n\times T}{T_0}, \frac{\eta - D_{Ring-NC}}{n\times T + T_0}\right)$, members nominate another cluster head.

Algorithm 5 Pseudocode for constructing minimum-cost ring topology.

Define:
$Members$ as set of multicast group members that are not in a cluster,
S as set contains information about the average distances between every member in the multicast group and all other members.
At group leader:
Computes the average distance between every member in the group and other members (S_i),
Find member Z that has the shortest average distance to all other members.
while $Members \neq \phi$, **do**
  Find the member Y in $Members$ which is the furthest to member Z,
  Find $(m^* - 1)$ members in set $Members$ that are closest to Y and request members to form cluster i and nominate a cluster head,
  if the distance between any member in cluster i and the nominated cluster head is less than x, **then**
  Eliminate members of cluster i from $Members$,
  else
  Nominate another cluster head.
  end if
end while

6.5.1 SINR Computation

So far, we cannot say that the constructed ring (i.e., the minimum-cost ring) guarantee QoS for the multicast streaming traffic with high confidence in probabilistic sense. This is because we only account for interference between direct neighbours on the ring topology when computing $D_{Ring-NC}$ in Section 6.4. Due to the broadcasting nature of wireless medium, traffic interference in wireless networks is defined as the summation of undesired signals collected at a receiving node. If the summation of noise and interference at a receiving node is larger than certain value, the receiver cannot decode the desired signal without error. In other words, if $SINR < \beta$, where $SINR$ is the ratio of power received from the desired signal to summation undesired signals and noise, the received signal cannot be decoded correctly. Hence, in order to construct a ring topology that support QoS with high confidence in probabilistic sense, we need to account for all possible interference in the multicast network.

Referring to our proposed enhanced IP routing algorithm (Algorithm 4), nodes in Green colour are the only nodes in the multicast network that can be in transmission mode at any instant of time; while nodes in Red and Orange colours are in receiving mode. Hence, nodes in Orange and Red colours may experience interference caused by nodes in Green colour. Hence, $SINR$ at any receiving node can be computed using Friis model as

$$SINR = \frac{P_{r_i}}{N + \sum_j I_j},$$

where P_{r_i} is received power of the signal that carries the desired data transmitted by node i, and can be computed as

$$P_{r_i} = \frac{K \times P_{t_i}}{D_i^n},$$

where K is a constant value (function of channel, environment, path loss and antenna gain), P_{t_i} is the power of signal transmitted by node i, D_i is the distance between sender i

and the receiver, and $n = 2$ for free space. N is thermal noise; while $\sum_j I_j$ is summation of received power of undesired signals (interference) caused by transmissions of other Green nodes in the network.

Hence, we can compute SINR at any receiving node on the ring topology (assuming all node are with the same transmitted power, $N \approx 0$, and K is constant) as follows.

$$SINR = \frac{K \times \frac{P_{t_i}}{D_i^2}}{N + \sum_j K \times \frac{P_{t_j}}{D_j^2}} = \frac{\frac{1}{D_i^2}}{\sum_j \frac{1}{D_j^2}}$$

6.5.2 Interference-aware Ring Construction

In this section, we propose an algorithm to construct an interference-aware ring topology. The algorithm modifies the topology of the minimum-cost ring to account for traffic interference in the WMCN that we did not account for in our analytical modelling and computation of end-to-end delay on the ring ($D_{Ring-NC}$). Every node on the the minimum-cost ring topology computes its SINR as follows. Consider a minimum-cost ring topology as shown in Figure 6.5. Suppose that at an instant of time, nodes c, f, i, l are with Green colour. Hence, every node on the ring can configure its sequence on the ring and determines the instants of time when it holds any of the colours (Green, Orange, and Red).

Every node computes its SINR in two cases, the case when it is in Orange colour (receiving traffic from the predecessor node in the direction of clockwise traffic flow), and the case when it is in Red colour (receiving traffic from the predecessor node in the direction of anti-clockwise traffic flow). Specifically, when a node is in Red colour (say node b in Figure 6.5), it determines its SINR as follows.

$$SINR_b^{Red} = \frac{\frac{1}{D_c^2}}{\sum_j \frac{1}{D_j^2}},$$

where D_c is the distance between nodes b and c; while D_j is the distance between nodes

b and f, i, l (i.e., $j \in \{f, i, l\}$. Similarly, node b can determines its SINR when it is in Orange colour as

$$SINR_b^{Orange} = \frac{\frac{1}{D_a^2}}{\sum_j \frac{1}{D_j^2}},$$

where $j \in \{k, h, e\}$. Nodes use their physical location information to calculate the distances between them. Nowadays, this is can be easily achieved by equipping mesh routers with GPS.

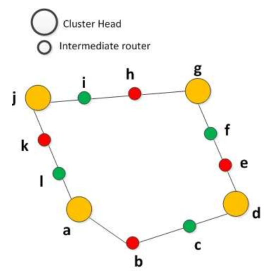


Figure 6.5: Interference-aware ring

Let us assume that a receiving node can decode the desired signal without error when $SINR > \beta$. Suppose that a node on the ring (say node b in Figure 6.5) has $SINR_b^{Orange}$ or $SINR_b^{Red} < \beta$, this implies that this node is not able to decode the desired signal and a re-transmission of the desired signal is required. This adds to the delay of the ring and, consequently, violates our computation of the ring delay ($D_{Ring-NC}$ in Eq. (6.2)). To avoid this problem, we propose interference-aware ring construction algorithm as follows. The interference-aware algorithm starts with the minimum-cost ring as an input (set R includes all nodes on the minimum-cost ring). If a node in set R (say node b in Figure 6.5) has $SINR_b^{Orange}$ or $SINR_b^{Red} < \beta$, it is added to a set called Q. The algorithm attempts to construct an alternative path in the minimum-cost ring between nodes a and d that satisfies three conditions. Firstly, the new path must be as short as possible (in terms of hop count) so as to keep the ring delay small. Secondly, the new path must not contain node b. Lastly, the new path must not add extra nodes to the set Q. In other words, nodes on the new path must not cause traffic interfere with other nodes on the ring (i.e.,

$Size(Q_{new}) < Size(Q)$).

Algorithm 6 Pseudocode for constructing interference-aware ring topology (QoS ring).

Define:
R as set of nodes on the minimum-cost ring,
Q as set of nodes $\in R$ that have $SINR_i < \beta$, where i is a node in the set R (i.e., $i \in \{1, 2, .., R\}$),
$Path_i$ as set of all available paths between the two adjacent cluster heads to node $i \in Q$,
R_{new} as set of nodes on the new constructed ring,
Q_{new} as set of nodes on the new constructed ring that have $SINR_i < \beta \; \forall i \in R_{new}$,
$i \leftarrow 1$,
while $Size(Q) > 0$, **do**
  h_i = number of hops on the path between adjacent cluster heads of node i on the ring (initial path),

  $j \leftarrow 0$,
  while $Size(Q_{new}) \geq Size(Q)$, **do**
    $Path_{i_j}$ = set of all paths $\in Path_i$ with number of hops $j + h_i$,
    $k \leftarrow 1$ {index for paths in set $path_{i_j}$},
    for path k, **do**
      {selecting best path}
      $R_{new_{j,k}} \leftarrow$ new ring that includes path k,
      Compute $Q_{new_{j,k}}$ {number of nodes on the new ring that have $SINR_i < \beta \; \forall i \in R_{new}$},
      $k \leftarrow k + 1$,
    end for
    Select the path k^* in the set of $Path_{i_j}$ that has the minimum $Size(Q_{new_{j,k}})$,
    $Q_{new} = Q_{new_{j,k^*}}$,
    if $Size(Q_{new}) \geq Size(Q)$, **then**
      $R_{new} \leftarrow R_{new_{j,k^*}}$,
      $j \leftarrow j + 1$,
    end if
  end while
  $Q \leftarrow Q_{new}$,
  $R \leftarrow R_{new}$,
  $i \leftarrow i + 1$,
end while

The algorithm works as follows. Every node i in the set Q sends a message to its adjacent cluster head in the direction of anti-clockwise traffic flow (e.g., b sends a message to cluster head d). When d receives the message, it broadcasts a route request message $RREQ$ to node a (message flooding) after inserting into the $RREQ$ node a's IP address. Intermediate nodes relay the RREQ after inserting their IP addresses in the header packet. This scheme is similar to the route discovery schemes used in the popular AODV routing protocol. $RREQ$ message reaches node a from multiple path carrying information about all available paths between nodes a and d. For all available paths, node a runs Algorithm 6 to select the best path between nodes a and d.

If a cluster head on the the ring (say node a in Figure 6.5) is with $SINR_a < \beta$, the path between its adjacent cluster heads (nodes j and d) needs to be replaced, and members belonging to cluster head a need to nominate a new cluster head on the new path. After constructing the interference-aware ring topology, the leader of the group multicast computes the end-to-end delay of the interference-aware ring ($D_{Ring-NC}$) as in Eq. (6.2). Given $D_{Ring-NC}$ and the required QoS delay (η), if $D_{Ring-NC} > \eta$, the leader must either ask members to reduce the rate at which multicast traffic is generated (i.e., reduce b), or reject some of join requests in order to reduce multicast traffic carried on the interference-aware multicast ring-based network topology. It is needless to mention that the ring-based multicast network topology must account for nodes mobility in the WMCN. Therefore, the ring construction scheme must be executed every time group members change their locations.

6.6 Evaluation

We simulated a stochastic group multicasting in static multi-hop wireless network using OPNET [83]. Our objectives are four folds. Firstly, we would like to show that our node colouring model is a good approximation of the performance of the MAC layer. Secondly, we would like to show that our derived bounds mimics the real values very good. Thirdly, we like to use simulations to evaluate the performance of our proposed algorithm for enhancing IP multicast routing on the ring multicast routing topology. Fourthly, we like to evaluate the performance of our algorithms proposed for constructing the interference-aware ring routing topology.

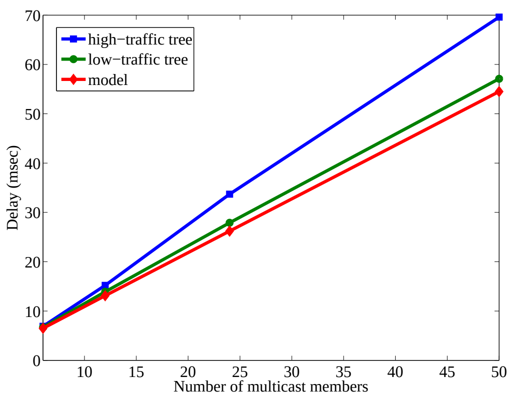
To achieve these goals, we simulated a static multi-hop wireless network consisting of number of static routers and end-users computing devices. Nodes were uniformly deployed on $5 \times 5Km^2$ area. The users participated in the group multicast session. The primary metrics for the performance evaluation were: end-to-end delay, power consumption, and traffic interference in terms of number of dropped packets, wireless medium access delay, and packet retransmission. We are not interested in investigating the impact of mobility of group members on the performance, and thus, each user was made to stay connected to a static router for large period of time exponentially distributed with mean 30 minute (the entire simulation time). The simulation parameters are summarized in Table 6.3.

Table 6.3: Simulation parameters

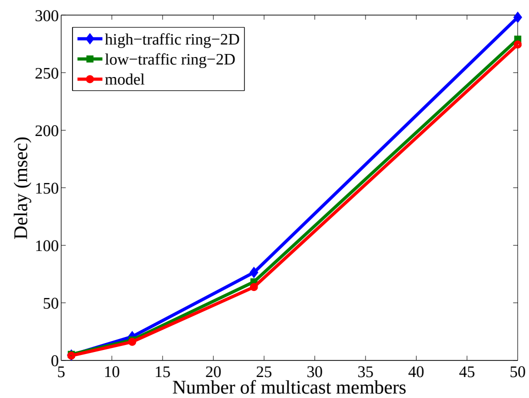
Parameter	Value
MAC layer	802.11b (Direct Sequence)
MAC data rate	$11Mbps$
Transmitted Power	0.02 (W)
Packet Reception (Power Threshold)	-95 (dbm)
Average packet size	50 (bytes)
MAC buffer size	1024 Kb
PIM-SM Hello packet interval	30 (sec)
PIM-SM holding time	105 (sec)

To simplify our simulation set-ups, we simulated n static routers playing the role of cluster heads. Each cluster head had only one member in its cluster $m = 1$ (i.e., $N = n$). We ran the PIM-SM (version two) protocol for IP multicast routing in all the investigated topologies [170], and the IGMP protocol for communication at the last hop between cluster heads and multicast members. The PIM-SM protocol establishes shortest path tree between multicast members and the root. Therefore, before we run the simulation for the ring-based topology, we had to disable this feature and manually establish the multicast routes for the ring-based topology. We implemented our proposed algorithm for enhancing IP multicast routing at nodes on the ring-based topology. The algorithm was implemented on top of the IP network layer to enable smooth compatibility with the PIM-SM protocol. The Rendezvous point (RP) and the multicast group address were statically selected and configured for all nodes. Multicast traffic starts 300 sec after running the simulation, while members join the group after 65(sec).

