REALIZING THE POTENTIAL OF THE NETWORK EDGE

DISSERTATION

Submitted in Partial Fulfillment of
the Requirements for
the Degree of

DOCTOR OF PHILOSOPHY (Electrical Engineering)

at the

POLYTECHNIC INSTITUTE OF NEW YORK UNIVERSITY

by

Ayaskant Rath

January 2014

REALIZING THE POTENTIAL OF THE NETWORK EDGE

DISSERTATION

Submitted in Partial Fulfillment of
the Requirements for
the Degree of

DOCTOR OF PHILOSOPHY (Electrical Engineering)

at the

POLYTECHNIC INSTITUTE OF NEW YORK UNIVERSITY

by

Ayaskant Rath

January 2014

Approved:

Jonathan Chao
Department Head
Department of Electrical and Computer Engineering
24 July 2013

Approved by the Guidance Committee:

Major: **Electrical Engineering**

S. S. Panwar

Shivendra Panwar
Professor of Electrical Engineering
02 July 2013

Yong Liu
Associate Professor of Electrical Engineering
02 July 2013

S. Rangan

Sundeep Rangan
Associate Professor of Electrical Engineering
02 July 2013

Minor: **Computer Science**

Justin Cappos
Assistant Professor of Computer Science & Engineering
02 July 2013

Microfilm or copies of this dissertation may be obtained from:

UMI Dissertation Publishing
ProQuest CSA
789 E. Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

VITA

Ayaskant Rath was born on 15 May 1984 in Keonjhar, INDIA. He received the B.Tech. (Hons.) in Computer Science and Engineering from Indian Institute of Technology Kharagpur, in Kharagpur, INDIA, in May 2005; and M.S. in Electrical Engineering from Polytechnic University, in Brooklyn, NY, in May 2008. He did an internship at Extenprise e-Solutions in Pune, INDIA, from April 2004 to June 2004; and worked as Senior Business Associate at Techspan India Ltd. (A Headstrong LLC Company) in Bengaluru, INDIA from July 2005 to August 2006. He has been a Ph.D. candidate in Polytechnic Institute of New York University in Brooklyn, NY, after passing his Qualifying Examination in August 2007. During this time, he also had an internship at InterDigital Communications, LLC, in Melville, NY, from May 2010 to September 2010, and served as a Teaching Assistant for the graduate level course Internet Architecture and Protocols in the Department of Electrical and Computer Engineering at Polytechnic Institute of NYU. During his dissertation study, he has published four technical papers, and one patent. He has conducted research in optimization of the efficiency of cellular networks with overlaid small cells. His research has been supported in part by the National Science Foundation (NSF Grants 1230773, 0905446, and IIP-1127960), the New York State Center for Advanced Technology in Telecommunications (CATT), and the Wireless Internet Center for Advanced Technology (WICAT), an NSF I/UCRC.

ACKNOWLEDGEMENT

My doctoral dissertation stands on contributions from several diligent supporters. The foremost among them is my dissertation advisor, Prof. Shivendra Panwar, who has consistently supported, inspired, guided, critiqued, and advised me throughout my study at Polytechnic Institute of NYU. Always giving me the freedom to choose the direction of research I wanted to pursue, and helping me stay on the right track, he is truly the most perfect advisor one could ask for. In addition to what I have learned from him academically, his values and practices will continue to guide me perpetually in my professional life. I am also indebted by the support and guidance of Prof. Yong Liu, especially in the initial stages of my research, when he helped me learn the fundamentals to build a strong foundation in my research. I am also immensely grateful to the other members of my Guidance Committee, Prof. Sundeep Rangan, and Prof. Justin Cappos, who devoted their time to reviewing my dissertation, and provided their unique perspectives respectively.

I would like to thank my colleagues Dr. Sha Hua, and Sanjay Goyal, for meticulously collaborating with me in my research projects. I am also thankful to my colleagues Dr. Shunyuan Ye, and Dr. Tan Ba Le, who also collaborated with me in some of my research work. Together, they helped me learn and develop my research to a large extent. I also thank Dr. Prabhakar Chitrapu, and Dr. John Tomici from InterDitial Communications; Prof. Keith Ross from the Department of Computer Science and Engineering; Dr. Peter Kubat, Dr. Donald Smith, Dr. Stevan Leiden, and Dr. Rina Schneur from Verizon Network and Technology, for their contributions to my research.

My deepest gratitude also extends to Valerie Davis, Budget and Operations Manager in my department, for her constant support and affectionate care throughout my graduate life. Her incessant efforts played a key role in solving every single administrative issue I faced; indeed my dissertation could not have been possible without her. I am also thankful to Toros Massarlian, Kiren Nayak, and Farhan Dalvi, for helping with all logistics and paperwork during my study at Polytechnic Institute of NYU.

The healthy nurturing environment of a lab, produced by its occupants, is indispensable for research. I was fortunate to have a perfect lab environment, which was only possible due to my lab colleagues. I thank Dr. Pei Liu, and Dr. Yanming Shen, for their invaluable advice and suggestions, and Dr. Chun Nie, Shu Luo, Zizhong Cao, and Tingting Lu, for their help and support. I also thank Fraida Fund, Seyedamirhossein (Amir) Hosseini, Chamint Nicolas Barati, and last, but far from the least, Georgios Kyriakou, for all the help, the discussions, and the intense debates and arguments, that made working in our lab a pleasure.

I also thank Srikanth Mone, Nandini Priya Rabelli, Abhijeet Kumar, Sandeep Saha, Manu Bhai, Amrender Pal Singhal, Shashi Raj Singh, and numerous other friends who have always provided the support that I needed. I am also eternally indebted to all my teachers who have made pivotal contributions to both my academic as well as personal life.

Finally, I would like to thank my parents, who have been a persistent source of support and encouragement throughout my life, and my sister, who has always been my best friend. Their essential unconditional love, care, devotion, and patience towards me forever play the vital role in everything I have and aspire to achieve.

To Pupli, Maa, Baba, and friends.

ABSTRACT

REALIZING THE POTENTIAL OF THE NETWORK EDGE

by

Ayaskant Rath

Advisor: Shivendra Panwar

Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy (Electrical Engineering)

January 2014

Data networks have a tendency to perennially experience evolution of technology and growth of infrastructure. Although the core of data networks has traditionally reaped a larger portion of this evolution and growth, the *edge*, or the *last mile* of data networks is fast catching up in recent times. In modern cellular networks, for example, introduction of small cells such as femtocells has helped boost the resources available at the network edge, which is the last link between the user device and the network. Similarly, modern cable network service providers have begun enriching the edge of their networks by deploying fiber optic lines all the way to the home of the subscribers. With these new resources available at the edge of the modern data networks, we investigate ways in which the real potential of the new network edge can be realized.

Small cells overlaid on existing cellular networks provide opportunities for us to be able to better manage the ever increasing traffic demand. We present novel ideas to alleviate the burden of traffic on the traditional cellular networks by offloading major portions of this traffic to the newly introduced femtocells at the network edge. We also present realizable deployments of the proposed ideas. Mobility being an essential characteristic of cellular users, the potential of the newly introduced femtocells cannot be fully realized until mobile users are allowed to benefit from them. Hence we modify existing handover procedures that result in completely seamless handovers between femtocells, reducing the duration of data interruptions experienced by mobile users during handovers to negligibly low levels. Finally, we move on to propose an innovative video delivery mechanism that is designed with specific focus on the properties of the network edge of the modern cellular networks, which serves high quality videos to mobile users in the network at lower costs.

Extension of high capacity lines by modern cable network service providers all the way to the subscriber's home, and installation of operator controlled devices with reasonable computational power and storage capacities at home provides opportunities for us to push centralized service functionalities to the edge of the network, thus gaining additional scalability. We propose simple but novel strategies for video on demand and premium voice services for such walled garden networks using peer-to-peer communication technologies that can be deployed on existing network infrastructure.

These improvements in modern data networks work together towards realizing the potential of the network edge by harnessing the resources at the last mile of the network, thus providing better service to users, while benefiting service providers by saving on the cost of upgrading the infrastructure.

Contents

1 Introduction **1**
1.1 Bottleneck in the Macrocell Backhaul 2
1.2 Femtocells for Mobile Users 2
1.3 Better Video Streaming with Femtocells 3
1.4 Data Networks with Resource Rich Edges 4
1.4.1 Video-on-Demand in Cable Data Networks 4
1.4.2 Voice-over-IP in Cable Data Networks 5
1.5 Outline 5

2 FemtoHaul: Traffic Offloading from Macrocells to Femtocells **6**
2.1 Related Work 7
2.2 System Model and Assumptions 8
2.2.1 Channel Allocation Model 8
2.2.2 Access Policy 9
2.2.3 Demand Model 9
2.2.4 System Architecture 9
2.2.5 Macrocell Base Station Decision Making 10
2.3 Performance Evaluation 11
2.3.1 Simulation Settings 11
2.3.2 Simulation Results 13
2.4 Implementation of FemtoHaul 16
2.4.1 3GPP LTE Network Architecture 17
2.4.2 FemtoHaul using the Uu Interface 18
2.4.3 FemtoHaul using the Un Interface 19
2.4.4 FemtoHaul using the X2 Interface 20

3 Near Instantaneous Handover between Femtocells **22**
3.1 Fast OTA Interface between Femtocells 24
3.1.1 LTE Network Architecture 24
3.1.2 Direct Femto-Femto Interface 25
3.1.3 Realizations of the Femto-Femto Interface 26
3.2 Near Instantaneous Handover 27
3.2.1 The Legacy LTE Handover Procedure 27
3.2.2 Speed of UE in Handover 30
3.2.3 The Prefetch-based Fast Handover Procedure 30

3.2.4 The Near Instantaneous Handover Procedure 35
3.3 Performance Evaluation 39
3.3.1 Simulation Settings 39
3.3.2 Results and Analysis 41

4 Streamloading: Video Streaming for Mobile Users 46
4.1 Modern Video Delivery Services 48
4.1.1 Scalable Video Coding 48
4.1.2 Streaming *v.* Downloading 49
4.2 Implementation of Streamloading 50
4.2.1 A Streamloading Example 51
4.2.2 Heuristics of Streamloading 51
4.3 Performance Evaluation 54
4.3.1 Simulation Parameters 54
4.3.2 Analysis of Results 57

5 Peer-to-Peer Video-on-Demand on Cable Data Networks 60
5.1 The VoD Transport Network 61
5.2 Proposed P2P VoD Service 62
5.2.1 The Source of an Ordered Video 63
5.2.2 Rate of Video Download 64
5.3 System Model 65
5.3.1 Video Catalogs 65
5.3.2 Demand Arrival 66
5.4 Performance Evaluation 66
5.4.1 System Parameters 66
5.4.2 Simulation Set Up 67
5.4.3 Results and Analyses 68

6 Peer-to-Peer Voice-over-IP for Premium Voice Services 71
6.1 Features in Today’s Telephone Service 72
6.2 Network Architecture 73
6.3 The Proposed P2P Premium Voice Service 75
6.3.1 Installation of an ONT 75
6.3.2 Regular Call Set Up 76
6.3.3 Implementation of other Features 77

7 Future Work 82
7.1 FemtoHaul 82
7.2 Near Instantaneous Handover 83
7.3 Streamloading 84
7.4 P2P VoD in FTTP Networks 85
7.5 P2P VoIP in FTTP Networks 85

Bibliography 89

List of Figures

1.1 Modern Cellular Network with Femtocells 2
1.2 Modern FTTP Cable Data Network 4

2.1 Traditional *v.* FemtoHaul Traffic Flows 8
2.2 FemtoHaul Simulation Area Map 12
2.3 Backhaul Supply Rate Comparison 14
2.4 Stationary User Data Rate Comparison 15
2.5 Mobile User Data Rate Comparison 16
2.6 Interfaces in 3GPP LTE Architecture 17
2.7 FemtoHaul via Uu 18
2.8 FemtoHaul via Un 19
2.9 FemtoHaul via X2 20

3.1 Mobile Cellular Traffic Projection 23
3.2 Components of 3GPP LTE Architecture 25
3.3 Legacy Handover Procedure 28
3.4 Prefetch-based Fast Handover: Proximity Add/Release Processes 31
3.5 Prefetch-based Fast Handover: Handoff Process 33
3.6 Near Instantaneous Handover: Proximity Add/Release Processes 36
3.7 Near Instantaneous Handover: Handoff Process 38
3.8 Near Instantaneous Handover Simulation Area Map 39
3.9 Handover Interruption Comparison 41
3.10 Handover Data Delivery Comparison 43
3.11 Skipped Handover and Resource Wastage Comparison 45

4.1 Cellular Video Traffic Projection 47
4.2 Scalable Video Coding 49
4.3 Demonstration of Streamloading 52
4.4 Heuristic Streamloading Notations 53
4.5 Streamloading Simulation Area Map 56
4.6 Video Quality Comparison 58
4.7 Quality Distribution Comparison 59

5.1 VoD Service Network Architecture 62
5.2 Popularity Ranks and Demands of Videos 67
5.3 VHO Bandwidth Usage Comparison 68
5.4 Effect of Cache Size on VHO Bandwidth Usage 69

6.1 Premium VoIP Service Network Architecture 74
6.2 Call Set Up Process 76

List of Tables

2.1 FemtoHaul Simulation Parameters 13

4.1 SVC Video Data Rates 55

5.1 VoD Service Simulation Parameters 66

Chapter 1

Introduction

Transformation has been inherent to cellular networks ever since their conception a few decades ago. Yet, the rate of growth and progress in the design and technology of cellular networks has never been higher. Advancement in technology for more resources in cellular networks has always run hand in hand with the growth in consumer demand. The introduction of small cells in existing cellular networks is one of the most discussed directions towards enhancing capacities. Such small cells include *picocells* strategically deployed by service providers, and *femtocells* deployed by consumers in an ad hoc fashion.

A femtocell has a low-power and low-cost base station overlaid on the existing cellular network [1]. This base station is normally installed indoors, connected to the broadband service modem in a manner similar to WiFi access points, to provide a high-speed data connection to subscribers within a small range. Due to spatial reuse of the wireless spectrum, the usage of femtocells provides a significant gain in capacity (throughput per unit area). As a result, subscribers are able to enjoy faster speeds and longer battery lives due to the shortened distance between the wireless transmitter and the receiver. The recent advancement in transmission technologies, together with introduction of femtocells has been a key factor in drastic increase in the resources available in the last mile, or the edge of the cellular networks.

Figure 1.1 shows a typical modern cellular network with femtocells overlaid on it. The introduction of such networks results in a new network edge that opens up numerous improvement opportunities. In recent times, there has been a significant amount of research tackling the issues brought about by the introduction of femtocells in cellular networks. The focus of our work however is more about the study and realization of the potential of femtocells over and above their natural benefit through the spatial reuse of resources. We focus on the question of what can we get out of femtocells when they are widely deployed, rather than that of how to make femtocells work better.

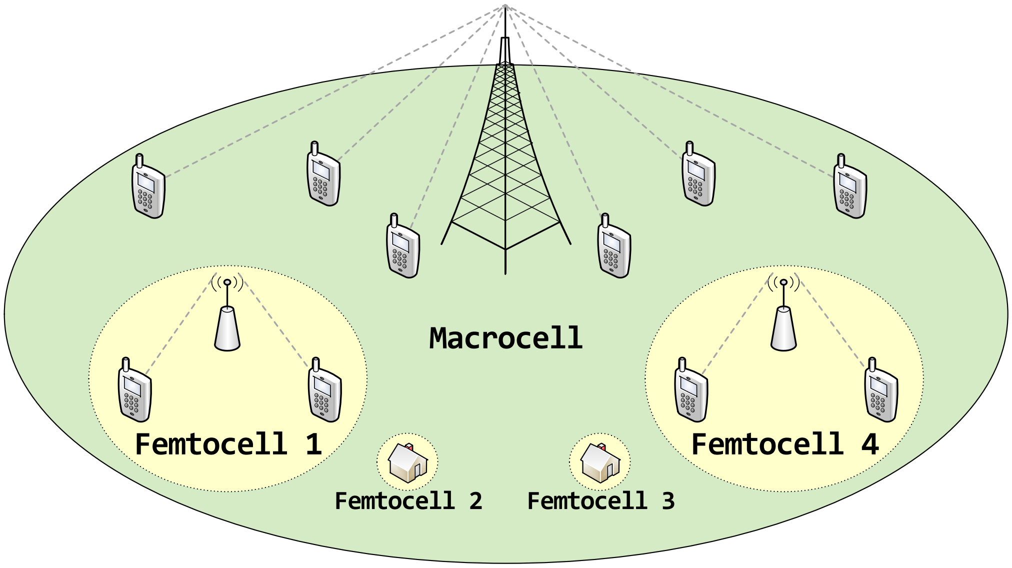


Figure 1.1: A typical modern cellular network with overlaid femtocells. The open access femtocells serve users falling under their range, while the macrocell serves all users who do not fall in the range of any femtocell in the region.

1.1 Bottleneck in the Macrocell Backhaul

The ever increasing user demand for highly data-intensive applications is motivating cellular operators to provide more data services. However, the operators suffer from the heavy budgetary burden of upgrading their infrastructure. Most macrocell Base Stations still connect to backhauls with capacities of less than 8 Mbps, much too low to be able to serve all voice and data users in the cell. This so-called macrocell backhaul bandwidth shortage problem is encumbering the growth of cellular data services. We propose a novel solution, *FemtoHaul*, which efficiently exploits the potential of femtocells to bear the macrocell backhaul traffic by using relays, thus enhancing the data rates of cellular subscribers. We design a system architecture and its related signaling and scheduling strategies. Extensive simulations demonstrate that FemtoHaul can effectively serve more users and support higher data demand with the existing macrocell backhaul capacity. We also study ways to implement FemtoHaul within the standardized network architecture.

1.2 Femtocells for Mobile Users

The explosion of demand for data in cellular networks has led to the macrocells limited also by capacity over the air, in addition to their backhaul. Although femtocells in cellular networks take a portion of this burden off the macrocell by serving stationary users within their coverage, more offloading is needed. With nearly a third of cellular traffic being mobile, it is desirable that mobile users too can tap into the potential of femtocell offload. This can be made possible by making sure every mobile user is able to handover to every femtocell along its path. In the absence of a handover procedure specifically catering to femtocells, the slow Legacy Handover procedure fails to achieve this goal, given the high latency backhaul link of femtocells to the core network. We propose to introduce an over-the-air fast control plane interface between pairs of femtocells, which leads to a near instantaneous handover between femtocells. The new fast interface is only rarely activated and is used for transmission of a few control messages per handover. *Near Instantaneous Handover* allows all mobile users, including those moving at high speeds, to take advantage of every femtocell in their path. It also enables the fast moving user to spend a larger proportion of time staying associated with the femtocell, rather than going through the handover procedure. We perform extensive system level simulations to evaluate Near Instantaneous Handover in comparison to Legacy Handover.

1.3 Better Video Streaming with Femtocells

Mobile users in modern wireless networks often face a difficult question when it comes to consuming video: *To download or to stream?* Most video downloading services tend to be more expensive by charging users for every video they download as opposed to monthly unlimited video viewing subscriptions typically offered by streaming services. At the same time, video downloads offer better video quality by insuring users against the low data rates they may experience in their path. Users viewing streaming videos experience lowered video quality while going through low bandwidth regions in cellular networks compared to the quality they experience when in high bandwidth femtocells; this is facilitated by adaptive video streaming techniques. We therefore propose *Streamloading*, a new type of video delivery service that aims to take advantage of both streaming and downloading. Streamloading uses Scalable Video Coding to encode the video into layers of bit streams, and allows subscribing users to *download* the enhancement layers, while restricting them to only *stream* the base layer. This allows streamloading to legally qualify as a streaming service, thus enabling it to be offered at low, streaming service level prices. In addition, streamloading can, with higher likelihood, achieve the video quality levels offered by downloading services. Heuristics are then applied to prioritize video data requests, and network transmission scheduling to enhance the performance of streamloading service. Extensive simulations are then performed to analyze and quantify the benefits of streamloading over streaming. We demonstrate that streamloading can deliver higher video quality to users and, by efficient use of available bandwidth, significantly alleviate the “bandwidth crunch” that cellular operators currently experience.

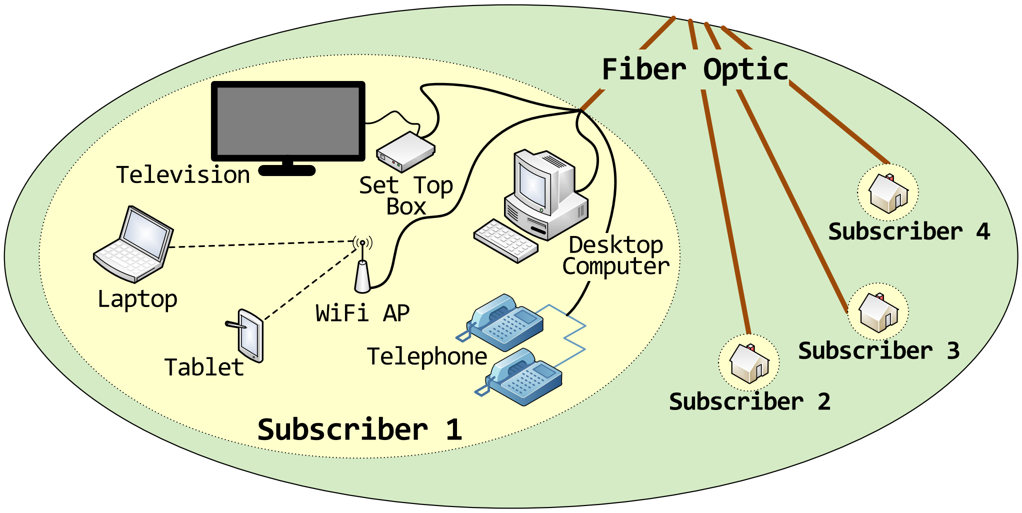


Figure 1.2: A typical modern cable data network with fiber optic connectivity to subscriber premise providing premium digital voice, television video, and high speed internet services.

1.4 Data Networks with Resource Rich Edges

In addition to looking at opportunities of improvement in the edge of modern cellular networks, we also study the improvement opportunities in services provided on modern cable data networks that have a resource rich last mile, as shown in Figure 1.2. Fiber optic lines go all the way up to the subscriber premise, and provide video, voice, and high speed internet services. In particular, we focus on today's Video-on-Demand and Premium Voice services.

1.4.1 Video-on-Demand in Cable Data Networks

Recent deployments of Fiber-to-the-Premises (FTTP) networks, with their high bandwidth potential, have enabled delivery of a variety of new multi-media services to a customer's home. In addition to broadcast TV, traditionally delivered by coax or fiber, new high bandwidth Internet connections are now available through such deployments. One of the very popular services is Video-on-Demand (VoD), which can bring standard or high definition video streams directly to the TV. However, VoD does not scale well, as each VoD stream traverses the core transport network every time it is requested, making its delivery rather expensive. We investigate an alternative delivery of VoD services by deploying a Peer-to-Peer (P2P) like protocol in a restricted local environment, where the peers are connected via a 'private' high bandwidth network, and peer storage is fully controlled by the P2P manager. This so called 'walled garden' scenario has been simulated in great detail to get quantitative system performance measures, to illustrate its feasibility and show the benefits of the approach.

1.4.2 Voice-over-IP in Cable Data Networks

With the integration of voice and data networks, service providers are gradually switching to Voice-over-IP (VoIP) for their landline voice services. Typically, today's centralized VoIP networks pay a high price for scalability, particularly, when they provide a modern premium voice service. However, as the Internet Service Providers are offering their customers high bandwidth access, scalability issues with VoIP systems can be addressed by using P2P schemes for VoIP services. In this article, we propose a scalable and robust P2P VoIP service system that can provide all the features and services a premium voice service provides today. We discuss the network architecture on which the system runs, as well as the implementation of various features.

1.5 Outline

We begin with Chapter 2, where we present FemtoHaul, a novel solution to the shortage of bandwidth at the backhaul of the macrocells of cellular networks. FemtoHaul strives to offload macrocell traffic to overlaid femtocells using various innovative methods, the fundamental one being the usage of a user relay. We use extensive simulations to show the benefits of FemtoHaul, and also propose various ways to implement FemtoHaul in existing cellular network infrastructure within the existing standards.

In Chapter 3, we propose to introduce a new interface directly connecting pairs of neighboring femtocells that then helps us design a Near Instantaneous Handover procedure for handovers of mobile users between femtocells in a cellular network. We begin by introducing a prefetch-based Fast Handover procedure, and then extend it using the proposed femto-femto interface to achieve Near Instantaneous Handover. System level simulations show the realizability of the seamlessness of the proposed handover.

We then propose in Chapters 5 and 6 P2P models of VoD and premium telephony services, respectively, on FTTP networks where the edge of the network has an abundance of bandwidth, storage and computational resources. The proposed VoD service is simulated with close to real world demand arrivals and catalog designs that helps quantify the benefits of using P2P over existing centralized solutions. P2P VoIP is proposed for the premium voice service, and implementation schema of all required service feature functionalities are proposed.

We conclude the dissertation with Chapter 7 where we also indicate future directions of the solutions presented.

Chapter 2

FemtoHaul: Traffic Offloading from Macrocells to Femtocells

Cellular services today are experiencing a transition from simple traditional voice services to highly data-intensive applications. The fundamental cause of this transition lies in the ever increasing user demand for higher data rates in cellular wireless networks, for applications such as content-rich multimedia and gaming. At the same time, the average growth of revenue per user for voice is declining due to the saturated market, which is driving the cellular operators to turn their focus to data services. Moreover, the storage and computational power of a cellular phone today is approaching those of a personal computer, allowing it to support more data-based applications. All these trends are driving the design and development of new data-oriented cellular systems.

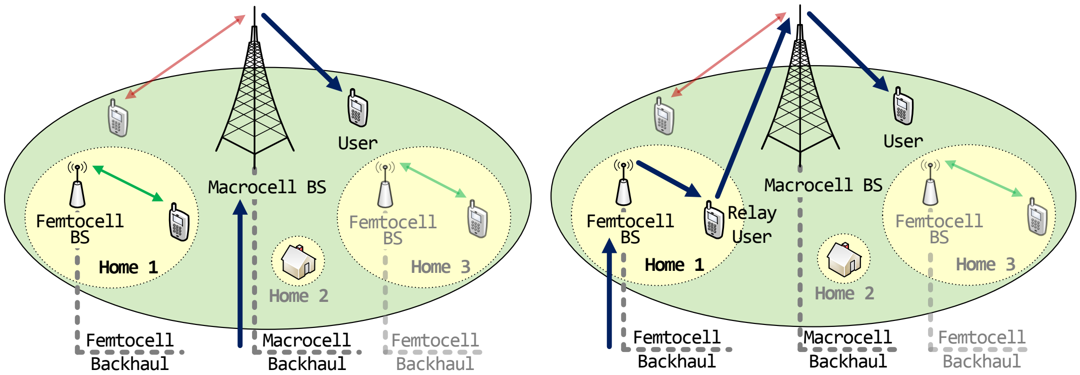
However, the lag between the traffic demand and the development of the cellular infrastructure is a major obstacle. Current 3G standards are pushing cumulative macrocell data rates to tens of Mbps or even more. For instance, with the highest modulation, Evolved High-Speed Packet Access (HSPA+) can achieve a 42.2 Mbps maximal data rate, while this parameter for WiMAX is 63 Mbps [2]. Newer cellular standards, such as 3GPP Long Term Evolution (LTE), support even higher data rates. However, most of the macrocell Base Station (BS) backhauls use only a few (2 to 4) T1/E1 lines [3] which account for only about 3 to 8 Mbps of bandwidth. Although adding multiple lines to the BS backhaul, or increasing the number of BSs may seem like an obvious solution to fill this gap, high CAPEX and OPEX costs (more than $10000 per line and $50000 per site annually) [3] make this an expensive option. The macrocell backhaul has thus become a bottleneck delaying the growth of cellular data services. This problem has been attracting much attention in industry lately [4, 5]. Companies like Jupiter Networks [6] and ADC [7] are already offering their solutions to this backhaul bottleneck problem.

The introduction of the femtocell offers a cost-effective solution to this problem. A femtocell provides high speed data connection to subscribers within its range, and is connected to the customer's broadband wireline network provided by their Internet Service Provider (ISP), which carries the traffic that originally went on the macrocell backhaul. This relieves the heavy burden on the operators to upgrade their infrastructure. As a result of this win-win situation, market research predicts femtocell market growth from $230 million in 2009 to $4.6 billion in 2014 [8].

