Schemes for Self-Management in Wireless Networks for Throughput and Reliability Improvement

THESIS

Submitted in partial fulfilment
of the requirements for the degree of
DOCTOR OF PHILOSOPHY

by

GITANJALI BHUTANI

Under the Supervision of
Dr. Mruthyunjaya H Kori

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
2014

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

CERTIFICATE

This is to certify that the thesis entitled **Schemes for Self-Management in Wireless Networks for Throughput and Reliability Improvement** and submitted by **Gitanjali Bhutani** ID No **2008PHXF418P** for award of Ph. D. Degree of the Institute embodies original work done by her under my supervision.

Signature in full of the Supervisor: ______________________

Name in capital block letters: Dr. Mruthyunjaya H Kori

Designation: Technical Director, Alcatel-Lucent

Date: 07 Dec 2014

Abstract

In the current age of Information Technology revolution, quick availability and use of information to make speedy decisions is becoming a competitive advantage for many businesses. In such an environment, to have all decision-makers connected in, ubiquitous communications has become the need of the hour. In turn, wireless networks are gaining importance as they are a key enabler of the ubiquitous communication paradigm. With businesses depending heavily on speedy decision-making, loss of communication for small amounts of time can potentially lead to very large losses to the business. Communication networks in general, and wireless networks in particular are thus needed to provide a high degree of availability and reliability, in addition to being able to adopt quickly to mobility and changing network conditions.

Several factors affect the quality of communication over a wireless network: bad network coverage, fading, interference, failure of different network elements or failure of just a single RF (Radio Frequency) channel. This is in contrast to wired networks, where loss of communication can be either due to congestion or faulty physical links. In spite of congestion effects, loss of communication in wired networks is much rarer than in wireless networks. Loss of communication can be stated to be more a norm than an exception in wireless networks. To keep the user transparent to such failures and to provide a high degree of availability, wireless networks must be able to recover from these failures and switch to alternate network resources so that the user sees no breakage in services. This recovery must be automated and should be carried out without the intervention of an operator. Also, effects like fading, radio interference and other changing network conditions lead to a lower-than-expected throughput being observed on wireless networks. In such cases, wireless networks should be able to change their operating parameters like power of transmission, RF channel in use, predictive buffering of packets and their ACKs, to maintain their quality, efficiency and performance. Note that the change in wireless conditions is very dynamic in nature, and such adjustment of operating parameters is best carried out in an automated manner, without the need of any type of human intervention.

Apart from demands placed by frequent loss of communication on the design of wireless networks, concepts like pervasive computing and deployment of sensor networks require extensive network infrastructure to be quickly deployed. Such deployments are required to be at least as fast as the growth and spread of the user-base. In order to support such quick deployments and to reduce the capital and operational expenditures of the network operator on manpower for these deployments, network elements must largely be able to configure themselves in an automated manner.

The ability of a network to discover, diagnose and fix problems like loss of communication, in an automated manner is termed as Self-Healing property of the network. The ability of a wireless network and its elements to adapt to changing network conditions and maintain quality, efficiency and performance is called as its Self-Optimizing property. Self-configuring is defined as the plug and play behavior of newly installed network elements, which reduces installation costs and simplifies installation procedures. The properties of self-configuring, self-optimizing and self-healing together impart the quality of autonomic management to a communication network and form the broader concept of Self-Management.

One of the main areas of wireless throughput degradation due to transient signal problems is associated with the usage of the transmission control protocol (TCP). Most of the Internet is TCP-based and hence, wireless devices use this protocol extensively as well. However, TCP being designed for wireline systems attributes all packet losses to congestion and triggers its congestion control mechanisms to reduce the packet loss. These congestion control mechanisms at the data source tends to throttle the flow of data to the client for a particular time period determined by TCP specific timers. In wireless networks, this means that a small period of temporary signal loss is followed by a longer period of lower packet input rate to the user, thus overall impacting the throughput of the wireless network even though the wireless conditions may have returned to ideal state. In this thesis, we propose a proactive scheme implemented at the base station, wherein the base station detects that it may lose connectivity to the user in the near future and thus, takes action to cache the TCP acknowledgements from the user. By spacing out these acknowledgements, the base station keeps the TCP server transparent to the disconnection of the user, thus preventing it from bringing any of its congestion control mechanisms to play. As part of the thesis, we address the two aspects of this scheme:

1. The TCP acknowledgement pacing algorithm that the base station employs to keep the server transparent to the loss. The scheme ensures that in the process of delaying the acknowledgements, the TCP retransmission timeout values at the server do not become too large so as to have later transmissions suffer, while at

the same time ensuring they stay large enough so as to prevent the server from invoking its congestion control mechanisms.

2. The disconnection prediction scheme that allows the base station to detect an impending disconnection in advance so as to be able to take proactive action to cache the TCP acknowledgements.

In order to predict disconnection, we model the user's movements using different mobility models and then arrive at the expression that can be used to predict disconnection well in advance, based on a user's mobility characteristics like velocity and direction of movement. We use the Gauss-Markov model and a modified version of it to arrive at different expressions for disconnection prediction. We also look at predicting disconnection using artificial intelligence techniques like Artificial Neural Networks, Radial Basis Function Networks (RBF) and Random Forests. We find that the accuracy of disconnection prediction is the best with RBF networks. These networks are also fairly simple to implement in resource limited network elements and hence, provide the most practical solution for disconnection prediction. In evaluating the TCP ACK pacing scheme based on the disconnection prediction models, we find that although the scheme needs these models in order to predict disconnection, it is not overly dependent on the accuracy of these models for its proper functioning. A disconnection prediction scheme that over-estimates the disconnection duration will work well with the ACK pacing scheme.