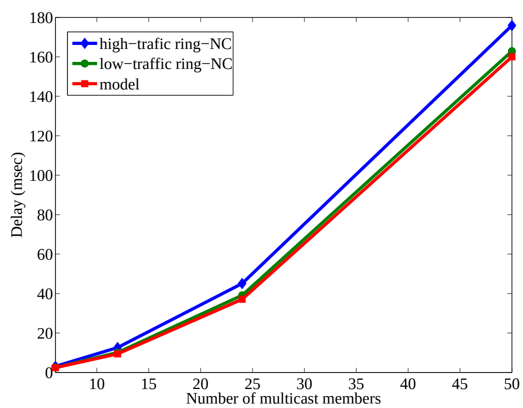
Recall that in our analytical modelling, interference between wireless links was determined based only upon their relative logical positions on the routing topology. To investigate the impact of this approximation on the accuracy of the derived bounds, we simulated two kinds of multicast traffic: Low traffic rate where each member generates $12Kb/sec$ of multicast traffic, and high traffic rate ($120Mb/sec$). Type of service was set to streaming multimedia. For each evaluated performance metric, we ran our simulator with different seed values including 40, 80, and 128 for 30 minutes, and the average of statistical results was computed.



(a) End-to-End delay in a tree topology



(b) End-to-End delay in a bidirectional ring-based topology

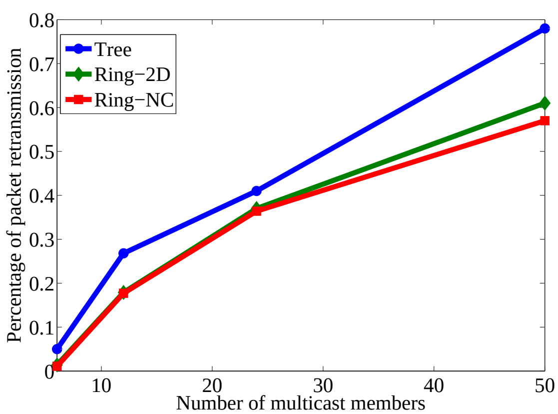


(c) End-to-End delay in a ring-based topology with the network coding algorithm

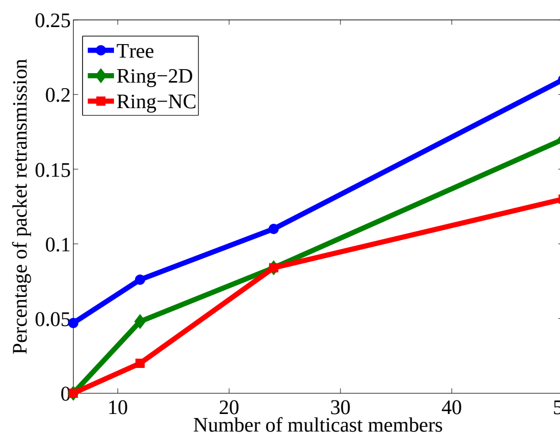
Figure 6.6: End-to-End delay

We computed the average end-to-end delay in the ring-based and tree routing topologies for varied number of group members N (N is equal to cluster heads (n) on the multicast routing topology). We compared the simulation results with the analytical results (the derived lower bound on the delay) in a tree topology (Figure 6.6(a)). Please see Appendix B for detailed description of the end-to-end computation on tree routing topology. We observe that our model provides a good approximation of the performance of the MAC layer, and our derived bound on the end-to-end delay is close to the one obtained using simulations. The gap between our derived delay and the simulations results is higher in the high-traffic case. This gap is a result of two reasons. Firstly, the approximation used in our modelling which only accounts for the interference between traffic transmitted from adjacent nodes on the routing topology. Secondly, the stochastic media access mechanism (CSMA/CA) used in the MAC layer which we did not account for in our modelling.

We also compared the simulation results with our bounds derived for a bidirectional ring-based topology using pure PIM-SM (no network coding) (Figure 6.6(b)), and in a bidirectional ring-based topology using PIM-SM coupled with our proposed network coding algorithm (Figure 6.6(c)). We can see that our end-to-end bounds provide a very good approximation of the real system delays.



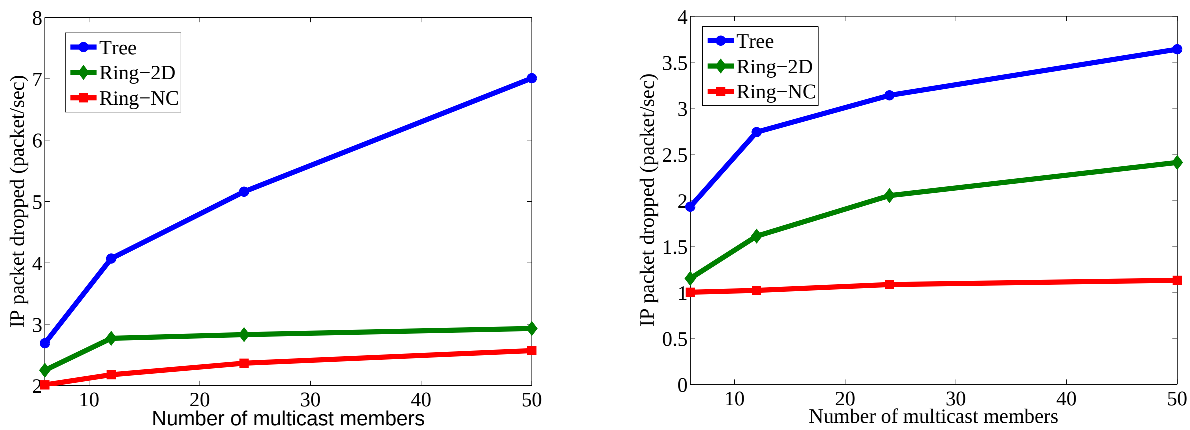
(a) High traffic



(b) Low traffic

Figure 6.7: Average percentage of retransmitted packets

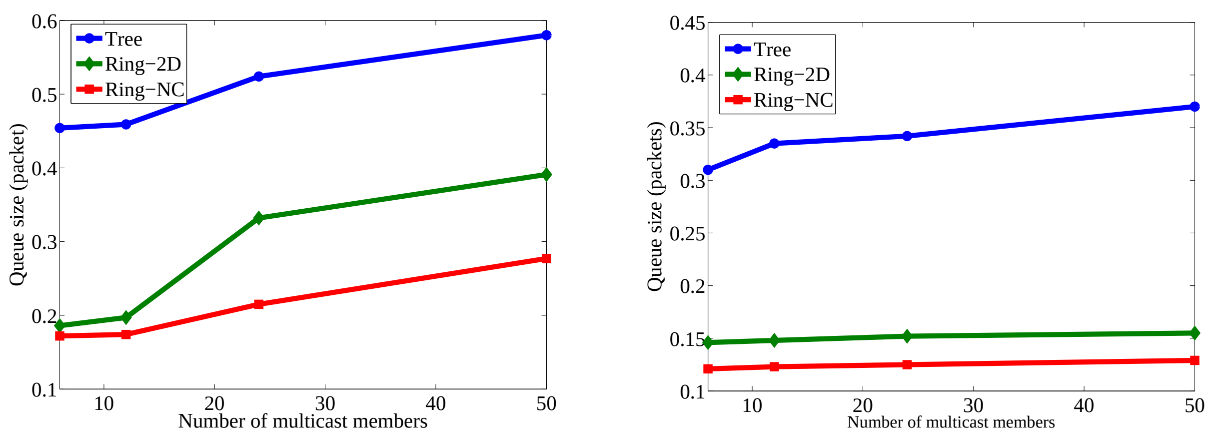
We observe that the gap between our derived bounds and the simulation results is higher in the tree routing topology as compared to the ring. This is due to interference between traffic on adjacent multicast paths (tree branches) on the tree topology, which we did not account for in our analytical modelling. This interference phenomenon was observed when we compared the percentage of retransmitted packets (Figure 6.7), number of dropped IP packets/sec (Figure 6.8), and size of queue at multicast routers on the routing topologies (Figure 6.9).



(a) High traffic

(b) Low traffic

Figure 6.8: Average number of dropped packets



(a) High traffic

(b) Low traffic

Figure 6.9: Average size of queue at mesh routers

We also compared the end-to-end delay that was obtained using simulations in the cases of tree, bidirectional ring, and bidirectional ring with our proposed network coding

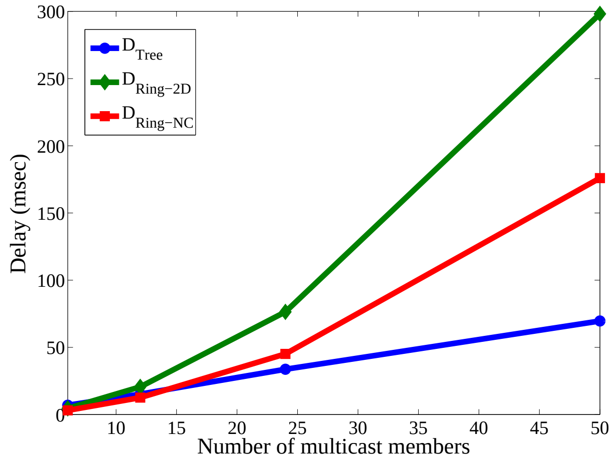
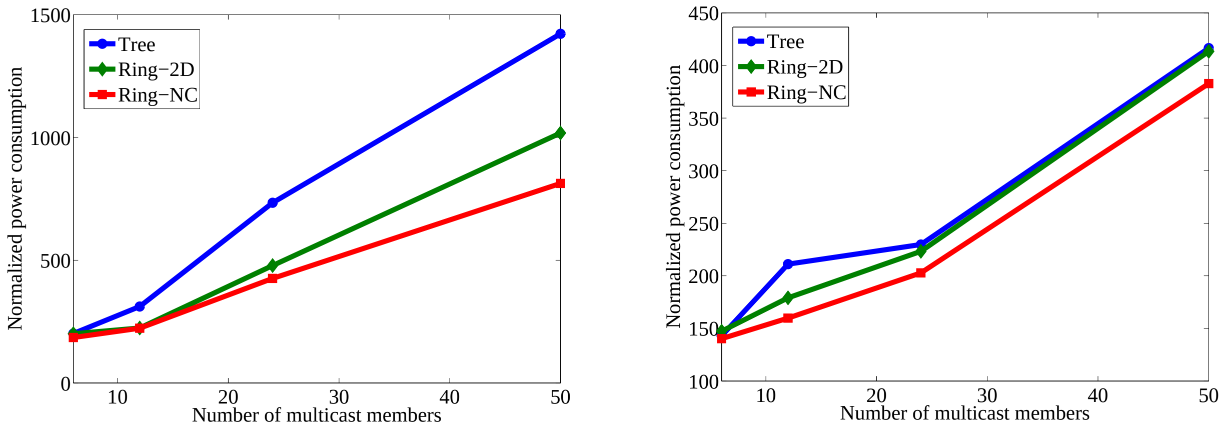


Figure 6.10: End-to-End delay

algorithm (Figure 6.10). The simulation results reveal that a bidirectional ring-based topology coupled with our proposed algorithm for traffic routing has the least end-to-end delay when the number of group members (i.e., number of cluster heads on the network topology n) is less than 12. However, the tree topology has the least delay when number of cluster heads is higher than 12. This is because the average number of wireless hops on paths between multicast members on a ring-based topology becomes higher for higher number of cluster heads on the ring as compared to a tree topology. We observe that a bidirectional ring-based topology with the proposed network coding algorithm outperforms a ring-based with pure PIM-SM routing protocol. This is because our algorithm enables multicast routers to better utilize the available channel bandwidth. It is important to note here that the performance of the ring routing topology coupled with our proposed routing algorithm can outperform the tree routing topology even for higher number of multicast nodes on the ring when amount of multicast traffic carried on the routing topology is high. This is because interference between parallel multicast paths on the tree topology deteriorates the performance of multicast streaming.