We propose a novel solution to the macrocell backhaul problem, named FemtoHaul, which efficiently exploits the potential of femtocells to bear the backhaul traffic, enhancing the data rates of cellular subscribers. In our solution, we propose to use any user in the range of a femtocell as a candidate to relay traffic. When mobile outdoor users are communicating with the macrocell BS, instead of fetching data through its own backhaul, the macrocell BS uses relays to obtain the data from the broadband wireline networks connected to the femtocells, which typically have more than adequate bandwidth, as shown in Figure 2.1. Note that the concept of relays has been accepted in standards such as IEEE 802.16j. Through analysis and simulations, we show that this solution can accommodate high data demand and a large number of users that would otherwise overload the macrocell BS. Specifically, we design the system architecture and devise corresponding protocols that handle the signaling between different entities, as well as efficient decision making and relay choosing strategies for the macrocell BS, while considering the interference, channel allocation and handoff issues, and conduct extensive simulations. The simulations demonstrate that our solution can significantly reduce the macrocell backhaul traffic while still guaranteeing a high data rate to the subscribers. We also present various realizations of FemtoHaul that are possible within the standardized 3GPP LTE architecture framework.

2.1 Related Work

As a promising technology, the femtocell has driven several research innovations. A survey in [9] described WiMAX femtocell deployment and showed a potential 150 times capacity gain and good indoor coverage. Despite the advantages a femtocell offers, various technical challenges still remain. Interference Management is the main issue considered. Claussen [10] analyzed the feasibility of femtocells in the same frequency band as a macrocell network and showed that femtocell power control mitigates interference to other users. Chandrasekhar and Andrews [11] derived exact outage probability at a macrocell and tightened lower bounds on the femtocell outage probability. They showed that interference avoidance through a TH-CDMA physical layer coupled with sectorized receive antennas consistently outperforms a split spectrum two-tier network with omnidirectional femtocell antennas. Choi et al. [12] studied the public access policy of the



(a) Traditional System: Traffic is fetched from macrocell backhaul for all users outside of any femtocell range.

(b) FemtoHaul: Traffic is fetched from femtocell backhaul, reaches a user outside of any femtocell range, through a relay user.

Figure 2.1: Traffic flow in a traditional cellular network system with femtocells, in contrast to FemtoHaul, where traffic burden on the macrocell is significantly reduced due to offloading. Users who do not fall in any femtocell range also benefit from femtocells when using FemtoHaul.

femtocell and showed that an adaptive policy which takes the instantaneous loads on the network into account can lead to improved performance. Resource management is another issue for femtocell overlay cellular network. A location-based solution for leveraging maximal spatial reuse from OFDMA-based femtocells by allowing them to reuse the macro resources is presented in [13]. None of these articles considers the macrocell backhaul problem. To the best of our knowledge, our work offers the first solution to mitigating this problem by intelligently using femtocells to reduce the macrocell backhaul traffic load.

2.2 System Model and Assumptions

In this section, we present a brief description to the channel allocation model and the access policy in this cellular overlay network, along with the assumptions we made. Note that we use the terms users and cellular phones interchangeably.

2.2.1 Channel Allocation Model

In a cellular network with femtocells, the channel allocation policy is an open problem. Fixed allocation of channels between the macrocell and the femtocells leads to a waste of spectrum. On the other hand, making them share the same spectrum causes interference. In our system, we let the macrocell use all the frequency channels, while allowing the femtocells to reuse some of them to communicate with their associated users, as presented in [13]. This can be achieved in most of the new OFDMA-based standards (WiMAX, LTE), which ensure an efficient channel usage with controlled interference. Under this model, users that behave like relays can use separate channels to communicate with the macrocell and the femtocell BSs.

2.2.2 Access Policy

Due to the low transmission power and signal penetration loss, femtocells always serve the users within only a small range, while we assume the entire region is served by the macrocell BS as shown in Figure 2.1. Every femtocell also has a set of registered users who can be thought of as those who own or lease the femtocell, and whose ISP is used by the femtocell as its backhaul. In our model, a femtocell gives absolute priority to traffic to/from its registered users.

In our system, we assume that each user (registered or unregistered) can sense all the channels for the pilot signals sent from the macrocell and the femtocell BSs, and connect to the one with the strongest pilot signal. Although there is some debate about whether femtocells should be open access, where any user in their range can associate with them, or closed, where only registered users can associate with them, both researchers, and service providers are now inclining towards adopting an open access policy for femtocells. The primary reason for this inclination is that having open access femtocells avoids the "loud neighbor effect" [12], in which the strong signal from an in-range, but unregistered user contributes significantly to the noise environment in the femtocell. Thus, for all work presented in this dissertation, we always assume femtocells to be open access. We refer to all users connected to a femtocell as its *associated users.*

2.2.3 Demand Model

We model two typical kinds of applications in our system, voice and data services. Since voice calls are highly delay-sensitive, we let the macrocell BS serve all the voice applications, except for the registered users of the femtocells, who use femtocells for their voice applications. For data applications however, femtocells serve all demand for their associated users, whether they are registered or not.

2.2.4 System Architecture

We consider a system that includes three kinds of entities, namely, cellular devices, femtocell BSs and a macrocell BS. In a normal scenario, when the users demand data, the BS serves them by fetching data using the macrocell backhaul. The principal idea we propose is to relieve the macrocell backhaul by redirecting traffic to femtocell backhauls using other users as relays. For example, as shown in Figure 2.1 (b), when a user requests a file download, the macrocell BS, instead of directly fetching it from the macrocell backhaul as in Figure 2.1 (a), may forward the request to a femtocell through a relay user, who is associated with the femtocell. The femtocell BS, upon receipt of the request, fetches the requested file from the femtocell backhaul, and sends it back through the path of the request. The process would be symmetric when the user requests a file upload instead. This way, the request from the user is served without using the macrocell backhaul. Note that though a normal system may use only a downlink channel from BS to download data, our scheme uses both downlink and uplink channels of the BS. This may appear wasteful, but note that we assume the system capacity to be bottlenecked by the backhaul capacity of the macrocell, and thus the wireless channel capacity is available to handle this additional load.

2.2.5 Macrocell Base Station Decision Making

The macrocell BS always maintains association information of users associated with femtocells. All voice requests from registered users are forwarded to the femtocell whereas those from others go to the macrocell. During a voice call, voice packets are transmitted in both directions. When the macrocell BS receives a data request, it makes a decision whether to serve the request from the macrocell backhaul or from the backhaul of a femtocell using an associated user as a relay. The macrocell BS makes this decision based on its average backhaul utilization in the near past, θ, and a given utilization threshold θ_0. When $\theta < \theta_0$, the macrocell BS uses the macrocell backhaul to serve any arriving data demand. In other words, when the backhaul of the macrocell is sufficiently under-utilized, its BS fetches the data packets for the new data application request from the macrocell backhaul. On the other hand, when $\theta > \theta_0$, the macrocell BS chooses a user who is associated with a femtocell as the relay and forwards the data request through it to the femtocell BS. When the femtocell BS responds with the data packets through the relay, the macrocell BS in turn forwards them to the requesting user. In either case, how the macrocell BS fetches the data packets it transmits to the requesting user remains transparent to the requesting user.

In order to choose a relay node, the macrocell BS continually maintains a list of candidate relay users ordered according to decreasing values of their fairness index f_i given by

$$f_i = u_i/r_i \tag{2.1}$$

for each user i. It is the ratio of a user's available uplink data rate to the macrocell BS, u_i, to its average uplink data rate usage in recent past, r_i [14]. Thus, the node with the highest fairness index $f_{max} = \max_i\{f_i\}$ sits at the top of the list. The macrocell BS chooses relays according to their order in the list as requests arrive. The intent is to choose those users who have a good uplink bandwidth to the BS, and have not been transmitting much on this link in recent past.

Whenever a relay moves out of the range of its femtocell during forwarding of data, the macrocell BS restarts the process of choosing a new substitute relay. When a new relay is found, the macrocell BS requests the corresponding femtocell only for the part of the file left to download. This way, the mobility of relay nodes does not result in any significant wastage of resources.

2.3 Performance Evaluation

For the performance evaluation of FemtoHaul, we simulated it using the C programming language, which better allows for a flexible implementation of the relay choosing and traffic routing algorithms, compared to other simulation platforms. We simulated the generation of demand, and the transmission and reception of packets through queues maintained by various entities in the network. For performance evaluation, the macrocell and femtocell *backhaul supply rate*, and the *average download rate* experienced by users are measured and compared in various scenarios. We analyze these statistics at the end of this section.

2.3.1 Simulation Settings

Network Settings

In our simulation, there is only one macrocell covering the entire region. The macrocell is considered to be a circle with a radius of 1000 meters, with a BS located at the center. Femtocells and users are deployed randomly within the coverage area following a uniform distribution. The macrocell contains 500 users (including registered users). An illustration of the area map considered is shown in Figure 2.2. The macrocell downlink transmission power is set such that the received SNR at the cell edge is 6.0 dB, which is the minimum requirement for decoding data in IEEE 802.16e [15]. The femtocell transmission power is controlled as in [10], achieving a consistent femtocell range of 50 meters. Two users are registered with each femtocell, which are fixed, and located within the corresponding range.

For the purposes of simulation, WiMAX (IEEE 802.16e) is adopted as the cellular standard. We simulate four channels (Ch1∼Ch4) for transmissions throughout the network, each representing a group of channels in a WiMAX OFDMA system. Ch1 and Ch3 are reserved for downlink to the users whereas Ch2 and Ch4 are for uplink from the users. The macrocell uses all the four channels and the femtocells reuse Ch3 and Ch4. Throughput statistics are obtained by calculating the SINR of each channel for the

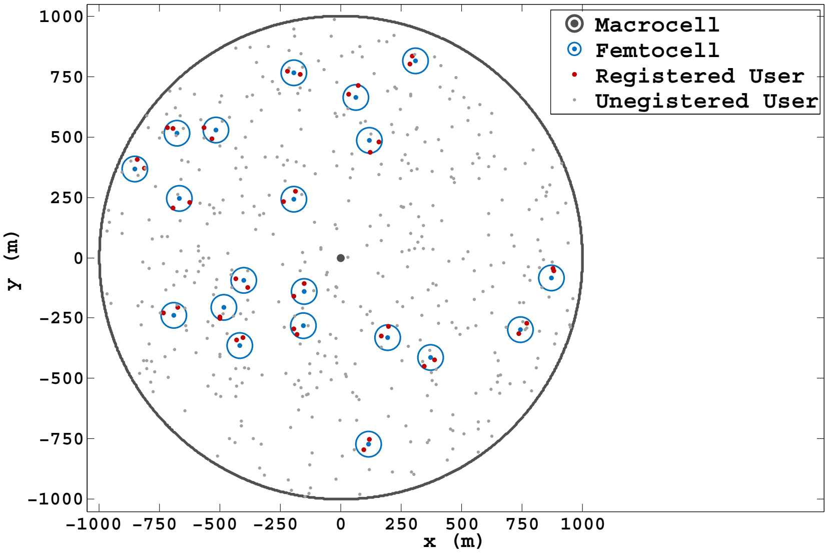


Figure 2.2: An example map of the area under consideration with twenty femtocells under one umbrella macrocell. Each femtocell has two registered users within its range, and there are 500 users in total.

receiver, then mapping it to the corresponding modulation/coding technique supported by WiMAX system, which gives the data rate. All user transceivers are considered half-duplex and hence cannot transmit and receive at the same time. The key simulation parameters are listed in Table 2.1.

Demand Generation

We assume that the requests for data download and the voice calls arrive according to i.i.d. Poisson processes for each user and the file sizes for data download are Pareto distributed.

Transmission/Reception Scheduling

It is assumed that the transmissions to multiple users are scheduled in a TDMA fashion with time slot lengths of 2 ms each. For each time slot, each station (the macrocell BS and all the femtocell BSs) schedules a user for a given channel based on the proportional fairness criterion [14], as described in Section 2.2.5. The user with the highest ratio of available link data rate on the channel to average link data rate usage in recent past (1 second window) is chosen for the channel.

Network Parameters
Outdoor path loss: $28 + 35\log_{10}(d)$ dB; indoor path loss: $38.5 + 20\log_{10}(d)$ dB. d is the distance from the BS in meters; Wall loss: 5 dB; Noise Power: -114.13 dB
Mobility Parameters
Random Walk Model with reflection; Users' speed: 0 to 2 m/s (random, uniform); Direction Change Periodicity: 0 to 100 sec.
Demand Generation Parameters
Packet Size: 8000 bits; Voice Call Frequency: 1 call per hour; Voice Call Duration: 3 min; Voice Data Rate: 10 kbps each way; Average File Size: 23.6 kB; Minimum File Size: 3.6 kB (as measured from W95 traces in [16]).
Transmission Parameters
Macrocell BS backhaul bandwidth: 6 Mbps; Broadband Service Bandwidth (for femtocell connection): 10 Mbps; BS Utilization Threshold (θ_0): 80%.

Table 2.1: Parameters used in the simulation for performance analysis of FemtoHaul.

Packet Transmission

Every voice/data stream is implemented as a packet queue. For the backhaul transmission, a packet is transmitted only if there is sufficient bandwidth for its complete transmission in the time slot. Moreover, we suppose femtocells share the bandwidth of the broadband service they connect to, with other services like WiFi. Thus the backhaul bandwidth of the femtocell is the leftover bandwidth.

For transmission, if the available bandwidth in a time slot is not sufficient to transmit all packets in all voice stream queues, then the bandwidth is equally distributed among all voice streams. On the other hand, if sufficient bandwidth is available, all voice stream packets are transmitted. The bandwidth leftover is used to transmit packets from the data streams by again allocating equal share to each data stream. For packet transmission on the air, since the standards allow fragmentation of packets into smaller blocks before transmission [14], in our simulation, partial transmission of a packet may occur in a time slot. The total bandwidth available for each channel in every time slot is the throughput computed as stated in Section 2.3.1.

2.3.2 Simulation Results

In this section, we simulated scenarios with number of femtocells ranging from 0 to 35. Please note that since we maintain the total number of nodes the system the same for all scenarios, the number of unregistered users in the system decreases as the number of femtocells increase. In stationary scenarios, all nodes are fixed. In mobile scenarios, all unregistered users are mobile while registered users are fixed. We apply the random walk mobility model with reflection at the boundary [17] to drive user movement in the

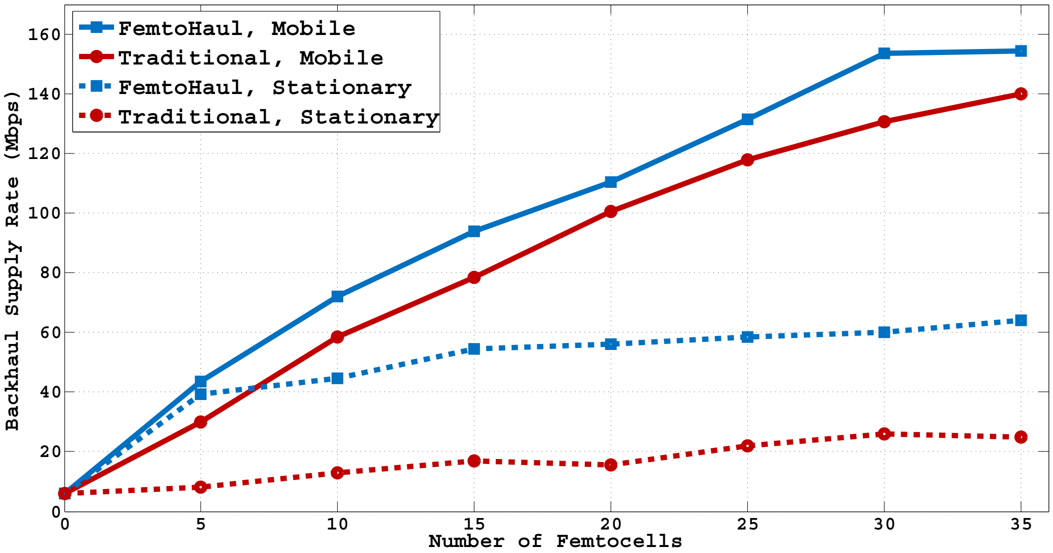


Figure 2.3: Rate of aggregate data supplied by all backhauls in the network in a traditional system and FemtoHaul, for scenarios with mobile and stationary users, respectively, with increasing number of femtocells in the region.

simulation, with the parameters shown in Table 2.1. In order to find the capacity of the network, we set the data demand rates of the users high enough to saturate the entire system. Exceptions to this are registered users who have absolute priority over other users; we let them have normal demand, otherwise they will block traffic from any other user. Thus, registered users' demand data rates are set at 100 requests per hour. Every scenario is run three times and the results are their average.

Backhaul Supply Rate

In Figure 2.3, we compare the backhaul supply rates, defined as the average rate at which data is pulled from all of the backhauls in the system, in FemtoHaul and the traditional scheme. We can see that when femtocells are first introduced in the network, there is a surge in the backhaul supply rate. For example, in the stationary scenario, when there are no femtocells in the system, the total data coming from the backhauls is only 6 Mbps. However, the introduction of 5 femtocells in the system increases the backhaul supply rate to 8.2 Mbps. This results from the users associated with femtocells downloading files from the femtocell backhaul, rather than the constrained macrocell backhaul. Moreover, FemtoHaul exploits the potential of femtocells further by using relays, which direct more traffic to the cheaper femtocell backhauls. In the same scenario with 5 femtocells, the backhaul supply rate increases from 8.2 Mbps to 39.3 Mbps due to usage of relays. The macrocell backhaul bandwidth shortage is thus relieved to a large extent. The same phenomenon can be observed in the mobile scenario. For instance, the backhaul

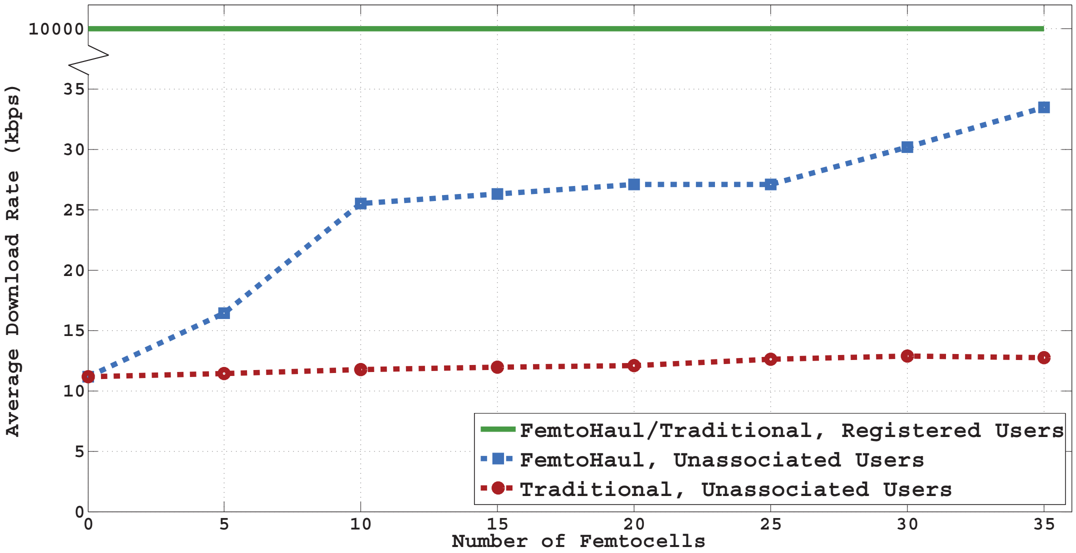


Figure 2.4: Average effective download rate experienced by unassociated stationary users in a traditional system and FemtoHaul with increasing number of femtocells in the network. Download rate experienced by registered users in the network is not affected by the use of FemtoHaul.

supply rate increases from 130.6 Mbps to 153.7 Mbps in a 30 femtocells scenario, when FemtoHaul is applied. Note that the additional backhaul supply rate is delivered to the users that are otherwise under-served in the traditional system, i.e. those who cannot connect to a femtocell directly.

The backhaul supply rates are higher when the nodes are mobile because mobile nodes get associated with femtocells periodically, and are able to get high download rates in such periods of association. Note that we assume here that mobile nodes are indeed able to quickly associate with every femtocell on their path, which is not currently feasible in cellular networks. However, we address this issue of fast handover of mobile nodes between femtocell in Chapter 3, by even allowing users with extremely high speeds to associate with femtocells, thus justifying the benefits of FemtoHaul presented here. Stationary nodes, if unassociated, stay unassociated and always get lower download rates.

Download Rate

The rate at which a user downloads a file is the ratio of the file size to the total time taken to download the entire file. We define the average download rate experienced by a user as the average of the rates at which the user downloaded all its files. Figure 2.4 shows the average download rates of the users who are not associated with any femtocell for stationary scenarios. In a traditional scheme, these users can only download files from the macrocell BS. Since the demand rates are high enough to saturate the system, congestion occurs and every user experiences a low download rate of less than 14 kbps.

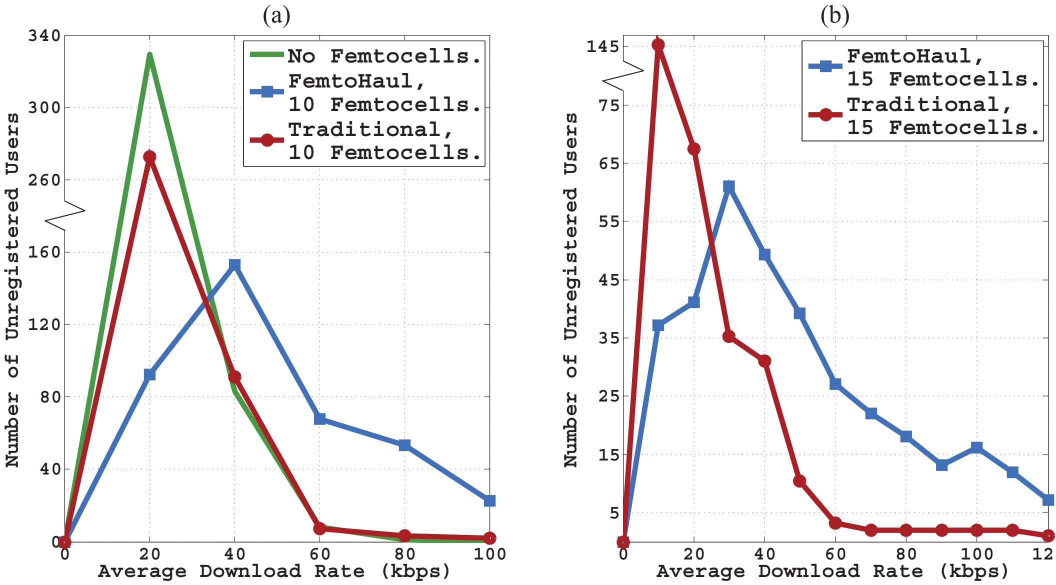


Figure 2.5: Distribution of average effective download rates experienced by mobile (unassociated) users in a traditional system and FemtoHaul. Distributions for networks with 0, 10, and 15 femtocells are presented, respectively.

FemtoHaul enables such users to download files from femtocells through relays. From the figure we can see that FemtoHaul leads to higher download rates of up to 33.5 kbps for similar saturating demands. Note that because we set relatively large number of users and a high demand rate in the network, our resulting average download rates are lower than those in a typical system. However, similar benefits can be expected when FemtoHaul is applied to a typical system. Finally, as shown in Figure 2.4, the registered users always get a rate of almost 10 Mbps as they are close to the femtocell and their traffic always has absolute priority.

We now analyze the effect of FemtoHaul on download rates of users in mobile scenarios. Figures 2.5 (a) and (b) show the distribution of download rates of users in systems with 0, 10 and 15 femtocells comparing the traditional system with FemtoHaul. The users in the low rate domain (0 to 100 kbps) are essentially those unregistered users who rarely associate with a femtocell during their move. As can be seen, such users in a traditional system suffer from low download rates. FemtoHaul, as is evident from the figures, benefits them by managing a significantly fairer rate distribution.

2.4 Implementation of FemtoHaul

Having demonstrated the inherent benefits of the idea of FemtoHaul, we elaborate this idea further in this section by presenting various implementations of FemtoHaul within

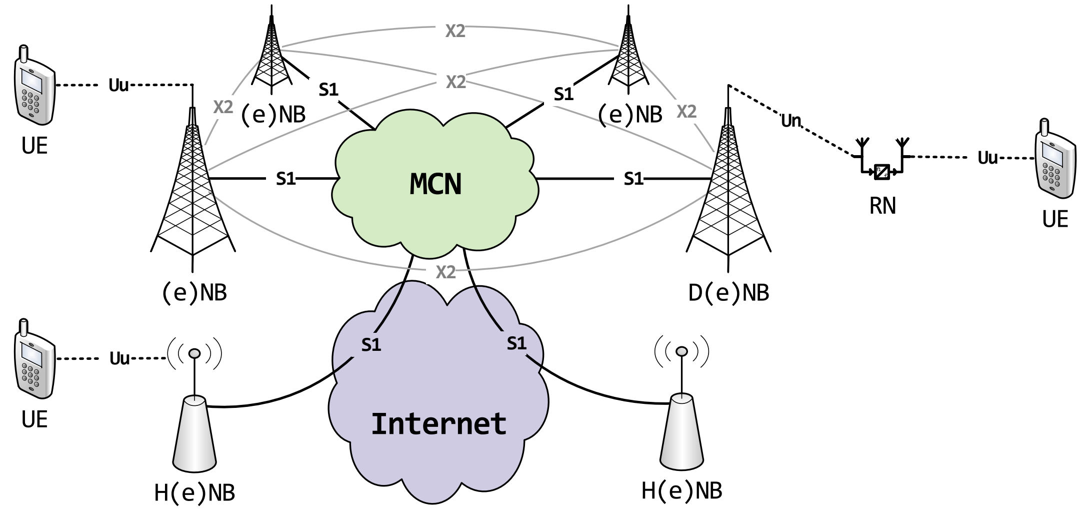


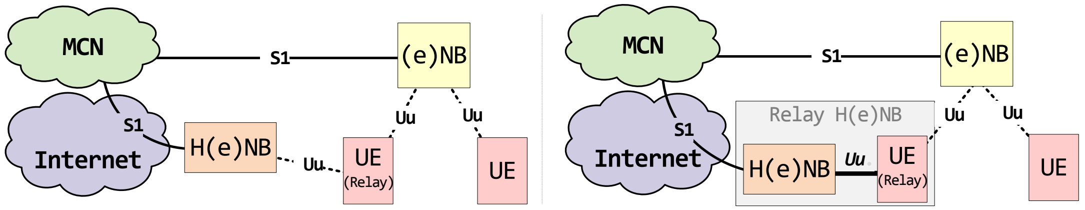
Figure 2.6: Relevant interfaces and components of the 3GPP LTE architecture from the perspective of offloading of traffic between the macrocells and the femtocells. The solid and dashed lines represent wired and over-the-air interfaces, respectively.

the 3GPP LTE architecture standards that achieve various levels of feasibility and address some of the issues raised by the fundamental concept of FemtoHaul presented thus far.

2.4.1 3GPP LTE Network Architecture

We begin by presenting in a concise manner, the relevant components of the present day standardized 3GPP LTE network architecture, illustrated in Figure 2.6. The Mobile Core Network (MCN) directly connects to all the macrocell BSs denoted by (enhanced) NodeBs ((e)NBs). A femtocell BS is similar to a macrocell BS and is denoted by a Home (e)NB (H(e)NB). All user devices, such as cellular phones, represented as User Equipment (UE) connect to their associated (e)NBs or H(e)NBs via a standard over-the-air interface called Uu that includes both control and data paths. All BSs have backhauls that connect to the MCN via an S1 interface for both control and data paths. The (e)NBs are normally interconnected with each other via X2 interfaces, containing both control and data plans, that provide signaling and data forwarding during handovers. The 3GPP LTE architecture also standardizes a Relay Node (RN), whose primary purpose is to extend a cell by relaying control and data plane signals between a Donor (e)NB (D(e)NB) and a UE located outside the macrocell of the D(e)NB. The air interface between the RN and the D(e)NB in this relay link is called Un.

With the above relevant entities and interface of the 3GPP LTE architecture, we proceed to present various possible implementations of FemtoHaul.



(a) A selfless UE serves as a relay, devoting its bandwidth and battery resources.

(b) Proposed relay UE hardware is embedded in the H(e)NB.

Figure 2.7: Offloading of traffic using a Uu interface. Requires modifications to the UE or H(e)NB architecture, in addition to the (e)NB.