The other area of self-management that the thesis looks at is that of self-configuring in overlay networks. Self-configuration requires co-operation among all participating nodes in order to achieve network operation that satisfies all users. In particular, we look at the problem of non-cooperation among the nodes of a P2PTV network. In a peer-to-peer network, disconnection of participating nodes leads to a re-configuration in the network which in turn leads to a negative impact on user experience especially if the content being distributed is streaming content like a TV programme and so on. In order to address this problem, we look at a scheme that provides a substantial incentive to participating users thus, ensuring that they cooperate. These incentives are provided by the service provider by adjusting content prices in such a way that overall, the revenue of the service provider is also substantially improved. The scheme is developed by modeling the scenarios in which multiple products are available at the same time for users, thereby making the demand interdependent on each other.

In summary, this research looks at making wireless networks self-managing by proposing solutions that allow the network to:

1. Detect a pending disconnection using mobility models and machine learning techniques. The use of machine learning makes the network completely self-sufficient and adaptive to whatever mobility pattern it may observe.

2. Control degradation in throughput observed by the user due to transient network conditions. The proposed TCP ACK Pacing mechanism uses inputs from the disconnection predictor to pace out the ACKs and to make the sender transparent to the wireless network condition. With this scheme, simulations showed a 7.5 times improvement in throughput observed by the user during degraded signal conditions as compared to when the ACK Pacing was not done. In contrast to most other schemes proposed in the literature, this scheme is proactive and serves to detect the problem in advance instead of reacting once the disconnection actually occurs. This ensures that the chances of a degradation are further reduced to minimum as compared to other schemes under which some degradation will still be observed during the time it takes it to react to the problem.

3. Self-configure and self-optimize by solving the problem of non co-operative participants in overlay networks like P2PTV networks. The proposed incentivization scheme encourages users to not leave the network in the middle of content dissemination thereby, ensuring a stable network. Experiments show that implementing this scheme allows the network operator to earn as much as 30% more revenue on average.

Acknowledgements

I am extremely thankful to my guide Dr. Mruthyunjaya Kori without whom this thesis would not have been possible. His depth of knowledge on the problems in the wireless world and his patience in reviewing several times the solutions I came up with, have taken my research work to where it is today. Despite his rigorous travel, conference and teaching schedule he has always taken the time out to work on this thesis whenever I have needed him.

My special thanks to my husband Atul Saroop for being my guiding light, always encouraging me to pursue my dreams of a doctorate even if that meant extremely limited time for him and my family. I am extremely grateful to my parents for always instilling in me the desire to do big things and to dream big. This research work would never have commenced had it not been for their encouragement and the faith they have in my abilities. My son Anmol & daughter Garima deserve special mention for being patient and tolerant children, letting me spend hours on my laptop while they waited patiently for their mother's time.

I am very grateful to Alcatel-Lucent for being supportive and encouraging of my research work. The organization and management has always provided all resources I ever needed to pursue this research.

A special thanks goes out to BITS, Pilani for providing me this opportunity to pursue my dreams of a doctorate.

Contents

List of Figures

List of Tables

Abbreviations

ACK	Acknowledgment
ACO	Ant Colony Optimization
ANN	Artificial Neural Networks
AODV	Adhoc On-Demand Distance Vector Routing
APS	Automatic Protection Switching
BART	Bayesian Additive Regression Trees
BSC	Base Station Controller
BTS	Base Transceiver Station
CART	Classification and Regression Trees
CDMA	Code Division Multiple Access
DSDV	Destination Sequenced Distance Vector
DSL	Digital Subscriber Line
DSR	Dynamic Source Routing
dupACK	Duplicate Acknowledgments
E2E	End to End
EEABR	Energy Efficient Ant-based Routing
ELN	Explicit Loss Notification
FEC	Forward Error Correction
GP	Genetic Programming
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications / Groupe Special Mobile
HPI	Handover Performance Indicators
ILC	Inter-Layer Collaboration Protocol
IP	Internet Protocol
I-TCP	Indirect TCP

KPI	Key Performance Indicators
L3	Layer 3
LAN	Local Area Network
LAPC	Load Adaptive Power Control
LL	Link-Layer
LL-TCP-Aware	TCP-aware Link Layer Scheme
LR	Logistic Regression
LTE	Long Term Evolution
MAC	Medium Access Control
MANET	Mobile Adhoc Networks
MSC	Mobile Switching Center
Nnet	Neural Network
NPC	Network Protection Codes
OAM	Operations and Maintenance
P2P	Peer to Peer (Networks)
P2PTV	Peer to Peer Television
PDCCH	Physical Downlink Control Channel
PRB	Physical Resource Block
QoS	Quality of Service
RBF	Radial Basis Function
RF	Radio Frequency / Random Forest
RFC	Request For Comments
RTO	Retransmission Timeout
RTT	Round Trip Time
SACK	Selective Acknowledgments
SCOMANET	Self Configuring and Optimizing Mobile Adhoc Networks
SCTP	Stream Control Transmission Protocol
SINR	Signal to Noise Ratio
SMACS	Self Organizing Medium Access Control for Sensor Networks
SON	Self Organizing Networks
SPLIT	Split Connection Schemes
SVM	Support Vector Machine
TCP	Transmission Control Protocol

TORA	Temporally Ordered Routing Algorithm
UE	User Equipment
UL	Uplink
UMTS	Universal Mobile Telecommunications System
WiFi	Wireless Fidelity
WSN	Wireless Sensor Networks

Chapter 1

Introduction

In the current age of Information Technology revolution, quick availability and use of information to make speedy decisions is becoming a competitive advantage for many businesses. In such an environment, to have all decision-makers connected in, ubiquitous communication has become the need of the hour. Ubiquitous computing and communication combine mobility with context awareness, adaptability, scalability and localization to create an environment where devices are smarter and take actions by predicting user behavior. It involves several smart devices communicating with each other to achieve a particular result simply by predicting what the user would do next, instead of waiting for the user to command the devices. Ubiquitous computing find its applications in a large variety of areas including energy conservation, manufacturing, healthcare, banking, education and telecommunication. In areas like energy conservation, smart devices can perform simple actions like switching off lights when the room is empty, leading to large savings and nature conservation. In education, distance learning already uses this concept as the basis for dissemination of educational content to the desired audience along with expert consultations. Even in critical areas like healthcare, ubiquitous computing has become a critical part of service delivery allowing for body conditions to be regularly monitored by wearable devices and reported to specialists in cases of anomaly.