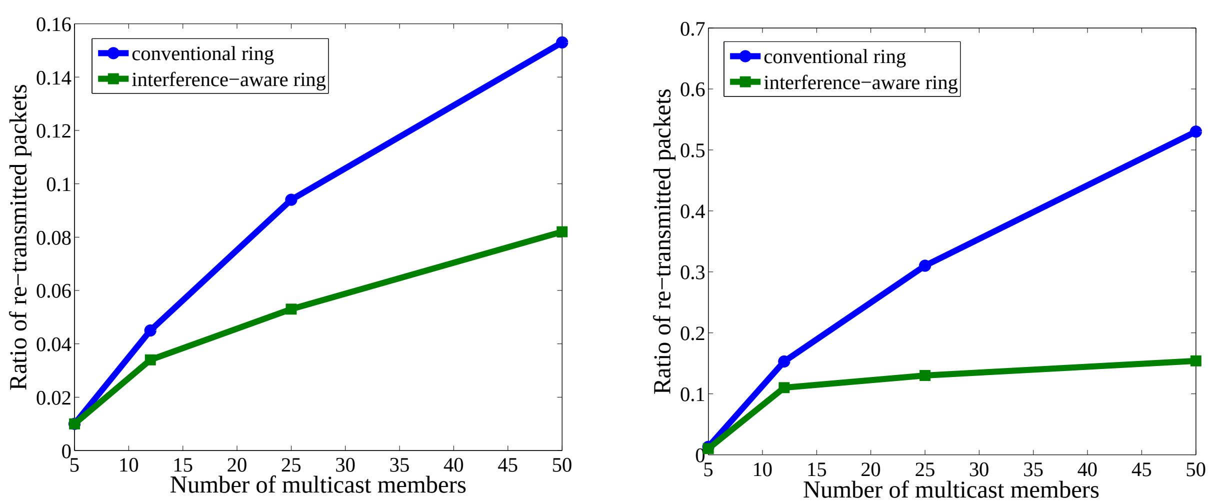
Another important performance metric is the power consumption in the network. Since we simulated multicast packets with fixed average size, we assume that the power



(a) High traffic

(b) Low traffic

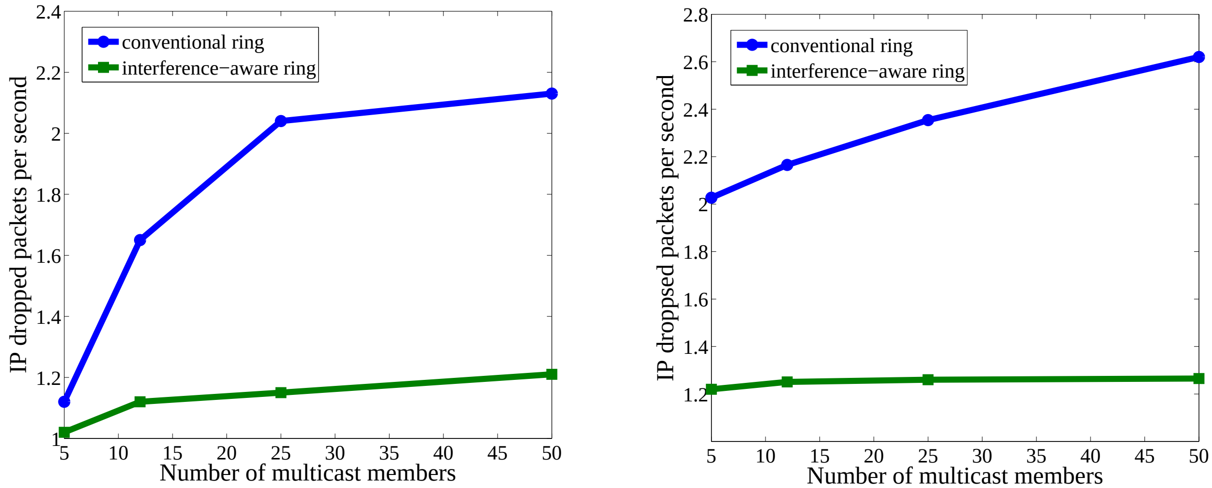
Figure 6.11: Average normalized power consumption



(a) Low traffic

(b) High traffic

Figure 6.12: Average ratio of packets re-transmitted



(a) Low traffic

(b) High traffic

Figure 6.13: Average number of dropped packets per second

consumption in the network is proportional to the total number of wireless packet transmissions. We observe that the tree consumes the highest amount of power, while the ring-based using our proposed algorithm for traffic routing consumes the least amount of power (Figure 6.11). This high power consumption in tree topology is mainly due to the high congestion at the cluster heads on the tree topology, which results in high interference, packet collisions, and packet retransmissions. The low power consumption in the ring-based topology using the proposed algorithm for traffic routing as compared to ring-based topology using pure IP mulitcast scheme for traffic routing is because our proposed scheme benefits from the structure of a ring topology. Thus, our proposed scheme enables node on the ring to encode the multicast traffic in such a way that allows for better bandwidth and power utilization.

To evaluate the performance of our proposed algorithm for constructing interference-aware ring (Section 6.5.2), we simulated two ring topologies. The first one is a ring topology constructed without using our proposed algorithm (i.e., the minimum-cost ring). The second one is a ring topology constructed using the proposed algorithm (Algorithm 6) (i.e., the interference-aware ring). We evaluated the performance of both topologies in terms of both the ratio of retransmitted packets to successfully received packets (Figure 6.12), and number of dropped IP packets/sec (Figure 6.13). The simulation results revealed the performance enhancement of the ring topology using our proposed algorithm. We can see that number of dropped packets and number of re-transmissions can be significantly reduced when our algorithm is used. We can see that the ring using our algorithm is still suffering from packet drop. However, most of these dropped packets are due to contention for the wireless medium and traffic collision. This mainly results from the 802.11 MAC multicast layer standard, which does not support any RTS/CTS mechanism.

6.7 Summary

This chapter investigated the advantages of using a ring-based multicast routing topology to support a reliable group communication over a wireless mesh network such as a WMCN. We have shown that support from the infrastructure nodes (mesh routers) on the ring to the group communication service by playing the role of cluster heads and implementing our proposed routing algorithm, results in significant performance enhancement for the group communication service.

We developed an analytical model which allows us to derive bounds on performance metrics such as end-to-end delay, capacity, and power consumption for a ring-based multicast routing topology. We have shown that the derived bounds are feasible, very useful, and mimic the real values very good. We have also proposed an algorithm to enhance IP multicast traffic routing on a ring routing topology. We have shown that our proposed algorithm can reduce the delay of a ring routing topology by a factor close to $\frac{2}{3}$ (33%), and increase the capacity of a ring routing topology by a factor of $\frac{3}{2}$ (50%).

We have further proposed algorithms for constructing efficient and interference-aware ring-based multicast network topology over a wireless mesh network. The proposed algorithms construct a ring multicast routing topology that achieves two goals. Firstly, it meets QoS requirements for the streaming multicast traffic with high confidence in probabilistic sense. Secondly, it reduces bandwidth and power that multicast traffic consumes in the network.

One possible direction for future research is to derive better bounds on performance analysis by considering geographic properties of nodes in the physical network topology (perhaps using unit disk graph) and finding an upper bound of non-interfered nodes.

Chapter 7

Conclusion

In this dissertation, we demonstrate that support for P2P applications at infrastructure nodes in WMCNs (e.g., mesh routers) results in significant performance improvements for P2P content sharing and media streaming. Such support from the infrastructure nodes benefits from both awareness of the underlying network topology at mesh routers, and the under-utilized resources at a large number of infrastructure nodes in the WMCN.

Our main contributions in this dissertation were two-fold. Firstly, we have considered the problem of P2P content sharing and distribution in wireless mesh community networks. We have shown that the stationary characteristics of mesh routers in a WMCN can be exploited to construct a topology-aware overlay network that peers can use to lookup desired contents. We have shown that such support from the mesh routers mitigates the traffic load imbalance in the network. We have also demonstrated that efficient download paths can be constructed by exploiting the underlying network topology information available at mesh routers. We have shown that such support from mesh routers enables peers to efficiently consume the limited network resources, reduces traffic interference in the network, while increases throughput at downloading peers. We have also shown that by utilizing idle resources at a large number of infrastructure nodes in a WMCN (bandwidth and storage), replicating content desired by the community at those infrastructure nodes enables peers to enjoy less content download times, while consuming less network resources.

Secondly, we have considered the problem of group communication in WMCNs, and argued for a ring-based routing topology for the multicast traffic distribution. We have demonstrated that support from mesh routers on the ring to the multicast application by playing the role of cluster heads, constructing cluster networks, and implementing a simple network coding technique results in significant performance enhancement for the group communication.

In this chapter, we conclude the dissertation by highlighting the contributions in each chapter and summarizing the results. Also, we present a number of possible directions for future research.

7.1 Summary of Contributions and Accomplished Work

- **Topology-aware schemes for efficient peer-to-peer content sharing in wireless mesh community networks**

  In Chapter 2, we have proposed P2PMesh: Topology-aware schemes for P2P content sharing in a WMCN. The proposed schemes benefit from the support of infrastructure nodes (mesh routers) to the P2P content sharing application in order to enable efficient P2P content lookup, efficient establishment of download paths, and efficient data dissemination in the WMCN. We have shown that P2PMesh mitigates the traffic load imbalance in the network, minimizes traffic interference, and increases throughput for downloading peers.

- **P2P content caching and replication strategies in wireless mesh community networks**

  We have further enhanced the performance of P2PMesh by proposing optimum replication strategies for P2P content at a number of mesh routers in a WMCN (Chapter 3). Our proposed strategy replicates a set of P2P contents at the participating mesh routers (helpers) such that the average access cost of all contents is minimized. We have shown that our replication strategy reduces the bandwidth and energy consumption in the network, while increases the network throughput.

We note here that our proposed strategies for P2PMesh and content replications can be extended to the case when the network is a mix of wireless and wired links. However, the problem is more difficult due to the complexity of the wired network topologies. Also, the proposed algorithms and schemes are specifically designed to account for the characteristics of the wireless medium and traffic interference. Therefore, the performance enhancements of our proposed solutions may not be the same.

- **Delicate design strategies for the P2P-with-helpers system that allocating optimum resources at helpers - based on prediction of the future evolution of content demand in the network**

  In Chapter 4 and Chapter 5, the performance of the proposed hybrid approaches for content sharing and media streaming in a community network (P2P-with-helpers system) have been evaluated. We have analytically characterized the evolution of interest in a P2P content in a community network. The derived model allows us to predict the future content demand in the community network. Based on the prediction of future demand, we developed design strategies for the P2P-with-helpers system that allocate optimum resources at helpers. We have shown that our design strategies for the P2P-with-helpers system have the potential to significantly reduce the average content download times, mitigate traffic load at the media content provider and CDNs, and reduce the energy and bandwidth consumption in the community network. One of the interesting observation was that, even when only few infrastructure nodes in the network play the role of helpers, significant saving in bandwidth and energy consumption in the network can be realized.

- **Analytical modelling for group communication in WMCNs**

In Chapters 6, the performance of group communication using a ring-based multicast routing topology over a WMCN has been evaluated. More specifically, we developed a model that allows us to derive upper/lower bounds on important performance metrics such as end-to-end delay, capacity of the routing topology, and energy consumption. Interestingly, we have shown that for a moderate multicast group size, a ring-based multicast routing topology coupled with our proposed algorithm for multicast traffic routing on the ring outperforms a tree multicast routing topology in terms of end-to-end delay and capacity. However, the performance enhancement of a ring-based multicast routing topology requires efficient construction of the ring. We, therefore, proposed algorithms for constructing efficient and interference-aware ring-based multicast routing topology that guarantees QoS for the multicast streaming with high confidence, in probabilistic sense.

7.2 Possible Future Research Directions

In the following, some interesting research directions for extending the work presented in this dissertation are introduced.

- **Incentives for helpers in a WMCN to participate in content sharing**

  Our proposed approaches for content sharing in a community network require that a number of infrastructure nodes in the network incorporate our proposed schemes and designs in order to participate in content sharing and distribution and play the role of helpers to the P2P application. In this dissertation, we only proposed incentives for a community network operator that owns the infrastructure nodes. We have shown that the reduction of bandwidth and energy consumption in the community network can encourage the network operator to incorporate our designs at the infrastructure nodes. However, a large number of WMCNs rely on nodes belonging to individuals (e.g., user home networks, WiFi APs) forwarding packets for one another. In this case, our approaches require individuals to contribute to the system, and voluntarily dedicate a fraction of their storage capacity and equip their infrastructure nodes with P2P aware devices to enable them to participate in content sharing and cache P2P content. However, a selfish user, in order to conserve its resources (e.g., limited bandwidth and storage resources), could decide not to participate in the content sharing, thus potentially leading to system collapse. Hence, we must discuss incentives for users of such networks to incorporate the required features at their infrastructure nodes. Moreover, mechanisms that do not require that end users (peers) to modify their peer-to-peer applications to benefit from support at infrastructure nodes must also be examined.

- **Heterogeneous peer capacities**

  In order to simplify our performance analysis and obtain useful insights about our proposed approaches for content sharing in community networks, we have assumed that all peers and all helpers have the same upload bandwidth (Chapter 4). An interesting future research direction is to relax this assumption and consider a more sophisticated P2P content distribution scheme where peers have different upload capacities.

- **Enabling secure content sharing in WMCNs**

  One of the disadvantages of sharing content in WMCNs includes propagation of malware. This is an open issue. Solutions are orthogonal to what this dissertation addresses and need to be addressed.

- **Measurements and experimental study for the evolution of content demand in community networks**

  We have analytically characterized the evolution of interest in a P2P content in a community network using fluid-flow approximation (Chapter 4). It is very important to perform an experimental study to validate the derived model and results, in order to make sure that the approximation used in the fluid-flow modelling does not impact the validity of the derived results.

- **Streaming of live media content**

  In Chapter 5, we only considered the case of media streaming of stored content (e.g., VoD-like service) in community networks. A possible future research work is to consider streaming of live media content. In live streaming, the media content is available only at one particular time and interested users have synchronous playback times; while media streaming of stored content allows users to watch any point of the video at any time. Media content in VoD-like service is often preloaded on the media content provider server and, hence, VoD offers more flexibility and convenience to users. In VoD service, although a large number of users may be watching the same video, they are asynchronous to each other and different users are watching different portions of the same video at any given moment. Moreover, some users may stay in the network after watching the content to serve other interested peers. The different characteristics of the live media streaming from the VoD media streaming require different designs for content distributions.