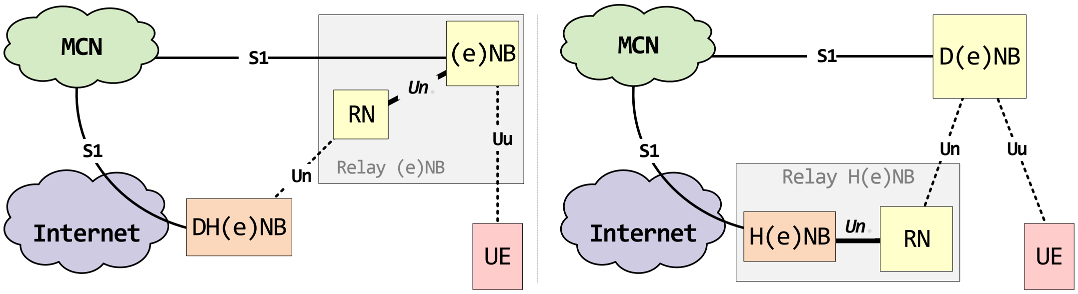
2.4.2 FemtoHaul using the Uu Interface

The first implementation of FemtoHaul replicates the basic concept presented in Section 2.2, where a UE associated with one of the femtocells in the region acts as a relay between the (e)NB and its associated H(e)NB to offload traffic. In other words, as shown in Figure 2.7 (a), the offloading of traffic in this scenario is performed via a Uu interface.

The Relay UE contributes its resources, such as bandwidth and battery, to help an unrelated UE with no benefit of its own. Thus, in this scenario, the UEs associated with femtocells may have to be given incentives to promote acting as Relay UEs. The traffic on the Uu interface between the Relay UE and the H(e)NB consists of traffic it is forwarding and the traffic of its own. It may be desirable to prioritize these two kinds of traffic. In systems with open access femtocells, in which provisions for prioritizing of traffic from registered and unregistered users already exist, they could reused for this purpose.

The architecture of the UE in this implementation needs to be modified for it to be able to act as a Relay UE. The Relay UE should be able to forward requests from the (e)NB to the H(e)NB. It should also be able to recognize traffic from H(e)NB that needs to be forwarded to the (e)NB. Finally, it should either be able to switch between the two Uu connections so that it can forward traffic between the (e)NB and the H(e)NB. Such fast switching associations may require new protocols. The introduction of an additional Uu link that uses the licensed band also adds to the interference management issues in the network, apart from consuming more air resources. Since the Relay UE is not the property of the service provider, there may also be issues regarding maintenance of security of the data that passes through it.

The above complications brought about by the proposed implementation of FemtoHaul can easily be avoided by some simple modifications, as shown in Figure 2.7 (b), that make the H(e)NB additionally perform the functions of the Relay UE. We call this new device a Relay H(e)NB. The direct link between the Relay H(e)NB and the (e)NB



(a) The (e)NB includes RN hardware that helps connects to the H(e)NB for traffic offload.

(b) The H(e)NB includes RN hardware that helps connect to the (e)NB for traffic offload.

Figure 2.8: Offloading of traffic using a Un interface. Requires modifications to the (e)NB or the H(e)NB architectures.

can stay alive at all times providing continual offloading opportunities irrespective of the availability of associated UEs. Continuous association of the Relay H(e)NB with the (e)NB will consume minimal resources and when not in use, the Relay UE component of the Relay H(e)NB will be in idle mode. Figure 2.7 (b) also shows the Uu interface between the Relay UE and the H(e)NB implemented with an internal *Uu* interface within the new Relay H(e)NB, that is a permanent hard wired interface, which thus saves air resources and reduces interference, and avoids the requirement for the Relay UE to continually switch between the two Uu interfaces. This modified implementation of FemtoHaul using the Uu interface also avoids requirements of modification to the UE architecture which is extremely difficult to realize in the real world. Instead it only requires the architecture of the H(e)NB to be modified, which is comparatively easier to realize, since the H(e)NB is the property of the service provider. Since the Relay UE component of the Relay H(e)NB is like any other UE, the (e)NB is already able to talk to the Relay H(e)NB. Also, since there is no traffic from the Relay UE itself on the Relay UE-H(e)NB link, there are no issues relating to prioritizing of traffic.

2.4.3 FemtoHaul using the Un Interface

We now present an alternate implementation of FemtoHaul where the offloading uses the Un interface to transport traffic from the (e)NB to the H(e)NB. The Un interface is currently defined by 3GPP for relay applications between an RN and a D(e)NB as shown in Figure 2.6. Figure 2.8 (a) illustrates a realization of a direct link between the H(e)NB and the (e)NB. When the (e)NB chooses to offload traffic and picks a femtocell to offload it to, it may connect to the corresponding H(e)NB via a Un based interface, thus acting as a Relay (e)NB. This implementation is different from the current concept of relay in 3GPP LTE, since here the (e)NB acts as a relay rather than as donor, i.e.,

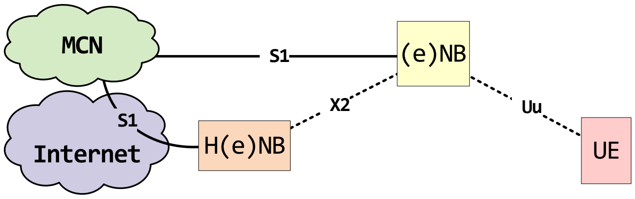


Figure 2.9: Offloading of traffic using a X2 interface. Requires suitable PHY layer realization of the X2 interface for H(e)NBs.

the (e)NB does not proxy the S1 signaling. Instead, the H(e)NB acts as an S1 signaling proxy via its backhaul. Once associated, the Relay (e)NB places a request for traffic to the H(e)NB, presented as the Donor H(e)NB (DH(e)NB), just like a UE associated with the femtocell would normally do. From a high level view, it may seem as if the (e)NB acts as a proxy for the UE it is serving, by forwarding its request to a femtocell. To the UE, it appears as though it is the (e)NB that is serving it, and to the DH(e)NB, it appears as though a UE (which is actually the (e)NB) is requesting traffic.

This architecture can be seen as a variant of the LTE relay architecture from Figure 2.6. However, here, it is the (e)NB that functions like the RN towards the UE, and as a UE to the DH(e)NB. This implementation of FemtoHaul needs Un interfaces between the (e)NB and each of the H(e)NBs in the region to stay alive all the time. This is to avoid delay and wastage of resources if the (e)NB were to set a Un interface with an H(e)NB up for every offload attempt.

Figure 2.8 (b) shows an alternate implementation of FemtoHaul with offloading via Un, where the Relay H(e)NB functions just like the RN in Figure 2.6, with an additional S1 interface with the MCN via the internet. This time, when the (e)NB chooses to offload traffic to a femtocell, it talks to the Relay H(e)NB directly. This implementation of FemtoHaul aims to minimize the modifications in the architecture of the (e)NB and focus most modifications to the architecture of the H(e)NB, as opposed to the other way around for the previous implementation.

2.4.4 FemtoHaul using the X2 Interface

Finally, we present a possible realization for FemtoHaul, where offloading between the macrocell and the femtocells occurs via X2 interfaces, which is normally an interface between any two (e)NBs, as shown in Figure 2.6. The X2 interface is used essentially when handover occurs between macrocells. The source (e)NB transmits remnant data packets to the target (e)NB after completion of handoff using the X2 interface. The details of the handover procedure are further discussed in Section 3.2.1 in Chapter 3.

The same X2 interface can be used by an (e)NB to offload traffic on to an H(e)NB.

The H(e)NB in turn, may forward traffic between this X2 interface and its backhaul. This implementation of FemtoHaul depends on how exactly, if at all, the X2 interface connecting H(e)NBs is realized in the network. Although the X2 interface between (e)NBs has already been standardized in 3GPP LTE, the existence of the same for H(e)NB is often questioned. In the presence of an X2 interface for H(e)NBs however, the above implementation of FemtoHaul becomes valid and feasible. Since the X2 interface specifications do not impose any restrictions on its PHY layer, it is possible to implement X2 sharing its PHY layer with the Uu (or even the Un for LTE relay) interface. In other words, the X2 interface could be implemented over licensed cellular spectrum. Note that the (e)NB and the H(e)NBs are already capable of using these frequencies. The X2 interface may also be built on a wired IP routed network independent of the cellular network. For this, either existing infrastructure (from ISPs for example) can be reused, or new infrastructure may be installed.

Chapter 3

Near Instantaneous Handover between Femtocells

The tremendous rate of evolution of modern cellular networks over the past few years has led to traffic demands pushing against its capacity limits, in spite of a manifold increase in over the air and backhaul capacities. Finding ways to offload macrocell traffic to alleviate congestion is fast becoming a priority, the most effective way of doing which is by adding pico and femtocells. WiFi hotspots and femtocells in today's cellular networks are alleviating the heavy burden on the macrocells.

A femtocell has a base station - the Home-(enhanced) NodeB (H(e)NB) - that is installed by the user at home. The H(e)NB backhaul connects to the Mobile Core Network (MCN) via the public internet using the user's broadband service modem, much like a WiFi access point. Femtocells are overlaid on the existing cellular network. As described in Chapter 2, H(e)NBs have a small cell range, and a comparatively lightly loaded and independent backhaul. As a result, when compared with the backhauls of the macrocell base stations, also known as the (enhanced) NodeB ((e)NB), femtocells provide a high-speed data connection to a user's mobile device, called the User Equipment (UE) [1]. We also continue to assume the femtocells to be open access as before.

As shown in Figure 3.1, Cisco predicts cellular data traffic to grow almost seven-fold in the next four years [18]. Thus, even with only a third of the cellular users being mobile, data traffic from *mobile* users will approach the levels of current *total* data traffic within the next two years. In recent years, WiFi hotspots, along with femtocells in cellular networks have helped offload macrocell traffic to a large extent by serving stationary indoor UEs. However, this is far from being sufficient. Hence the number of users being benefiting from femtocells must be further increased by allowing even the mobile users to associate with them, in order to maximize the advantage of femtocells. To make this possible, it is essential to enable all mobile UEs to handover to femtocells on their paths.

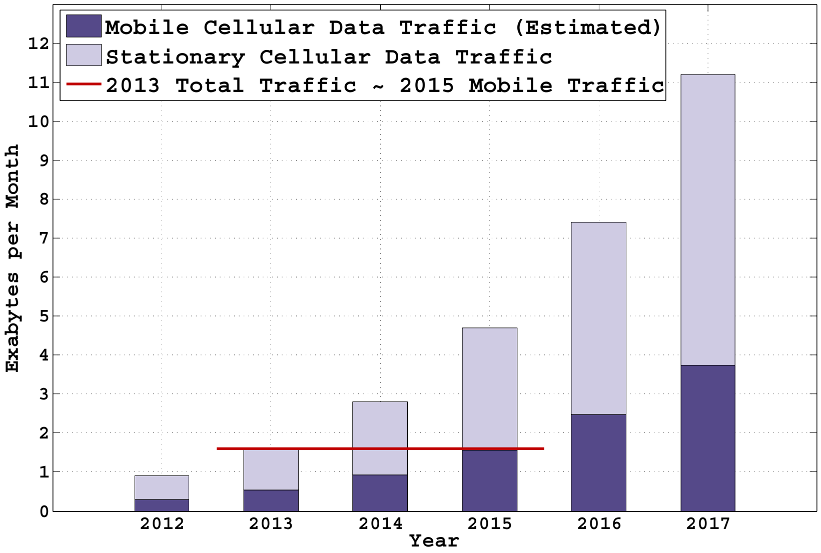


Figure 3.1: Cellular data traffic projections for the years 2012-2017 [18] separated into mobile and stationary components, assuming a third of cellular users to be mobile at any time. Mobile cellular data traffic is expected to become comparable to total cellular data traffic in 2013, by 2015.

Handover is defined as the process of switching from associated cell to another for a UE. A handover process is efficient when it is fast enough to maintain seamlessness to the applications running on the UE. While the Legacy Handover procedures are optimally designed for handover between macrocells, they fall short when applied to the newly introduced femtocells in the network. Femto to femto handovers are inherently more frequent, given the small size of femtocells, and take longer, given their high latency backhaul connection to the MCN. The Legacy Handover procedure typically takes less than 100 ms for a handover between macrocells [19] because of the minimal latency between the (e)NB and other entities of the MCN [20]. However, as we will show here, the Legacy Handover procedure, when applied on handovers between femtocells, can take up to 1.74 s. This is because of the latency experienced by messages having to reach the MCN via the public internet. It is because of this slow handover procedure that mobile users passing through femtocells either end up spending most of their time going through handover, or having to skip femtocells because they simply cannot keep up. Thus it has become necessary to re-design the existing handover procedures so they can be fast and efficient for handovers between femtocells as well.

The inefficiencies in handover procedures as applied to femtocells have started drawing research attention lately. Apart from some recent work trying to improve handover efficiency in Wireless Metropolitan Area Networks (WMAN) by using a cross-layer design for fast IPv6 handover [21], and to draw attention to various handover related research issues in WiMAX networks [22], there has also been some focus on handovers between WLANs and cellular networks using mobile IPv6 [23], and reducing latency in IPv6 handovers with localized authentication [24]. Other ideas such as proactively triggering handover procedures by predicting user mobility [25], modification of network architecture and signal flows to avoid unnecessary handovers [26][27] and caching of recently visited cell information to reduce scanning time [28] have also appeared recently.

We begin with proposing a *prefetch-based fast handover* procedure that substantially reduces the time spent in signaling as well as data exchange between femtocells and the MCN during the handoff, with very little change to the existing MCN architecture [29]. This procedure prefetches higher layer data to nearby femtocells by decoupling portions of the existing handover procedures that occur before and after the actual handoff process, from the handoff process itself. We then go a step further, and propose the introduction into the LTE architecture, a new fast control plane interface between two H(e)NBs. With the help of this new interface, which only requires a small proportion of network resources; we build on our Fast Handover procedure to achieve *near instantaneous* femto-femto handovers. This Near Instantaneous Handover procedure minimizes any interruptions to the higher layer applications, thus making the handover truly seamless. Also, the negligible time spent in handover ensures a longer period of time spent by UEs remaining associated with femtocells, which in turn not only helps offload more of the macrocell traffic on to femtocells, but also effectively increases the total network capacity. In addition, the introduction of the new interface also helps reducing the wastage of network resources due to prefetching of data to femtocells.

3.1 Fast OTA Interface between Femtocells

We begin this section with a brief overview of the current LTE network architecture, following which we propose the introduction of a new interface into this architecture discussing its requirements, benefits, and drawbacks. We also present possible realizations of the proposed interface within the scope of reasonable assumptions.

3.1.1 LTE Network Architecture

The LTE network architecture includes components from within the MCN extending over to the public internet and other networks [30]. The entities of the architecture relevant to handover, along with various interfaces connecting them, are illustrated in Figure 3.2. The Serving Gateway (SGW) supports user data and provides routing and forwarding functionality between (e)NBs (or H(e)NBs) and the Packet Data Network (PDN). It also acts as the mobility anchor during handovers between LTE and other 3GPP sys-

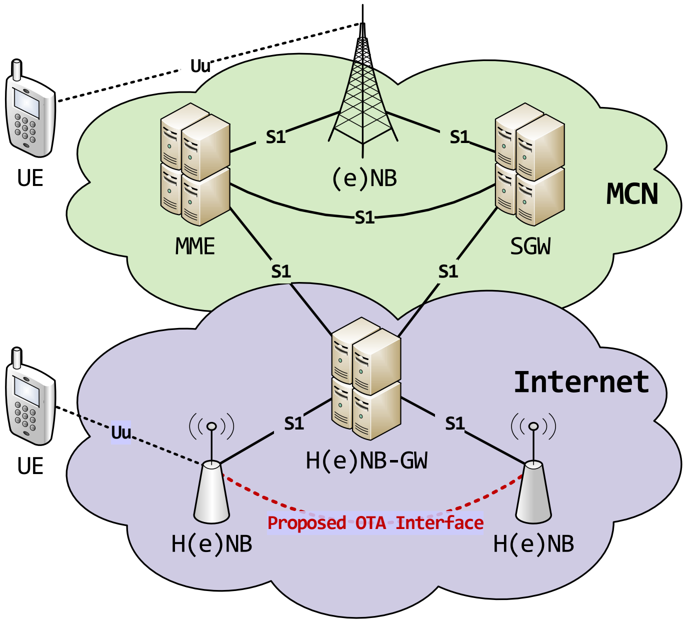


Figure 3.2: Relevant components of 3GPP LTE architecture with the proposed OTA Interface. The solid and dashed lines represent wired and over-the-air interfaces, respectively. Prefetch-based Fast Handover procedure, presented in Section 3.2.3 works on the existing standard LTE architecture. The newly proposed OTA interface is only used by the Near Instantaneous Handover procedure presented in Section 3.2.4.

tems [31]. All base stations ((e)NBs and H(e)NBs) connect to the MCN and the PDN through the SGW for control signaling. The Mobility Management Entity (MME) provides the control plane function for mobility between LTE and other access networks, and is responsible for choosing the right SGW for a UE and its authentication. A H(e)NB Gateway (H(e)NB-GW) is used to provide interface scalability and support to a large number of H(e)NBs [32]. The H(e)NB-GW works as a concentrator for the control plane.

3.1.2 Direct Femto-Femto Interface

We define the *associable region* of a femtocell as the region where the signal from its H(e)NB is stronger than that from the umbrella (e)NB as well as all neighboring H(e)NBs, as in [29]. In other words, in normal circumstances, a UE in the associable region of a femtocell always associates with it. We then define the *proximity region* of a femtocell to consist of its associable region and the region surrounding it, where the strength of the signal from its H(e)NB is within δ of the strongest of the signals from the umbrella (e)NB and the neighboring H(e)NBs. Thus, while associable regions of various femtocells are exclusive to each other, their proximity regions overlap. With these definitions in mind, as shown in Figure 3.2, we propose to introduce a new control plane interface between a pair of femtocells with a set of basic minimal requirements as listed below.

- The interface is required for every pair of femtocells that have overlapping proximity regions. In other words, this interface is not required between femtocells that are far away from each other.
- The interface between a pair of femtocells is activated only when there is an active UE located in the overlapping portion of their proximity regions.
- While active, the interface is only required to bear one control plane message for every UE entering the overlapping portion of the proximity regions of the corresponding femtocells, and three control plane messages per handover between them.
- The interface must be fast enough to transmit a control plane message within a few milliseconds.

Given these minimal requirements from the proposed femto-femto interface, the reader may note here that the introduction of the new interface, in any kind of realization, would limit the changes in the existing LTE architecture only to the H(e)NBs.

The LTE architecture includes a direct interface between two (e)NBs for buffer and context transfer during handovers, standardized as an X2 interface [33]. However, an X2 interface for H(e)NBs has not been standardized yet. Since the standard does not impose any restriction on the physical layer realization of X2, any realization of X2 for H(e)NB, if added to standards, that satisfies the above listed conditions may also be used as the fast femto-femto interface we propose here. Current proposals for the implementations of X2 go through the existing S1 interface via the H(e)NB-GW [34] shown in Figure 3.2.

3.1.3 Realizations of the Femto-Femto Interface

We now describe two possible realizations of the interface within the existing LTE architecture.

Over-the-Air Femto-Femto Interface

One of the most suitable realizations of the proposed fast femto-femto interface can be over-the-air (OTA), using the same licensed band spectrum as that used by the Uu interface between the base stations and the UEs [30]. Hence the interface closely resembles a Uu interface in the control plane. As specified earlier, the interface remains inactive when there is no UE in the intersection of the proximity regions of the two femtocells. The activation of the interface is triggered when the first UE enters this region (discussed in Section 3.2.4). When this happens, the H(e)NB the UE is not associated with initiates the interface with the associated H(e)NB. Since the interface resembles a Uu interface, when active, transmission of a control plane message only takes a few milliseconds.

Even though the OTA realization of the femto-femto interface has certain obvious benefits, it does have a few drawbacks too. Firstly, it requires a modification of the H(e)NB architecture, which now needs to incorporate the new interface. H(e)NBs will also need to be able to make more powerful transmissions, now that they need to communicate, although rarely, with other H(e)NBs having overlapping proximity regions with them. This may also add to the existing interference management issues with femtocells. However, the benefits of adding the interface will very likely outweigh these drawbacks.

Femto-Femto Interface via Local ISP Network

A second possible realization of the proposed fast femto-femto interface can be via the femtocell backhauls. Since the only requirement from the realization is that the interface should be fast enough, and since the interface only connects femtocells geographically close to each other, the implementation of the interface via the backhauls of the femtocells facilitated by the ISP in the region is very feasible. This realization of the femto-femto interface, among others discussed in Section 2.4.4, can also work as an X2 interface between two H(e)NBs.

3.2 Near Instantaneous Handover

We will begin this section with a description of the Legacy LTE Handover procedure, and a discussion of its shortcomings when applied to handovers between femtocells. We then briefly describe the prefetch-based Fast Handover procedure [29]. We will then show how the newly proposed femto-femto interface above can be used to build on the prefetch-based Fast Handover to achieve Near Instantaneous Handover.

3.2.1 The Legacy LTE Handover Procedure

In an LTE network, a UE uses a *Measurement Report* message to periodically report signal strengths of surrounding base stations to its associated (e)NB or H(e)NB by periodically scanning all available channels. When a *Measurement Report* suggests that the best signal received by the UE is from an (e)NB or H(e)NB that it is not associated with, a positive *Handover Decision* is taken by the associated (e)NB or H(e)NB thus triggering a handover procedure [35]. Figure 3.3 illustrates the Legacy Handover procedure employed in LTE networks extended on to handover between femtocells. Control

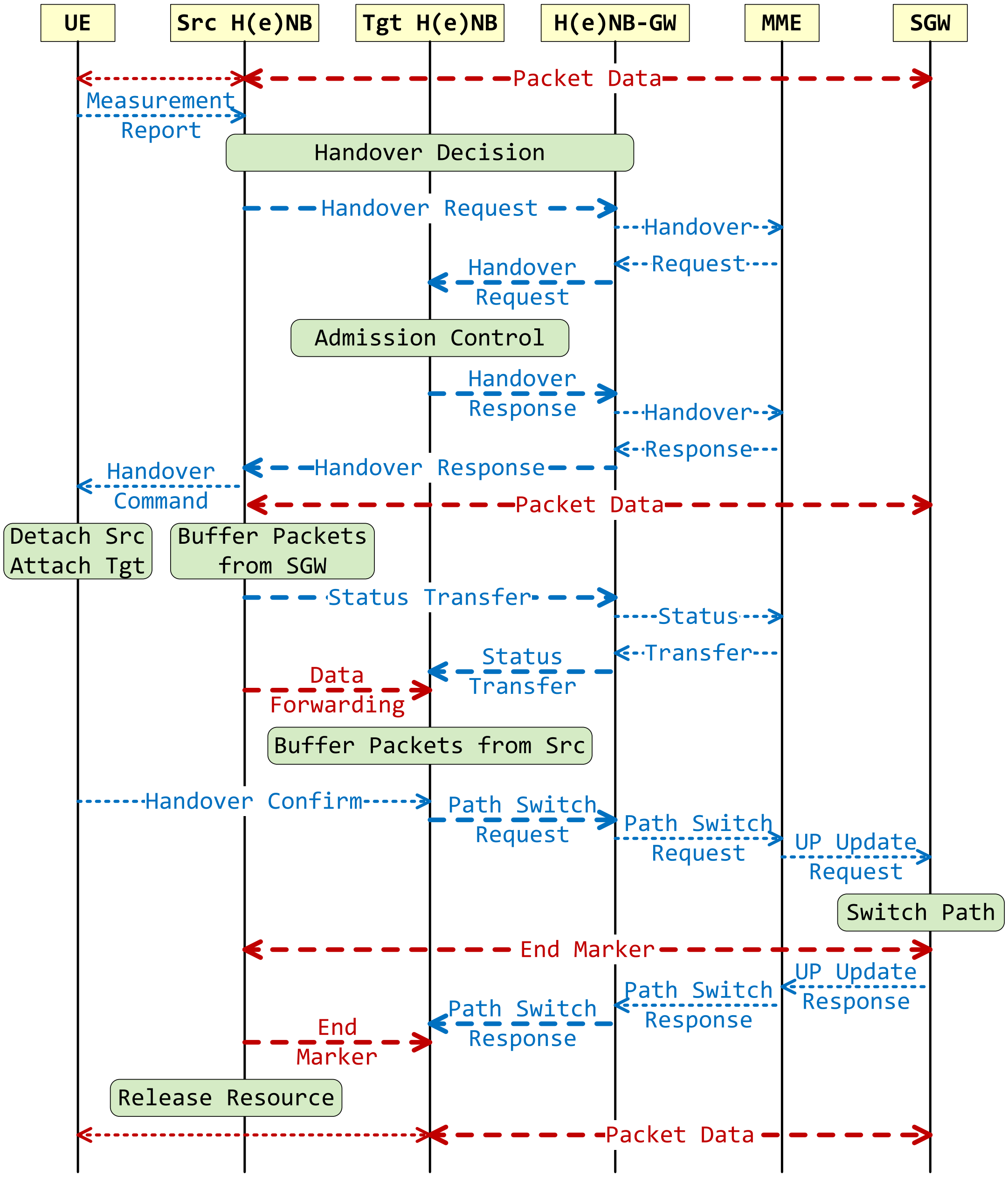


Figure 3.3: The Legacy Handover Procedure applied to handover between femtocells. The UE is being handed-over from the Src H(e)NB to the Tgt H(e)NB. Control message transmissions are shown with blue dashed lines and data transmissions in red. The bolder dashed lines represent messages transmitted via the internet with higher latency.

message exchanges are shown in blue, while the data packet exchanges are in red. The slower transmissions going via the public internet are shown using bold dashed arrows. The UE associated with the *Src H(e)NB* is being handed-over to the femtocell of the *Tgt H(e)NB*.

- Following a positive *Handover Decision*, Src H(e)NB sends a *Handover Request* message to the Tgt H(e)NB via the MME.
- The Tgt H(e)NB then performs *Admission Control* for the UE, and responds with a positive *Handover Response* message.
- The Src H(e)NB, having received the *Handover Response*, issues the *Handover Command* to the UE, which then tries to handoff to the new femtocell detaching from it.
- In the meantime, the Src H(e)NB starts to buffer the application layer data it continues to receive from the SGW. It also sends out a *Status Transfer* message to the Tgt H(e)NB via the MME and then begins forwarding the data to it.
- The Tgt H(e)NB, having received the *Status Transfer* message, begins to buffer the data being forwarded by the Src H(e)NB. It also accepts the *Handover Confirm* message from the UE when feasible, thus allowing the UE to associate with itself.
- After the UE is attached to the Tgt H(e)NB, it begins transmitting the buffered data to the UE. At this time, the data goes from the SGW to the Tgt H(e)NB via the Src H(e)NB, effectively traversing through the public internet twice.
- Finally, the Tgt H(e)NB issues a *Path Switch Request* to the SGW, which then updates the user plane so that it now streams the data directly to the Tgt H(e)NB, and responds with a *Path Switch Response* message.
- The SGW also sends an *End Marker* data packet to the Src H(e)NB marking the point of switching of the user plane. The Src H(e)NB forwards this *End Marker* packet to the Tgt H(e)NB when it is done forwarding all the data from the SGW it has been buffering.
- The Tgt H(e)NB, having received the *End Marker* packet, begins to transmit data received from the SGW directly to the UE. This marks the end of the Legacy Handover procedure.

The Legacy Handover procedure has been sufficiently optimized for a cellular network consisting solely of macrocells [20]. However, when this procedure is extended on to femtocells, the process becomes unacceptably long due to the large number of control messages having to go through the public internet, as shown in Figure 3.3 with bold dashed arrows. Consequently, the interruption experienced by the UE during the handover, when it is unable to receive data at all is also lengthened. Furthermore, the period of time during which the data traverses the internet twice before being transmitted to the UE introduces additional interruptions.

3.2.2 Speed of UE in Handover

In a network consisting solely of macrocells, for a UE to stay connected, it must always be associated with the macrocell it is in. Since Legacy Handover is originally designed for handovers between macrocells, it naturally forces the UE to handover to a new macrocell when the UE moves into it, irrespective of the speed of the UE, to maintain connectivity. However, with the introduction of femtocells into the network, a UE within a femtocell has an option to either associate with the femtocell or with the umbrella macrocell. Although associating with the femtocell may be the favorable action for a UE in most cases, sometimes associating with the umbrella macrocell instead may be more favorable. This may happen when the user is moving too fast to be able to handover to femtocells on its path since the user spends insufficient time within their ranges. Thus, we propose to factor in the speed of the mobile UE for handovers in cellular networks with femtocells. A UE is said to be in *Swift Mode*, when its estimated speed is higher than a pre-defined threshold θ, and in *Free Mode* otherwise. Only when a UE is in Free Mode does the *Handover Decision* function allow it to handover to a femtocell. When a UE is in Swift Mode, it is only allowed to handover to a macrocell.

The speed of the mobile UE can be estimated based on the *Measurement Report* messages from it. The speed threshold θ can be set based on the length of the handover procedure. A θ value that is too high may cause unreliable UE connectivity because of failed handover attempts, while θ value too low results in missed opportunities to associate with femtocells, increasing the burden on the macrocell. Some of the present day service providers only allow stationary users to be served by femtocells because of the slow handover, which is equivalent to setting an extremely low threshold θ. We aim to maximize θ within the above bounds so that the potential of femtocells in the network can be tapped to the maximum.