In order to make ubiquitous computing a reality, certain devices are absolutely necessary. These include sensors or smart phones with sensor functionality, which form the bottom of the ubiquitous computing stack. These are responsible for collecting information from the surrounding environment and reporting it to a decision making entity. In order to allow these sensors to communicate, the next layer is the wireless communication layer which can be provided by the 802.11 family of networks or any other communication technology. The final level in the stack includes the applications that collect, mine and analyze these data for patterns in order to make decisions. Wireless networks, being a key enabler of the ubiquitous communication paradigm are gaining in importance. With businesses depending heavily on speedy decision-making, loss of communication for small amounts of time can potentially lead to very large losses to the business. Communication networks in general, and wireless networks in particular are thus needed to provide a high degree of availability and reliability, in addition to being able to adapt quickly to mobility and changing network conditions.

Several factors affect the quality of communication over a wireless network: bad network coverage, fading, interference, failure of different network elements or failure of just a single RF (Radio Frequency) channel. This is in contrast to wired networks, where loss of communication can be either due to congestion or faulty physical links. In spite of congestion effects, loss of communication in wired networks is much rarer than in wireless networks. Loss of communication can be stated to be more a norm than an exception in wireless networks. To keep the user transparent to such failures and to provide a high degree of availability, wireless networks must be able to recover from these failures and switch to alternate network resources so that the user sees no breakage in services. This recovery must be automated and should be carried out without the intervention of an operator. Also, effects like fading, radio interference and other changing network conditions lead to a lower-than-expected throughput being observed on wireless networks. In such cases, wireless networks should be able to change their operating parameters like power of transmission, RF channel in use, predictive buffering of packets and their acknowledgements, to maintain their quality, efficiency and performance. Note that the change in network conditions is very dynamic in nature, and such adjustment of operating parameters is best carried out in an automated manner, without the need of any type of human intervention.

Apart from demands placed by frequent loss of communication on the design of wireless networks, concepts like pervasive computing and deployment of sensor networks require extensive network infrastructure to be quickly deployed. Such deployments are required to be at least as fast as the growth and spread of the user-base. In order to support such quick deployments and to reduce the capital and operational expenditures of the network operator on manpower for these deployments, network elements must largely be able to configure themselves in an automated manner.

The ability of a network to discover, diagnose and fix problems like loss of communication, in an automated manner is termed as Self-Healing property of the network. The ability of a wireless network and its elements to adapt to changing network conditions and maintain quality, efficiency and performance is called as its Self-Optimizing property. Self-configuring is defined as the plug and play behavior of newly installed network elements, which reduces installation costs and simplifies installation procedures. The properties of self-configuring, self-optimizing and self-healing together impart the quality of autonomic management to a communication network and form the broader concept of Self-Management. In this thesis, we look at some self-management techniques in wireless networks and their application in solving the reliability and efficiency problems of these networks.

1.1 Related Work

Before delving into the details of our research, we perform a brief survey of the work that has already been done in the area of self-management of networks. The data from this survey brings out the research gaps in the area of self-management of networks, some of which we try to address in this thesis. A more detailed overview of the literature in this area will be discussed in the next chapter.

1.1.1 Self Configuring Networks

Salehie and Tahvildari [1] and Aliu et al. [2] define self-configuring as the capability to adapt autonomically and dynamically to environmental changes. Self-configuring has two main aspects as follows: 1) installing, (re)configuring and integrating large, complex network intensive systems, and 2) adaptability in architecture or component level to re-configure the system for achieving the desired quality factors such as improving performance in response to environmental changes. The literature on self-configuring can be divided into three broad areas:

1. Self-Configuring in Wireless Networks: The literature in this area mostly covers the need for self-configuring in wireless sensor networks. The literature discusses the problems to be addressed as a part of self-configuring in these sensor networks, such as topology control [3] and routing [4], [5]. In addition, a substantial portion of the literature discusses the architecture of self-configuring systems in terms of the roles of specific nodes, their communication and the organizing algorithms used [6].

2. Self-configuring in wired IP networks: The literature in this area mostly addresses the self-configuring requirements of peer-to-peer networks [7]. In most cases, the self-configuring properties of peer-to-peer networks are used to setup more advanced self-configuring networks. One such notable example is that of Peer-to-Peer TV (P2PTV) networks. The main research in these areas looks at ways to set up stable and long-lasting peer-to-peer networks by encouraging nodes to participate and share content.

3. Self-Deploying networks: The literature in this area covers mobile base stations that are self-deploying and hence, solve several wireless network problems such as coverage etc by moving closer to the site of high user densities. Research in this area primarily looks at robotic base stations that can achieve all of the self-deploying functionality without manual intervention [8].

1.1.2 Self Optimizing Networks

The literature in this area looks at the self-optimizing properties of different types of wireline and wireless networks. Although self-optimizing in networks is prevalent now because of the advent of 4G networks, it has also been used to some extent in wired IP networks. However, this forms a very small portion of the literature, so we do not discuss it here. The literature on self-optimizing networks can be divided into three categories:

1. Self-optimizing in LTE networks - This is the predominant usage of self-optimization in communication networks. Important use cases for self-optimization in LTE include coverage and capacity optimization [9], energy savings and mobility load balancing optimization. However, a large portion of the literature such as [10],

[11] and [12] on self-optimizing in LTE networks discusses self-optimization as applied to load balancing in these networks.