  In our analysis in Chapter 5, we assumed that peers do not leave the network for the entire life-time of the content. Investigating the impact of peer and helper departures is also an interesting future research direction.

- **Heuristics for the replica placement problem**

  It has been shown that the content placement problem is NP-Hard [112]. Hence, our optimum content replication strategy proposed in Chapter 3 for P2P content in a WMCN has been derived based on a set of assumptions about placement of replicas in the network. Thus, more accurate solutions for the P2P content replication problem that account for the replica placement problem must be considered in a future research.

- **Better bounds on performance metrics of group communication in a wireless mesh network**

  We have derived bounds on a number of important performance metrics for a ring-based routing topology over a WMCN (Chapter 6). However, the major drawback in our derived results is that we only account for interference between multicast traffic at nodes that are neighbours in the multicast routing topology. Although this drawback is accounted for in our proposed algorithm for the ring construction, an interesting future research work is to derive better bounds by considering geographic properties of nodes in the physical network (perhaps using unit disk graph) and finding an upper bound of non-interfered nodes.

Bibliography

[1] S. Bury and N.J.P Race, "Towards resilient community wireless mesh network," in *Proc. of the 2nd International Conference on Autonomous Infrastructure, Management and Security*, pp. 195–199, 2008.

[2] Champaign-Urbana Community Wireless Network (CUWiN), http://cuwireless.net/

[3] P. Antoniadis, B. L. Grand, A. Satsiou, L. Tassiulas, R. L. Aguiar, J. P. Barraca, and S. Sargento, "Community building over neighborhood wireless mesh networks," *IEEE Technology and Society Magazine, Special Issue on Potentials and Limits of Cooperation in Wireless Communications*, vol. 27, no. 1, pp. 48–56, 2008.

[4] Microsoft project, "Self organizing wireless mesh networks," white paper, available at: http://research.microsoft.com/en-us/projects/mesh/.

[5] J. P. Barraca, P. Fernandes, S. Sargento, and R. Rocha, "An architecture for community mesh networking," in *Proc. of IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications Conference (PIMRC)*, pp. 1–6, 2008.

[6] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," in *Proc. of the 11th Annual International Conference on Mobile computing and Networking (MobiCom)*, pp. 31–42, 2005.

[7] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, no. 4, pp. 445–487, March 2005.

[8] A. Banchs, N. Bayer, D. Chieng, A. O. de la, B Gloss, M. Kretschme, S. Murphyk, M. Natkaniec, and F. Zdarsky, “CARMEN: Delivering carrier grade services over wireless mesh networks,” in *Proc. of the IEEE PIMRC*, pp. 1–6, 2008.

[9] P. Pathak and R. Dutta, “A Survey of network design problems and joint design approaches in wireless mesh networks,” *IEEE Commun. Surveys Tuts*, vol 13, no. 3, pp. 396–428, 2010.

[10] J. D. Camp, E.W. Knightly, and W.S. Reed, “Developing and deploying multihop wireless networks for low-income communities,” *Journal of Urban Technology*, vol. 13, no. 129, pp. 1063–0732, 2006.

[11] F. Licandro and G. Schembra, “Wireless mesh networks to support video surveillance: architecture, protocol, and implementation issues,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, no. 1, pp. 21-34.

[12] M. Portmann and A. A. Pirzada, “Wireless mesh networks for public safety and crisis management applications,” *IEEE Internet Computing*, vol. 12, no. 1, pp. 18-25, 2008.

[13] K. Al Agha, M-H Bertin, T. Dang, A. Guitton, P. Minet, T. Val, and J. B. Viollet, “Which wireless technology for industrial wireless sensor networks? The development of OCARI technology,” *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4266–4278, 2009.

[14] K. C. Lan, Z. Wang, M. Hassan, T. Moors, R. Berriman, L. Libman, M. Ott, B. Landfeldt, and Z. Zaidi, ”Experiences in deploying a wireless mesh network testbed for traffic control,” *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 5, pp. 17–28, 2007.

[15] D. Wu, D. Gupta, and P. Mohapatra, “QuRiNet: A wide-area wireless mesh testbed

for research and experimental evaluations," *Ad Hoc Networks*, vol.9, no. 7, pp. 1221–1237, 2011.

[16] Seattle Wireless. *http://www.seattlewireless.net*.

[17] NYCwireless, *http://www.nycwireless.net*.

[18] OpenAirBoston, *http://www.openairboston.net*,

[19] FON community, *http://www.fon.com*.

[20] Athens wireless, *http://www.awmn.net*.

[21] Fred-e Zone, *http://www.westnet.ca*.

[22] , L. Katrina and H. Balakrishnan, "Measurement and analysis of real-world 802.11 mesh networks," in *Proc. of the 10th ACM SIGCOMM Conference on Internet Measurement*, pp. 123–136, 2010.

[23] Cisco Systems Inc., "Cisco Visual Networking Index: Forecast and Methodology, 20112016," *white paper*, 2012.

[24] T. Karagiannis, A. Broido, N. Brownlee, K.C. Claffy, and M. Faloutsos, "Is P2P dying or just hiding? [P2P traffic measurement]," in *Proc. of the IEEE GLOBECOM Conference*, pp. 1532–1538, 2004.

[25] C. Canali, M.E. Renda, P. Santi, and S. Burresi, "Enabling efficient peer-to-peer resource sharing in wireless mesh networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 3. pp. 333–347, 2010.

[26] P.A. Frangoudis, G.C. Polyzos, and V.P. Kemerlis, "Wireless community networks: An alternative approach for nomadic broadband network access", *IEEE Communications Magazine*, vol. 49, no. 5, 2011.

[27] F.A. Elianos, G. Plakia, P.A.Frangoudis, and G.C. Polyzos, “Structure and evolution of a large-scale Wireless Community Network,” in *Proc. of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks Workshops (WoWMoM)*, pp. 1–6, 2009.

[28] H. Xie, Y.R. Yang, A. Krishnamurthy, Y.G. Liu, and A. Silberschatz, ”P4P: Provider portal for applications,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 351–362, 2008.

[29] M. Adler, R.K. Sitaraman, and H. Venkataramani, “Algorithms for optimizing the bandwidth cost of content delivery,” *Comput. Netw.*, vol. 55, no. 18, pp. 4007–4020, 2011.

[30] Community grid, http://www.worldcommunitygrid.org.

[31] C. Dale and J. Liu, “apt-p2p: A Peer-to-Peer distribution system for software package releases and updates,” in *Proc. of the IEEE INFOCOM Conference*, pp. 864–872, 2009.

[32] D. S. Bernstein, Z. Feng, B. N. Levine, and S. Zilberstein, “Adaptive peer selection,” *Peer-to-Peer Systems II, Lecture Notes in Computer Science*, vol. 2735, pp. 237–246, 2003.

[33] M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron, “Topology-aware routing in structured peer-to-peer overlay networks,” *Technical Report MSR-TR-2002-82*, Microsoft Research, 2002.

[34] X. Jin, W.-P. Ken Yiu, S.-H. Gary Chan, and Y. Wang, “On maximizing tree bandwidth for topology-aware peer-to-peer streaming,”, in *IEEE Trans. in Multimedia*, vol. 9, no. 8, pp. 1580–1592, 2007.

[35] M.K Sbai, C. Barakat, J. Choi, A.A. Hamra, and T. Turletti, "Adapting BitTorrent to wireless ad hoc networks, in *Proc. of the AdHoc-Now Networks and Wireless Conference*, pp. 189–203, 2008.

[36] P.M. Ruiz, F.J. Galera, C. Jelger, and T. Noel, "Efficient multicast routing in wireless mesh networks connected to internet," in *Proc. of the ACM InterSense Conference*, ACM, 2006,

[37] X. Zhu, and B. Girod, "Distributed rate allocation for video streaming over wireless networks with heterogeneous link speeds," in *Proc. of ACM IWCMC*, pp. 296–301, 2007.

[38] F. Wang, J. Liu, and Y. Xiong, "On node stability and organization in Peer-to-Peer video streaming systems," in *IEEE Systems Journal*, vol. 5, no. 4, pp. 440–450, 2011.

[39] L. Yin and G. Cao, "Supporting cooperative caching in ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 1, pp. 77–89, 2006.

[40] B. Tang, H. Gupta, and S. R. Das, "Benefit-based data caching in ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 3, pp. 289–304, 2008.

[41] S.M. Das, H. Pucha, and Y.C. Hu, "Mitigating the gateway bottleneck via transparent cooperative caching in wireless mesh networks" *Ad Hoc Netw.*, vol. 5, no. 6, pp. 680–703, 2007.

[42] S. Jin and L. Wang, "Content and service replication strategies in multi-hop wireless mesh networks," in *Proc. of the ACM MSWiM*, pp. 79–86, 2005.

[43] E. Cohen and S. Shenker, "Replication strategies in unstructured peer-to-peer networks", *ACM SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 4, pp. 177–190, 2002.

[44] J. Wang, C. Yeo, V. Prabhakaran, and K. Ramchandran, "On the role of helpers in peer-to-peer file download systems: Design, analysis and simulation," in *Proc. of the IPTPS*, 2007.

[45] D. Qiu and R. Srikant, "Modeling and performance analysis of BitTorrent-like peer-to-peer networks," in *Proc. of ACM SIGCOMM Conference*, pp. 367–377, 2004.

[46] J. Wong, "Enhancing collaborative content delivery with helpers," Master's thesis, University of British Columbia, Nov 2004.

[47] H. Zhang, J. Wang, M. Chen, and K. Ramchandran, "Scaling Peer-to-Peer Video-on-Demand systems using helpers," in *Proc. of the IEEE International Conference on Image Processing*, pp. 3053–3056, 2009.

[48] T. Karagiannis, A. Broido, N. Brownlee, K.C. Claffy, and M. Faloutsos, "Improving the streaming capacity in P2P VoD systems with helpers," in *Proc. of the ICME 2009 Conference*, pp. 790–793, 2009.

[49] J. Wang and K.Ramchandran, "Enhancing peer-to-peer live multicast quality using helpers," in *Proc. of the 15th IEEE International Conference on Image Processing (ICIP 2008)*, pp. 2300–2303, 2008.

[50] Y. Liu, Y. Guo, and C. Liang, "A survey on peer-to-peer video streaming systems," *Peer-to-Peer Networking and Applications*, vol. 18, no. 1, pp. 18–28, 2008.

[51] X. Xiang, X. Wang, and Y. Yang, "Supporting efficient and scalable multicasting over mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 4, pp. 544–559, 2011.

[52] C-Y Chiu, E. Wu, and G-H Chen, "A reliable and efficient MAC layer broadcast (multicast) protocol for mobile ad hoc networks," in *Proc. of the IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 5, pp. 2802–2807, 2004.

[53] S. Roy, D. Koutsonikolas, S. Das, and Y. Hu, "High-throughput multicast routing metrics in wireless mesh networks," in *Proc. of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp. 48–53, 2006.

[54] I. Stoica, R. Morris, D. Liben-Nowell, D.R. Karger, M.F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup protocol for internet applications", *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 17–32, 2003.

[55] A. Mawji and H. S. Hassanein, "Optimal path selection for file downloading in P2P overlay networks on MANETs," in *Proc. of the IEEE ISCC*, pp. 1133–1138, 2010.

[56] O. Abboud, A. Kovacevic, K. Graffi, K. Pussep, and R. Steinmetz, "Underlay awareness in P2P systems: Techniques and challenges," in *Proc. of IEEE IPDPS Conference*, pp.1–8, 2009.

[57] Y. Bejerano, S.J. Han, and A. Kumar, "Efficient load-balancing routing for wireless mesh networks," *Comput. Netw.*, vol. 51, no. 10, pp. 2450–2466, 2007.

[58] D. Tian and V. C.M. Leung, "Analysis of broadcasting delays in vehicular ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 11, no. 11, pp. 1433–1445, 2011.

[59] S. Surana, B. Godfrey, K. Lakshminarayanan, R. Karp, and I. Stoica, "Load balancing in dynamic structured peer-to-peer systems," *Perform. Eval.*, vol. 63, no. 3, pp. 217–240, 2006.

[60] J. Broch, D.A. Maltz, D.B. Johnson, Y-C Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pp. 85–97, 1998.

[61] A. A Pirzada, M. Portmann, and J. Indulska, “Evaluation of multi-radio extensions to AODV for wireless mesh networks,” in *Proc. of the ACM MOBIWAC*, pp. 45–51, 2006.

[62] C. Canali, M.E. Renda, and P. Santi, “Evaluating load balancing in peer-to-peer resource sharing algorithms for wireless mesh networks” in *Proc. of the IEEE MeshTeck*, pp. 603–609, 2008.

[63] H. Hassanein and A. Zhou, “ Load-aware destination-controlled routing for MANETs,” *Computer Communications*, vol. 26, no. 14, pp. 1551–1559, 2003.