3.2.3 The Prefetch-based Fast Handover Procedure

In prefetch-based Fast Handover, parts of the Legacy Handover procedure are segregated to prefetch higher layer data to all H(e)NBs in the proximity of the UE. The Admission Control process (taking place before the actual handoff), and the Path Switching process

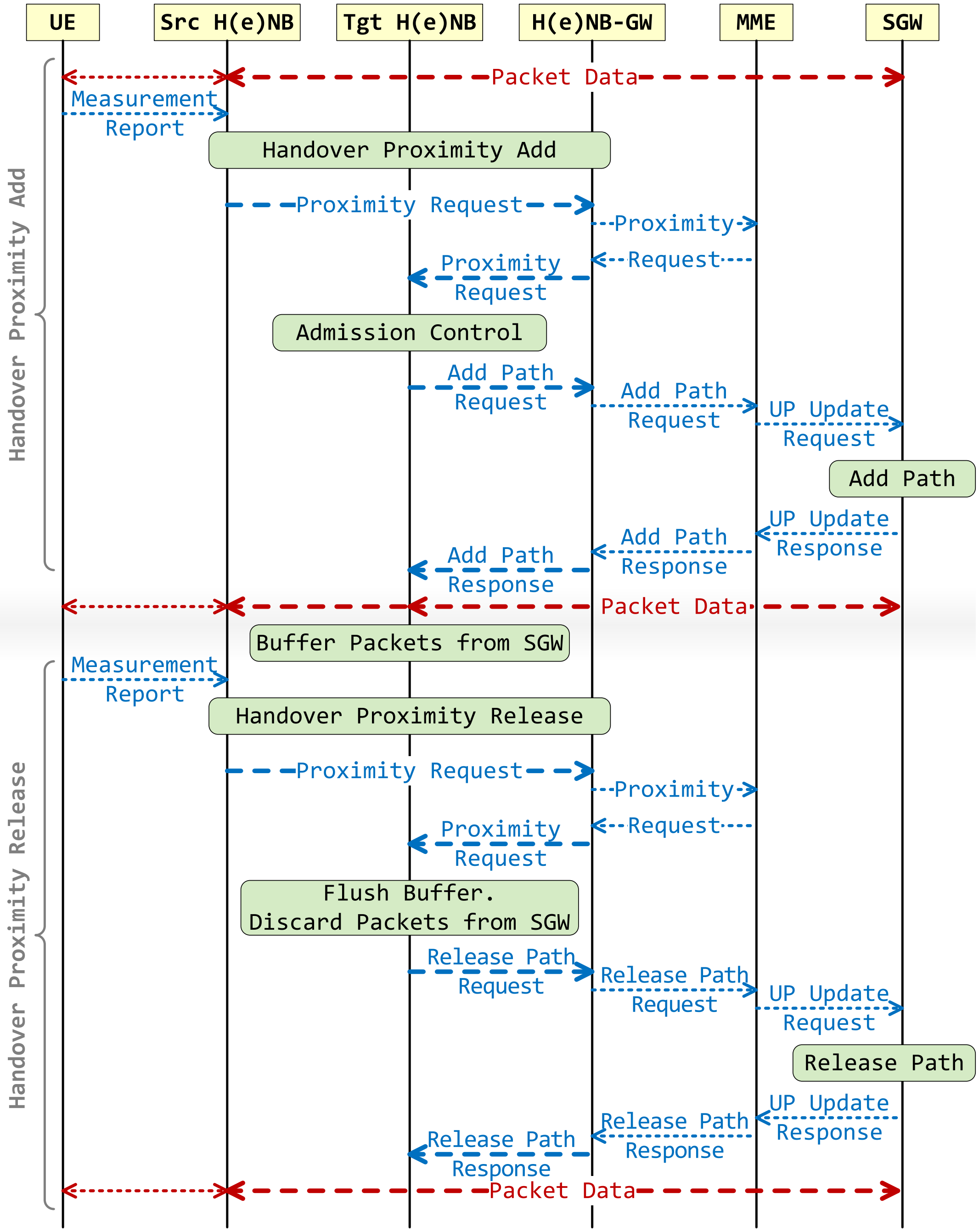


Figure 3.4: The Proximity Add/Release Processes in the prefetch-based Fast Handover procedure. The UE, associated with Asc H(e)NB, moves into/out of the proximity region of Prx H(e)NB. Control message transmissions are shown with blue dashed lines and data transmissions in red. The bolder dashed lines represent messages transmitted via the internet with higher latency.

(taking place after the actual handoff) in the Legacy Handover procedure are tightly coupled with the actual handoff itself. In prefetch-based Fast Handover, the actual handoff process is decoupled from these processes taking place before and after it. When a UE is in the proximity region of one or more H(e)NBs, copies of higher layer data are streamed from the SGW to each of those H(e)NBs. While only the associated H(e)NB transmits the data to the UE, the others buffer this data. Each of the H(e)NBs in proximity of the UE is ready for the UE to be associated with it in the near future and stands prepared for the handover. The actual handoff however, only takes place when the UE moves into the associable region of a femtocell much like in Legacy Handover.

The Proximity Add process, shown in Figure 3.4, takes place when a UE first moves into the proximity region of a femtocell, and the Proximity Release process takes place when a UE moves out of it. A UE periodically sends *Measurement Report* messages to its associated H(e)NB (labeled as *Asc H(e)NB*) just like in Legacy Handover. The Asc H(e)NB makes the *Handover Proximity Add* decision when the *Measurement Report* received from the UE suggests that it has now entered the proximity region of a new H(e)NB (labeled as *Prx H(e)NB*), which triggers the Proximity Add process.

- When a positive *Handover Proximity Add* decision is made by the Asc H(e)NB, it sends a *Proximity Request* message to the Prx H(e)NB via the MME.

- The Prx H(e)NB then performs *Admission Control* for the UE, and issues a *Add Path Request* message to the SGW via the MME. The MME keeps track of the femtocells in proximity of the UE.

- The SGW responds with an *Add Path Response* message to the Prx H(e)NB with information about the new data stream to be initiated, and then creates a copy of the data stream already being sent to the Asc H(e)NB and starts streaming it to the Prx H(e)NB simultaneously.

- The new H(e)NB starts to buffer the received data to prepare for a possible near future association with the UE.

Similarly, the Proximity Release process is triggered when a UE moves out of the proximity region of a femtocell resulting in a positive *Handover Proximity Release* decision by the Asc H(e)NB. Here the Prx H(e)NB flushes out the buffered data and the SGW stops streaming duplicate data to it.

The Proximity Add and Release processes occur decoupled from the actual Handoff process shown in Figure 3.5. Here, the H(e)NB the UE is handing-over from is labeled as the *Src H(e)NB* and that the UE is handing-over to as the *Tgt H(e)NB*.

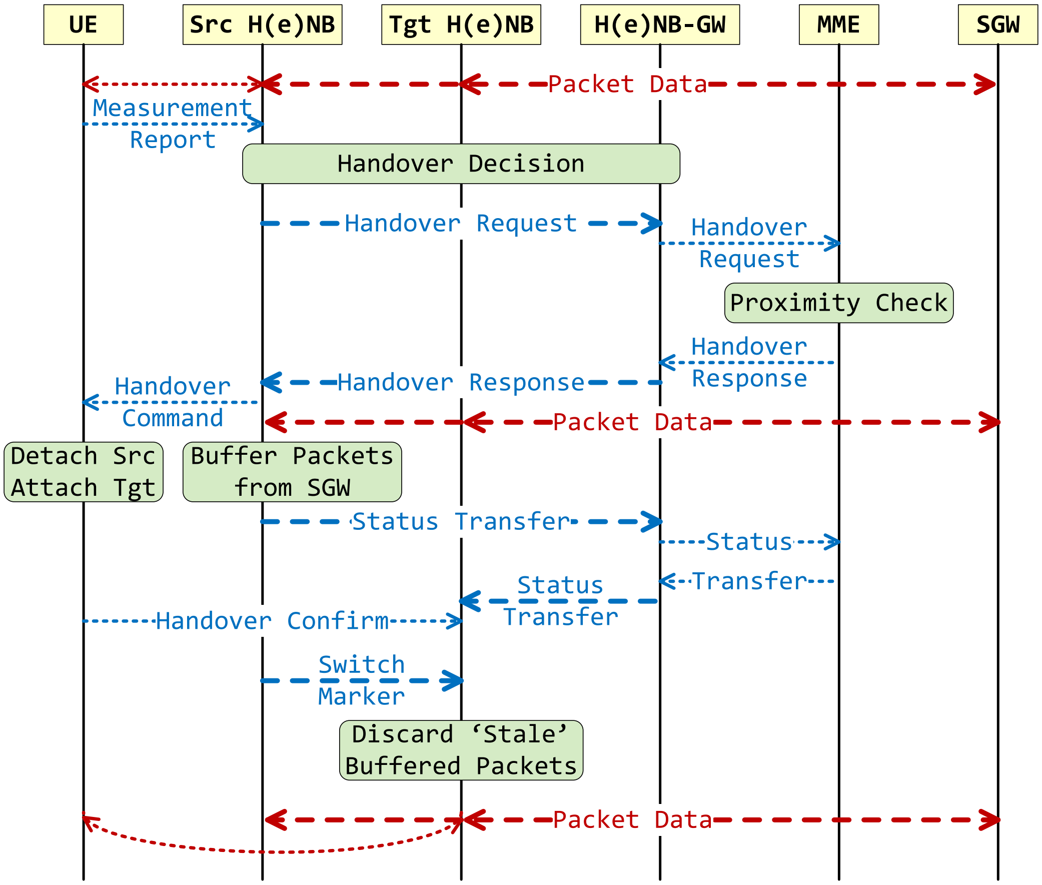


Figure 3.5: The Handoff Process in the prefetch-based Fast Handover procedure. The UE is being handed-over from the Src H(e)NB to the Tgt H(e)NB. There is no change in user planes from the MCN during the Handoff process. Forwarding of data between the Src H(e)NB and the Tgt H(e)NB is completely avoided, replaced by transmission of a single Switch Marker control message over the internet. Control message transmissions are shown with blue dashed lines and data transmissions in red. The bolder dashed lines represent messages transmitted via the internet with higher latency.

- Following a positive *Handover Decision* resulting from a *Measurement Report* suggesting that the best signal received by the UE is from the Tgt H(e)NB rather than the Src H(e)NB [35] then, just like in Legacy Handover, the Src H(e)NB sends a *Handover Request* message to the MME.

- The MME then performs the *Proximity Check* function to ensure that the Tgt H(e)NB has already gone through the Proximity Add process for the UE, and responds with a positive *Handover Response* message to the Src H(e)NB.

- The Src H(e)NB, having received the *Handover Response*, issues the *Handover Command* to the UE, which then detaches from the Src H(e)NB and tries to handoff to the Tgt H(e)NB, just like in Legacy Handover.

- The Tgt H(e)NB, which is already buffering higher layer data meant for the UE only needs a marker to know the point from which data needs to be transmitted to the UE. The Src H(e)NB sends a *Status Transfer* message to the Tgt H(e)NB via the MME, followed by this *Switch Marker* packet.

- Upon receiving the *Switch Marker* packet, the Tgt H(e)NB discards those packets which have already been received by the UE (Stale Buffered Packets), and starts transmitting new data. This marks the end of the Handoff process in Fast Handover procedure.

By decoupling the Proximity Add and Release processes from the actual Handoff process, the Fast Handover procedure effectively separates out the back-end tasks of the procedure from the front-end ones. In other words, the changes of user plane that involve the MCN are performed in the Proximity Add and Release processes, while the switching of association of the UE is performed in the Handoff process. While the Proximity Add and Release processes remain transparent to the UE, the Handoff process remains transparent to the SGW. Since the femtocell connects to the MCN via the public internet, it is the back-end tasks of the handover procedure that consume more time. Consequently, the Handoff process, which now involves significantly fewer message exchanges through the public internet, is much faster than the Legacy Handover procedure with smaller interruptions experienced by the UE. In addition, by avoiding data forwarding between the Src H(e)NB and the Tgt H(e)NB via the internet, and replacing it with a *Switch Marker* packet instead, Fast Handover also reduced the number of interruptions experienced by the UE during a handover.

While the above Prefetch-based Fast Handover procedure is significantly faster than the Legacy Handover procedure, as shown in [29], it is still significantly slower than

Legacy Handover between macrocells, and is still not fast enough to be completely seamless to higher layer applications. This is mainly because there are a few messages that still need to go through the internet during the handover. In the next section, we modify the Fast Handover procedure to address this issue.

3.2.4 The Near Instantaneous Handover Procedure

The Fast Handover procedure aims to quicken the handovers between femtocells while not disturbing the LTE network architecture even in the slightest. The only modifications it proposes are in the control and data message flows on the existing architecture. The Near Instantaneous handover procedure builds on the Fast Handover procedure by using the fast direct femto-femto interface proposed in Section 3.1 as shown in Figure 3.2. This implies a slight modification to the network architecture, strictly limited to the H(e)NBs.

The Near Instantaneous handover procedure, just like the Prefetch-based Fast Handover procedure, consists of the Proximity Add, Release, and Handoff processes. The Proximity Add and Release processes in Near Instantaneous Handover are almost identical to those in prefetch-based Fast Handover, as shown in Figure 3.6. For each active femto-femto interface of an H(e)NB in Near Instantaneous handover, it constantly maintains a list of all the UEs located in the intersection of proximity regions of the two femtocells.

In Near Instantaneous handover, the proposed femto-femto interface (in Section 3.1) may be initiated in the Proximity Add process, and may be torn down in the Proximity Release process. As specified in Section 3.1.2, the interface is initiated if the UE going through the Proximity Add process is the first UE to enter the intersection of the proximity regions of the Asc H(e)NB and the Prx H(e)NB. Otherwise, the interface must already be active (initiated earlier by another UE entering the region). The former scenario is represented by the heavily shaded region in Figure 3.6, the initiation of the femto-femto interface occurring after the UE passes *Admission Control* at the Prx H(e)NB. In the latter scenario, when the OTA interface already exists between the Asc H(e)NB and the Prx H(e)NB, the *Proximity Request* message can be directly transmitted to the Prx H(e)NB from the Asc H(e)NB rather than having to go via the MME through the public internet.

In the Proximity Release process in Near Instantaneous on the other hand, the *Proximity Request* message sent from the Asc H(e)NB to the Prx H(e)NB is always directly transmitted via the femto-femto interface unlike in Fast Handover. The Asc H(e)NB tears down the femto-femto interface after sending the *Proximity Request* if the UE is the last to leave the intersection of the proximity regions of the Asc H(e)NB and Prx H(e)NB.

Now that the Proximity Add process ensures that an active direct femto-femto in-

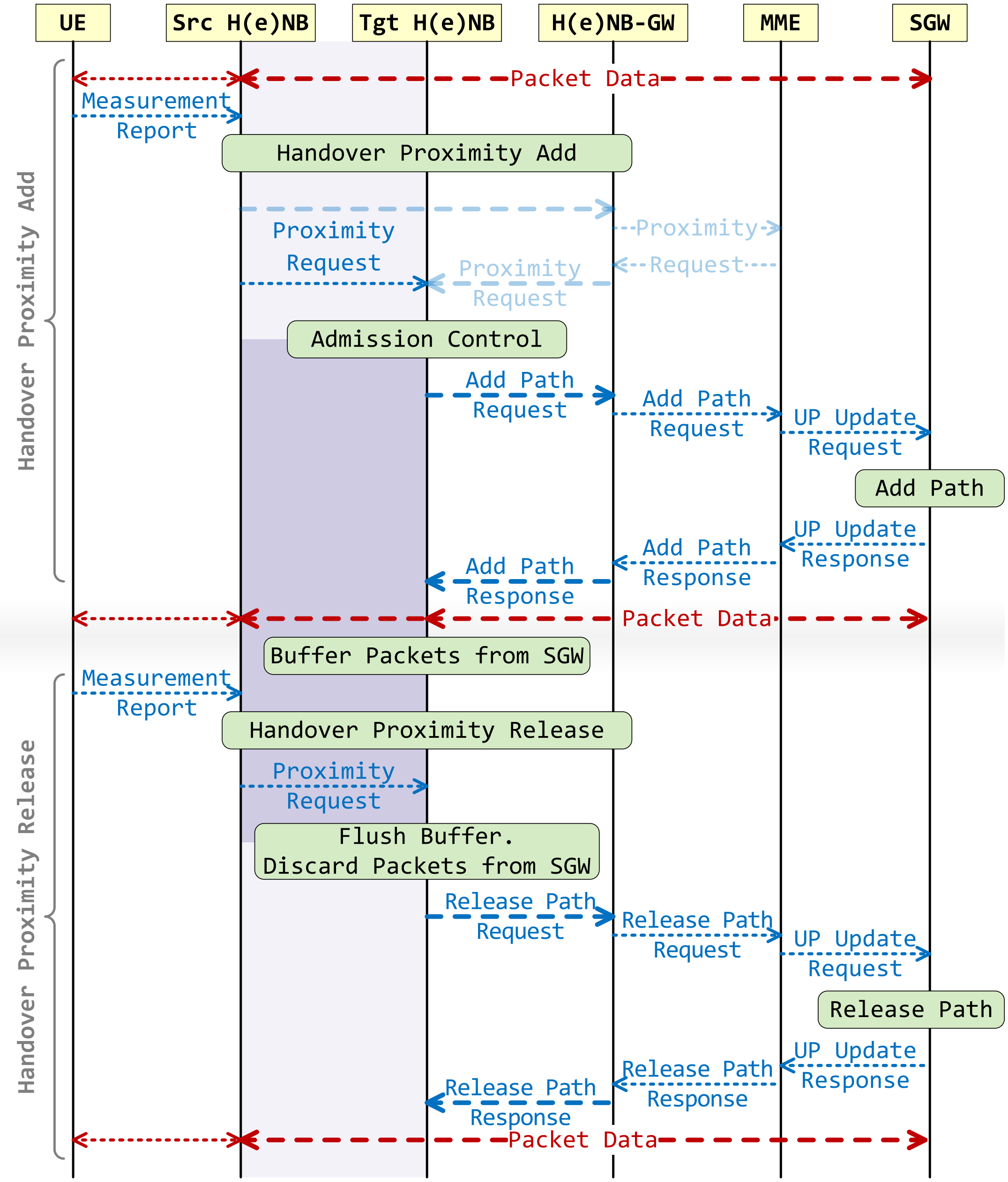


Figure 3.6: The Proximity Add/Release Processes in the Near Instantaneous Handover procedure. The UE, associated with Asc H(e)NB, moves into/out of the proximity region of Prx H(e)NB. The shaded region shows when the direct femto-femto interface proposed in Section 3.1, is active. The lightly shaded region is when the interface remains active because of an unrelated UE also present in the intersection of the proximity regions of the two femtocells, while the heavily shaded region shows when the interface is activated or torn down due to the UE in consideration. Control message transmissions are shown with blue dashed lines and data transmissions in red. The bolder dashed lines represent messages transmitted via the internet with higher latency.

terface between the Src H(e)NB and the Tgt H(e)NB for the Handoff process has been initiated, we propose the Handoff process for Near Instantaneous Handover, by making a few simple modifications to the Handoff process in prefetch-based Fast Handover. This new Handoff process is illustrated in Figure 3.7.

First, we propose to move the *Proximity Check* function to the Tgt H(e)NB. This is because the *Handover Request* and *Handover Response* messages can now be directly exchanged between the Src H(e)NB and the Tgt H(e)NB via the femto-femto interface. The H(e)NB is already capable of performing this function as it maintains a list of all UEs in its proximity at all times, as discussed above. Secondly, instead of the MME having to forward the *Status Transfer* message from the Src H(e)NB to the Tgt H(e)NB, the Src H(e)NB can now combine a copy of the *Status Transfer* message with the *Switch Marker* message and send it directly to the Tgt H(e)NB via the femto-femto interface. These two simple modifications result in a Handoff process that is completely devoid of waiting for any message having to go through the internet, thus making it *near instantaneous*.

As we will demonstrate in Section 3.3.2 ahead, a near instantaneous Handoff process helps increase the value of θ, the speed threshold for a UE to be in Free Mode. A higher value of θ ensures that a larger proportion of of mobile UEs can now handover to femtocells and associate with them, thus reducing the burden on macrocells further, as discussed in Section 3.2.2.

Apart from making the Handoff process near instantaneous, Near Instantaneous Handover also reduces the lengths of Proximity Add and Release processes by reducing the number of messages having to go through the public internet. Only in cases where the Proximity Add process involves initiation of the femto-femto interface, the Proximity Add process takes the same amount of time in Near Instantaneous Handover as it does in Fast Handover. The reduction in time taken for Proximity Add and Release procedures in turn allows for reduction in the value of δ, which is an indication of the width of the proximity region of the femtocell beyond its associable region. Consequently, this helps in reducing the wastage of network resources due to duplication of streaming data introduced by Fast Handover.

Please note here that the back-end of the network, including the SGW, is still completely unchanged by the Handoff process in Near Instantaneous Handover, and from the perspective of the UE, the entire Near Instantaneous Handover procedure continues to be exactly the same as the Legacy Handover procedure. Additionally, note that the newly introduced femto-femto interface is only used to carry three control messages during every Handoff process, and one control message during some of the Proximity Add processes.

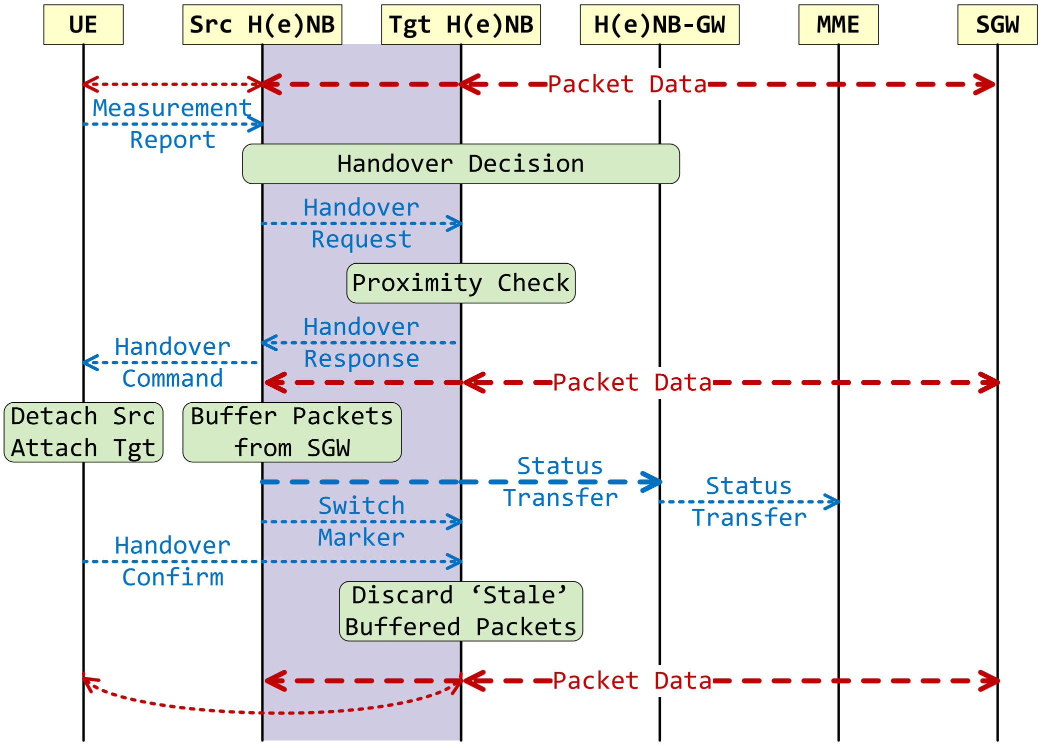


Figure 3.7: The Handoff Process in the Near Instantaneous Handover procedure. The UE is being handed-over from the Src H(e)NB to the Tgt H(e)NB. An active femto-femto interface between the Src H(e)NB and the Tgt H(e)NB is shown by the shaded region. This interface helps the process to completely avoid waiting for any message exchange over the public internet, thus making it near instantaneous. Control message transmissions are shown with blue dashed lines and data transmissions in red. The bolder dashed lines represent messages transmitted via the internet with higher latency.

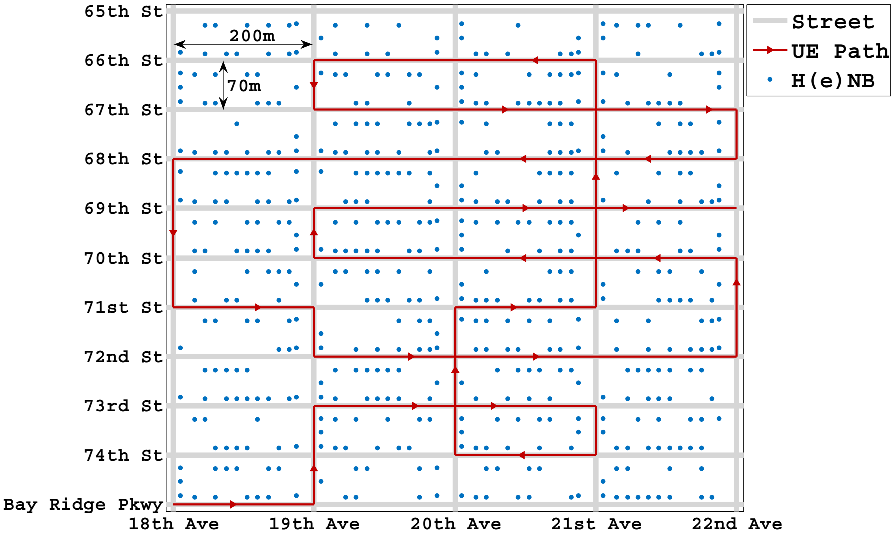


Figure 3.8: Area map representing a residential neighborhood in Brooklyn, NY, with 50% of residence buildings having femtocells installed. The UE drives through the streets in the neighborhood passing through 239 open access femtocells in this scenario.

3.3 Performance Evaluation

In order to evaluate the benefits of the Near Instantaneous Handover procedure, we simulated the system using the C programming language, which allows for more flexible implementation of procedure modifications, along with user mobility, compared to other simulation platforms. As shown in Figure 3.8, the implementation simulates a mobile UE driving at varying average speeds through streets in a specific residential neighborhood in Brooklyn, NY. A varying number of residential buildings in the neighborhood have femtocells installed, shown as blue dots. Half of the buildings in the neighborhood have femtocells installed in the setting shown in Figure 3.8. The UE is assumed to be running a single application layer session that continuously consumes downlink data. Thus a continuous data stream is assumed to flow to the UE throughout the duration of the simulation.

3.3.1 Simulation Settings

Network Settings

The entire area under consideration is covered under one macrocell of radius 400 m with its (e)NB located at the intersection of 20th Ave and 70th St. Selected residential buildings have H(e)NBs installed, following a random uniform distribution, with a femtocell range of 50 m. Given that a residential building may consist of several residences, the reader may note here, that even with femtocells installed in over 50% of the residence buildings in the region, the number would be less than the number of WiFi access points in similar suburban areas more than six years ago [36]. Downlink transmission powers of all base stations are set such that the received SNR at the cell edge is 6 dB, as in [37]. Also, the setting of outdoor path loss and noise power parameters, and calculation of throughput statistics are the same as in [37]. Two randomly located stationary UEs per femtocell are also introduced into the network in order to add background data traffic and interference.

Scheduling of Transmission and Reception

The simulation runs in slotted time, with a time slot length of 2 ms. The downlink channel is simultaneously used by the macrocell (when the UE is associated with it), and other randomly selected femtocells. Thus, the UE under consideration suffers from interference from the surrounding H(e)NBs. When associated with a femtocell, the UE is scheduled with equal priority as the two stationary UEs in the femtocell, following proportional fairness scheduling with a throughput window length of 1 s.

Control Message Exchanges

The UE sends Measurement Reports every 100 ms when associated with any base station. We assume the average message transmission time over the internet to be 200 ms, based on round trip time data collected from the internet. The message transmission time over the direct femto-femto interface depends on the congestion in the femtocell. The *Handover Decision*, *Handover Proximity Add/Release*, *Admission Control*, *Proximity Check*, and *Switch/Add/Release Path* functions involved in the handover procedures are assumed to consume 2 ms on an average.

User Mobility

The UE under consideration moves along a pre-defined specific path shown in Figure 3.8. At every turn, it randomly picks a speed, and moves at that speed until it arrives at the next turn. The speed threshold θ is assumed to be high enough for the UE to remain in Free Mode all the time unless otherwise mentioned. The proximity threshold, δ, is set such that the width of the proximity region of the femtocells beyond their associable region is approximately 15 m for prefetch-based Fast Handover, and 9 m for Near Instantaneous Handover.