2. Self-optimizing in Ad-hoc and Sensor Networks - Due to the nature of these networks in terms of absence of centralized control and the need to optimize resource usage, self-optimization is essential in these networks. The main problem in these networks is that of routing and understandably, a large portion of the research focuses on solving this problem [13]. The other major problem is energy optimization [14] since all the nodes of these networks run on limited power and must be able to perform routing as well as their own functionality with minimal power usage.

3. Frameworks for self-optimization - Literature in this category proposes how self-optimizing systems must be built and the essential components of such systems [15].

Having briefly looked at the fundamentals of self-configuring and self-optimizing networks here, we proceed to look at a specialized application of these properties in P2PTV Networks.

1.1.3 Peer-to-Peer TV Networks

Ad-hoc networks and overlay networks like P2P networks rely heavily on their user penetration and participation in networks for realizing their promised benefits. Hence, a significant portion of literature in these areas focuses on ensuring co-operation between nodes in these networks. The literature in this area can be divided into three sub-parts. One large sub-part talks of selfish peers who significantly undermine the advantages of P2P networks, and deals with the problem of free-riders. Studies show that as much as 70% of users in a single Gnutella network are free-riders [16], [17]. Wierzbicki [18] and Wongrujira et al. [19] talk of various incentivization schemes to overcome this and to encourage users to share resources. Various incentive schemes have been proposed to encourage user cooperation in P2P systems - Inherent generosity [16], [20], Monetary Payment Schemes [21] and Reciprocity based schemes [22], [23]. The second sub-part of the user behavior related part of literature deals with the problem of non-cooperation of peers in forwarding lookup messages. In [24], a scheme is setup to allow all lookup-message forwarders to share profits. In [25], forwarders get monetary incentives and schemes are established to prevent peers from overcharging. The third sub-part of literature deals with the problem of topology in P2P networks. Peers exploit the principle of locality and usually attempt to minimize costs by selecting peers that are closest to them. However, such behavior adversely impacts the topology of P2P networks [26]. This sub-part of the literature also studies various algorithms for selection of peers to optimize performance of P2P networks.

In the survey of the various incentivization schemes developed, we find that all the literature on P2P networks assumes a static demand model with very minimal dynamic pricing adjustments based on demand. Schemes that dynamically adjust content pricing based on changes in demand are more realistic and are a notable gap in this area of literature. In addition, the problem of pricing and incentivization in P2PTV networks becomes even more challenging because the content can be considered to be perishable in nature. The content holds value only in a certain time interval after which the value drops. Hence, incentivization schemes devised for P2P networks cannot be reused as-is for P2P networks and must take perishable content into account.

1.1.4 Self-Healing in Wireless Networks

Self-healing in communication networks has a variety of applications, ranging from improving performance and availability of wired systems, to being the foundation principle for ad-hoc networks in providing capabilities like improving the QoS. Kant and Chen [27] define survivability as the ability to provide uninterrupted services to applications amidst unanticipated network element failures in communication networks. The authors specify two main reasons for requiring self-healing and survivability: a) the dynamic and unpredictable nature of wireless networks wherein network resource availability can diminish sporadically, and b) wireless resources are expensive. Since wireless networks are increasingly used for applications like video-conferencing where delays affect the end-user experience, self-healing and survivability mechanisms must take into account criticality of applications in order to prove most effective. Psaier and Dustdar [28] survey various self-healing systems and define self-healing as the ability of the system to recover from abnormal state to normal state, to function in the same fashion as it did before the failure.

The literature in the area of self-healing and survivability can be divided into three main categories:

1. Literature that proposes a framework for self-healing: This category comprises a major portion of the literature on self-healing. Generic frameworks and architectures are proposed to facilitate self-healing in different types of networks [29].

2. Self-healing in the context of ad-hoc networks: Self-healing forms the basis of ad-hoc networks and hence, there is a large amount of literature that discusses the use of self-healing mechanisms to address the various types of problems in these networks. As discussed earlier, the main problem in ad-hoc networks is that of routing. In the context of self-healing, the research looks at the behavior of routing algorithms when some nodes of the network are lost [30], [31] . Another prominent problem discussed in the literature is that of secure communication between nodes [32], [33].

3. Self-healing using biological foundations: While this category is similar to the first category that proposes a framework for self-healing, it requires a special mention due to the amount of research dedicated to it. This category of literature attempts to base the self-healing of networks on the way self-healing is accomplished in the human body. Various artificial intelligence mechanisms attempt to emulate the human body and accomplish self-healing in communication networks such as the one proposed in [29].

Wireless networks are inherently unreliable in nature. This put together with the need for wireless devices to inter-operate with the wireline infrastructure that has existed for decades poses several interesting self-healing challenges. One such challenge is the throughput degradation brought about by the Transmission Control Protocol (TCP) when there are transient packet losses in wireless environments. TCP was originally designed for the wired world, where packet losses only occur because of congestion. As such, the protocol has been designed to treat packet losses as congestion and react accordingly. This poses a challenge in wireless networks. We look at the background of this protocol design and the research on this problem in the next sub-section.

1.1.5 Overview of the Transmission Control Protocol

TCP - Transmission Control Protocol is a reliable transport layer protocol. It provides a connection-oriented byte stream service. It provides reliability by having the receiver acknowledge received packets and having the sender maintain timers for unacknowledged packets [34]. The time interval for the which the sender waits for acknowledgements before retransmitting the packet is called the retransmission timeout and the corresponding timer is called the retransmission timer. Acknowledgements are also used by the sender to estimate the round-trip time (RTT) to the receiver. The retransmission timeout (RTO) is updated using the RTT estimates, every time an acknowledgement for new data is received. In addition, TCP also provides flow control in order to prevent the sender from overwhelming the receivers buffers. In order to achieve this, the receiver advertises a window size, to inform the sender of the number of bytes it is ready to receive at any given time. TCP on the sender uses this advertised window to determine its congestion window size, i.e., the number of outstanding unacknowledged packets at any given time.