[64] S. Waharte, B. Ishibashi, R. Boutaba, and D. Meddour, “Interference-Aware Routing Metric for Improved Load Balancing in Wireless Mesh Networks,” in *Proc. of the IEEE ICC*, pp. 2979–2983, 2008.

[65] M. Ripeanu, I. Foster, and A. Iamnitchi, “Mapping the Gnutella network: Properties of large-scale peer-to-peer systems and implications for system design,” *IEEE Internet Computing Journal*, vol 6, no. 1, pp. 50–57, 2002.

[66] A. Nandan, S. Das, G. Pau, M. Gerla, and M. Sanadidi, “Co-operative downloading in vehicular ad-hoc wireless networks,” in *Proc. of the IEEE WONS*, pp. 32–41, 2005.

[67] Y. Zhu and Y. Hu, “Efficient, proximity-aware load balancing for DHT-based P2P systems,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 4, pp. 349–361, 2005.

[68] H. Xie, Y.R. Yang, A. Krishnamurthy, Y.G. Liu, and A. Silberschatz, “P4P: provider portal for applications,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 351–362, 2008

[69] J. Peterson, V. Gurbani, E. Marocco, et al., “ALTO Working Group Charter,” http://www.ietf.org/html.charters/alto-charter.html.

[70] J. Seedorf, S. Kiesel, and M. Stiemerling, “Traffic localization for P2P-applications: The ALTO approach,” in *Proc. of the IEEE Ninth International Conference on Peer-to-Peer Computing*, pp. 171–177, 2009.

[71] H. Wang and J. Liu, “Exploring Peer-to-Peer Locality in Multiple Torrent Environment,” in *IEEE Transactions on Parallel and Distributed Systems*, vol 23, no. 7, pp. 1216–1226, 2012.

[72] M. Castro, P. Druschel, Y.C. Hu, and A. Rowstron, “Exploiting network proximity in distributed hash tables,” in *Proc. of FuDiCo*, pp. 52-55, 2002.

[73] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica, “The impact of DHT routing geometry on resilience and proximity,” in *Proc. of the ACM SIGCOMM Conference*, pp. 381–394, 2003.

[74] P. Ruiz and A. Gomez-Skarmeta, “Approximating optimal multicast trees in wireless multihop networks,” in *Proc. the IEEE ISCC*, pp. 686–691, 2005.

[75] Z. Wang and J. Crowcroft, “Quality-of-service routing for supporting multimedia applications,” *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 7, pp. 1228–1234, 1996.

[76] P. Maymounkov and D. Mazi‘eres, “Kademlia: A Peer-to-peer information system based on the XOR metric,” in *Proc. of the IPTPS*, Springer-Verlag, pp. 2002.

[77] D.B Johnson, D.A. Maltz, and J. Broch, “DSR: the dynamic source routing protocol for multi-hop wireless ad hoc networks,” *Ad Hoc Networking*, Chapter 5, 139–172, Addison-Wesley, 2001.

[78] L. Ji and M. Corson, “Differential destination multicast - A MANET multicast routing protocol for small groups,” in *Proc. of the IEEE INFOCOM Conference*, vol 2, pp. 1192–1201, 2001.

[79] C-K Chiang and C.T. King, "Source routing for overlay multicast in wireless ad-hoc and sensor networks," in *Proc. of the IEEE ICPPW*, 2007.

[80] C. Cramer and T. Fuhrmann, "Performance evaluation of chord in mobile ad hoc networks," in *Proc. of the ACM MobiShare Conference*, pp. 48–53, 2006.

[81] I. Baumgart, B. Heep, and S. Krause, "Oversim: A flexible overlay network simulation framework," in *Proc. of the IEEE GI '07*, pp. 79–84, 2007.

[82] M. Castro, E. Villanueva, I. Ruiz, S. Sargento, and A. Kassler, "Performance evaluation of structured P2P over wireless multi-hop networks," in *Proc. of The Sixth International Conference on Sensor Technologies and Applications (SENSORCOMM)*, pp. 796–801, 2008.

[83] Opnet. *http://www.opnet.com/*.

[84] P. Felber and E.W. Biersack, "Cooperative content distribution: Scalability through self-organization," in *Self-star Properties in Complex Information Systems*, Lecture Notes in Computer Science, vol. 3460, Springer, 2005.

[85] G. Tyson, A. Mauthe, S. Kaune, M. Mu, and T. Plagemann, "Corelli: A peer-to-peer dynamic replication service for supporting latency-dependent content in community networks," *Proc. of the 16th Multimedia Computing and Networking Conference (MMCN)*, 2009.

[86] L. Xiao, and X. Zhang, and A. Andrzejak, and S. Chen, "Building a large and efficient hybrid Peer-to-Peer Internet caching system," *IEEE Trans. on Knowl. and Data Eng.*, vol. 16, no. 6, pp. 1041–4347, 2004.

[87] Y. Mao, Z. Zhu and W. Shi, "Peer-to-peer web caching: hype or reality?," in *Proc. of the IEEE Conference on Parallel and Distributed Systems (ICPADS)*, pp. 171–178, 2004.

[88] W. Shi, K. Shah, Y. Mao, and V. Chaudhary, "Tuxedo: A peer-to-peer caching system," in *Proc. of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*, pp. 981-987, June 2003.

[89] A. Weirzbicki, N. Leibowitz, M. Rapeanu, and R. Wozniak, "Cache replacement policies revisited: The case of P2P traffic," in *Proc. of the International Workshop on Global and Peer-to-Peer Computing (GP2P)*, pp. 182–189, 2004.

[90] G. Shen, Y. Wang, Y. Xiong, B. Zhao, and Z. Zhang, "HPTP: Relieving the tension between ISPs and P2P," in *Proc. of the International Workshop on Peer-to-Peer Systems (IPTPS)*, 2007.

[91] M. Hefeeda and O. Saleh, "Traffic modeling and proportional partial caching for peer-to-peer systems," *IEEE/ACM Trans. on Networking*, vol. 16, no. 6, pp. 1447–1460, 2008.

[92] O. Saleh and M. Hefeeda, "Modelling and caching of Peer-to-Peer traffic," in *Proc. of the IEEE International Conference on Network Protocols*, pp. 249–258, 2006.

[93] F. R. Dogar, A. Phanishayee, H. Pucha, O. Ruwase, and D.G. Andersen, "Ditto: A system for opportunistic caching in multi-hop wireless networks," in *Proc. of the 14th ACM International Conference on Mobile computing and Networking (MobiCom)*, pp. 279–290, ACM, 2008.

[94] J. Zhao, P. Zhang, G. Cao, and C. R. Das, "Cooperative caching in wireless P2P networks: Design, implementation, and evaluation" *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 7, pp. 229–241, 2010.

[95] M. Denko, T. Nkwe, and M. Obaidat, "Efficient cooperative caching with improved performance in wireless mesh networks," in *Proc. of the IEEE ICC*, pp. 1–5, 2010.

[96] M. Fiore, F. Mininni, C. Casetti, and C. F. Chiasserini, "To cache or not to cache?," in *Proc. of IEEE INFOCOM Conference*, pp. 235-243, 2009.

[97] "OverCache MSP Home Page," http://www.oversi.com/products/pvercache-msp.

[98] "PeerApp UltraBand Home Page," http://www.peerapp.com/products-ultraband.aspx.

[99] M. Hefeeda and B. Noorizadeh, "On the Benefits of Cooperative Proxy Caching for Peer-to-Peer Traffic," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 7, pp. 998–1010, 2010.

[100] K. Xu, M. Zhang, J. Liu, Z. Qin, and M. Ye, "Proxy caching for peer-to-peer live streaming," in *Computer Networks*, vol. 54, no. 7, pp. 1229–1241, 2010.

[101] M. Hefeeda, C-H Hsu, and K. Mokhtarian, "Design and evaluation of a proxy cache for Peer-to-Peer traffic, *IEEE Transactions on Computers*, vol. 60, no. 7, pp. 964–977, 2011.

[102] Tolga Balci, "Detecting P2P activity in your network," white paper, available at: http://www.brighthub.com/computing/smb-security/articles/48875.aspx.

[103] S. Sen, O. Spatcheck, and D. Wang, "Accurate, scalable in-network identification of P2P traffic using application signatures," in *Proc. of the International Word Wide Web Conference*, pp. 512–521, 2004.

[104] A. Spognardi, A. Lucarelli, and R. Di Pietro, "A methodology for P2P traffic classification," in *Proc. of the IEEE International Workshop on Hot Topics in Peer-to-Peer Systems*, pp. 52–61, 2005.

[105] T. Karagiannis, A. Broido, M. Faloutsos, and Kc. claffy, "Transport layer identification of P2P traffic," in *Proc. of the 4th ACM SIGCOMM Conference on Internet Measurement*, pp. 121–134, 2004.

[106] R. Asorey-Cacheda, F. J. Gonzalez-Castanio, M. Crespo-Alonso, L. Caviglione, and F Davoli, "A Packet sniffing and synchronization technique to boost P2P satellite networks," in *Proc. IEEE HOT-P2P Conference*, pp. 28–32, 2005.

[107] V. Aggarwal, A. Feldmann, and C. Scheideler, "Can ISPS and P2P users cooperate for improved performance?," *ACM SIGCOMM Computer Communication Review Archive*, vol. 37, no. 3, pp. 29–40, 2007.

[108] BT partners with Motorola for wireless city initiatives, *Internet Business*, http://www.internetbusiness.co.uk/2006/12/05/.

[109] http://pdoscsailmitedu/roofnet/design/.

[110] http://wwwberlinfreifunknet/.

[111] J. Kangasharju, K.W. Ross, D.A. Turner, and T. Darmstadt, "Adaptive content management in structured P2P communities," in *Proc. of the International Conference on Scalable Information Systems (INFOSCALE)*, Hong Kong, 2006.

[112] K. Jain and V.V. Vazirani, "Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation," *Journal of the ACM*, vol. 48, no. 2, pp. 274–296, 2001.

[113] B.J. Ko and D. Rubenstein, "Distributed self-stabilizing placement of replicated resources in emerging networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 3, pp. 476–487, June 2005.

[114] P. Nuggehalli, V. Srinivasan, and C.F. Chiasserini, "Energy-efficient caching strategies in ad hoc wireless networks," in *Proc. of the 4th ACM International Symposium on Mobile Ad-hoc Networking & computing (MobiHoc)*, pp. 25–34, 2003.

[115] L. Massouli and M. Vojnovic, "Coupon replication systems," in *Proc. of ACM SIGMETRICS*, pp. 2–13, 2005.

[116] S. Sanghavi, B. Hajek, and L. Massoulie, "Gossiping with multiple messages," in *Proc. of the IEEE INFOCOM Conference*, pp. 2135–2143, 2007.

[117] B. Hajek and J. Zhu, "The missing piece syndrome in peer-to-peer communication," in *Proc. of the IEEE International Symposium on Information Theory (ISIT)*, pp. 1748–1752, 2010.

[118] M. Zulhasnine, C. Huang, and A. Srinivasan, "Favorable peer supported throughput optimization in wireless mesh network," in *Proc. of the IEEE GLOBECOM Conference*, pp. 1–5, 2010.

[119] K. Gummadi, R. Dunn, S. Saroiu, S. Gribble, H. Levy, and J. Zahorjan, "Measurement, modeling, and analysis of a peer-to-peer file-sharing workload," in *Proc, of the 19th ACM Symposium on Operating Systems Principles (SOSP)*, pp. 314–329, 2003.

[120] S. Al-Kiswany, M. Ripeanu, A. Iamnitchi, and S. Vazhkudai, "Beyond music sharing: An evaluation of peer-to-peer data dissemination techniques in large scientific collaborations," *Journal of Grid Computing*, vol. 7, no. 1, pp. 91–114, 2009.

[121] M. Izal, G. Urvoy-Keller, E. Biersack, P. Felber, A. Al Hamra, and L. Garces-Erice, "Dissecting BitTorrent: Five months in a torrent's lifetime," in *Passive and Active Network Measurement*, Lecture Notes in Computer Science, vol. 3015, pp. 1–11, 2004.

[122] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, "A performance study of BitTorrent-like peer-to-peer systems," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 1, pp. 155–169, 2007.

[123] Y. Yue, C. Lin, and Z. Tan, "Analyzing the performance and fairness of Bittorrent-like networks using a general fluid model," *Computer Communications*, vol. 29, no. 18, pp. 3946–3956, 2006.

[124] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, "Measurements, analysis, and modeling of BitTorrent-like systems," in *Proc. of the ACM/SIGCOMM (IMC '05)*, pp. 3548, 2005.

[125] N. Andrade, M. Mowbray, A. Lima, G. Wagner, and M. Ripeanu, "Influences on cooperation in BitTorrent communities," in *Proc. of the ACM SIGCOMM Workshop on Economics of peer-to-peer systems (P2PECON)*, pp. 111–115, 2005.

[126] Y. Tian, D. Wu, and K. W. Ng, "Modeling, analysis and improvement for BitTorrent-like file sharing networks,", in *Proc. of the IEEE INFOCOM Conference*, pp. 1–11, 2006.