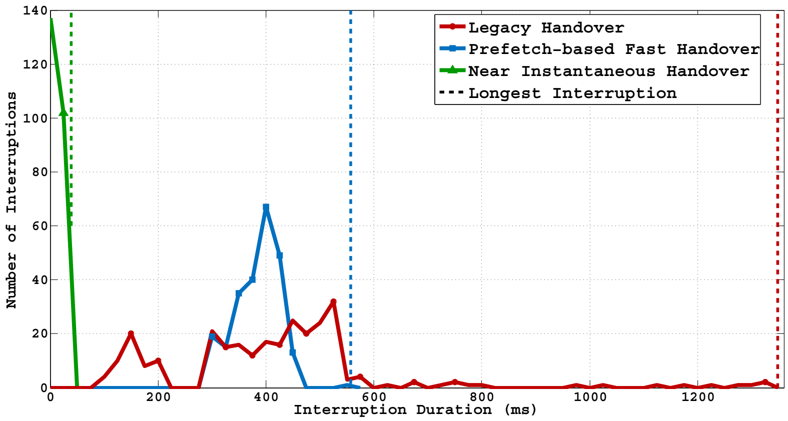


Figure 3.9: Distribution of durations of interruptions experienced by the UE while going through Legacy, prefetch-based Fast, and Near Instantaneous Handovers between femtocells on its path, respectively. The dashed lines indicate the longest interruption for a given kind of handover. The average speed of the UE is maintained at 35 mph, and 50% of residence buildings in the region have femtocells installed.

3.3.2 Results and Analysis

We simulate the system described above in Section 3.3.1 and measure four different criteria for evaluation of the proposed handover procedures, namely the lengths of the interruptions experienced by the UE, the improvement in network capacity, the number of missed handover opportunities, and the time spent in streaming data that remained unused to the femtocells. The results of these measurements are reported below.

Near Instantaneous handover resulting from the introduction of the direct fast femto-femto interface shows tremendous improvements in handover and interruption durations. While the average time taken for the full handover process across all simulation settings for Legacy Handover was found to be an unacceptable 1.81 s, for prefetch-based Fast Handover, it was much lower at 820 ms, which although significantly lower, cannot be termed as seamless for many applications. For Near Instantaneous Handover however, handover duration was found to be drastically reduced to a mere 28.3 ms, hence justifying the name. Thus, when compared with Legacy Handover and Prefetch-based Fast Handover, Near Instantaneous Handover was found to take a negligible amount of time. In fact the handover duration experienced by the UE in Near Instantaneous Handover is significantly lower than the 80-90 ms [20] experienced by the UE in even the much optimized Legacy Handover between macrocells.

Figure 3.9 compares the effective interruption lengths experienced by the UE, which is the time during these handovers for which the UE was unable to receive any higher layer data packets. The average speed of the UE was set at 35 mph, uniformly randomly distributed between 20 and 50 mph, and 50% of residence buildings had femtocells installed. Figure 3.9 shows the distribution of these interruptions. While the average interruption in Legacy Handover was 517.31 ms, it was significantly lower for prefetch-based Fast Handover at 396.72 ms, and negligibly low at 23 ms for Near Instantaneous Handover. Even the largest interruption experienced by the user, marked by the dashed lines in Figure 3.9, was only 39.47 ms for Near Instantaneous Handover as compared to 557 ms for Prefetch-based Fast Handover and over 1.3 s for Legacy Handover. The interruptions experienced by the UE in Legacy Handover are sometimes extremely high, mainly because of the UE skipping femtocells on its path, and therefore being unable to finish the handover procedure before moving out of the femtocell. The negligibly small interruptions experienced by the UE going through Near Instantaneous Handover are consequently transparent to almost any higher layer application, including demanding high bandwidth low-latency applications like video conferencing. In other words Near Instantaneous Handover is virtually completely seamless to any application.

Figure 3.10 compares the total amount of data served to the UE in systems with various modes of handover employed, with increasing density of femtocells in the region. The average speed of the UE was maintained at 35 mph as in the previous scenario. Even though Legacy Handover treats all UEs equally irrespective of their speed, we have included a version of Legacy Handover where the UE is not allowed to handover to femtocells when moving at speeds higher than 35 mph in the comparison, in addition to comparing the normal Legacy Handover with prefetch-based Fast Handover and Near Instantaneous Handover, for a better perspective.

Figure 3.10 (a) shows the total amount of data delivered to the UE in normal Legacy Handover. First, we observe how the macrocell's burden of delivering data to a mobile UE drops with increasing density when we allow the UE to handover to femtocells on its path. This is evident from the declining heavily shaded area on the plot, as the density of femtocells increases. In fact, by the time 50% of the residence buildings have femtocells installed, the macrocell does not need to serve any data to the mobile UE any more. This is because the femtocells already have complete coverage of the entire path of the UE at this density. We next observe that the total amount of data served to the mobile UE increases at first, as more femtocells are introduced into the network, but saturates and begins dropping after a while because the UE needs to go through an increasing number of handovers, and hence starts spending more time going through the handover procedure, rather than receiving data. The data served by the femtocells drops all the way down to zero when the density of femtocells becomes extremely high. When 50% of the residence buildings have femtocells installed, femtocells already cover the entire path of the UE. Increasing density of femtocells further only reduces the size of the femtocells effectively, while maintaining complete coverage. At extremely high femtocell density, the UE spends all its time trying to handover to the femtocell it is in,

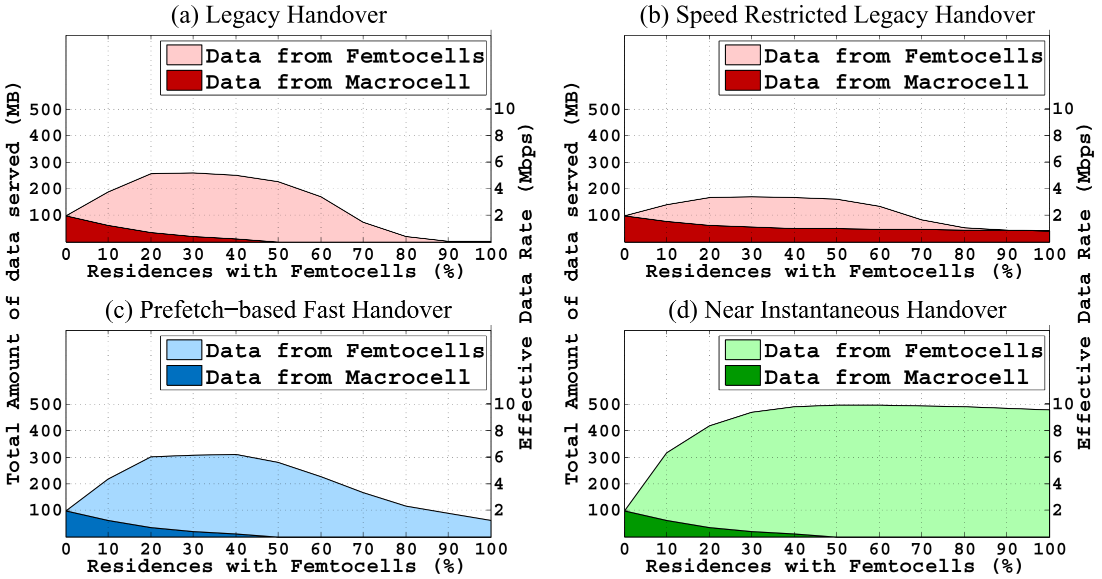


Figure 3.10: Amount of data served to the mobile UE by the macrocell and the femtocells in systems with Legacy, prefetch-based Fast, and Near Instantaneous Handovers, respectively, with increasing density of femtocells. The average speed of the UE is maintained at 35 mph. The UE is always allowed to handover to femtocells with the exception of (b), when there is a speed restriction of 35 mph.

but given the small cell size, is already out of the femtocell by the time the handover procedure concludes. Given complete femtocell coverage, the UE never tries to handover to the macrocell instead.

In order to avoid the above undesirable effect, we present a speed restricted Legacy Handover system in Figure 3.10 (b). The UE is only allowed to handover to the macrocell whenever its speed is higher than 35 mph in this system. Consequently, we avoid the situation where the total data delivered to the mobile UE drops all the way down to zero at extremely high femtocell densities. However, in this case, the maximum data delivered to mobile UE for any femtocell density is lower than that in normal Legacy Handover. This is because even when the region has manageable femtocell densities, the UE spends a significant amount of time associated with the low bandwidth macrocell because of its speed. Also, the burden on the macrocell is not reduced as much because of the restrictions on the mobile UE handing over to femtocells.

Figure 3.10 (c) shows the amount of data delivered to the mobile UE in a system employing prefetch-based Fast Handover. Here, we observe a similar effect as in Legacy Handover. However, the total amount of data served to the mobile UE is higher here, essentially contributed by femtocells. This is because of the UE being able to spend a comparatively longer amount of time associated with femtocells, given the reduced handover procedure duration. At extremely high femtocell densities, the total amount of data served to the mobile UE again drops significantly because of the small cell sizes. The macrocell's burden of data delivery to the mobile UE, is reduced in this system just as in normal Legacy Handover, because the mobile UE spends almost identical time being associated with the macrocell.

Finally, Figure 3.10 (d) shows amount of data delivered to the mobile UE in a system with Near Instantaneous Handover. We see a significant increase in the data served here. This is because the UE now spends negligible amount of time going through handovers, and hence spends as much time associated with high bandwidth femtocells as possible. The data served to the mobile UE by the femtocells increases until the femtocells have complete coverage. Moreover, when the density of femtocells increases further, only effectively reducing the size of the femtocells, making the UE go through higher and higher number of handovers, the amount of data served by the femtocells does not drop significantly. Since the handovers are near instantaneous, even drastically increasing the number of handovers does not waste a lot of the UE's time in the handover procedure. Thus the UE almost seamlessly receives data from the femtocells irrespective of how many handovers it needs to go through.

The effective increase in the capacity of the network is also evident from Figure 3.10. The total amount of data served to the mobile UE increases substantially when switched form a system with Legacy Handover to a system with Near Instantaneous Handover. The increase in capacity comes from the handover procedure allowing the mobile UE to efficiently tap into the femtocell resources in the network.

Figure 3.11 (a) compares the number of femtocells skipped by the mobile UE in Legacy, prefetch-based Fast, and Near Instantaneous Handover procedures, respectively, with increasing average UE speeds. In this simulation scenario, all speeds of the mobile UE are uniformly randomly distributed 10 mph above and below the average, and 50% of the residence buildings have femtocells installed. The path of the UE in this scenario presents it with 239 handover opportunities. While Legacy Handover starts skipping cells at average UE speeds of as low as 30 mph, prefetch-based Fast Handover is able to keep up with femtocells at speeds as high as 60 mph. Near Instananeous Handover however, allows the UE to keep up with femtocells on its path, even when it is moving at extremely high average speed of 120 mph. Thus, the value of the speed threshold θ can be extremely high for Near Instantaneous Handover when compared to prefetch-based Fast Handover. While such speeds may be unrealistic in a dense urban environment, this comparison is only to indicate the benefits of employing Near Instantaneous Handover in other scenarios with desirable results. For example, deployment of femtocells along high speed train lines could potentially provide commuters with seamless high bandwidth connectivity with Near Instantaneous Handover.

Because of the user data being prefetched to the H(e)NBs, prefetch-based Fast and

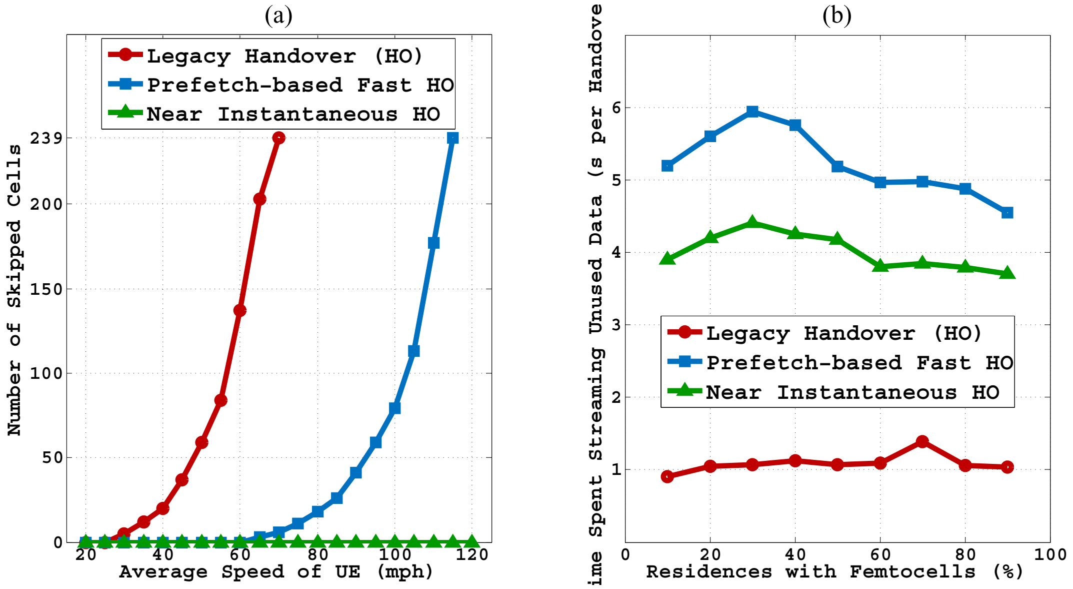


Figure 3.11: (a) Number of skipped cells (missed handover opportunities) by the mobile user with increasing speeds. (b) Amount of time spent in streaming unused data in the proposed handover procedures, compared to the amount of time spent forwarding data from the source to the target femtocells in Legacy Handover with increasing femtocell density.

Near Instantaneous Handover procedures introduce wastage of streamed data. This is the real cost of the benefits brought about by the proposed handover procedures. Figure 3.11 (b) evaluates this wastage. The wastage in Legacy Handover is measured by the time during which data to the target femtocell was streamed via the source femtocell thus traversing the internet twice. The average speed of the UE was set at 35 mph. For every handover experienced by the UE in Prefetch-based Fast Handover, an average of about 5.2 s of time was wasted in streaming unused data to femtocells, which is almost 4.8 times higher than the 1.09 s of time spent in transmitting data through the internet for Legacy Handover. The amount of time wasted per handover in prefetch-based Fast Handover initially increases with increasing femtocell density as the number of femtocells in the proximity of the mobile UE at any point of time increases. After a certain level however, increasing density of femtocells does not affect the number of proximal femtocells any more, and thus the amount of time wasted also saturates. The slight drop in wastage per handover is essentially because of the increasing number of handovers the UE needs to go through because of higher femtocell density. Because of a reduction in the value of Proximity Threshold δ for Near Instantaneous Handover, this wastage was reduced to an average of 4 s spent streaming unused data to femtocells. Thus, Near Instantaneous Handover not only makes the handover instantaneous, but also reduces the wastage of network resources by over 23% when compared to prefetch-based Fast Handover.

Chapter 4

Streamloading: Video Streaming for Mobile Users

Video has come to be the most dominant component of network traffic in recent years. Various kinds of video delivery services ranging from simple download-and-watch type of services, to streaming live television, constitute nearly 55% [18] of the traffic on the internet. Almost all of these video delivery services have succeeded in spilling over to modern wireless networks too. A cellular network user today can not only buy or rent movies online, and watch them on their mobile device, but can also stream high quality videos on the go. Cisco predicts mobile video to generate over 66% of total mobile data traffic by 2017 [18]. In fact, as shown in Figure 4.1, Cisco predicts that mobile video traffic every year would exceed the total mobile traffic the year preceding it.

The recent advances in cellular network technology discussed in Chapter 2 are a major factor contributing to this exponential growth of mobile video traffic. The bandwidths of cellular networks over the air have increased in the last few years, with modern LTE networks offering data rates of up to tens of Mbps to end users. This trend is expected to continue in the near future. This trend, along with the sudden rise in the use of smartphones has led to an ever increasing use of video delivery services by mobile users. In addition, the introduction of high bandwidth small cell hotspots in cellular networks (such as femtocells, for example), along with recent research guaranteeing seamless mobile user handover to these small cells [29] has introduced further variability in the bandwidth a mobile user may experience during the consumption of a video. These trends together motivate us to look for ways to modify traditional video delivery services so that the experience of mobile wireless users can be significantly improved.

In recent years, several types of adaptive video streaming that adapt to the variability of a wireless channel, have been developed by the industry. The video bit rate is switched on-the-fly to provide the best video quality for the user for the given net-

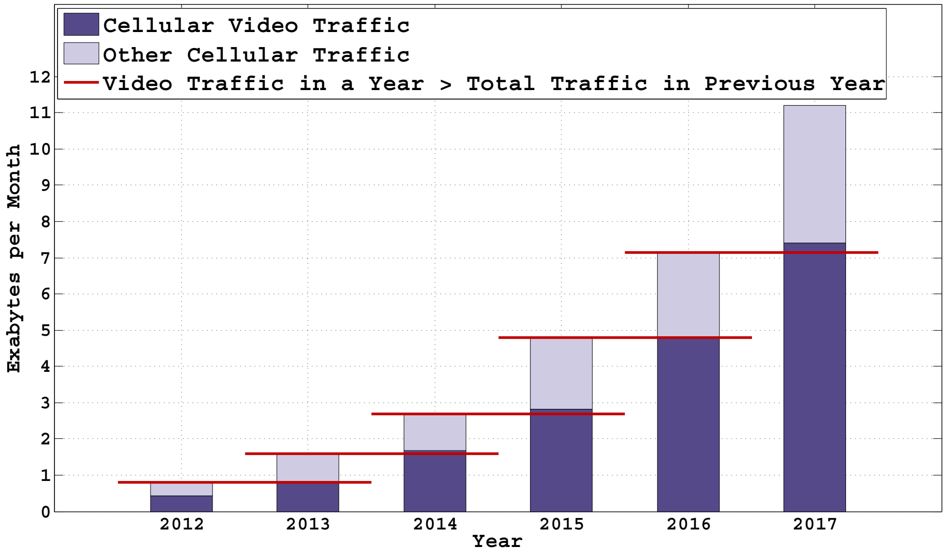


Figure 4.1: Cellular data traffic projections [18] for the years 2012-2017 separated into video and non-video components. Video is expected to constitute over two thirds of mobile traffic within a couple of years. The amount of cellular video traffic every year is expected to be higher than the total cellular traffic of the preceding year.

work resources. Microsoft's IIS Smooth Streaming, Adobe's Flash Dynamic Streaming, and Apple's HTTP Adaptive Bit-rate Streaming are the most common adaptive video streaming services [38]. They use various techniques to dynamically switch between different quality level (bit-rate) streams of the video to deliver smooth and seamless video to users. The research community has also been very active in this field. For example, by exploiting the advanced bit stream switching capabilities using SP/SI picture defined in the H.264/MPEG-4 AVC standard, an optimized bit-stream switching policy for mobile video streaming has been proposed in [39]. Similarly, a bit-rate switching based video streaming to provide the best possible video quality to users with minimum replay interruptions is proposed in [40].

Most modern video delivery services can broadly be classified into *streaming* and *downloading*. For legal reasons, a streaming video service is one where the user is strictly limited to cache video data corresponding only to a short period of video ahead of what they are currently watching [41]. This is to prevent (or at least discourage) illegal copying of content. Netflix, Amazon Instant Video, and Hulu are popular streaming video services [42]. A downloading video service on the other hand allows users to cache in advance as much of the future video data as possible. All online video rental and sale services employ video downloading. Popular example of these services are Google Play Movies, YouTube, and iTunes Movie Rentals [42]. As can be verified on the websites of these streaming and downloading services, a video downloading service almost always costs a lot more (often as high as ten times more) than a video streaming service. While a typical video streaming service may charge a fixed monthly fee for unlimited amount of video streaming, a typical video downloading service charges for every video downloaded.

We present a novel video delivery service, called *streamloading* [43], which qualifies legally as a streaming video service [41], and thus can be offered to users at streaming video service prices. At the same time, streamloading offers video quality levels potentially as high as those provided by more expensive video downloading services. Streamloading video delivery service is then developed extensively, introducing heuristics to modify the algorithms used to prioritize video data requests from each user, as well as to schedule these requests from various users in the network.

4.1 Modern Video Delivery Services

Video delivery services today operate in a wide variety of ways. Some services allow users to buy or rent videos that they can watch at their convenience at a later time, while others only allow users to stream videos over a live connection so they can only watch what has just been transmitted to them. In a cellular network with mobile users the video delivery services need to cater to users with different bandwidth availability. Many video services allow a compromise in the quality of video when serving users with low available bandwidths [38].

4.1.1 Scalable Video Coding

Scalable Video Coding (SVC) for encoding videos is one of the more natural choices [44] for video delivery services where the user is served with varying quality of video depending on the resources available to them. SVC, an extension of the H.264 video coding standard [39], allows lowering the quality of a video by lowering its bit rate. It allows a high quality video to be decomposed into multiple bit streams of lower bit rates. Subsets of these bit streams can then be decoded to get lower quality versions of the video. Thus, SVC can be used to split the high quality video into layers such that each additional higher layer improves the video quality. The layers can be designed such that higher layers consist of predictions based on data decoded from its lower layers. Thus, a high quality video can be encoded using SVC in such a way that every additional upper layer can only be decoded (and hence contribute to the quality of the video) when all its lower layers have already been decoded. The lowest layer of the video, referred to as its *Base Layer*, can be decoded all by itself. The upper layers of the video that depend on their lower layers for successful decoding are referred to as the *Enhancement Layers* of the video.

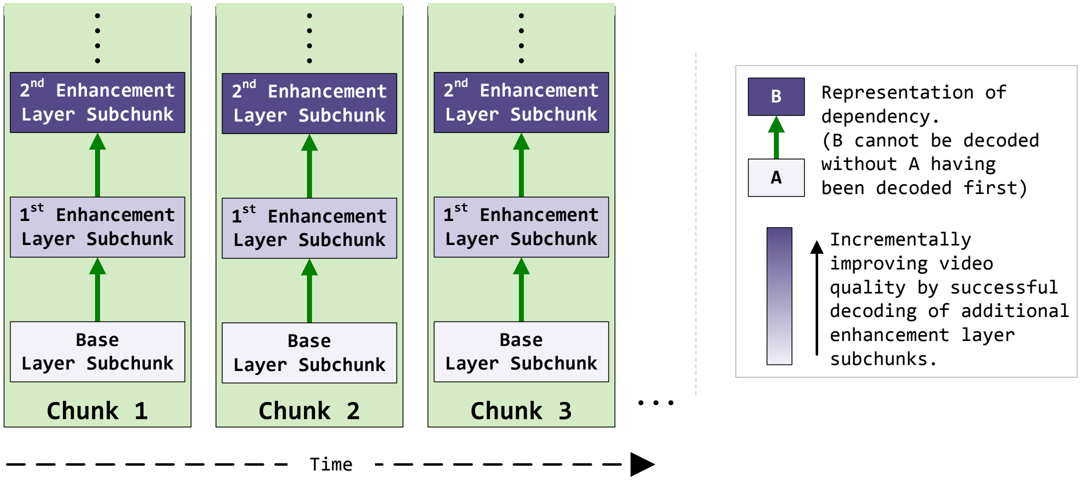

2nd Enhancement Layer Subchunk
1st Enhancement Layer Subchunk
Base Layer Subchunk
Chunk 1
Chunk 2
Chunk 3
...
Time
B
A
Representation of dependency.
(B cannot be decoded without A having been decoded first)
Incrementally improving video quality by successful decoding of additional enhancement layer subchunks.

Figure 4.2: An illustration of Scalable Video Coding, suitable for encoding videos to be delivered over wireless networks. Video is divided into chunks along its playback timeline. Each chunk is then split into subchunks by being encoded into layers, namely, a base layer, and a set of enhancement layers. Base layer subchunks can be decoded independently, resulting in low quality video. Decoding of each additional enhancement layer incrementally improves the quality of the video. An enhancement layer subchunk can only be decoded using the decoded video data of all its lower layer subchunks.

It is these properties of SVC that make it suitable for delivering streaming video to mobile users in a cellular network. As shown in Figure 4.2, the video to be served can first be divided into a sequence of *chunks* where each chunk contains video data playing for a certain length of time. In other words, the original video can be considered as a sequential playlist of its chunks. Each chunk, which is now a small video by itself, is then encoded into layers, using SVC, thus splitting the chunk into *subchunks*, where each subchunk corresponds to a given layer of the chunk. The video is then delivered to the mobile user as subchunks: more subchunks per chunk being delivered when the user has a high available bandwidth, and fewer subchunks per chunk otherwise. This way, the user can enjoy uninterrupted video as they move through varying network bandwidth regions.

4.1.2 Streaming *v.* Downloading

Streaming and downloading are two major categories of video delivery services. Streaming video services restrict users to cache ahead only a small amount of video data [41]. Thus, a user needs to maintain continuous network connectivity throughout their video viewing time. Also, seeking ahead (or even backwards) into the video forces the user to download the corresponding video data again. In contrast, a downloading video service is typically more expensive, and allows unlimited caching, and even allows for offline viewing.

When it comes to delivery of video services over cellular networks, streaming and downloading services have very different behaviors. The bandwidth in a modern cellular network (3G/LTE) may vary from tens of Mbps to a few hundred kbps from the center of the cell to its edge [45]. Moreover, these macrocells formed by the base stations are overlaid with small cells such as femtocells, typically deployed by users at their homes, which offer very high bandwidths to associated users within their short ranges. Such high bandwidth hotspots, along with the inherent differences in bandwidths offered by macrocells from the center of the cell to their edge contribute to mobile users experiencing highly variable data rates as they move around the macrocell passing through femtocells on their path.

For a mobile user to be able to view an uninterrupted video, SVC is used as the encoding choice. Since a user watching a streaming video only downloads immediately viewable video data, the fluctuation in bandwidth experienced by the user is almost instantaneously reflected on the quality of video they are viewing. For example, when a user is close to the macrocell base station, or within a femtocell range, the bandwidth available may be more than sufficient to download subchunks corresponding to all layers of every chunk they are viewing. However, when the user then moves away to the edge of the cell, the bandwidth drops to a level that is just enough to download the base layer subchunks, which results in low quality for the chunks viewed at the time. A user watching a downloading movie however, can take full advantage of any high bandwidth regions they come across, downloading as many future subchunks as they can, thus being able to sustain better quality video even when the bandwidth drops eventually.

Thus, it is clear that in a cellular network, downloading video services can potentially offer much better video quality to users than that offered by streaming video services. In fact, even in the worst case scenario, video quality of a downloading video is at least as high as that of a streaming video. In other words, *from the video quality perspective, it is desirable for users to opt for a video downloading service, while from a cost point of view, streaming services are preferable.*

4.2 Implementation of Streamloading

The issues with video delivery services in cellular networks discussed in Section 4.1.2 above, motivate us to design a video delivery service specifically for delivery in wireless networks, such that the quality of videos experienced by users can match that of existing downloading services, and the price of the service can match that of existing streaming services. To this end, we propose a new video delivery service called *streamloading.*

The basic idea behind streamloading is to allow users to freely *download* enhancement layer subchunks of chunks of video they are watching, as far into the future as they can, while restricting them to *stream* base layer subchunks as limited by existing video streaming services. Thus, at any point of time, a user watching a video is only allowed to have in their cache a few base layer subchunks, and unlimited enhancement layer subchunks. Since decoding of a chunk of video is only possible with its base layer subchunk, any amount of future enhancement layer subchunks remain useless until the corresponding base layer subchunks are streamed to the user. This allows streamloading to legally qualify as a streaming service [41]. Just like a streaming service, streamloading requires the user to maintain continuous network connectivity (to be able to stream base layer subchunks), and seeking forward (or backward) into the video requires the user to download base layer subchunks of the video again. These properties of streamloading enable it to be offered at prices similar to streaming services. Moreover, since the user is now allowed to take full advantage of any surplus bandwidth available to them, by downloading future video data, streamloading can potentially offer video quality similar to those offered by video downloading services.

4.2.1 A Streamloading Example

Let us consider the example illustrated in Figure 4.3, where the quality of video experienced by a user is compared for streaming and streamloading services. The user moves from the center of the macrocell to its edge and thus experiences falling data rate during this time. Each chunk is labeled with an index i, and each subchunk is labeled with the index of the chunk it belongs to. Both streaming and streamloading systems are capable of delivering high quality video when the user is at the center of the macrocell. However, as is clear from the figure, because of the restrictions on what the user is allowed to cache in a streaming service, the surplus bandwidth available to them when they are at the center of the macrocell is wasted. In a streamloading service, on the other hand, the user uses this surplus bandwidth to download future enhancement layer subchunks. Thus, when the user eventually moves towards the cell edge, and the data rate available to them drops, the quality of video served by a streaming service drops too. A streamloading service though, is still able to keep up with the video quality because the current enhancement layer subchunks were downloaded earlier, when surplus data rate was available. This illustrates how a streamloading service can potentially deliver downloading service quality videos to a mobile user.