To provide a reliable service, a protocol must have a robust mechanism to handle packet losses. TCP handles packet losses by invoking its congestion control mechanisms as described by [35]. This is because TCP was originally designed for wired networks where the majority of packet losses are due to congestion. TCP determines that a packet is lost either when a timeout occurs or when it receives a fixed number of duplicate acknowledgements. If a timeout occurs, TCP invokes its slow start algorithm. As a part of the slow start algorithm, it reduces its congestion window (the number of outstanding unacknowledged packets at any given time) to one. Subsequently, as acknowledgements are received for new data, the size of the congestion window is increased. The slow start algorithm is not invoked when the packet loss is indicated by a duplicate acknowledgement. This is because a duplicate acknowledgement is only generated when the receiver has received a TCP segment, albeit an out of sequence one. Since, this indicates that there is still data flowing between the two ends, slow start is not invoked. Instead, when the third duplicate acknowledgement is received, the congestion window is reduced to half its value and the missing segment is retransmitted. Since, TCP does not wait for the timer to expire before retransmitting, this algorithm is called the fast retransmit algorithm. Reducing the congestion window to half instead of invoking slow start, is called the fast recovery algorithm.

The slow start, fast retransmit and fast recovery algorithms together constitute TCP's congestion control mechanisms. Reduction in window sizes to handle packet losses due to congestion reduces the throughput of the network, as is expected in congestion scenarios. However, with most of the Internet being TCP-based and with wireless networks being extensively used to access the Internet, the throughput reduction is not always preferred in the face of packet losses. In wireless systems, packet losses can be due to a number of reasons like poor signal quality, interference, temporary fading and so on. These packet losses are taken as an indication of congestion by the protocol, leading to its congestion control mechanisms dropping the transmission rates to near zero levels. This is not a desirable behavior for wireless networks, which observe such temporary fading very often. To enable a seamless movement from the wired to the wireless access domain, it is important to solve impending problems that reduce the achievable throughput and the efficiency of wireless networks. One such problem is the reduction in throughput observed due to TCP's congestion control mechanisms that come into operation during periods of mobility or fading of a wireless link.

1.1.6 TCP Adaptations for Wireless Networks

Since, finding a solution to this throughput degradation is important for wireless networks to succeed in providing ubiquitous communication, a lot of research has been done to address this problem. The literature in this area can be categorized into two broad categories - reactive schemes versus proactive schemes. Most of the research in the area of communication networks involves finding different ways of tackling a problem once it has occurred. We call these mechanisms as reactive schemes. An alternative is, for the elements in computer networks to be able to forecast the onset of a problem and take preventive action. We call these mechanisms as proactive schemes. In the proactive scheme, a node may learn from previous "experiences" of the start of an adverse condition. The node may then take preventive action, which may be able to reduce the adverse effect of this condition. A node may also use some pre-configured parameters to detect such conditions. For example, a base station may be configured to predict an impending disconnection, if the strength of the signal from the mobile falls below a certain threshold.

The literature pertaining to enhancements to the TCP protocol to improve its performance on wireless networks can be divided into three classes: Link-Layer schemes, End-to-End schemes and Split Connection schemes. Link-layer schemes use link layer retransmissions to prevent TCP from invoking its congestion control mechanisms. The Snoop protocol [36] and the Delayed DupAcks scheme [37] are examples of link-layer schemes. End-to-End schemes, like Selective Acknowledgments (SACK) [38], Explicit Loss Notification (ELN) [39] and Freeze-TCP [40] try to speed up the recovery mechanism of TCP to improve its performance after the congestion control mechanisms have been triggered. Split-connection schemes, like Indirect TCP [41], attempt to solve the problem by splitting the TCP connection between the fixed host and mobile host into two separate connections - one from the fixed host to the base station and the second from the base station to the mobile host. The main aim of this technique is to keep the fixed host independent of the problems of wireless connectivity.

Having looked at the various schemes proposed in the literature, we realize that most of these schemes are reactive in nature and only kick-in when the congestion control mechanisms have already started operating. This means that they never get rid of the degradation completely. In addition, a lot of the schemes either require the addition of new fields or packet types to TCP or they require that the peer be aware of a wireless receiver being on the other end of the connection. This makes the design of the TCP protocol more convoluted, while also risking a lot of the existing infrastructure that relies on TCP.

1.1.7 Research Gaps

The survey of the previous sections highlighted the research done in each of the areas, together with the prominent problems that stay un-addressed in each of these areas. In this section, we summarize the gaps we discovered as part of the survey:

1. The survey of literature in the P2PTV area reveals the presence of a lot of incentivization mechanisms to ensure the nodes share content and maintain the stability of the network [17]. All of the research in this area assumes a static demand model

and the models do not learn or provide dynamic pricing adjustments based on the demand and the shared content.

2. The utility of the content served in P2PTV networks for its users can be expected to be time-sensitive. As the scheduled timing of the program content being offered passes by, the utility for the users would fall down significantly, and it is in this respect that the content can be characterized more or less as perishable in nature. It is due to this perishable nature of the content on offer that we contend that the problem of pricing and inducing socially correct behavior in P2PTV networks should differ from that in P2P networks. Use of this data to formulate an incentivization scheme can prove to be beneficial for both the operator and the user and is a notable gap in this area.