[127] X. Yang and G. de Veciana, "Performance of peer-to-peer networks: Service capacity and role of resource sharing policies," *Performance Evaluation*, vol. 63, no. 3, pp. 175-194, 2006.

[128] J. Pouwelse P. Garbacki, D. Epema, and H. Sips, "The BitTorrent P2P file-sharing system: Measurements and analysis," in *In International Workshop on Peer-to-Peer Systems (IPTPS)*, 2005.

[129] Z. Chen, C. Lin, H. Yin, and B. Li, "On the server placement problem of P2P live media streaming system," in *Proc. of the 9th Pacific Rim Conference on Advances in Multimedia Information Processing*, pp. 178–187, Springer-Verlag, 2008.

[130] S.H. Kwok, K.Y. Chan, and Y.M. Cheung, "A server-mediated peer-to-peer system," *ACM SIGecom Exch*, vol. 5, no. 3, pp. 38–47, 2005.

[131] M. Hefeeda, B.K. Bhargava, and D.K. Yau, "A hybrid architecture for cost-effective on-demand media streaming," *Computer Networks*, vol. 44, no. 3, pp. 353–382, 2004.

[132] D. Xu, S. Kulkarni, C. Rosenberg, and H.K. Chai, "Analysis of a CDN-P2P hybrid

architecture for cost-effective streaming media distribution," *Multimedia Systems*, vol. 11, no. 4, pp. 383–399, 2006.

[133] S. Shakkottai and R. Johari, "Demand-aware content distribution on the Internet," *IEEE/ACM Transactions on Networking*, vol. 18, no. 2, pp. 476–489, 2010.

[134] A. Iamnitchi, M. Ripeanu, E. Santos-Neto, and I. Foster, "The small world of file sharing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 1120–1134, 2011.

[135] F.M. Bass, "A new product growth for model consumer durables," *Management Science*, vol. 15, no. 5, pp. 215–227, 1969.

[136] J. Lorincz, A. Capone, and M. Bogarelli, "Energy savings in wireless access networks through optimized network management," in *Proc. of ISWPC Conferecne*, pp. 449–454, May 2010.

[137] L. Chiaraviglio, M. Mellia, and F. Neri, "Energy-aware backbone networks: A case study," in *Proc. of the ICC Communications Workshops*, pp. 1–5, June 2009.

[138] A. Capone, F. Malandra, and B. Sanso, "Energy savings in wireless mesh networks in a time-variable context," *Mobile Network Applications (MONET)*, vol. 17, no. 2, pp. 298–311, 2012.

[139] S. Boiardi, A. Capone, and B. Sanso, "Energy-Aware Planning and Management of Wireless Mesh Networks," in *Proc. of IEEE Globecom Conference*, pp. 3073–3079, 2012.

[140] Cisco Systems Inc., "Data center virtualization and orchestration: Business and financial justification," *white paper*, 2007.

[141] Cisco Systems Inc., "Cisco visual networking index: Forecast and methodology, 2010-2015," *white paper*, 2010.

[142] C. Xu and J. Liu, "Exploring interest correlation for peer-to-peer socialized video sharing," in *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 8, no. 1, pp. 5:1–5:20, 2012.

[143] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "Towards predictable datacenter networks," in *Proc. of the ACM SIGCOMM Conference*, pp. 242–253, 2011.

[144] T. T. Do, K. A. Hua, and M. A. Tantaoui, "Robust video-on-demand streaming in peer-to-peer environments," *Computer Communications*, vol. 31, no. 3, pp. 506–519, 2008.

[145] S. Islam, J. Keung, K. Lee, and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," *Future Generation Computer Systems*, vol. 28, no. 1, pp. 155–162, 2012.

[146] Y. C. Lee and Y. Albert, "Zomaya: Rescheduling for reliable job completion with the support of clouds," *Future Generation Computer Systems*, vol. 26, no. 8, 1192–1199, 2010.

[147] D. Niu, Z. Liu, B. Li, and S. Zhao, "Demand forecast and performance prediction in peer-assisted on-demand streaming systems," in *Proc. of the IEEE INFOCOM Conference*, pp 421–425, 2011.

[148] S. Peichang, W. Huaimin, Y. Gang, L. Fengshun, and W. Tianzuo, Prediction-based federated management of multi-scale resources in cloud," *AISS: Advances in Information Sciences and Service Sciences*, vol. 4, no. 6, pp. 324–334, 2012.

[149] G. Gursun, M. Crovella, and I. Matta, "Describing and forecasting video access patterns," in *Proc. IEEE INFOCOM Conference*, pp. 16–20, 2011.

[150] D. Niu, H. Xu, B. Li, and S. Zhao, "Quality-assured cloud bandwidth auto-scaling

for video-on-demand applications," in *Proc. of IEEE INFOCOM Conference*, pp. 421–425, 2012.

[151] Z. Wang, L. Sun, X. Chen, W. Zhu, J. Liu, M. Chen, and S. Yang, "Propagation-based social-aware replication for social video contents," in *Proc. of the 20th ACM International Conference on Multimedia* pp. 29–38, 2012.

[152] N. Laoutaris, P. Rodriguez, and L. Massoulie, "ECHOS: Edge capacity hosting overlays of nano data centers," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 1, pp. 51–54, Jan. 2008.

[153] M. Cha, H. Kwak, P. Rodriguez, Y. yeol Ahn, and S. Moon, "I tube, you tube, everybody tubes: Analyzing the worlds largest user generated content video system," in *Proc. of the 5th ACM/USENIX Internet Measurement Conference (IMC)*, 2007.

[154] M. J. Freedman, E. Freudenthal, and D. Mazières, "Democratizing content publication with coral," in *Proc. of the 1st Conference on Symposium on Networked Systems Design and Implementation (NSDI)*, vol. 1, pp. 18–18, USENIX Association, 2004.

[155] X. Zhu, T. Schierl, T. Wiegand, and B. Girod, "Video multicast over wireless mesh networks with scalable video coding (SVC)," in *Proc. of the Visual Communications and Image Processing*, vol. 6822, 2008.

[156] F. Xie, K. A. Hua, and N. Jiang, "A cross-layer framework for video-on-demand service in multi-hop WiMax mesh networks," *Computer Communications*, vol. 31, no. 8, pp. 1615–1626, 2008.

[157] M. Portmann and A. Pirzada, "Wireless mesh networks for public safety and crisis management applications," *IEEE Internet Computing*, vol. 12, no. 1, pp. 18–25, 2008.

[158] A. Yarali, B. Ahsant, and S. Rahman, "Wireless mesh networking: A key solution for

emergency and rural applications," in *Proc. of International Conference on Advances in Mesh Networks*, pp. 143–149, 2009.

[159] X. Liu, L. Junahai, Y. Danxia and F. Mingyu, "A survey of multicast routing protocols for mobile ad-hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, pp. 78–91, 2009.

[160] T. Kunz and E. Cheng, "On-demand multicasting in ad-hoc networks: comparing AODV and ODMRP," in *Proc. of the IEEE ICSCS*, pp. 453–454, 2002.

[161] M. Masoudifar, "A review and performance comparison of QoS multicast routing protocols for MANETs," *Ad Hoc Netw*, vol. 7, no. 1, pp. 1150–1155, 2009.

[162] R. Biradar and S. Manvi, "Ring mesh based multicast routing scheme in MANET using bandwidth delay product," *Wireless Personal Communications*, vol. 38, no. 1, pp. 1–30, 2011.

[163] Y. Boujelben, A. Girard, and G-C. Gregoire, "A sequential algorithm for constructing delay-constrained multirings for multipoint-to-multipoint communications," *Telecommunication Systems*, vol. 31, no. 1, pp. 43–59, 2006.

[164] J. Wang and W. Yurcik, "A survey and comparison of multi-ring techniques for scalable battlespace group communications," in *Proc. of SPIE Conference*, 2005.

[165] D. Koutsonikolas and Y. C. Hu, "Exploring the design space of reliable multicast protocols for wireless mesh networks," *Ad Hoc Networks*, vol. 7, no. 5, pp. 932–954, 2009.

[166] X. Wang, L. Wang, Y. Wang, Y. Zhang, and A. Yamada, "Supporting MAC layer multicast in IEEE 802.11n: Issues and solutions," in *Proc. of the IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2009.

[167] T. Kunz and T. Kunz, "Reliable multicasting in MANETs," in *Contractor Report*, Communications Research Centre Ottawa Canada, 2003.

[168] S. Jain and S. Das, "MAC layer multicast in wireless multihop networks," in *Proc. of First International Conference on Communication System Software and Middleware (Comsware)*, pp. 1–10, 2006.

[169] C. Campolo, A. Molinaro, C. Casetti, and C.-F. Chiasserini, "An 802.11-based MAC protocol for reliable multicast in multihop networks," in *Proc. of IEEE 69th Vehicular Technology Conference*, pp. 1–5, 2009.

[170] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas "Protocol independent multicast-sparse mode (PIM-SM): Protocol specification (revised)," *www.ietf.org*, August 2006.

[171] O. S. Badarneh and M. Kadoch, "Multicast routing protocols in mobile ad hoc networks: a comparative survey and taxonomy," *EURASIP J. Wirel. Commun. Netw.*, vol. 2009, pp. 26–42, January 2009.

[172] P. Jacquet, P. Muhlethaler, T. Clausen, and A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for Ad hoc networks," in *Proc. of IEEE INMIC*, pp. 62–68. August, 2002.

[173] C. Cordeiro, H. Gossain, and D. Agrawal, "Multicast over wireless mobile ad hoc networks: Present and future directions," *IEEE Network*, vol, 17, no. 1, pp. 52–59, 2003.

[174] C. A.S Oliveira and P. M. Pardalos, "A survey of combinatorial optimization problems in multicast routing," *Computers and Operation Research*, vol. 32, no. 8, pp. 1953–1981, 2005.

[175] J. Wang, M. Song, and Y. Zhao, "Interference-aware multicast in wireless mesh networks with directional antennas," in *Proc. of the IEEE Fifth International Conference on Networking, Architecture and Storage (NAS)*, pp. 171–180, 2010.

[176] I. Tinnirello and G. Bianchi, "Interference estimation in IEEE 802.11 networks," *IEEE Control Systems Magazine*, vol. 30, no. 2, pp. 30–43, 2010.

[177] C. Rosenberg, A. Iyer and A. Karnik, "What is the right model for wireless channel interference," *IEEE Transactions on Wireless Communication*, vol. 8, no. 5, pp. 2662–2671, May 2009.

[178] S. Guo, O. W.W Yang, and V. Leung, "Tree-based distributed multicast algorithms for directional communications and lifetime optimization in wireless ad hoc networks," *EURASIP J. Wirel. Commun. Netw.*, vol. 2007, no. 1, 2007.

[179] P. Gupta, and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[180] X-Y Li, S-J Tang, and O. Frieder, "Multicast capacity for large scale wireless ad hoc networks," in *Proc. of the 13th Annual ACM International Conference on Mobile Computing and Networking*, pp. 266–277, 2007.

[181] S. Shakkottai, L, Xin, and R. Srikant, "The multicast capacity of large multihop wireless networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 6, pp. 1691–1700, 2010.

[182] S. Murthy, A. Goswami, and A. Sen, "Interference-aware multicasting in wireless mesh networks," in *Proc. of the 6th International IFIP-TC6 Conference on Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, Springer-Verlag, pp. 299–310, 2007.

[183] D. Koutsonikolas, S. M. Das, and Y. Charlie Hu, "An interference-aware fair scheduling for multicast in wireless mesh networks," *J. Parallel Distrib. Comput.*, vol. 68, no. 3, pp. 372–386, 2008.

[184] H. L. Nguyen and U. T. Nguyen, "Minimum interference channel assignment for multicast in multi-radio wireless mesh networks," in *Proc. of International Conference on Wireless Communications and Mobile Computing (IWCMC)*, pp. 626–631, 2008.

[185] G. Zeng, B. Wang, Y. Ding, L. Xiao, and M. Mutka, "Efficient multicast algorithms for multichannel wireless mesh networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 1, pp. 86–99, 2010.

[186] H. Nicanfar, M. Jaseemuddin, N-W Ma, and V. C.M. Leung, "GORby: Interference-Aware Multicast for Wireless Mesh Network," in *Proc. of the IEEE ICC SCPA workshop*, 2012.

[187] A. Alasaad, S. Gopalakrishnan, and V. C.M Leung, "An architecture with QoS support for application layer multicasting over wireless mesh networks," in *Proc. of the IEEE PIMRC*, Tokyo, Japan, 2009.

[188] K. Xu, M. Gerla, and S. Bae, "How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks," in *Proc. of the IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 1, pp. 72–76, 2002.

[189] H. Tebbe and A. Kassler, "QAMNet: Providing quality of service to ad-hoc multicast enabled networks," in *Proc. of the 1st Int. Symp. on Wireless Pervasive Computing*, pp. 1–5, 2006.

[190] L. Layuan, and L. Chunlina, "A QoS multicast routing protocol for clustering mobile Ad hoc networks," *Computer Communications*, vol. 30, no. 7, pp. 1641–1654, 2007.