4.2.2 Heuristics of Streamloading

Now that the inherent benefits of streamloading are clear, we now focus on the implementation of a streamloading system. Since streamloading allows a user to use their

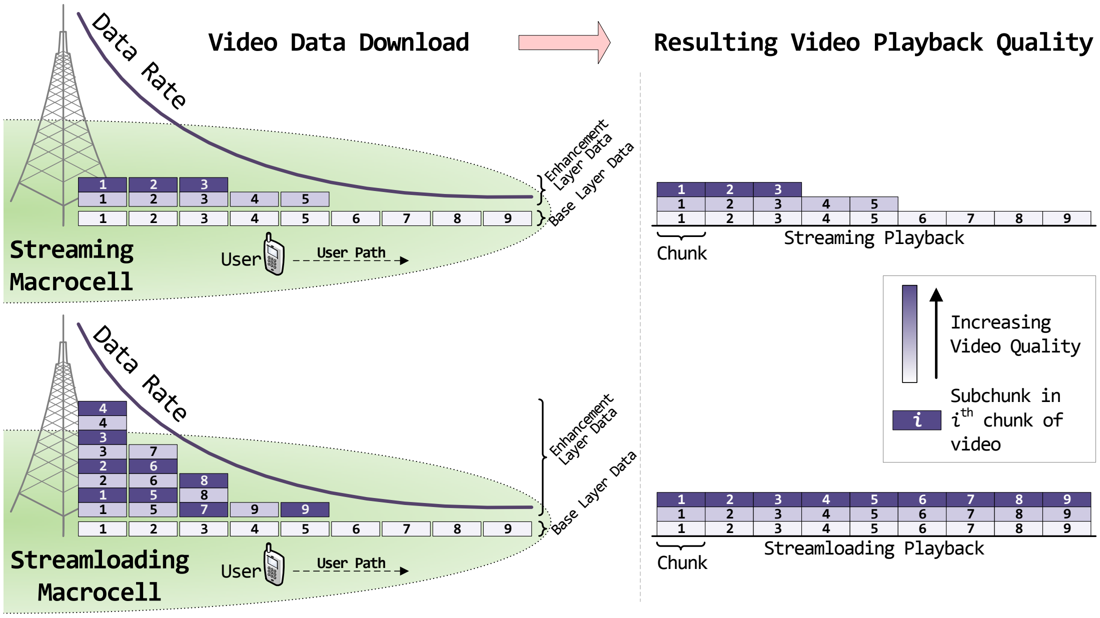


Figure 4.3: An example of Streaming *v.* Streamloading when a user moves from the center of a cell to its edge experiencing dropping data rates. The quality of video experienced by the user in a streaming service drops with decreasing available bandwidth, while the same user using a streamloading service is able to maintain high video quality even after the bandwidth drops. This is because streamloading allows the user to use the surplus bandwidth initially available to cache a sufficient number of future enhancement layer subchunks. This surplus bandwidth available at the beginning is wasted by streaming because it is never allowed to cache future video data.

surplus bandwidth to download future enhancement layer subchunks, the exact choice of the enhancement layer subchunks to be downloaded is of critical importance. If a user greedily downloads enhancement layer subchunks of the video to the highest layer chunk by chunk, running into a very low bandwidth region in the future may result in a drastic drop of the video quality. On the other hand if an overly cautious user chooses to download future enhancement layer subchunks layer by layer, starting with the lowest enhancement layer, they may lose the opportunity to enjoy highest quality video even while sufficient bandwidth was indeed available to them. It is thus important to strike the right balance between these two scenarios.

We therefore formulate a utility index for the enhancement layer subchunks of the video, and use this utility index to evaluate the value of downloading a given subchunk of a video, at a given time. Going through the layers of a video from base layer upwards, the contribution of a layer to the quality of the video drops logarithmically with its data rate [46]. In addition, at any given point of time, the value of a near future subchunk is always higher than that of a subchunk further into the future. In other words, the values

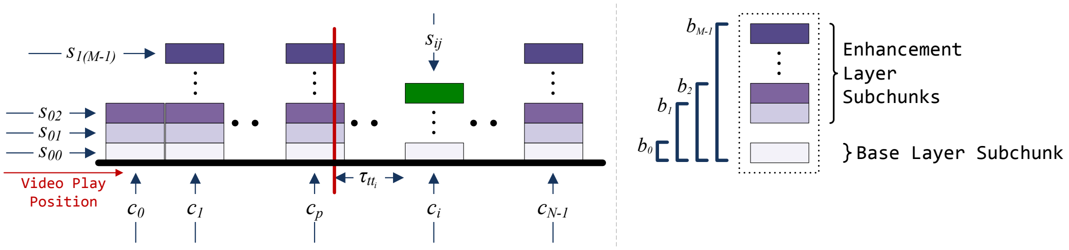


Figure 4.4: Notations used for heuristics of streamloading. Here, c_i denotes the i^{th} chunk of the video, s_{ij} denotes the j^{th} layer subchunk of chunk c_i, c_p denotes the playing chunk at current time t, τ_{tt_i} denotes the time remaining before time t_i when chunk c_i starts playing, γ denotes the time normalizing factor, and b_j denotes the chunk size of the video with quality of layer j.

of the subchunks drop as we go further into the future from the point at which the video is currently playing. Thus, we formulate a heuristic utility index for a given subchunk at a given time to consist of two factors - a video quality factor logarithmically increasing with data rates, and a time decay factor exponentially decaying with increasing time distance between the subchunk under consideration, and the chunk being currently played.

The video is assumed to consist of N chunks, where each chunk is encoded into subchunks of M layers, i.e., one base layer subchunk, and $M-1$ enhancement layer subchunks. Every chunk has the same playback time. Chunk c_i denotes the i^{th} chunk of the video where $0 \leq i < N$, and subchunk s_{ij} denotes the j^{th} layer subchunk of chunk c_i, where $j = 0$ for base layer, and $1 \leq j < M$ for enhancement layers. These notations are illustrated in Figure 4.4. The utility index of subchunk s_{ij} of a video at time t is given by

$$U_t^{ij} = (\ln b_j - \ln b_{j-1}) \cdot e^{-\tau_{tt_i}/\gamma}, \tag{4.1}$$

where $j > 0$, and $i > p$ when c_p is the currently playing chunk. Here, b_j denotes the data rate of a chunk containing the lowest j of the $M-1$ enhancement layer subchunks along with the base layer subchunk, τ_{tt_i} is the amount of time before time t_i when chunk c_i begins playback, and γ is a time normalizing factor that dictates the rate of decay of the utility with time. A low γ ensures that the difference in the number of subchunks downloaded for various enhancement layers at any given time is low, especially for small values of τ_{tt_i}.

At any point of time, the future enhancement layer subchunk with the highest utility index is downloaded. The design of the heuristic utility index ensures that at any point of time, the user has a larger number of lower enhancement layer subchunks than higher enhancement layer subchunks. This feature allows for insurance against short term reductions in bandwidth by being able to maintain the highest quality video, as well as against longer term reductions in bandwidth by being able to avoid drastic drops in video quality. When the user is still downloading near future enhancement layer subchunks, the difference between the number of lower enhancement layer subchunks downloaded and the number of higher enhancement layer subchunks downloaded is much higher than the difference when the user is downloading subchunks far into the future.

We also make use of the heuristic utility index of subchunks for scheduling transmissions to various users within the cell in the network. Our transmission scheduling is designed such that requests for base layer subchunks are always given priority to requests for enhancement layer subchunks, since delivery of base layer subchunks is critical to uninterrupted viewing of the video. Among the requests for base layer subchunks, requests for subchunks closer to the chunk being played are scheduled before others. Since the heuristic utility index of the subchunk a user is requesting to download is also an indicator of the criticality of the enhancement layer video data requests among users, users requesting subchunks with higher heuristic utility indices are scheduled before those with lower indices. Thus, by using the heuristic utility index as the fairness criterion for scheduling, we ensure the average video quality experienced by users in the network is maximized.

4.3 Performance Evaluation

We evaluate the performance of the proposed streamloading system in comparison with a traditional streaming system by building an extensive system level simulation using the C programming language, which allows for a more flexible implementation of the lower layers of the network stack as well as the application layer, when compared to other simulation platforms. The simulation setup consists of a single macrocell with a set of femtocells overlaid on it. All cells deliver video data packets to mobile users via transmissions interfering with each other, in streaming and streamloading systems. Note that the femtocells simulated here could also be replaced by WiFi hotspots, as long as the unrelated problem of maintaining seamless mobile connectivity for a single connection across these two technologies, cellular and WiFi, has been solved.

4.3.1 Simulation Parameters

In this section, we present in detail the relevant parameters used in the simulation model for performance analysis.

Video Data

Each video is 100 min long, and is split into chunks of 1.2 s playback time each. Thus, every video consists of 5000 chunks, each of which is coded into four layers. Thus,

Quality / Video Type	**HD**	**ED**	**CIF**
	(kbps)	(kbps)	(kbps)
Base Layer Only (b_0)	1067	533	67
Up to 1 Enhancement Layer (b_1)	1600	800	107
Up to 2 Enhancement Layers (b_2)	1867	933	120
All Four Layers (b_3)	2133	1067	133

Table 4.1: Data rates of HD, ED, and CIF quality videos used in the simulations. The base layer in this scenario, always has the highest data rate when compared to the data rate of any individual enhancement layer.

$N = 5000$, and $M = 4$. We consider three kinds of videos, namely, High Definition (HD), Enhanced Definition (ED), and Common Intermediate Format (CIF). The data rates of these videos are shown below in Table 4.1.

The maximum allowed buffer size used for HD and ED video delivery systems is 4.8 s while that for CIF video delivery systems is 3.6 s. A lower buffer size of CIF videos is used because of their significantly lower bit rates. These are the lengths of playable video data that streaming limits users to buffer for base layer as well as all enhancement layers. Streamloading on the other hand, allows users to buffer unlimited amount of enhancement layer video data, while restricting playable base layer video data to the buffer size specified.

Network Model

The single macrocell spanning the region under consideration is a circle of radius 1000 m, with the base station located at the center of this circle. A total of twenty femtocells are randomly overlaid on this macrocell following a uniform random distribution for simulation of HD, and ED video delivery systems. A network with no femtocells is also simulated with CIF video delivery systems. The downlink transmission power of the macrocell base station is set such that the received SNR at the edge of the macrocell is 6 dB, adhering to the minimum requirement for decoding data in IEEE 802.16e (WiMAX) [15]. The transmission powers of the femtocells are controlled as described in [37] so that a consistent approximate range of 50 m is achieved. Although WiMAX is adopted as the cellular standard for our simulation, we expect similar results if the LTE standard, which is based on similar OFDMA technology, was simulated instead. The femtocells together cover only approximately 5% of the macrocell region. We simulate two downlink channels for transmissions throughout the network, each representing a group of channels in a WiMAX OFDMA system. The macrocell uses both channels and the femtocells reuse only the second. Throughput statistics, the path loss, and other network parameters are set as in [37].

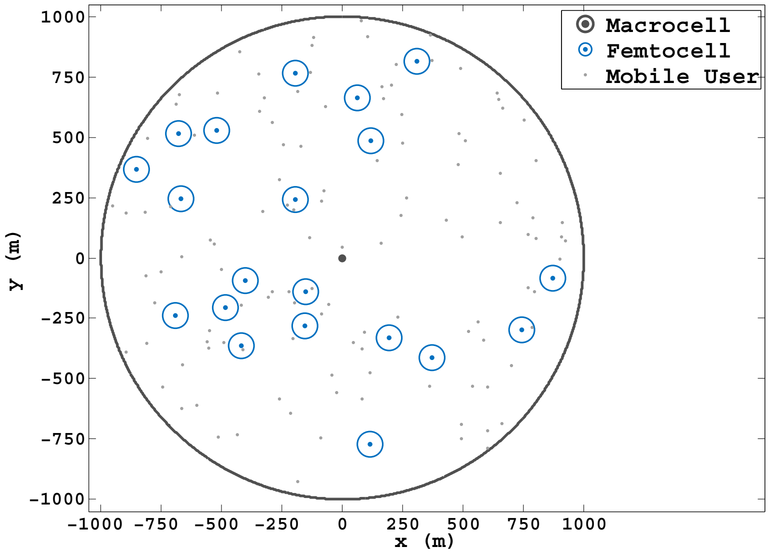


Figure 4.5: An example map of the area under consideration with twenty femtocells under one umbrella macrocell, and 120 users.

User Behavior

All users are mobile, following the Random Walk mobility model, and associate with cells based on their location at any point of time. The average speed of the users is set to 11 mph, corresponding to the average traffic speeds in congested urban areas. The periodicity of change of direction for the users is distributed between 0 and 100 s following the uniform random distribution. The first video demands from users arrive at uniformly distributed time points (to avoid undesirable synchronization), following which every user demands a new video as soon as they finish watching one.

Heuristic Parameters

The time normalizing factor γ in the heuristic utility index is set at 164 s, an optimum value derived from numerous simulations for the video data rates used. However, we make the value of γ increase linearly to the above value at the beginning of each video playback, because lower values of γ are desirable at the beginning of the video to avoid initial low playback qualities. This also helps reduce the wastage of downloaded content in the real world systems where the probability of a user abandoning viewing a video is high in the initial playback duration.

Transmission Scheduling

Packet transmissions to users are scheduled in each cell in TDMA fashion with 2 ms long time slots. One user on every available channel in each cell is scheduled for transmission in every single time slot. Base layer video data transmissions are scheduled with absolute priority over enhancement layer video data transmissions. This ensures identical video playback times (identical initial delays, and interruptions) for users in both streaming and streamloading scenarios, given identical user mobility and user demands, allowing for a fair comparison between streaming and streamloading systems. When scheduling base layer data transmissions, the user with the most time critical base layer subchunk request is scheduled with higher priority. When scheduling enhancement layer transmissions, the user requesting a subchunk with highest heuristic utility index is scheduled first, as discussed in Section 4.2.2. In the scenarios with CIF video delivery, for networks with no femtocells however, the high user density required to saturate the network for better analysis of results may lead to starvation. This is because the large number of users present at the edge of the cell may always have more urgent video data requests waiting. Thus, in such scenarios, we use proportional fairness as the transmission scheduling policy instead.

4.3.2 Analysis of Results

We begin by comparing the average video data rates (which directly affect the average video quality) achieved by users in streaming and streamloading services, respectively. Figure 4.6 presents this comparison as the number of users in the network increases, where Figures 4.6 (a) and 4.6 (b) are for networks with twenty femtocells each, for HD and ED video delivery systems, respectively. Figure 4.6 (c) presents the same comparison for a network with no femtocells, delivering CIF quality video to users. The number of users is varied from 17, 32, and 220 users, respectively, corresponding to an underloaded network, all the way up to 36, 76, and 400 users, respectively, when the network is almost fully saturated for HD, ED, and CIF video delivery systems. Any further increase in the number of users starts to cause interruptions in videos. As can be seen in the figure, streamloading consistently sustains a significantly higher video data rate when compared with streaming.

For example, for a network trying to deliver HD videos to 29 users, streaming is just about able to serve video quality of up to one enhancement layer, while streamloading is able to serve almost perfect quality video. Similarly, for a network serving HD videos to 72 users, while average video data rates served by streaming are barely above the base layer quality, those served by streamloading are significantly higher with up to close to all four layers of video delivered. For a network with no femtocells, the bandwidth fluctuations experienced by users, which is what streamloading tries to take advantage of, is not as much as in the above scenarios. Yet, for such a network with 340 users, streamloading is able to deliver almost two additional layers of video on an average.

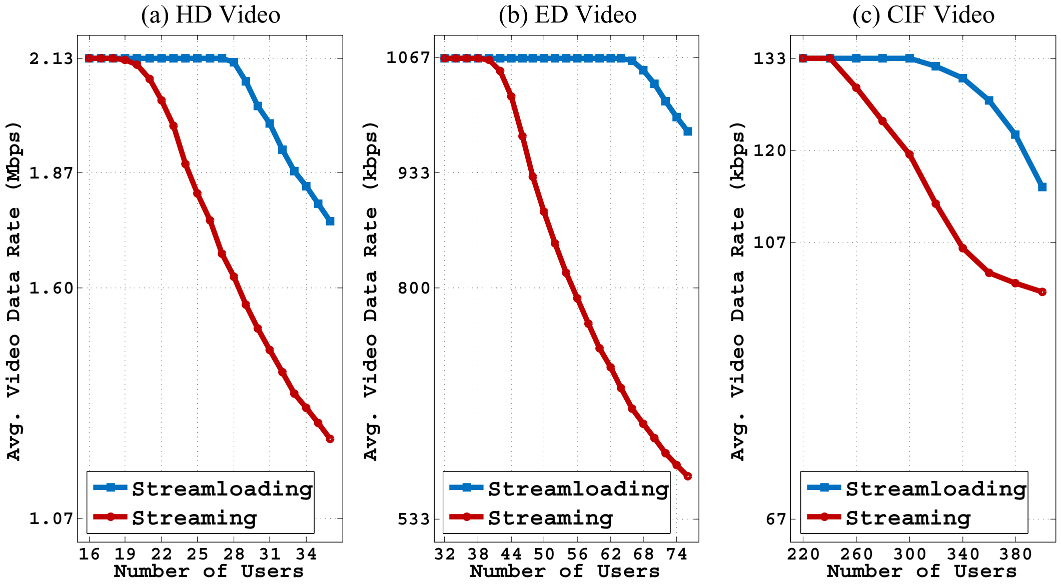


Figure 4.6: Video Quality Measurement for Streamloading and Streaming systems with increasing number of users consuming HD, ED, and CIF videos, respectively. The networks delivering HD and ED videos have twenty femtocells each, and the one delivering CIF videos has no femtocells. As the number of users in the network increases, the performance of streaming drops in quality, while streamloading is able to maintain significantly better video quality for a much higher number of users.

An observation by us, that may not be sufficiently apparent in Figure 4.6, is that the maximum user density for which a video delivery service can ensure serving 99.99% perfect quality video (all four layers) to every user was 16, 36, and 220 users, respectively, for streaming, while it was 26, 64, and 280 users (an increase of 62%, 77%, and 27% respectively) in streamloading, for ED, HD, and CIF video delivery systems, respectively. This observation also shows how streamloading can also be a more suitable video delivery service for the cellular network operators. From their perspective, by using streamloading, the number of users in the network can be significantly increased while still maintaining the same quality of video delivery.

Since presentation of average video quality alone may not always give a complete picture, we show the distribution of quality of chunks of videos delivered for a few scenarios in Figure 4.7. Figure 4.7 (a) compares the distribution of number of chunks of various quality levels in a system serving HD videos to 31 users for streaming and streamloading services. Figures 4.7 (b) and (c) present the same comparison for systems serving ED and CIF videos to 70 and 360 users, respectively. As shown in the figure, streamloading always delivers a significantly larger number of chunks at perfect (all four layers) quality, when compared to those delivered by streaming.

Note that for fair comparison, identical transmission scheduling policies are applied to

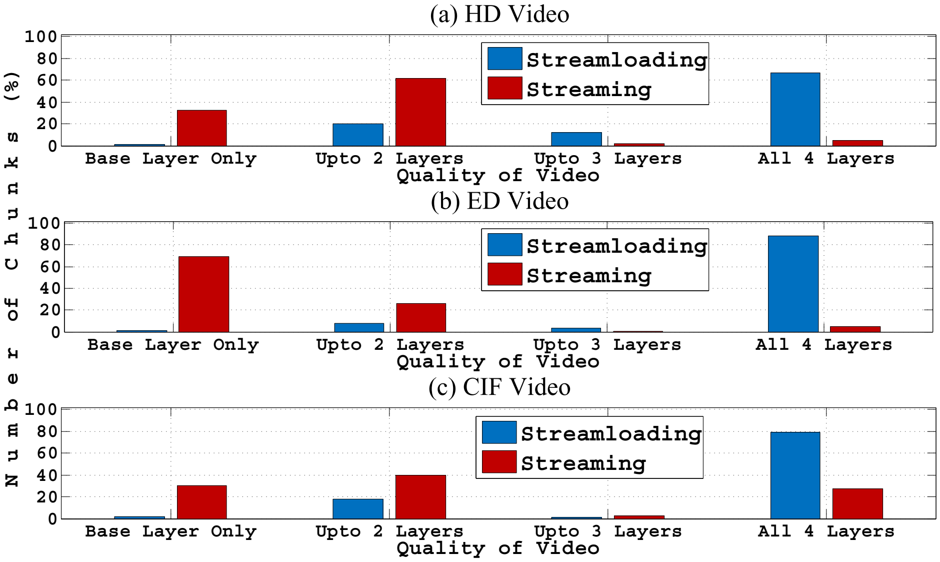


Figure 4.7: Video Quality distribution for Streamloading and Streaming in systems with 31, 70, and 360 users consuming HD, ED, and CIF videos, respectively. The networks delivering HD and ED videos have twenty femtocells each, and the one delivering CIF videos has no femtocells. Even though the difference between average video quality delivered by the two services may not always appear sufficiently significant, the distribution of quality among the chunks makes the benefit of streamloading much more apparent.

both streaming and streamloading systems. For the streaming system delivering ED and HD videos, the application of traditional scheduling policies, rather than the heuristic utility index based scheduling proposed here could reduce its performance, thus further increasing the amount of improvement brought about by streamloading.

Chapter 5

Peer-to-Peer Video-on-Demand on Cable Data Networks

Video-on-Demand (VoD) services, where customers get to choose the videos they want to watch from a catalog, have gained tremendous popularity in recent years. A superior VoD service is one which has a large catalog, and allows the subscriber to begin watching any video from the catalog as soon as they place the order, and continue without interruptions. Thus, VoD services provide customers the flexibility to choose and pay for exactly what they want to watch. In most of the VoD services today, each video delivery stream uses its own media streamer at the server, and thus consumes high bandwidth to transport the video all the way from the server to the subscriber's TV. Such services lack scalability - as the number of subscribers increases, so must the corresponding serving and transport facilities. Both of these kinds of resources could be very expensive. To avoid transport network expansion, a service provider may deploy smaller, satellite VoD servers closer to the subscriber locations [47]. The satellite servers pick videos to store from the catalog, depending on user behavior in the serving locality. However, such solutions still have to deal with the replication of server hardware (and its corresponding storage), which is both expensive and difficult to manage optimally. This has led to the idea of using Peer-to-Peer (P2P) solutions for VoD services. Instead of all users downloading files from a single source, P2P schemes reduce the load on the source by distributing it to the users or peers. Such schemes have been successfully deployed in applications such as BitTorrent [48] and SopCast [49], and have proved their efficiency by facilitating both load reduction on the source server and achieving scalability.

Using P2P solutions for VoD services is a field of interest for many researchers. A comparatively lower number of peers concurrently downloading the same video make VoD more challenging than streaming of live videos. Normally, a P2P VoD service uses a small amount of cache memory contributed by every peer along with the upload bandwidth available at each peer to be able to serve a small part of the demand from the system. There have been efforts to optimally use the available resources at the peers in order to achieve the best performance [50, 51, 52]. However, most of these efforts concentrate on how the small amount of cache available at peers can be used to the best of its potential; how the chunks from a video should be replicated; or how multiple source peers should be optimally chosen to use their small upload bandwidths most efficiently. There have also been strategies proposing pushing of content, in anticipation of future demand, to the peers at a time when there is low traffic on the network [53]. Thus, with time, the P2P schemes proposed to be used for VoD services have become increasingly complicated to be able to achieve better efficiency.

Departing from our focus on cellular networks with resource rich femtocells, we note that lately, due to the impact of resource rich Fiber-to-the-Premise (FTTP) networks, today's VoD subscribers enjoy much higher bandwidths (both download and upload) at home. Moreover, advances in digital storage media may provide inexpensive and substantial storage, which can be easily engineered into a Set Top Box (STB). Thus, with STBs coming with hundreds of GBs of storage, the amount of cache each peer can contribute is anticipated to be much higher. As a result, instead of using sophisticated P2P schemes available today, it has become essential to step back and re-examine simpler techniques.

We propose a simple, but highly robust P2P scheme for VoD services in a resource rich environment. A 'resource rich' network is one that provides high upload and download bandwidths, as well as large cache sizes at peers. The network is called a 'walled garden' network, if all caches at all peers are centrally managed and controlled, and peer upload/download bandwidths are not being shared with third parties. We then move on to simulate the proposed scheme in a close to real world scenario. The results are very encouraging - bandwidth savings at the server of more than 95% can be expected.

5.1 The VoD Transport Network

Before looking at the proposed P2P scheme, it is important to understand the delivery process of a VoD service. We focus on the parts of the network that are relevant for the VoD transport. Note that, in principle, the same network, or part of it may also be used to carry data and voice traffic, although these services end on different gateways.

We describe a typical regional scenario using FTTP access, shown in Figure 5.1. The video originates from servers at the Video Head Office (VHO), which stores every videos listed in the catalog. In a traditional server-client model, the VHO serves all the demands placed by the VoD customers assigned to it. At the other end, the subscriber's home is connected to the local central office, also called the Video Service Office (VSO),

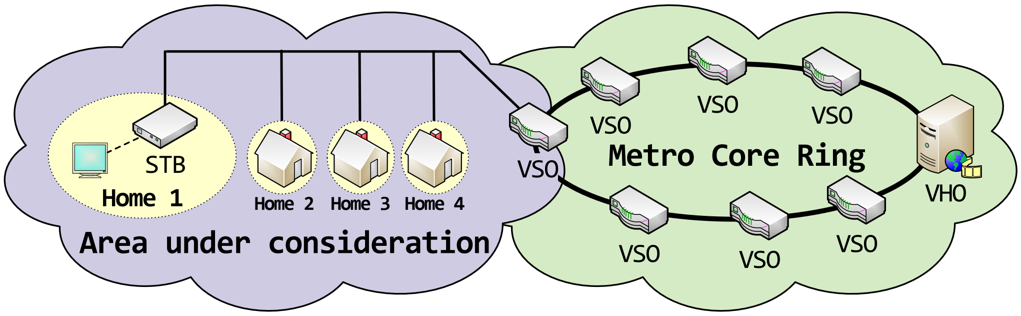


Figure 5.1: Relevant components of the network architecture of a typical FTTP network used for a VoD service. Each VSO in the Metro Core Ring handles all subscribers in a given geographic neighborhood.

via a Passive Optical Network (PON). Each VSO serves a group of customers in a certain locality. All the video data being fed to the customers to serve their demands comes from the VHO through the metro core ring to the VSO and then via PON to the customer's STB [54].

The aim of a P2P service scheme is to reduce the traffic on the metro core ring which hosts the VHO thus avoiding expensive expansions of transmission facilities. Since any traffic from one VSO to another also goes through metro core ring, it will not be advisable for a peer under one VSO to serve one under another. In other words, any P2P service scheme should aim to choose peer uploaders under the same VSO locality as the demand generating peer.

When a peer orders a video via the VoD service, the video is downloaded to the disk partition on the STB that is completely controlled by the service provider. Whether the customer is given the right to transfer the video to their partition of the hard disk is out of the scope of this chapter. It depends on the copyright agreements pertaining to the video. How many times the customer is allowed to watch the video is a similar issue, and therefore will not be considered.

5.2 Proposed P2P VoD Service

The proposed P2P service aims to reduce the load on the VHO and the traffic on the metro core ring by trying to make peers connected to the same VSO, serve most of each others' demands. Note that from the network traffic perspective this, in effect, is equivalent to having a high bandwidth video server in the VSO. In the P2P scenario however, this storage and high bandwidth required from the VSO video server is virtually provided by a large number of peer STBs, each with a smaller storage cache. The following assumptions are made for simplicity.

- All peers under a VSO have equal and dedicated upload bandwidths for the VoD service.
- All peers have equal cache size. The cache at a peer is the service provider controlled disk space in their STB.

5.2.1 The Source of an Ordered Video

When a peer orders a video, a space equivalent to the size of the entire video is immediately reserved in their cache. If sufficient space is not available, the oldest video in the cache is erased first. In the (rare) case when the downloading of the oldest video is still in progress, then the new order is denied. After the space reservation is done, a source is randomly selected by the service provider from a list of potential uploader peers.

A peer is a potential uploader peer if

1. it has the entire video in its cache; and
2. it is either not serving any order or serving an order at a rate at least as high as twice the minimum serving rate. (This is further clarified in Section 5.2.2)

The first condition checks for availability of content at the peer and the second checks for availability of spare upload bandwidth. Since all peer cache is service provider controlled, the service provider is always aware of the contents of all peer cache. Note that both erasing of an old video as well as downloading of a new video are service provider assisted. The service provider is also always aware of all downloads in progress, and always maintains the list of potential uploaded peers for every video in the catalog.