3. After looking at the various schemes to improve TCP performance in wireless networks, it is clear that the schemes proposed are all reactive schemes. This means that the algorithms mostly always kick in once TCP has already invoked its congestion control mechanisms. Thus, all the schemes will only start to operate once the throughput degradation has already started to happen. Pro-active schemes that do not require fundamental changes to the TCP protocol have not been explored by the research in this area.

4. All the TCP schemes that we have discussed differ in the criteria that is used to detect the degradation. Some schemes use existing fields in the header to indicate to the peer that there is a wireless link degradation [40]. Other schemes introduce new packet types or fields in the header to achieve the same [38], [39]. A prominent gap in this area is that of prediction of a possible disconnection in order to take early corrective measures at the TCP layer, without having the peer content provider know of the wireless link condition. This ensures that the content provider sees the mobile as any other device it is serving content to, and does not need to interpret or understand any wireless-specific indications of disconnection.

Section 1.2 discusses the objectives of the thesis and how the gaps discussed above will be addressed in this thesis.

1.2 Thesis Objectives

The overall objective of this thesis is to design mechanisms and algorithms for self-management of wireless networks, while causing minimal changes to the now widely deployed Internet infrastructure, preferably through the use of schemes that lead to self-selection amongst the various nodes on the network. Our aim is to carry out such investigations for both automated systems and human actors. While investigating automated systems, we attack the research problem where a network element interoperating with many other legacy networking elements changes its own behavior to elicit the best overall response from the entire system. The network element in question here is the base station of a cellular network. We propose mechanisms that require the base station to take actions in case of an impending disconnection of a mobile. This ensures that the servers dispatching data over TCP continue to remain transparent to how the peer is connected to the network. While investigating human actors, we explore the case where a P2PTV service provider designs pricing schemes to elicit response from individual human actors (the customers), that while maximizing the benefits that the human actor observes, also improves upon the gains observed by the service provider.

As discussed in the previous section, one of the prominent research gaps is the absence of proactive TCP schemes that look at wireless conditions to prevent a throughput degradation in advance. Most of the current schemes to cater to the adaptation of TCP to wireless networks rely on different TCP indications such as dup-ACKs, fields in the TCP header and window sizes to detect and react to the problem. The ACK Pacing scheme that we propose in this thesis, in contrast, relies on the prediction of mobile disconnection by the network in advance. As part of developing this scheme, we have developed different mechanisms for predicting disconnection in wireless networks. This includes developing expressions for predicting a mobile's disconnection time and duration, when its movement is modeled by different mobility models. In order to make the wireless network a truly adaptive system, we have used machine learning techniques such as Radial Basis Function Networks, Artificial Neural Networks and Random Forests to predict mobile disconnection. We compare and contrast these techniques to find the most efficient and implement-able one.

Another objective of the thesis is to develop and design mechanisms for self-configuration on ad-hoc, and possibly overlay networks that lead to self-selection among the various nodes of the network, thereby implementing distributed control on node behavior and making redundant the need for a central controlling authority. An excellent example of overlay networks is P2P networks, where it may not be possible to have a central authority to carry out the required topology configuration for efficient content sharing. The absence of a central authority coupled with nodes being allowed to leave the network at any time severely degrades the reliability and user experience in these networks. The problem gets further compounded for applications like P2PTV that try to serve real-time content over P2P networks. Most of such problems can be addressed by providing monetary incentives to users who display appropriate network behavior. Various incentive schemes have been proposed to encourage user cooperation in P2P systems Inherent generosity [16], [20], Monetary Payment Schemes [21] and Reciprocity based schemes [22]. However, most these schemes assume an independent, static demand model for an individual information product on the network. In this thesis, we look to enhance the reliability of these networks using incentivization schemes to ensure that nodes stay on in the network for the period of time that they are involved in content transfer. We have thus, designed learning mechanisms for adjusting content prices to encourage users to show behavior consistent with that beneficial for the overall network [42]. Also, we have carried out our studies under dynamic models of demand, where the nature of the demand is not necessarily known apriori to the content pricing mechanism. We model the scenario where users are exposed to multiple information products at the same time, thereby making their demand interdependent on each other. Encouraging users to stay connected and participate in content dissemination ensures that the self-configured network has enhanced reliability and user-experience where the users will be provided uninterrupted services.

1.3 Key Contributions of the Thesis

In resolving the research gaps discussed in the previous sections, the key contributions of this thesis are the following:

1. A detailed review of the self-management schemes proposed for wireless networks over the past several years, together with an analysis of the schemes proposed to

counteract the problem of wireless throughput degradation due to inter-operation of TCP and wireless networks.

2. Development of a TCP ACK Pacing scheme that serves to heal the after-effects of a transient wireless network condition, by restoring the throughput experienced by the user as quickly as possible. In contrast to most other schemes proposed in the literature, this scheme is proactive and serves to detect the problem in advance instead of reacting once the disconnection actually occurs. This ensures that the chances of a degradation are further reduced to minimum as compared to other schemes under which some degradation will still be observed during the time it takes it to react to the problem. This scheme provides for the self-healing aspect of wireless networks.

3. Development of schemes based on mobility models to predict that the user would be out of signal range allowing measures for it to be taken in advance. This allows the optimization of the operating parameters for that user and reduces data loss to a minimum.

4. Development of schemes based on machine learning techniques to predict disconnection. The use of machine learning makes the network completely self-sufficient and adaptive to whatever mobility pattern it may observe.

5. Development of an incentivization scheme that allows for self-configuration and self-optimization of overlay networks such as P2PTV networks. The proposed incentivization scheme encourages users to not leave the network in the middle of content dissemination thereby, ensuring a stable network. Experiments show that implementing this scheme allows the network operator to earn a higher revenue, as well as ensure a higher quality of user experience.