[191] J. Ng, C.P. Low, and H.S. Teo, "On-demand QoS multicast routing and reservation protocol for MANETs," in *Proc. of the IEEE PIMRC*, pp. 2504–2508, 2004.

[192] A. Sobeih, J. Wang, and W. Yurcik, "Vring: A case for building application layer multicast rings (rather than trees)," in *Proc. of the IEEE MASCOTS*, pp. 437–446, 2004.

[193] B. Griese, A. Brinkmann, and M. Porrmann, "Selfs: A real-time protocol for virtual ring topologies," in *Proc. of IEEE IPDPS*, pp. 1–8, 2008.

[194] G. Hueter, "Solution of the traveling salesman problem with an adaptive ring," in *Proc. of IEEE Int. Conf. on Neural Networks*, pp. 85–92, 1988.

[195] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, 'Interference-aware channel assignment in multi-radio wireless mesh networks," in *Proc. of the 25th IEEE INFOCOM Conference*, pp. 1–12, 2006.

[196] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proc. of the ACM 10th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 114–128, 2004.

[197] A. Subramanian, H. Gupta, S. Das, and J. Cao, "Minimum interference channel assignment in multi-radio wireless mesh networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 12, pp. 1459–1473, 2008.

[198] J. Lee, S-J Lee, W. Kim, D. Jo, T. Kwon, and Y. Choi, "RSS-based carrier sensing and interference estimation in 802.11 wireless networks," in *Proc. of the IEEE 4th Annual Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pp. 491–500, June 2007.

[199] J. Padhye, S. Agarwal, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill, "Estima-

tion of link interference in static multi-hop wireless networks," in *Proc. of the 5th ACM SIGCOMM Conference on Internet Measurement (IMC)*, pp. 28–28, 2005.

Appendix A

Optimum Replication Strategy for P2P Content in WMCNs with Heterogeneous Peer Capacities

In Section 3.4, we assumed that a downloading peer in a WMCN retrieves segments of file i from all regular file providing peers at the same rate. Since this assumption is restrictive and impractical (specially when considering varied wireless channel condition and different number of wireless hops on download paths between a downloading peer and file provides), we relax this assumption in this appendix and consider a general and more feasible case.

We consider a set of classes (S) for content i providing peers residing in the retrieval-area according to the average rate at which a downloading peer retrieves segments of content i from those peers. In particular, class $j \in S$ contains content providing peers from which a downloading peer retrieves segments of content i at rate μ_{p_j}. Define $P_{i,j}$ as the number of peers which possess replicas of file i, residing in the retrieval-area of a downloading peer, and belonging to class j. Let η_j be ratio of the number of peers belonging to class j to the number of all peers in the WMCN during time period T.

Define ϵ_h as the average fraction of segments of file i that a downloading peer retrieves from a replica stored at helpers, and can be computed as follows.

$$\epsilon_h = \frac{\mu_{h_i}}{\mu_{h_i} + \sum_{j=1}^{S} \mu_{p_j} \cdot P_{i,j}},$$

where μ_{h_i} is the upload bandwidth that is allocated for file i at helpers.

$\epsilon_{p_{i,j}}$ is defined as the average fraction of segments of file i that a downloading peer retrieves from peers in class j, and can be computed as follows.

$$\epsilon_{p_{i,j}} = \frac{\mu_{p_j} \cdot P_{i,j}}{\mu_{h_i} + \sum_{j=1}^{S} \mu_{p_j} \cdot P_{i,j}},$$

Similar to Eq. (3.2), we can compute the average cost of accessing file i by any peer in the WMCN as function of number of replicas of file i at helpers (r_i) as follows.

$$c_i = \epsilon_h \cdot \frac{d}{\sqrt{r_i}} + \frac{1}{2} \sum_{j=1}^{S} \epsilon_{p_{i,j}} \cdot \frac{d}{\sqrt{r_i}}.$$

Hence,

$$\begin{aligned} c_i &= \frac{d}{\sqrt{r_i}} \cdot \frac{\mu_{h_i}}{\mu_{h_i} + \sum_{j=1}^{S} \mu_{p_j} \cdot P_{i,j}} \\ &+ \frac{1}{2} \frac{d}{\sqrt{r_i}} \cdot \sum_{j=1}^{S} \frac{\mu_{p_j} \cdot P_{i,j}}{\mu_{h_{i+\sum_{j=1}^{S} \mu_{p_j} \cdot P_{i,j}}}}. \end{aligned}$$

As we have discussed earlier, we can approximately write $P_{i,j}$ as function of r_i as follows $P_{i,j} = \frac{\eta_j \times P_i}{4 \times r_i}$. Recall that P_i is the average number of peers in the WMCN that possess replicas of file i, and can be approximately computed as $P_i \approx k \times p_i \times M \times r_i$ (Section 3.4).

Let N be the total number of P2P files in the network, and every file is with popularity p_i. Thus, the average total access cost for all P2P files in the WMCN during time period T can be computed as $c = \sum_{i=1}^{N} p_i \times c_i$. Hence, we can write

$$c = \sum_{i=1}^{N} \frac{d \cdot p_i}{\sqrt{r_i}} \cdot \left(\frac{\mu_{h_i} + \frac{p_i}{2} \sum_{j=1}^{S} \frac{\mu_{p_j} \cdot \eta_j \cdot k \cdot M}{4}}{\mu_{h_i} + p_i \sum_{j=1}^{S} \frac{\mu_{p_j} \cdot \eta_j \cdot k \cdot M}{4}} \right).$$

After omitting negligible terms we can write

$$c = \sum_{i=1}^{N} \frac{p_i}{\sqrt{r_i}} \cdot \frac{d}{1 + k \cdot \frac{\sum_{j=1}^{S} \mu_{p_j} \cdot \eta_j}{\mu_{h_i}} \cdot p_i \cdot M}.$$

We can formulate the optimization problem as in Section 3.4, and the solution can be written as follows

$$r_i \propto \left(\frac{p_i}{1 + k \cdot \frac{\sum_{j=1}^{S} \mu_{p_j} \cdot \eta_j}{\mu_{h_i}} \cdot p_i \cdot M} \right)^{\frac{2}{3}}.$$

Appendix B

Group Communication on Different Multicast Routing Topologies: Performance Comparisons

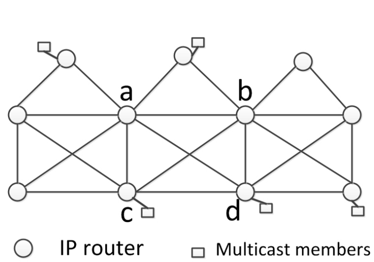
In this appendix, we use the node colouring model proposed in Chapter 6 to derive bounds on performance metric of group communication for star and tree routing topologies over a wireless mesh network when 802.11s with RTS/CTS mechanism is used at the MAC mulitcast layer to enable reliable multicasting. Our analysis is based on the list of assumptions described in Section 6.3 and Section 6.4.

As we have discussed earlier, streaming traffic is delay sensitive and thus, it is important to evaluate the performance of multicast streaming in terms of metrics such as delay, throughput, and capacity in order to ensure that the multicast routing topology constructed by the IP multicast routing protocol achieves the QoS for the multicast streaming. To evaluate the performance of multicast streaming in a wireless multi-hop network, one must have some knowledge of which links in the network interfere with one another, and to what extent. However, the problem of estimating the interference among links of a multi-hop wireless network is a challenging one [195–198]. It involves accurate modelling of radio signal propagation, which is difficult since many environment and hardware-specific factors must be considered. Moreover, empirically testing every group of links is not practical: a network with n nodes can have $O(n^2)$ links, and even if we consider only pairwise interference, we may have to potentially test $O(n^4)$ pairs [199].

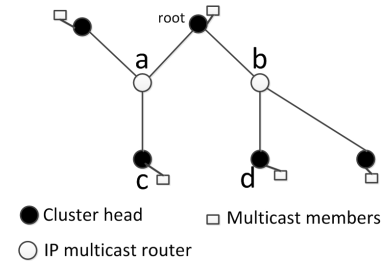
Therefore, it is hard to accurately evaluate the performance of multicast streaming in a wireless multi-hop network. However, lower/upper bounds on performance metrics are always easier to compute and can be very useful in deciding whether or not the constructed multicast routing topology can achieve the QoS guarantees.

Interference between wireless links is determined by their relative locations in the physical network topology. In our modelling in the appendix, however, interference between links is determined based only upon their logical positions on the multicast routing topology. Thus, our modelling ignores interference that can occur between multicast traffic transmitted from nodes that are not direct neighbours on the multicast routing topology. For illustration, let us consider the physical network topology depicted in Figure B.1(a). Since node c is a direct neighbour to node b (i.e., c is located in the coverage area of node b), traffic transmitted from node b to node d interferes with the traffic transmitted from node a to node c. However, since nodes c and b are not direct neighbours in the tree multicast routing topology (Figure B.1(b)), we ignore the effect of their traffic interference. We understand that this assumption is restrictive and not accurate. However, this assumption allows us to derive feasible bounds on performance metrics for star and tree multicast routing topologies when 802.11s with RTS/CTS mechanism is used at the MAC mulitcast layer to enable reliable multicasting.

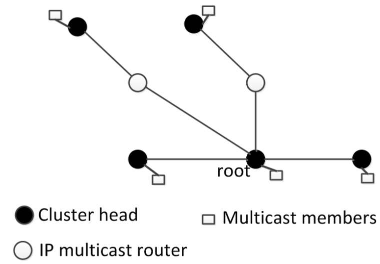
The purpose of this approximate performance analysis for star and tree multicast routing topologies that we present in this appendix is to motivate a ring-based multicast routing topology; and show that a ring-based routing topology is ideal for reliable many-to-many (group) communication in a wireless mesh network, especially for large number of group members.



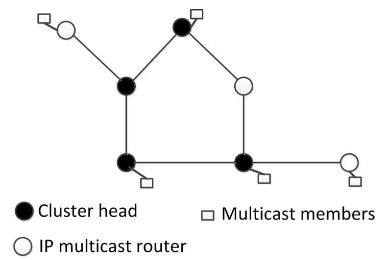
(a) Physical network topology



(b) Tree multicast routing topology



(c) Star multicast routing topology



(d) Ring-based multicast routing topology

Figure B.1: Possible multicast routing topologies on top of the underlying physical network topology

Star topology

We consider a multicast routing network that is connected in a shared star topology (Figure B.2(a)). We refer to IP multicast routers that connect group members to the multicast network as ***cluster heads***. We refer to the cluster head that connects all cluster heads together in the star topology as the root. We assume that the number of intermediate routers between a cluster head and the root is h. We note here that we carried out our performance analysis based on the star topology that is shown in Figure B.2(a). However, our analysis can be easily extended to any given star routing topology.

Proposition: The end-to-end delay in a star IP multicast routing topology is

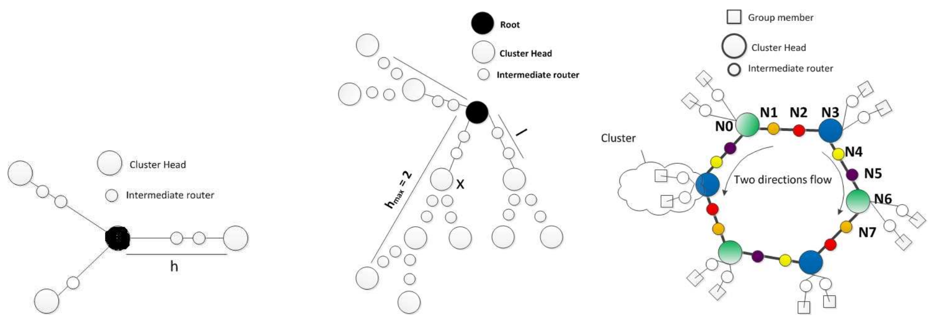
$$D_{Star} \geq 3 \times n \times T(h+1).$$

Proof. In a star topology, every cluster head delivers traffic generated by group members in its cluster to the root in order to get distributed to other clusters. We consider the following delays. Cluster heads can simultaneously receive traffic from their members. Due to interference between traffic, every cluster needs at least T time period to receive traffic from its members. Since the path from a cluster head to the root consists of h intermediate nodes, the time that a multicast packet spends on the path to reach the root is at least $3 \times h \times T$, where the factor 3 accounts for number of colours ($C = 3$) that are used to label nodes on the path ($h \geq 2$).

Due to interference between traffic, the root can receive traffic from one of its neighbouring nodes at a time. Thus, the root needs at least $(n-1)T$ time period to receive traffic from all neighbouring nodes. When the root receives all cluster heads traffic, it uses the multicast address at the IP layer to transmit (broadcast) the traffic to all neighbouring nodes during $n \times T$ time period. The relayed traffic then needs to traverse every path from the root to every cluster head. The time that is needed to deliver the traffic from the root to every cluster head is at least $3 \times h \times (n-1) \times T$, where $n-1$ accounts for the duplicate traffic that is eliminated at the IP relay nodes. Every cluster head (relays) broadcasts traffic using IP multicast address to all members in its cluster during $(n-1) \times T$ time period. Hence, the summation of all above delays gives us a feasible lower bound on the end-to-end delay as follows.

$$
\begin{aligned}
D_{Star_{LB}} &= T + 3 \times h \times T + (n-1) \times T + n \times T \\
&+ 3 \times h \times (n-1) \times T + (n-1) \times T \\
&\approx 3 \times n \times T(h+1).
\end{aligned}
$$

To achieve the end-to-end QoS guarantees (η), D must not exceed η (i.e., $D \leq \eta$). Moreover, the multicast routing topology must be able to support multicast traffic with rate of at least $N \times b$, where N is the number of group members, and b is the rate at



(a) Star (b) Tree (c) Ring-based

Figure B.2: Common multicast network topologies

which multicast traffic is generated at each member (frame/second).