Every order is always served by a single download stream. In other words, there is only one source for every downloading video at any point of time. However, the rate of the downloading stream, as well as the source itself may change several times during the course of the download.

If no potential peer uploader is available for an order, the video is downloaded from the VHO. During the course of the download of a video from the VHO, if and when a potential uploader peer becomes available, the source is switched from the VHO to that peer immediately.

The source of a video download can only be changed by the service provider due to one of the following reasons:

- The original uploader peer erased the video. As a result, a new source is picked.
- The VHO was the original source, but now a new potential uploader peer has appeared. A new potential uploader peer may appear because of one of the following reasons:

  - A peer just finished downloading the video under consideration and can sustain a new upload as described below in Section 5.2.2.
  - A peer uploading the same video just finished an upload service, thus freeing up upload bandwidth.

5.2.2 Rate of Video Download

Every video ordered by a peer is always served at a data rate no less than a specified *Minimum Serving Rate*, which is assumed to be lower than the toal upload bandwidth available to each peer. The download of a video begins as soon as the peer places the order. Assuming that the playback rate of the ordered video is lower than the minimum serving rate, the user can begin watching the video as soon as they place the order and continue without any interruptions. When a user orders a video, the entire video is downloaded whether or not the peer chooses to watch it in one go.

Every serving peer always uses all its available upload bandwidth. In other words, when a peer is serving a single order, it uploads the video at a rate equal to its total upload bandwidth. A peer may serve multiple orders when it has sufficient upload bandwidth to sustain every upload stream at the minimum serving rate. Any spare upload bandwidth is added to the first of the orders it is serving.

If a newly chosen uploader peer is not already serving any order, then it is set to serve the new order using up all its upload bandwidth. Otherwise, given the conditions on potential sources for an order, the chosen uploader peer must be serving at least one order at a rate no less than twice the minimum serving rate. The upload rate of this order is reduced by the minimum serving rate, and the uploader peer begins to serve the new order at the minimum serving rate. Every demand served by the VHO is served at the minimum serving rate. There is no upper limit to the number of orders that the VHO can serve.

Just like the source, even the rate of a video download may change several times during the download. The reason for a change in the download rate can be one of the following:

- The source of the download has changed.
- The source peer is assigned a new order to serve, thus reducing the rate by the minimum serving rate.
- The source peer just finished/stopped an upload service, thus increasing the rate.

By using a minimum rate for any download, the proposed scheme ensures an upper bound on the time taken by a peer to download an ordered video. Unlike many other access networks, the 'resource rich' nature of the network edge makes this simple and robust scheme produce very good performance. The high upload bandwidths at peers ensure that a single download stream is sufficient to stream any video. Also, ample peer cache storage helps the peers to serve more demands than otherwise possible, thus avoiding the necessity of breaking videos into chunks and implementing a complicated chunk management scheme. The proposed P2P VoD service also scales well; there is almost minimal cost associated with adding new VSOs to the system, or handling significant increases in the demand rate.

5.3 System Model

Due to the complex nature of the system itself, and the additional complexity of the P2P scheme, it is fairly difficult to obtain analytical quantitative measures of performance of the proposed P2P VoD service. Hence, to obtain quantitative performance statistics, we simulate the system in extensive detail. In this section, we describe the system model used for this simulation, which includes modeling of the catalogs of videos served by the system, and the arrival of demands from the peers in the system.

5.3.1 Video Catalogs

For the simulation to reflect a scenario close to reality, it is important that the catalog of available videos and its updating is well modeled. The nature of the catalog updating greatly affects the performance of any P2P VoD service. A catalog update simulation that is too rapid could result in minimal uploading from peers, while one that is too stagnant could result in the VHO never being used.

We assume the popularity distribution of videos in the catalog at any point of time to follow the Zipf Law [55] with a fixed parameter. This implies that throughout the duration of the simulation, the popularity of a particular video simply depends on its popularity rank in the catalog at that time. The catalog size, which is the number of videos in the catalog, is always fixed. The catalog is updated every time there is a new video released. Such an event involves the addition of the newly released videos into the Catalog, removal of an equal number of the most unpopular videos from the Catalog, and a change of ranks of videos in the Catalog.

The VoD service offers two different catalogs - the Children's Catalog and the Adults' Catalog. The Children's Catalog contains videos for children and the Adults' Catalog contains videos for adults. Videos in the Children's Catalog are of shorter duration (and hence smaller size) than those in the Adults' Catalog. All videos in a given catalog however, are of the same size. The demands for each catalog during the day varies

W	8 weeks	N	4000 peers	M	3000 videos
ν	4 videos	ν'	6 videos	μ	30 ranks
C	13.18 GB	τ_c	450 MB	τ_a	900 MB
r	2 Mbps			k	0.3

Table 5.1: Parameters used in the simulation for performance analysis of the proposed P2P VoD service.

following its own given distribution, as described in Section 5.3.2. New releases occur daily at midnight, and take high ranks in their respective catalogs, thus pushing all videos below further down. The number of new releases is the same every midnight, except for the one occurring at midnight of Friday into Saturday, when there are a slightly higher number of new releases.

5.3.2 Demand Arrival

All peers who are part of the network participate by ordering and watching videos. The video ordered by a peer at any given time depends on the popularity rank distribution of the videos in the catalog. The probability of a video being ordered is proportional to its popularity rank in its catalog, which in turn is dictated by the underlying Zipf law. A peer may only order a video that is not already in their cache. In other words, peers do not re-order recently ordered videos. The probabilities of selection of the videos are scaled up accordingly for a given peer at any given time. The demand arrivals for each catalog follow a Poisson process. The average demand arrival rate depends on the time of day. For example, demands arrive at a higher rate in the evenings compared to mornings.

5.4 Performance Evaluation

We evaluate the performance of the proposed P2P VoD service using the system model presented above, by simulating it using the C programming language. We begin this section describing the various parameters used in the simulation, followed by the simulation setup. We then present the results of the simulation and analyze them.

5.4.1 System Parameters

The time slotted simulation system consists of N peers. The number of videos at any point of time in each catalog is M. The number of new releases every midnight, except going into Saturday, in each catalog is ν and that on the midnight going into Saturday is ν'. The popularity of videos in each catalog follows a Zipf Distribution curve with

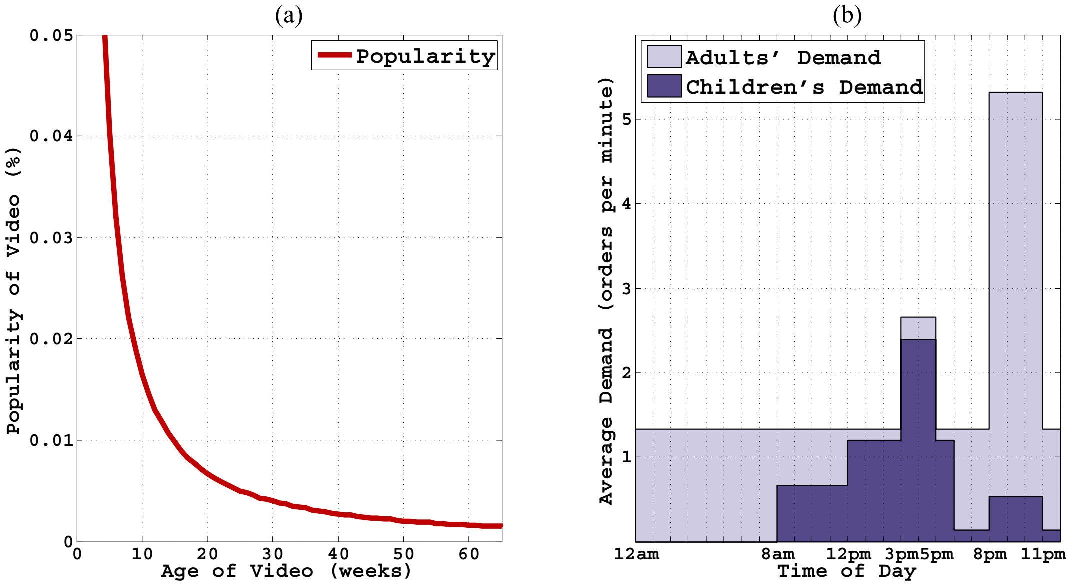


Figure 5.2: (a) Drop in the popularity rank of video with its age (in weeks) following a Zipf curve. (b) Distribution demand during the day for the two catalogs used.

parameter k as commonly used for videos in VoD services [56]. Thus, the popularity of a video from the Children's or the Adults' Catalog, whose popularity rank is m, is

$$p_c(m) = \frac{m^{-(1+k)}}{\sum_{i=0}^{M_c} i^{-(1+k)}}, \tag{5.1}$$

or

$$p_a(m) = \frac{m^{-(1+k)}}{\sum_{i=0}^{M_a} i^{-(1+k)}}, \tag{5.2}$$

respectively. At any midnight, each of the newly released videos takes one of the top μ ranks in their respective catalogs.

The size of videos is measured in chunks where a chunk is the portion of video that can be downloaded in one time slot at the minimum serving rate, r. Recall that all videos in a given catalog are of equal playback length and size. The size of the videos in the Children's Catalog is τ_c and that in the Adults' Catalog is τ_a. The cache size of every peer in the network is C chunks. The maximum upload bandwidth of every peer in the network is $2r$. The simulation runs for a period of time of length W.

5.4.2 Simulation Set Up

Time slots in the simulation are 1 min long. The values of basic parameters used in the simulation described above are presented in Table 5.1.

Since every newly released video pushes everything in the catalog under it further

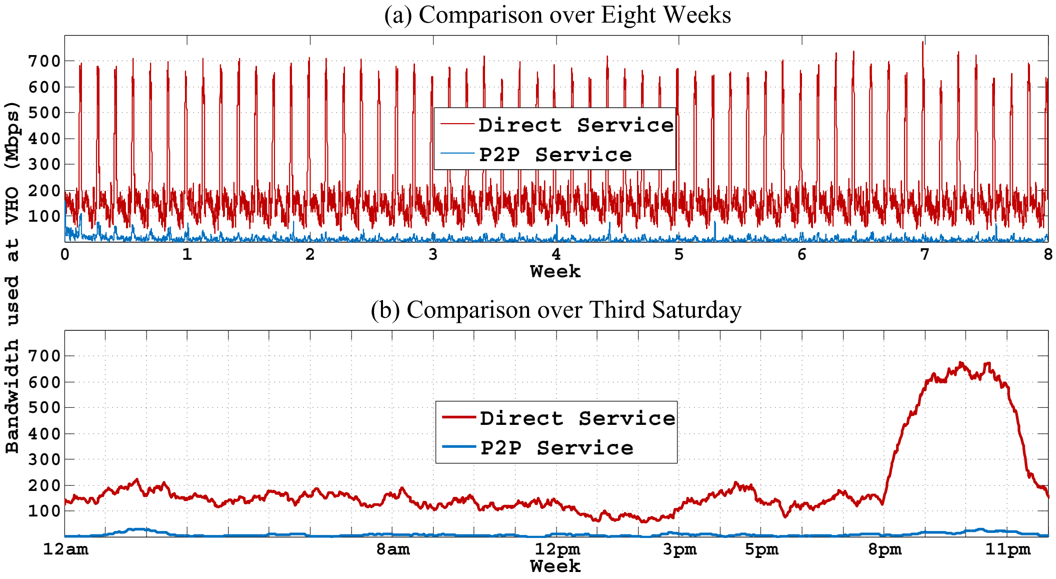


Figure 5.3: VHO bandwidth used in the direct server and P2P VoD services. The bandwidth usage fluctuates with fluctuation in demand, yet the peaks of such fluctuations are significantly suppressed in a P2P service.

down, a video just released into the Children's catalog, for example, stays in the catalog for approximately $\frac{M}{6\nu+\nu'}$ weeks. The popularity of such a video goes down the Zipf curve day after day. Figure 5.2 (a) shows the popularity of a video over its life time for the parameters set as above.

Figure 5.2 (b) shows how the average total demand arrival rate varies during the day (the envelope), and the portion of the total demand from each of the two catalogs. As specified in Section 5.3.2, the arrival rate are specific to each catalog, and depend on the time of the day. The variation is kept identical for every day in the week. The parameters are set taking into account the real world scenario, as evident from the figure. There is a surge in demand from the Children's catalog in the afternoon hours, when the children return from school, and there is a surge in demand from the Adult's Catalog in the night around the popular prime time entertainment hours.

5.4.3 Results and Analyses

The first and most important result of the simulation is the comparison of the VHO bandwidth usage in a direct server download VoD service to that in the proposed P2P VoD service. Figure 5.3 (a) shows this comparison during the 8 weeks of the simulation period, and figure 5.3 (b) is a magnified version of it, for a Saturday. The comparison of the peak VHO bandwidth usages in the two schemes helps us estimate the capacity required from the infrastructure. Reducing the peak bandwidth of the server is potentially

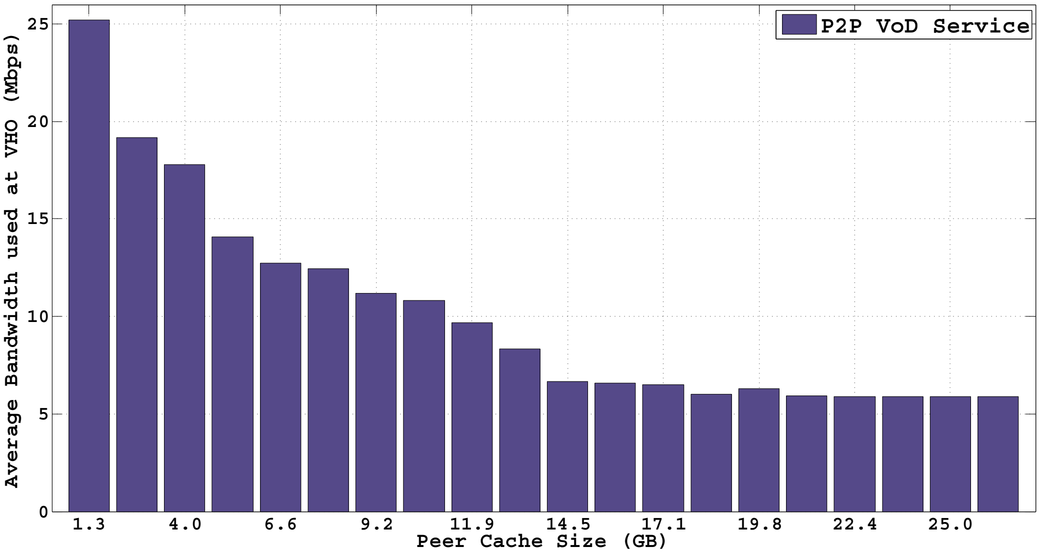


Figure 5.4: Average VHO bandwidth usage with increasing cache size at peers. After a certain threshold, the increase in peer cache only minimally effects the bandwidth usage.

an important cost-cutting measure. The peak VHO Bandwidth usage in a direct server download VoD service is found to be 774 Mbps and that in the proposed P2P Scheme is a mere 80 Mbps. Thus, the required capacity of the VHO is reduced by about 90%.

Recall that new releases occur at midnight. From the Figure 5.3 (b), it is clear that even though there is an addition of six extremely popular new videos in each catalog at midnight going into Saturday, there is no sharp spike in the blue curve, which is an indicator of VHO utilization in the proposed P2P VoD service. Apart from this, even in the afternoon and the evening hours, when there is a spike in the demand, which is closely reflected in the red curve representing a direct server download VoD service, the corresponding spikes in the blue curve representing the proposed P2P VoD service are negligible. This shows that a surge in demand during peak viewing hours, does not get directly propagated to the VHO in the proposed P2P VoD service. The rate consistently stays below 80 Mbps, about a tenth of the peak rate in a direct service.

Secondly, the average VHO bandwidth usage in direct server download VoD service is as high as 197 Mbps, while the same for the proposed P2P VoD service is a mere 8.2 Mbps. The simple and robust P2P VoD service is able to reduce the load on the VHO by more than 95% on an average. The VHO usage in a direct server download service actually gives us an estimate of the aggregate demand. Approximating the VHO bandwidth usage in direct server download service to the total demand in the proposed P2P VoD service (due to identical order arrivals) lets us conclude that in the proposed P2P scheme, close to 95% of the demand is handled by the peers. In other words, the metro core ring only sees 5% of the demand.

The simulation is also used to analyze the tradeoff between the cache size at the peers and the bandwidth savings at the VHO. It is natural that more VHO bandwidth savings can be achieved by increasing the peer cache. Figure 5.4 shows the decreasing average VHO bandwidth usage in the proposed P2P VoD service, with increasing size of the peer cache. When the cache size is increased from 1.32 GB per peer to 13.18 GB per peer, the average bandwidth used at the VHO is reduced by almost two thirds. However, further increases in the cache size do not help as much. These results give us an idea about how much cache in the STB should be partitioned for service provider control, as one would like to obtain the maximum benefit for the minimum cache on the STBs.

Chapter 6

Peer-to-Peer Voice-over-IP for Premium Voice Services

A landline telephone service that includes just a voice service has now become a service of the past. Today's premium telephone services provide much more than just voice. Features such as voicemail, address books, contact specific ringtones and customized diversion of incoming calls are common features of a telephone service today. With the advent of Voice-over-IP (VoIP) the voice network is becoming increasingly integrated with the data network. Although VoIP started off as a by-product of the cheap and high trunk bandwidth provided by Internet Service Providers (ISPs), it has now become a standard application that service providers offer. However, most of the VoIP usage by telephony service providers is done by using centralized servers that provide the voice services and host subscriber features. The problem with such a system is that it can only scale with increasing number of subscribers and/or service requests at great cost. There is an upper limit to the number of subscribers or service requests a centralized system can handle. As this maximum number is approached, particularly over 80% utilization, performance generally degrades in a non-linear way. Therefore, a second similar service host must be added to the system, making it an expensive scaling process. Apart from the scalability servers, the core network infrastructure must also be upgraded to handle the excessive traffic due to the increase in service requests. Often there are delay and bandwidth constraints involved in meeting the service requirements.

Today, ISPs are deploying Fiber-to-the-Premises (FTTP) networks [54] that support voice, data and video over a single fiber to the home. The Optical Network in such systems terminates right at the subscriber premise where a device called the Optical Network Termination (ONT) is installed. The ONT takes care of demultiplexing data to various devices such as the Telephone Handset(s), the Set Top Box (STB), an in-home WiFi Access Point, etc. Thus, the ONT, located at the subscriber premise, potentially has substantial computational power.

With such systems fast evolving, and having addressed the video delivery aspect of such a network with a resource rich edge in Chapter 5, we proceed to address the scaling issue of a large scale VoIP telephony service next, by utilizing Peer-to-Peer (P2P) technology. If the services provided by the centralized servers can be pushed to the peers, making them work in a distributed manner, the system will not only scale well, but also will be far more resilient to failures. In this article we discuss how a P2P system can be used to replace the existing centralized system in order to cut costs, attain scalability, and enhance resilience while not compromising on the services provided to subscribers.

6.1 Features in Today's Telephone Service

We describe in this section the most common features offered by today's premium voice services such as Verizon FiOS Digital Voice [57], and Google Voice [58]. Making a simple telephone call is merely one of the services provided. Listed below are the other most important features normally included in a modern premium voice service bundle.

- *Service Account*: The subscriber is provided with an online service account, which they can manage from any location with an internet connection. The account includes the subscribed and customized service features that can be modified by the user on the go.

- *Voicemail*: A caller may leave a voice message, referred to as a voicemail, when the subscriber does not respond to a call. This voicemail should be accessible from the telephone handset(s), the TV, or even from a computer connected to the internet in or outside the subscriber premise (using the service account).

- *Address Book*: The subscriber is provided with an address book, which stores information about other subscribers including their telephone numbers, pictures, and addresses, for example. The address book can also be edited by the subscriber using their service account.

- *Caller ID*: When the subscriber receives an incoming call, the caller telephone number is displayed on the telephone handset(s). Also, depending on the subscriber preferences, the caller's picture (and possibly other information) is also displayed if the handset(s) has the necessary interface.

- *Ringtones*: The subscriber may set specific ringtones for specific entries in their address book. Ringtone preferences can also be edited using the service account.

- *Ringback Tones*: A ringback tone is what a caller hears when they call the subscriber and the subscriber's phone begins to ring. The subscriber may choose specific ringback tones for specific callers calling them. The ringback tones can be one of several provided by the service provider, or even a customized tone made available by the subscriber.

- *Click-to-Dial*: The subscriber may log into their service account, open their address book and click on a contact to call them. When clicked, the subscriber's telephone handset(s) rings, indicating that the call is ready to be set up, and when the subscriber picks the phone, the call is set up.

- *Three-Way Call*: The subscriber may call two distinct parties at the same time and merge the two calls to form a three-way call where all three parties can hear each other.

- *Call Forwarding and Scheduling*: The subscriber may choose when and which calls they want to receive on their telephone handset(s). They can choose calls to ignore or to forward to other telephone numbers. The subscriber may also specify which calls are directed to which telephone handset(s) within the premise. These preferences are set through the service account.

- *Emergency Calls*: The subscriber can make emergency calls from their telephone handset(s). When the subscriber makes an emergency call, the service provider must provide the subscriber's location information to the emergency service.

- *CALEA* (Communications Assistance for Law Enforcement Act): The service provider must have built-in surveillance capabilities, allowing federal agencies to monitor all voice traffic in real-time to enhance the ability of law enforcement and intelligence agencies to conduct electronic surveillance.

- *Billing*: The subscriber may get real time account billing information.

- *Softphone*: Softphone is Skype [59] like software that can be used by the subscriber as a telephone handset(s). The subscriber can install the software on a computer connected to the internet at any location, and then log in to make calls, just as though they were making calls from their telephone handset(s) on premise. (Such features, usually, do not provide emergency call services.)

6.2 Network Architecture

The P2P premium voice service that we propose [60] runs on an FTTP network, where the fiber optic line terminates at the user premise. The network is assumed to be used

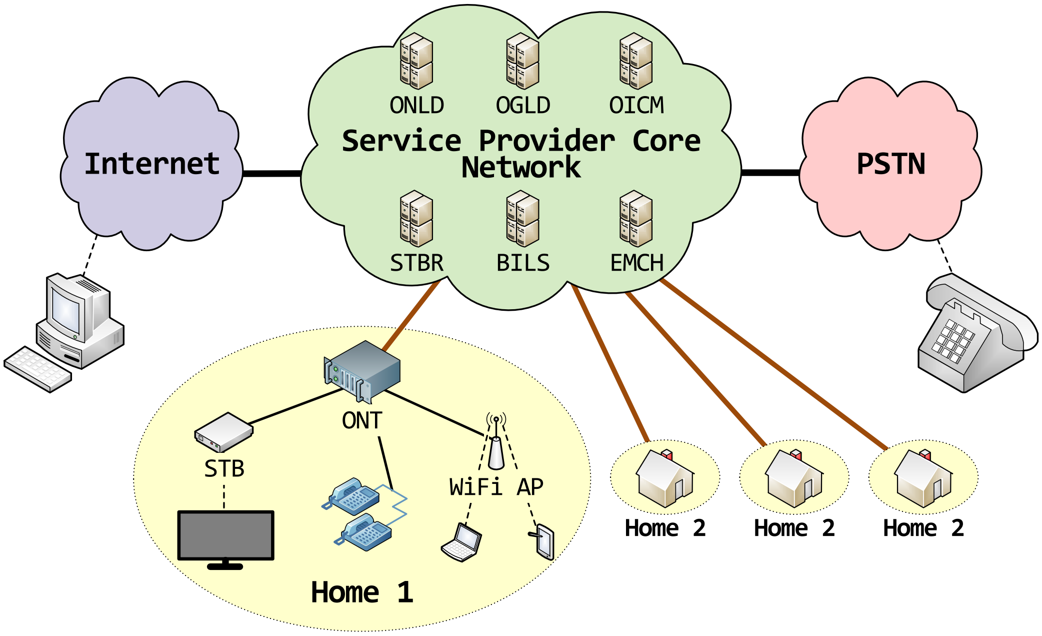


Figure 6.1: Relevant components of the network architecture of a typical FTTP network used for a premium VoIP service. All subscribers may not have an STB installed on premise.

to also provide other services such as television, video-on-demand, high speed internet, etc. The ONT located at the user premise demultiplexes data for various services and is assumed to have sufficient computational power. The ONT does not have any permanent memory storage, but has a battery backup to sustain power failures. Though located at the user premise, it is part of the service provider's network infrastructure and is entirely controlled by the service provider. The ONT is assumed to be sufficiently secure, so that the subscriber or anyone else cannot manipulate it. Every ONT in the network is associated with a *Network Location* and a *Geographic Location*. The network location of an ONT, which could be its IP address for example, is its identity in the network. The geographic location of the ONT is the set of its geographic coordinates. Figure 6.1 shows the network architecture of a typical FTTP network, on which the proposed P2P premium voice service is assumed to run.

The ONT Network Location Database (ONLD) is a set of servers storing the network location information of all the ONTs in the network. We propose to implement the ONLD using a Distributed Hash Table (DHT) [61], where each entry has the telephone number as the key and the network location of the corresponding ONT as the value. Each server in the ONLD stores information about a set of ONTs (possibly covering a particular area). These ONTs are said to be associated with this particular ONLD server. A set of servers similar to the ONLD, called the ONT Geographic Location Database (OGLD) is used to store the Geographic Locations of the ONTs. We also propose to store the information in the OGLD as a DHT. Use of a set of servers instead of a single one reduces individual load and adds a layer of redundancy. Any of the open source implementations available for DHTs [62] can be used to build the servers.

The ONT Initiation and Configuration Manager (OICM) is responsible for the configuration of the ONTs and the input of data into the ONLD and OGLD. OICM is the only server in the network that has the permission to write on the DHTs described above. The nodes in the ONLD and OGLD can only read (or fetch) data from the DHTs they store. For subscribers who subscribe to television services, their STB is responsible for any permanent data storage required at the ONT. For subscribers who do not subscribe to television services and hence do not have an STB on their premise, and for subscribers who subscribe to television services, but do not have an STB, an STB Replacer (STBR) server is responsible for permanent data storage.

The Billing Server (BILS), maintains the billing information for all users in the network. The ONTs are responsible for updating and sending information to this server. The Emergency Call Handler (EMCH) is a server that manages emergency calls. It is allowed to retrieve geographic location of a given ONT from the OGLD.

6.3 The Proposed P2P Premium Voice Service

In this section we describe the functional procedures used by the proposed P2P premium voice service that are required to achieve the features described in Section 6.1. We then present the implementation of these features on the network architecture presented above, in Section 6.2.

6.3.1 Installation of an ONT

When a new ONT subscribing to the voice service is installed, the OICM takes care of adding its information into the ONLD, the OGLD and the BILS. The OICM also configures the new ONT. Once this is done, the new ONT becomes part of the network and calls can be made from, and to it. Configuration of an ONT includes loading and running the application software, loading connectivity information about the associated ONLD, EMCH, BILS, STBR (when there is no STB on premise), and the OICM itself, loading the default user preferences and settings, the Emergency Call Filters and the ringtones and Universal Ringback Tones. Connectivity information loaded is used by the ONT so it can connect to the servers when necessary. Emergency Call Filters are used by the ONT to distinguish emergency calls from the regular ones. Universal Ringback Tones are the standard set of ringback tones made available to the subscriber, by the

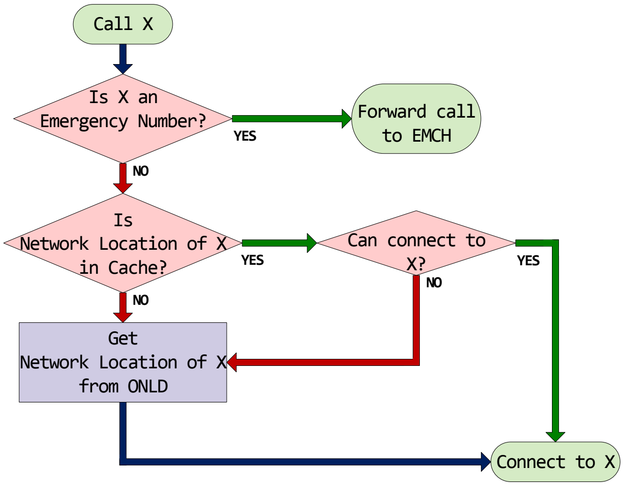

Call X
Is X an Emergency Number?
YES
Forward call to EMCH
NO
Is Network Location of X in Cache?
YES
Can connect to X?
YES
NO
NO
Get Network Location of X from ONLD
Connect to X

Figure 6.2: Steps followed by an ONT during a call setup process when calling X before successfully connecting to X.

service provider. The ONT in turn forwards the default user preferences and settings, the ringtones and the Universal Ringback Tones downloaded as part of initiation process to the STB or the associated STBR for permanent storage.