6. Development of a new mobility model - the modified Gauss-Markov mobility model. Currently existing mobility models in literature model mobility by discretizing the time dimension, and measuring variable location jumps the mobile makes in these equi-sized time slots. However, the modified Gauss-Morkov mobility model models mobility by discretizing locations, and measures variable time durations needed for transitioning amongst those locations.

1.4 Scope and Limitations

In looking at self-management schemes for wireless networks, we solve two independent problems - that of self-healing and self-optimization of wireless networks when hit by transient disconnections and consequent TCP throughput degradation, and the second problem of self-configuring of overlay networks to enhance the reliability of these networks.

Through the course of the thesis, we rely heavily on the use of Monte Carlo simulations to measure the impact of various algorithms and mechanisms. Such simulations depend heavily on assumptions made about the statistical behavior of the systems involved, and we are limited by our knowledge on their behavior. For example, the disconnection prediction schemes are devised using data generated from various mobility models, assuming these provide a realistic representation of moving nodes. We have tried to minimize the impact of such assumptions on the results obtained by carrying out cross-validation against a variety of choice of such assumptions. For example, we use the same machine learning models with a series of mobility models to conclude on the best one, rather than relying on data from a single mobility model.

1.5 Research Methodology

In looking at the various ways of building self-managing wireless networks, we use different research techniques to determine the optimal solution in each case. This section outlines the various research methodologies used in the thesis.

In Chapter 3, we develop the TCP ACK Pacing scheme to reduce the throughput degradation caused due to transient wireless network conditions. We use Monte-Carlo simulations to quantify the improvement brought about by introduction of this scheme. Network simulator-2 (ns2) [43] is an open-source software that allows for simulation of wireless networks and the different conditions in these networks, such as, fading, interference, mobility and so on. ns-2 has different TCP implementations such as TCP-NewReno [44] and TCP-Westwood [45]. The TCP ACK Pacing scheme is implemented as part of the TCP-NewReno protocol and used to run experiments in which we simulate disconnections in wireless networks and quantify the advantages of the scheme.

In Chapter 4, we look at the different mobility modeling schemes to predict disconnection in wireless networks. This chapter primarily focuses on deriving the expression to predict disconnection using these mobility models. We propose a new mobility model called the modified Gauss Markov model to search for a more tractable expression for disconnection duration prediction. We mathematically derive the time in which a mobile would return back within range when its movement follows the proposed mobility model. Similarly, we formulate a city-section model using metropolitan traffic conditions and mathematically derive the disconnection duration expression in this case as well.

In Chapter 5, we look at different machine learning techniques to solve the problem of disconnection prediction. We use Random Walk and Random Direction mobility models in order to generate mobile movement data and data to indicate if the mobile is disconnected at each point of movement. We use 90% of this data as the training set for the machine learning models and 10% of it as the test data set. Weka [46] is used to construct the machine-learning models. In each case, we tune the parameters of the model until we reach a combination that provides us the best accuracy of prediction for the given data set.

In Chapter 6, we develop an incentivization scheme for P2PTV networks, that serves to optimize the network provider's revenue and stabilizes the network. We use Monte-Carlo simulations to quantify the profits it brings to the network provider. We devise a simulator for the P2PTV network that allows for configurable number of users, TV programmes and pricing rounds. This simulator is used to arrive at the optimal number of pricing rounds and to determine the network provider's profit.

1.6 Organization of the Thesis

The thesis is organized as follows: Chapter 2 contains a detailed review of the various self-configuring, self-optimizing and self-healing schemes proposed in the literature. It also reviews the various schemes proposed to overcome the problem of wireless throughput degradation due to TCP's congestion control mechanisms. Chapter 3 details the TCP ACK Pacing scheme - one of the most important contributions of this thesis. The chapter contains the implementation details and results of simulations carried out to measure the improvements brought by these schemes. It compares this scheme to other reactive schemes proposed in the literature to prove that the mix of cross-layer and proactive measures helps the scheme score over most others. Chapter 4 deals with the development of disconnection predictors based on mobility models. In this chapter, we first devise two new mobility models - the modified Gauss-Markov model and the city-section model. Chapter 5 surveys the various machine learning techniques in detail and discusses the application of Artificial Neural Networks, Radial Basis Function Networks and Random Forest techniques to predict disconnection. The chapter provides experimental results which are used to compare these schemes based on the prediction accuracy. Chapter 6 discusses the problem of non-cooperation of users in overlay networks like P2P networks. The incentivization scheme to encourage users to not disconnect in the middle of content dissemination in P2PTV networks is discussed in detail in this chapter. It details the simulations performed and results that prove that the scheme is beneficial both to the network operator and the network users. Chapter 7 concludes the self-management discussion and discusses the scope for future research work in this area.

Chapter 2

A Survey of Self-Management in Communication Networks

In this chapter, we look at the research done in the area of self-management of networks. Specifically, we look at the various self-configuring, self-optimizing and self-healing mechanisms proposed in the literature and the problems that are addressed using these mechanisms. In addition, we survey the various techniques proposed to adapt TCP to wireless networks in order to mitigate the problem of throughput degradation. In each case, we look at the salient features of each proposed mechanism and the result of the discussed research. Finally, we survey the use of prediction techniques in handling wireless network problems as a specific case of self-healing networks. The data from this survey brings out the research gaps in the area of self-management of networks, some of which we try to address in this thesis.

2.1 Self-Configuring Networks

With the growing customer base for 3G wireless networks and the large amount of expectation from the 4G networks, the operators are now focused on trying to provide a large variety of services to the users while trying to maintain their capital expenditure and operating expenditure relatively constant. Increasing deployments means increase in the amount of man-power needed to first deploy and then monitor the systems. This need is overcome by developing self-configuring systems that are capable of configuring and installing themselves depending on operator network needs. In this section we discuss self-configuring in wireless and wired networks. In addition, we also mention self-deploying networks which at this point may seem a little progressive but the idea is fast catching on.