We can see that due to bandwidth constraint at the root and interference between traffic (traffic congestion at the root), the rate (in frame/second) at which the root can relay (forward) multicast traffic is limited. Referring to the D_{Star} computation, we can see that the maximum channel bandwidth that the root can use to relay the multicast traffic is

$$\left(\frac{n \times T}{3 \times n \times T(h+1)}\right) \times B.$$

Hence, in order for the star topology to support the multicast traffic, we have

$$\left(\frac{n \times T}{3 \times n \times T(h+1)}\right) \times B \geq N \times b.$$

We also know that b must exceed $\frac{1}{\eta}$ (i.e., $b \geq \frac{1}{\eta}$). Combining the above two constraints, a feasible upper bound on the number of group members that a star topology can support with QoS guarantees is

$$N_{Star_{UB}} = \frac{B \times \eta}{3 \times (h+1)}.$$

This implies that for relaxed QoS (i.e., high η), higher channel bandwidth B, or less h, a star topology can support a larger number of group members and vice versa.

The power that every multicast packet consumes in the star topology to reach all group members can be computed as $P_{Star} = P_{topology} \times (h + n \times h)$.

Tree topology

We consider a shared tree that is shown in Figure B.2(b). We refer to IP multicast routers that connect group members to the network as ***cluster heads***. We assume that each cluster head is with d node degree on average (d is the number of tree branches generated from a cluster head, $d = 3$ in Figure B.2(b)). We further assume that the maximum number of cluster heads on a path between any leaf cluster head and tree root is h_{max} ($h_{max} = 2$ in Figure B.2(b)). We assume that the average number of intermediate routers between adjacent cluster heads on the multicast routing topology is I. We note here that we carried out our performance analysis based on the tree topology that is shown in Figure B.2(b). However, our analysis can be easily extended to any given tree routing topology.

Proposition: The end-to-end delay of a tree IP multicast routing topology is

$$D_{Tree} \geq 2 \times n \times T(h_{max} + 1)(3 \times I + 1).$$

Proof. In a tree topology, cluster heads can simultaneously receive multicast traffic from members in their clusters during at least T time period. Every cluster head needs to receive multicast traffic from every neighbouring cluster head on the tree. Due to interference between traffic, a cluster head can receive traffic from one adjacent node at a time. Since the path between neighbouring cluster heads consists of I routers ($I \geq 2$), every cluster head needs at least $\sum_{i=1}^{d} 3 \times I \times P_i \times T$ time period to receive multicast traffic from all neighbouring cluster heads, where the factor 3 accounts for the number of colours ($C = 3$) that are used to label routers on a path between neighbouring cluster heads, i refers to a neighbouring cluster head on a tree branch, and P_i is number of descendants of cluster head i, $P_x = 4$ in Figure B.2(b).

When every cluster head receives multicast traffic from all of its adjacent cluster heads, cluster heads simultaneously transmit (broadcast) the received traffic using IP multicast address to all adjacent nodes on the tree topology and to members in their clusters during $n \times T$ time period. Every adjacent node on the tree eliminates duplicate packets and relays the traffic. Hence, the time that is required to deliver a multicast packet between two neighbouring cluster heads on the tree topology is at least

$$T + \sum_{i=1}^{d} 3 \times I \times P_i \times T + n \times T \approx 3 \times I \times n \times T + n \times T.$$

The end-to-end delay for a tree topology is the time that a multicast packet spends in the network to traverse a path on the tree topology between the two group members that are connected to the farthest leaf cluster heads from the root. Hence, a feasible lower bound on the end-to-end delay can be computed as

$$\begin{aligned} D_{Tree_{LB}} &= (2 \times (h_{max} + 1))(3 \times I \times n \times T + n \times T) \\ &= 2 \times n \times T(h_{max} + 1)(3 \times I + 1). \end{aligned}$$

As we have discussed earlier, in order to achieve the end-to-end QoS guarantees (η), D must not exceed η (i.e., $D \leq \eta$). Moreover, the multicast routing topology must be able to support multicast traffic with rate of at least $N \times b$, where N is the number of group members, and b is the rate at which multicast traffic is generated at each member (frame/second).

We can see that due to the bandwidth constraint at cluster heads and interference between traffic (traffic congestion at cluster heads), the rate (in frame/second) at which any cluster head can relay (forward) multicast traffic is limited. Referring to D_{Tree} computation, the maximum channel bandwidth that the any cluster head can use to relay (forward) the multicast traffic is

$$\left(\frac{n \times T}{3 \times I \times n \times T + n \times T}\right) \times B.$$

Hence, in order for the tree topology to support the multicast traffic, we have

$$\left(\frac{n \times T}{3 \times I \times n \times T + n \times T}\right) \times B \geq N \times b.$$

We also know that $b \geq \frac{1}{\eta}$. Thus, a feasible upper bound on the number of group members that a tree topology can support with QoS guarantees is

$$N_{Tree_{UB}} = \frac{B \times \eta}{3 \times I + 1}.$$

We can see that the average power consumption in the network when using tree topology is $P_{Tree} = P_{topology} \times ((n-1) \times I + n)$.

Performance comparisons and discussion

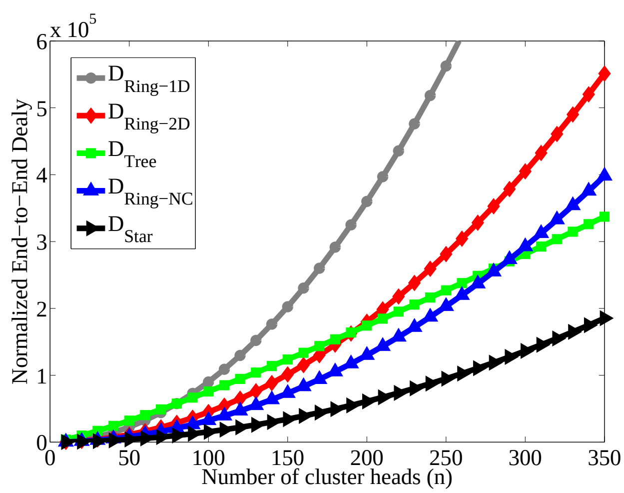


Figure B.3: End-to-End delay comparison

We summarize the results of our analytical results derived for different multicast routing topologies over a wireless mesh network in Table B.1. We can see that our proposed algorithm for traffic routing on the ring-based topology mitigates the end-to-end delay by a factor close to $\frac{2}{3}$ (33%). Furthermore, our proposed algorithm increases the capacity of the ring-based multicast routing topology (i.e., maximum group size that the ring-based topology can support with QoS guarantees) by a factor of $\frac{3}{2}$ (50%) (Table B.1).

We also compared the lower bound end-to-end delays (for $T = 1$ unit of time) and the average power consumptions (for $P_{topology} = 1$ unit of power) for all routing topologies, where we set $I = 3$, $d = 4$, $h = n/2$, and $h_{max} = 5 \times h_{avg}$, where $h_{avg} = log_{d-1}(n)$ (Figure B.3). We observe that for $n \leq 280$, a ring-based topology coupled with the proposed routing algorithm for traffic routing outperforms all other topologies in terms of the end-to-end delay except the star topology.

Interestingly, we can conclude that for a moderate multicast group size, a ring-based topology coupled with our proposed algorithm for traffic routing outperforms a tree routing topology in terms of both the end-to-end delay and capacity. Although a star topology has the lowest delay, the power consumption in a star topology is the highest and capacity of a star topology is the lowest. Hence, a star topology is not practical for group communication in a limited resources network such as static a wireless multi-hop network.

Table B.1: Performance comparison of multicast streaming for different routing topologies

Topology	End-to-End Delay	Power consumption	maximum group members
Star	$3 \times n \times T(h+1)$	$P_{topology} \times (h + n \times h)$	$\frac{B \times \eta}{3 \times (h+1)}$
Tree	$2 \times n \times T(h_{max}+1)(3 \times I + 1)$	$P_{topology} \times (n + n \times I)$	$\frac{B \times \eta}{3 \times I + 1}$
Bidirectional ring	$T(\frac{3 \times n^2}{2} + 3 \times I \times n + \frac{n}{2})$	$P_{topology} \times (n + I)$	$\frac{B \times \eta}{3}$
Bidirectional ring with network coding	$T(n^2 + 3 \times I \times \frac{n}{2} + \frac{n}{2})$	$P_{topology} \times \times (n + \frac{I}{2})$	$\frac{B \times \eta}{2}$

Ring topology with MAC scheduling

Although IP multicast protocols were developed more than a decade ago, IP multicasting is not deployed in today's wireless multi-hop networks. Most of IP routers in the Internet do not support IP multicasting. However, the structure of a ring-based multicast routing topology creates the possibility of supporting reliable group communication (traffic multicasting) in wireless mesh networks using the IP unicast scheme.

We propose a protocol for traffic multicasting in a ring-based routing topology over a wireless mesh network, wherein the IP unicast scheme is used to route the multicast traffic over the ring topology. In particular, we assume scheduling capability at the MAC layer based on time slots (e.g., TDMA), and propose a MAC scheduling protocol at mesh routers on the ring that maximally utilizes the available bandwidth at those nodes. We label nodes on the ring according to their MAC operating states as follows:

- Green (G): when the node receives multicast traffic of the clockwise flow.
- Orange (O): when the node receives multicast traffic of the anti-clockwise flow.
- Red (R): when the node relays multicast traffic of the anti-clockwise flow.
- Blue (B): when the node relays multicast traffic on the clockwise flow.

Consider four consecutive routers on a ring ($R1$, $R2$, $R3$, $R4$). Our proposed MAC scheduling protocol works as follows: consider a cycle that consists of 4 time slots. Let router $R1$ be with colour G at time slot $t = i$, while routers $R2$, $R3$, $R4$ are with colours O, R, B, respectively. At time slot $t = i + 1$, nodes $R1$, $R2$, $R3$, $R4$ switch to colours B, G, O, R, respectively. At time slot $t = i + 2$, nodes $R1$, $R2$, $R3$, $R4$ switch to colours O, R, B, G, respectively. At time slot $t = i + 3$, nodes $R1$, $R2$, $R3$, $R4$ switch to colours R, B, G, O, respectively (Figure B.4). The cycle repeats every 4 time slots, and all routers that are four hops away on the ring follow the same scheduling (same colour) (Table B.2). We can see that the number of colours in this case is $C = 4$. Hence, we can write

$$D_{Ring-MAC-scheduling_{LB}} = (\frac{n}{2} + I) \times (4 \times (\frac{n}{2}) \times T) + \frac{n}{2} \times T = T(n^2 + 2 \times I \times n + \frac{n}{2}).$$

To demonstrate the superiority of our proposed protocol, we analytically evaluated the performance of our proposed protocol against the case when IP unicast scheme is used to route multicast traffic on a unidirectional ring, and the case when IP unicast scheme is used to route multicast traffic on a bidirectional ring. For each case, we computed

Table B.2: Node colouring

Node	Time Slot				
R1	G	B	O	R	G
R2	O	G	R	B	O
R3	R	O	B	G	R
R4	B	R	G	O	B

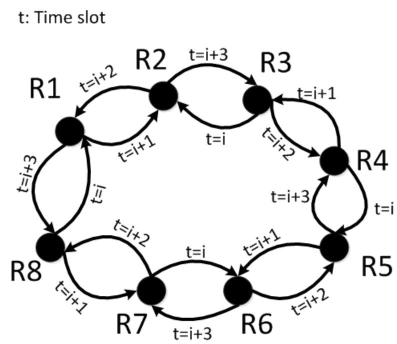


Figure B.4: MAC scheduling at mesh routers on the ring

end-to-end delay and the maximum group member that the ring can support with QoS guarantees. The results are summarized in Table B.3. We can see that the proposed protocol reduces the end-to-end delay of a ring multicast routing topology by a factor close to $\frac{2}{3}$. It can also be shown that the capacity of the ring is increased by a factor close to $\frac{7}{5}$.

Table B.3: Performance comparison of routing over a ring topology

Topology	End-to-End Delay	Maximum group members
Unidirectional ring	$T(3 \times n^2 + 3 \times I \times n + n)$	$\frac{B.\eta}{4}$
Bidirectional ring	$T(\frac{3 \times n^2}{2} + 3 \times I \times n + \frac{n}{2})$	$\frac{B.\eta}{3.5}$
Ring with MAC scheduling	$T(n^2 + 2 \times I \times n + \frac{n}{2})$	$\frac{B.\eta}{2.5}$