The above installation process also takes place in the rare occasion of an ONT being rebooted. In the event of an associated server going down, the ONT connects to the OICM to load connectivity information of its new associated server.

6.3.2 Regular Call Set Up

When subscriber A makes a call to subscriber B, the ONT-A (which is the ONT of subscriber A) first checks if the number being called is an emergency number using its Emergency Call Filters. If the number is not an emergency number, ONT-A then checks its cache for the network location of the ONT-B, and if found, ONT-A uses it to connect to ONT-B. If this connection attempt fails because of stale cache, or if the network location of ONT-B is not found in the cache of ONT-A, ONT-A connects to the ONLD to retrieve the network location of ONT-B. The normal call set up process followed by ONT-A in the example described here, is shown in Figure 6.2. Note that in a centralized server system, the network location of ONT-B is never revealed to ONT-A as all communication happens via the centralized server. In contrast, in the P2P voice service we propose, the network location of ONT-B is provided to ONT-A so it can attempt a P2P connection directly. However, the network location of ONT-B continues to be invisible to subscriber A. Since we assume that ONT cannot be compromised by the subscriber, an ONT being given access to the network location of another ONT does not raise any security issue.

Once ONT-A is provided with the network location of ONT-B, it makes a connection to the called ONT and proceeds with the necessary exchange of control signal, thus initiating the call. The voice data between the two ONTs is then exchanged through the direct connection between them. During the call, both ONTs record necessary information about the call such as duration of the call, for billing and other accounting purposes.

When ONT-B receives the call, it first checks if it is indeed the intended recipient of the call (the called number is actually the number corresponding to ONT-B). If not, it notifies ONT-A of the incorrect call set up, resulting in a connection failure at ONT-A. If on the other hand ONT-B finds the number to be correct, it looks for an entry in its subscriber's address book for the phone number corresponding to ONT-A, and also checks its subscriber's general and caller specific incoming call preferences. If the preferences allow for the call to be accepted, ONT-B accepts the call, sends the correct ringback tone back to ONT-A, and forwards the necessary caller ID and picture information to the telephone handset(s) and/or TV as per the subscriber's preferences. More call receiving options are presented in Section 6.3.3.

At the end of the call, both ONTs add the network location of each other to their respective cache, and report the call duration and other accounting information about the call to BILS. The caching of network locations helps avoid connection to ONLD and DHT lookup for every call.

Compared to the centralized VoIP service architecture, the proposed service significantly reduces the DNS load on the servers by using DHTs. The service also reduces the load on the bottleneck servers, which are otherwise responsible to forward all voice traffic, by making ONTs directly connect to each other. This also helps reduce voice traffic on the backbone network to a large extent. The central hardware infrastructure requirement is also reduced consequently.

6.3.3 Implementation of other Features

All the other features commonly offered in a premium voice service presented in Section 6.1 can be also implemented within the proposed architecture. The detailed aspects of these implementations are given below.

Service Account

The Service Account manager, which normally sits on a centralized server, instead runs on the ONT. The information it uses, the user preferences for example, is stored in the STB and retrieved by the ONT when necessary. For subscribers who do not have an

STB, the information is stored in the STBR instead. Initial Service Account parameters and default user preferences are downloaded as part of the ONT initialization process. Users can access all the information from their PC by logging into their service account. When the user logs in using a PC outside home, they connect to their ONT and work just as though they were working from home.

Since all user information, address book and preferences for example, are stored in the STB or STBR, other fringe features, like call notification via email, or via sending SMS to a cellular phone device, can be handled by the ONT very easily.

Voicemail

If the called subscriber is busy or if the called subscriber's preferences require it, the incoming call is forwarded to the called subscriber's Voicemail box. A greeting is passed on to the caller. A voice message from the caller is then recorded by the called subscriber's ONT, which then stores the recorded voicemail in the STB instead of a central Voicemail server. This is not unlike recording of voicemails in a cellular phone service, because the ONT is not under the user's control and the data cannot be lost. Also, voicemail can now be accessed by the user via their service account from any computer, in addition to traditional access via the telephone handset(s) on premise. In order to make the system more robust, there may be an option for backing up voicemail data onto STBs of neighboring ONTs. Once a Voicemail is received, depending on user preference, the ONT sends out notifications to the subscriber's telephone handset(s), TV (via STB), and/or their online account portal.

Address Book

The address book of the subscriber is cached in the ONT and stored in the STB/STBR. The ONT refers to the address book every time there is an incoming call for caller ID, picture, and other preferences. Also, in order to make a call, the subscriber can access the address book from the telephone handset(s), which is directly connected to the ONT. The subscriber can also access and edit their address book from their service account.

Caller ID

When an ONT receives a call, it checks for the caller's entry in the subscriber's address book. If found, depending on the preferences, the ONT transfers the name and picture of the caller to the subscriber's telephone handset(s), and/or TV (via STB).

Ringtones

When an ONT receives a call, it chooses the ringtone associated with the caller from the address book and/or user preferences, retrieves the tone from the STB/STBR and transmits it to the telephone handset(s). Alternatively, a set of standard ringtones can be added to the telephone handset(s), in which case the ONT simply needs to signal the telephone handset(s) about which ringtone it needs to play, when there is an incoming call.

Ringback Tones

As described before, ringback tones can be chosen from a specific set of Universal Ringback Tones or can be custom made. Every time an ONT accepts an incoming call, it checks for a ringback tone entry in the address book. If the ringback tone corresponding to the caller is one of the Universal Ringback Tones, the called ONT sends a signal to the caller ONT for the particular tone. The caller ONT then retrieves the corresponding tone from its own STB/STBR and plays it to the caller. If, on the other hand, the tone is a custom tone, the called ONT retrieves it from its STB and streams it to the caller ONT which in turn plays it on the caller's telephone handset(s).

Click-to-Dial

When the subscriber logs into their service account and clicks on a number to make a call, the ONT rings the subscriber's telephone handset(s). When the subscriber picks it up, the call set up process is automatically initiated by the ONT like a normal outgoing call.

Three-Way Call

When an ONT needs to make a three-way call, it needs to simultaneously connect to two ONTs. The caller ONT thus initiates two call set up procedures simultaneously. Once both connections are established, the caller ONT signals the two called ONTs informing them about the three-way call in progress, and forwards to each of them information about the other's network location. The caller ONT is then responsible for forwarding voice traffic from each of the two parties to the other ONTs. Note that if not for the additional signaling involved, the called ONTs cannot distinguish a three-way call from a regular call. The received voice data appears similar to that in any regular call to the called ONTs.

Also, note that the above procedure may not scale well enough to accommodate conference calls with a larger number of members as the caller ONT may not have sufficient resources to handle several simultaneous calls. In order to handle conference calls with more members, it is best to use a separate media server infrastructure, that handles these features including multimedia.

Call Forwarding and Scheduling

The call forwarding and scheduling preferences of the user are part of their profile data that is cached in the ONT and stored in the STB/STBR. Every time an ONT receives an incoming call, it checks for this information. If the call is meant to be forwarded, then the ONT sets up a second connection to the right destination.

Emergency Calls

When a subscriber makes an emergency call, the number is identified by the ONT using its Emergency Call Filters. Once identified as an Emergency Call, the ONT does not go through the normal call set up process. Instead, it connects to its associated EMCH. The EMCH in turn connects to the OGLD to retrieve necessary information about the caller's geographic location from the DHT. Finally, the EMCH forwards the call to the emergency service with the necessary emergency information attached.

CALEA

The OICM has access to all the ONTs in the network. Whenever calls from and/or to an ONT need to be intercepted, the OICM signals the ONT informing it of the CALEA request. When the ONT receives this request, it forwards all necessary information to the OICM for every call involved, similar to a three-way call. The information is then packaged and forwarded to the appropriate body. The interception remains completely transparent to the user.

Billing

The duration of every call is recorded at the caller's ONT as well as at the called ONT. As soon as a call is terminated the ONTs report the necessary information about the call to the BILS. The BILS then sends back billing information to the ONTs and the ONTs store this data on their corresponding STBs. The subscriber can access this information via their service account. Whenever the subscriber logs into their service account to check the current billing status, for example, minutes used, free minutes left, etc., the information is provided by the ONT (after being retrieved from the STB/STBR). However the BILS also always maintains all billing information, and updates it at the ONTs from time to time.

Unlike other features, billing features are not entirely left on to the ONTs because this information is critical to the service provider, and the service provider needs to be able to access/modify information frequently. This approach also provides an extra layer of security.

Softphone

When a subscriber starts the softphone application from any computer connected to the internet, the application, after having been authenticated, automatically connects to the subscriber's ONT. A direct connection between the softphone and the ONT is maintained as long as the user stays logged in. All the features in the softphone can then be provided by the ONT. All signaling is backhauled to the ONT, but voice paths are established client to client depending on whether the interface between the service provider's network and the external network allow it. Whether other approaches are feasible, or even needed, requires further research.

Chapter 7

Future Work

We have studied various opportunities for the realization of the potential of the edge of the modern cellular and high speed cable networks. The addition of femtocells at the edge of the cellular networks reduces the burden on the macrocells by offloading macrocell traffic to femtocells. We began by allowing even those users in the macrocell who are not associated with any femtocell, to take advantage of the femtocells. We then presented ways in which highly mobile users can tap into the potential of the femtocells in the network by allowing them to handover to the femtocells efficiently, and sufficiently quickly. Next, we suggested modifications in the modern video delivery services so that their design can specifically take advantages of the characteristics of the last mile of the modern cellular network. We then focused on modern Fiber-to-the-Premise (FTTP) networks that provision an abundance of resources all the way to the subscriber's home, and presented ways in which high quality Video-on-Demand (VoD) and premium voice telephony services can be offered using Peer-to-Peer (P2P) schemes, thus tapping into the potential of the network edge.

7.1 FemtoHaul

We presented FemtoHaul, a novel way to offload macrocell traffic on to the overlaid femtocells in the region, so that the macrocell backhaul bandwidth problem can be solved.

We started with a basic idea of offloading of traffic with user relays. We designed a system architecture with open access femtocells, which takes channel allocation and interference into consideration. The user transmission scheduling and relay choosing strategies were devised. Extensive packet-level simulations demonstrated that FemtoHaul is able to accommodate high data demand and a large number of users that would otherwise saturate the macrocell backhaul. In addition, the download rates for the unregistered users were improved due to the use of traffic relays, while the registered users continued to experience broadband data rates. We then presented various feasible options for real world deployment of FemtoHaul within the standard LTE architecture by reusing existing control and data plane interfaces so that proposed modifications in the architecture and equipment can be minimized.

An alternate FemtoHaul-like system could be one where Wifi access points replace the femtocells, and users' cellular phone devices (like smartphones) have multiple interfaces to be able to simultaneously communicate with the BS as well as an associated WiFi access point. After FemtoHaul is made to be more robust, even voice traffic can be offloaded to the femtocells. The offloaded traffic may also include other popular low-latency applications such as live video streaming and video conferencing then. Also, FemtoHaul can be designed to handle the data for each application differently based on its delay sensitivity and other properties.

It may appear to the reader that FemtoHaul aims to solve a problem that is only temporary, because the issue of the macrocell backhaul bandwidth bottleneck is only transient, since these backhauls will eventually be upgraded. However, the provisioning of the offloading of data from a macrocell to overlaid femtocells can always prove useful because of the flexibility it brings to traffic management for the service provider. Moreover, the same concept can also be used in the future, to *onload* traffic from femtocells back to macrocells, if and when the ISPs choose to begin charging cellular service providers for the data they carry to the femtocells, for example.

7.2 Near Instantaneous Handover

We presented Near Instantaneous Handover, a modified handover procedure that aims to make femto-femto handovers completely seamless even to the most demanding higher layer applications.

We introduced a new, sporadically used, fast direct femto-femto control plane interface, which could be implemented over-the-air. We considered a comparatively dense deployment of femtocells, no denser than the deployment of WiFi access points from over six years ago, and focus on fast moving users who have delay sensitive higher layer applications running, and who must handover to every passing femtocell on their paths without causing any noticeable interruption to the application. By enabling such users to handover to a larger number of femtocells, and by making the handover process near instantaneous, we allowed fast moving UEs to take maximum advantage of femtocells in the network, rather than relying mainly on the macrocell, thus offloading more of the macrocell traffic on to femtocells, a very desirable result. We restricted changes only to the femtocell architecture, and made modifications to the handover message flows.

Extensive simulations demonstrated a drastic reduction in handover and interruption durations, and better data service to the users when the Near Instantaneous Handover procedure is deployed.

The introduction of the proposed femto-femto control plane interface can open several other opportunities in the future to make the handover procedure even more efficient and reliable. It may also allow improvements in other message flow procedures. For example, an X2 interface between base stations may no longer be needed. Adding more resources to the new interface would also allow the Proximity Add/Release processes proposed to become more efficient, which in turn may allow us to further reduce the wastage of resources on duplication of data streams.

Although Near Instantaneous Handover successfully achieves complete seamlessness in handovers between femtocells, it might not be directly usable to achieve the same results for handovers that involve macrocells unless a direct control plane interface connecting a femtocell to a macrocell, as proposed in FemtoHaul, is deployed. Introduction of such an interface would allow Near Instantaneous handovers to be implemented not only for handovers between a femtocell and a macrocell, but also between two macrocells.

7.3 Streamloading

We proposed a novel video delivery service called streamloading that is specifically designed for delivery of video in wireless networks where the mobile users experience fluctuating data rates.

We designed streamloading so that it has the advantage of the low cost of a streaming video service, while potentially achieving the high quality of downloaded video at the same time. We also demonstrated how streamloading can help cellular network operators to effectively increase the capacity of their networks, thus alleviating the ongoing "bandwidth crunch". Heuristics are applied to the scheduling of video data packet transmissions in the network so that delivery of the best possible video quality can be achieved. We also performed extensive simulations to evaluate the performance of the proposed streamloading service, as compared to streaming.

In addition to the system level simulations presented, we are also in the process of implementing the proposed streamloading system in our locally deployed WiMAX cell at Polytechnic Institute of New York University in Brooklyn [63]. Encoding of raw videos using SVC is done offline, and the encoded video is then delivered on the go to WiMAX enabled user nodes walking around the neighborhood while viewing the video. Simulating real video delivery over a real cellular network will help us evaluate the performance of streamloading more accurately, and may give us more insights on the optimum encoding parameters for SVC suitable for this scenario.

Better ordering of the requested video data for a given bandwidth profile can be achieved by analytical results from a Markov process model for the system, which in turn can help maximize the benefit of streamloading over streaming. Also, incorporating utility functions to measure and compare qualities of the videos experienced by users from streamloading and streaming services can provide a better insight on the benefits of streamloading.

7.4 P2P VoD in FTTP Networks

We presented a simple, yet robust P2P VoD service for modern walled garden networks that have a resource rich network edge.

Based on our comprehensive simulations, we showed that the deployment of this P2P distribution protocol for VoD delivery on the FTTP network can bring down the traffic on the metro core rings by as much as 95%. In addition, reduction in the traffic load on the VoD servers was also demonstrated. Such traffic reduction helps delay the need for network infrastructure and server upgrades and expansions, thus significantly reducing capital and operations spending. We showed how such benefits can be obtained even with a very small amount of peer cache (typically a partition of the disk space on the Set Top Box). Also, it was evident from the simulations that with the proposed service deployed, such a system can withstand sudden and significant spikes in demand. Sharp surges of local demand were not carried over to the VoD servers.

The efficiency of the service can be further improved by allowing peers to concurrently download from multiple sources, and by allowing a peer to start uploading a video, even if it has not yet downloaded it completely. Such improvements come at the price of more complicated data management among peers.

7.5 P2P VoIP in FTTP Networks

We proposed a premium P2P Voice-over-IP service for modern FTTP networks where the network edge is resource rich, thus trying to push the load on the central infrastructure back to the edge.

We reduced the load on the centralized voice servers by enabling the devices at the network edge to perform most tasks fairly independently, and exchange voice traffic in a P2P fashion. The few servers that remain are simple databases running Distributed Hash Tables, and are mostly used for metadata retrieval. In cases where users involved in a call are located close to each other in the network, the voice data exchanged stays at the edge of the network. Such calls also save a significant amount of network bandwidth in the network core. In addition, the proposed service has extremely high fault tolerance.

Bibliography

[1] V. Chandrasekhar, J. Andrews, and A. Gatherer, "Femtocell Networks: A Survey," *IEEE Communications Magazine*, vol. 46, no. 9, pp. 59–67, September 2008.

[2] "Mobile WiMAX - Part I: A Technical Overview and Performance Evaluation," WiMAX Forum, August 2006. [Online]. Available: http://goo.gl/z0SHZ

[3] H. Claussen, L. T. W. Ho, and L. Samuel, "Financial analysis of a pico-cellular home network deployment," in *IEEE International Conference on Communications, 2007. ICC '07.*, 2007, pp. 5604–5609.

[4] Doug Webster. (2009, April) Solving the Mobile Backhaul Bottleneck. Cisco Blogs. [Online]. Available: http://blogs.cisco.com/sp/comments/solving_the_mobile_backhaul_bottleneck

[5] Craig Mathias. (2008, December) Fixing the Cellular Network: Backhaul is the Key. Network World. [Online]. Available: http://www.networkworld.com/community/node/35920

[6] "Mobile Backhaul Solution with ACX Series Universal Access Routers," Solution Brief, Juniper Networks, January 2012. [Online]. Available: http://goo.gl/p1lIx

[7] "Breaking the Wireless Backhaul Bottleneck," White Paper, ADC Backhaul Solutions, 2007. [Online]. Available: http://goo.gl/JlfT3

[8] (2010, March) Global Femtocell Market (2009-2014). Online. Markets and Markets. [Online]. Available: http://www.marketsandmarkets.com/Market-Reports/femtocell-advanced-technologies-and-global-market-59.html

[9] S. ping Yeh, S. Talwar, S.-C. Lee, and H. Kim, "WiMAX Femtocells: A Perspective on Network Architecture, Capacity, and Coverage," *IEEE Communications Magazine*, vol. 46, no. 10, pp. 58–65, 2008.

[10] H. Claussen, "Performance of Macro- and Co-Channel Femtocells in a Hierarchical Cell Structure," in *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, ser. PIMRC'07, 2007, pp. 1–5.

[11] V. Chandrasekhar and J. Andrews, "Uplink capacity and interference avoidance for two-tier femtocell networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 7, pp. 3498–3509, 2009.

[12] D. Choi, P. Monajemi, S. Kang, and J. Villasenor, "Dealing with Loud Neighbors: The Benefits and Tradeoffs of Adaptive Femtocell Access," in *IEEE Global Telecommunications Conference*, ser. GLOBECOM'08, 2008, pp. 1–5.

[13] K. Sundaresan and S. Rangarajan, "Efficient Resource Management in OFDMA Femtocells," in *Proceedings of the tenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '09. ACM, 2009, pp. 33–42.

[14] E. Dahlman, S. Parkvall, J. Skold, and P. Beming, *3G Evolution: HSPA and LTE for Mobile Broadband*, 2nd ed. Academic Press, October 2008.

[15] *Air Interface for Fixed and Mobile Broadband Wireless Access Systems: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands*, IEEE Std. 802.16e, February 2006.

[16] P. Barford, A. Bestavros, A. Bradley, and M. Crovella, "Changes in Web Client Access Patterns: Characteristics and Caching Implications," Boston, MA, USA, Tech. Rep., 1998.

[17] T. Camp, J. Boleng, and V. Davies, “A Survey of Mobility Models for Ad Hoc Network Research,” *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.

[18] (2013, February) Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012-2017. Cisco. [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html

[19] K. Dimou, M. Wang, Y. Yang, M. Kazmi, A. Larmo, J. Pettersson, W. Muller, and Y. Timner, “Handover within 3GPP LTE: Design Principles and Performance,” in *IEEE 70th Vehicular Technology Conference Fall*, ser. VTC’09-Fall, September 2009, pp. 1–5.

[20] A. Racz, A. Temesvary, and N. Reider, “Handover Performance in 3GPP Long Term Evolution (LTE) Systems,” in *16th IST Mobile and Wireless Communications Summit, 2007*, July 2007, pp. 1–5.

[21] Y.-H. Han, H. Jang, J. Choi, B. Park, and J. McNair, “A Cross-Layering Design for IPv6 Fast Handover Support in an IEEE 802.16e Wireless MAN,” *IEEE Network*, vol. 21, no. 6, pp. 54–62, November-December 2007.

[22] S. Ray, K. Pawlikowski, and H. Sirisena, “Handover in Mobile WiMAX Networks: The State of Art and Research Issues,” *IEEE Communications Surveys Tutorials*, vol. 12, no. 3, pp. 376–399, 2010.

[23] M. Bernaschi, F. Cacace, G. Iannello, S. Za, and A. Pescape, “Seamless Internetworking of WLANs and Cellular Networks: Architecture and Performance Issues in a Mobile IPv6 Scenario,” *IEEE Wireless Communications*, vol. 12, no. 3, pp. 73–80, June 2005.

[24] R. Li, J. Li, K. Wu, Y. Xiao, and J. Xie, “An Enhanced Fast Handover with Low Latency for Mobile IPv6,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 1, pp. 334–342, January 2008.

[25] A. Ulvan, R. Bestak, and M. Ulvan, “The Study of Handover Procedure in LTE-based Femtocell Network,” in *Third Joint IFIP Wireless and Mobile Networking Conference*, ser. WMNC’10, October 2010, pp. 1–6.

[26] M. Chowdhury, W. Ryu, E. Rhee, and Y. M. Jang, “Handover between Macrocell and Femtocell for UMTS based Networks,” in *11th International Conference on Advanced Communication Technology*, ser. ICACT’09, vol. 01, February 2009, pp. 237–241.

[27] J.-S. Kim and T.-J. Lee, “Handover in UMTS Networks with Hybrid Access Femtocells,” in *The 12th International Conference on Advanced Communication Technology*, ser. ICACT’10, vol. 1, February 2010, pp. 904–908.

[28] H.-Y. Lee and Y.-B. Lin, “A Cache Scheme for Femtocell Reselection,” *IEEE Communications Letters*, vol. 14, no. 1, pp. 27–29, January 2010.

[29] A. Rath and S. Panwar, “Fast Handover in Cellular Networks with Femtocells,” in *IEEE International Conference on Communications*, ser. ICC’12, June 2012, pp. 2752–2757.

[30] *Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN): Overall description; Stage 2*, 3GPP Std. TS 36.300, Rev. 11.5.0, March 2013.

[31] *Improved network controlled mobility between E-UTRAN and 3GPP2/mobile WiMAX radio technologies*, 3GPP Std. TR 36.938, Rev. 9.0.0, December 2009.

[32] *UTRAN architecture for 3G Home Node B (HNB): Stage 2*, 3GPP Std. TS 25.467, Rev. 11.3.0, June 2013.

[33] *Evolved Universal Terrestrial Radio Access Network (E-UTRAN): X2 Application Protocol (X2AP)*, 3GPP Std. TS 36.423, Rev. 11.5.0, June 2013.

[34] *Evolved Universal Terrestrial Radio Access (E-UTRA): FDD Home eNode B (HeNB) Radio Frequency (RF) requirements analysis*, 3GPP Std. TR 36.921, Rev. 11.0.0, September 2012.

[35] *General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access*, 3GPP Std. TS 23.401, Rev. 12.1.0, June 2013.

[36] K. Jones and L. Liu, “What Where Wi: an Analysis of Millions of Wi-Fi Access Points,” Georgia Institute of Technology, Tech. Rep., 2006.

[37] A. Rath, S. Hua, and S. Panwar, "FemtoHaul: Using Femtocells with Relays to Increase Macrocell Backhaul Bandwidth," in *29th IEEE Conference on Computer Communications Workshops*, ser. INFOCOM'10, March 2010, pp. 1–5.

[38] Chris Knowlton. (2010, January) Adaptive Streaming Comparison. IIS. [Online]. Available: http://www.iis.net/learn/media/smooth-streaming/adaptive-streaming-comparison

[39] T. Schierl, T. Stockhammer, and T. Wiegand, "Mobile Video Transmission Using Scalable Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1204–1217, September 2007.

[40] X. Qiu, H. Liu, D. Li, S. Zhang, D. Ghosal, and B. Mukherjee, "Optimizing HTTP-based Adaptive Video Streaming for Wireless Access Networks," in *3rd IEEE International Conference on Broadband Network and Multimedia Technology*, ser. IC-BNMT'10, October 2010, pp. 838–845.

[41] "Rates and Terms for use of Musical Works under Compulsory License for Making and Distributing of Physical and Digital Phonerecords," *Title 37 Patents, Trademarks, and Copyrights; Chapter III Copyright Royalty Board, Library of Congress; Subchapter E Rates and Terms for Statutory Licenses; Part 385*, vol. 37 CFR 385.11, February 2009.

[42] Christopher Breen. (2012, July) Where to look for streaming movies and TV shows. Tech Hive. [Online]. Available: http://www.techhive.com/article/2000200/where-to-look-for-streaming-movies-and-tv-shows.html

[43] A. Rath, S. Goyal, and S. Panwar, "Streamloading: Low Cost High Quality Video Streaming for Mobile Users," in *Proceedings of the 5th Workshop on Mobile Video*, ser. MoVid '13. ACM, February 2013, pp. 1–6.

[44] S. Hua, Y. Guo, Y. Liu, H. Liu, and S. Panwar, "Scalable Video Multicast in Hybrid 3G/Ad-Hoc Networks," *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 402–413, April 2011.

[45] "LTE: The Future of Mobile Broadband Technology," White Paper, Verizon Wireless, 2009. [Online]. Available: http://goo.gl/l10jh

[46] H. Hu, Y. Guo, and Y. Liu, "Peer-to-Peer Streaming of Layered Video: Efficiency, Fairness and Incentive," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 8, pp. 1013–1026, 2011.

[47] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng, "Understanding User Behavior in Large-Scale Video-on-Demand Systems," in *Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006*, ser. EuroSys'06. ACM, 2006, pp. 333–344.

[48] "BitTorrent," Website, http://www.bittorrent.org.

[49] "SopCast," Website, http://www.sopcast.org.

[50] S. Annapureddy, S. Guha, C. Gkantsidis, D. Gunawardena, and P. Rodriguez, "Exploring VoD in P2P Swarming Systems," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications*, 2007, pp. 2571–2575.

[51] K.-M. Ho and K.-T. Lo, "A simple model for peer-to-peer video-on-demand system in broadcast environment," in *International Conference on Information Networking, 2008*, ser. ICOIN 2008, 2008, pp. 1–5.

[52] Y. Boufkhad, F. Mathieu, F. de Montgolfier, D. Perino, and L. Viennot, "Achievable Catalog Size in Peer-to-Peer Video-on-Demand Systems," in *Proceedings of the 7th International Conference on Peer-to-Peer Systems*, ser. IPTPS'08. USENIX Association, 2008, pp. 4–4.

[53] S. Barnett and G. Anido, "A Cost Comparison of Distributed and Centralized Approaches to Video-on-Demand," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 6, pp. 1173–1183, 1996.

[54] M. Abrams, P. C. Becker, Y. Fujimoto, V. OByrne, and D. Piehler, "FTTP Deployments in the United States and Japan - Equipment Choices and Service Provider Imperatives," *IEEE Journal of Lighwave Technology*, vol. 23, no. 1, pp. 236–246, January 2005.

[55] E. W. Weisstein, "Zipf Distribution," Mathworld – A Wolfram Web Resource, http://mathworld.wolfram.com/ZipfDistribution.html.

[56] Y. Kim, T. Choi, K. O. Jung, Y. K. Kang, S. Park, and K.-D. Chung, "Clustered Multimedia NOD: Popularity-based Article Prefetching and Placement," in *1999 16th IEEE Symposium on Mass Storage Systems*, 1999, pp. 194–202.

[57] "Verizon FiOS Digital Voice Service," Website, http://www22.verizon.com/home/phone/fiosdigitalvoice.

[58] "Google Play Movies," Website, http://www.google.com/googlevoice/about.html.

[59] "Skype," Website, http://www.skype.com.

[60] S. H. Leiden, A. Rath, Y. Liu, S. Panwar, and K. Ross, "Peer-to-Peer Voice over Internet Protocol," U.S. Patent 8 315 521, November, 2012.

[61] H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Looking up Data in P2P Systems," *Communications of the ACM Magazine*, vol. 46, no. 2, pp. 43–48, 2003.

[62] S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu, "OpenDHT: A Public DHT Service and its Uses," in *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM'05. ACM, 2005, pp. 73–84.

[63] "Wireless Implementation Testbed (WITest) Laboratory - WiMAX Overview: WiMAX as a Research Tool," Website, http://witestlab.poly.edu/index.php/wimax.html.