Salehie and Tahvildari [1] and Aliu et al. [2] define self-configuring as the capability to adapt autonomically and dynamically to environmental changes. Self-configuring has two main aspects as follows: 1) installing, (re)configuring and integrating large, complex network intensive systems, and 2) adaptability in architecture or component level to re-configure the system for achieving the desired quality factors such as improving performance in response to environmental changes. The literature on self-configuring can be divided into three broad areas:

1. Self-Configuring in Wireless Networks: The literature in this area mostly covers the need for self-configuring in wireless sensor networks. The literature discusses the problems to be addressed as a part of self-configuring in these sensor networks.

2. Self-configuring in wired IP networks: The literature in this area mostly addresses the self-configuring requirements of peer-to-peer networks.

3. Self-Deploying networks: The literature in this area covers mobile base stations that are self-deploying and hence, solve several wireless network problems such as coverage etc by moving closer to the site of high user densities.

2.1.1 Self Configuring in Wireless Sensor Networks

Wireless sensor networks connect deeply embedded sensors, actuators and processors. These networks are used for tasks such as surveillance, widespread environmental sampling, security and health monitoring. Sensor networks typically consist of hundreds to thousands of nodes that are generally stationary after deployment except for a few mobile nodes in the network. However, unlike cellular networks and mobile ad-hoc networks, these networks operate under severe energy constraints. While the operation of cellular networks is optimized for QoS, the operation of sensor networks has to be optimized to expend minimal energy in order to prolong the life of the network. Since, sensor networks consist of an enormous number of nodes, deploying and configuring of these networks will be impossible unless it is done in an automated fashion. Hence, self-configuring is a necessary property to enable operation of sensor networks, since they have to operate mostly unattended. Self-configuring protocols for sensor networks must be able to enable network operation during startup, steady state and failure. For ad-hoc sensor networks, energy depletion is the primary factor in connectivity degradation and length of operational lifetime. Therefore, overall performance of a protocol becomes highly dependent on its energy efficiency. In conventional wireless networks, the primary area of focus is providing high quality of service and high bandwidth efficiency when mobility exists. However in ad-hoc sensor networks, the primary area of focus is prolonging the lifetime of the network and, compromising QoS and bandwidth utilization to achieve this is acceptable. Hence, the literature in the area of self-configuring wireless networks discusses different protocols that enable network operation in an energy efficient manner. In addition to this, the other areas of research in this field are various mechanisms for adaptive power control and schemes for topology control.

Sohrabi et al. [4] proposes different protocols at each layer of the network stack for wireless sensor network nodes. It discusses a MAC protocol - Self Organizing Medium Access Control for Sensor Networks (SMACS), for setup of links between various nodes while also allowing nodes to switch off and on their transceivers periodically to allow energy efficiency. This is primarily a protocol for neighbor discovery and link formation. The neighbor discovery phase is combined with the channel assignment phase in the SMACS protocol. The paper specifies the need for SMACS as opposed to the traditional link layers used in wireless networks since, these schemes require nodes of the network to be listening to the network at most times, thus, involving energy exemption even during idle periods. The 2 main types of link layer protocols used in traditional wireless networks which are unsuitable for wireless sensor networks:

1. Contention-based channel access: The MAC layer design of 802.11 is an example of such a scheme. These schemes are not suitable for sensor networks since, radio transceivers need to monitor the radio channels at all times.

2. Organized channel access: This requires nodes in the network to be synchronized at some level. This network-wide synchronization requires extensive message passing among the nodes which is a very expensive procedure for sensor networks.

Like link-layer protocols, there are also several routing protocols discussed in the literature on sensor networks. These protocols are required to be able to detect quickly and work around a node failure while placing low processing and communication demands on the nodes. Two main protocols discussed for ad-hoc networks are Ad-Hoc On-Demand Distance Vector Routing (AODV) and Temporally Ordered Routing Algorithm (TORA). Both these algorithms are on-demand and hence, the route setup phase is costly. Singh et al. [47] compares the performance and efficiency of these algorithms. Jeon et al. [48] discuss a variation of the AODV algorithm in order to setup backup routes to handle failures and packet collisions. Another algorithm called Power-Aware Routing [5] finds minimum metric paths on two different power metrics:

1. Minimum energy per packet

2. Minimum cost per packet.

Sohrabi et al. [4] discuss a table-driven multi-path approach for routing in sensor networks with limited mobility. This algorithm creates multiple paths from each node to the sink, multiple trees, each rooted from a 1-hop neighbor of the sink, are built. Each tree will be forced to grow outward from the sink by successively branching, whenever possible, to neighbors at higher hop-distance from the sink while avoiding nodes with very low QoS and energy reserve. At the end of the tree building procedure, most nodes will belong to multiple trees and thus have multiple paths that are disjoint inside the 1-hop neighborhood of the sink. The advantage of this structure is that it allows each sensor indirect control of which 1-hop neighbor of the sink will relay a message. For each node, two parameters are associated with each path: (1) energy resource estimated by maximum number of packets that can be routed without energy depletion if it has exclusive use of the path, (2) additive QoS metric where higher metric implies lower QoS.

As discussed earlier, topology control is another important area of research in self-configuring networks. Karnik and Kumar [3] address the problem of building an optimal network topology using distributed algorithms. The paper states that it is not only important that the sensor nodes self-organize but also that they do so to form an optimal topology. The paper treats this as an optimization problem and describes a distributed algorithm to achieve it. Santi [49] present a survey on topology control in sensor networks. Cerpa and Estrin [50] propose adaptive techniques that permit applications to configure the underlying topology based on their needs while trying to save energy to extend network lifetime. It proposes a scheme wherein each node assesses its connectivity and adapts its participation in the multihop network topology based on the measure operating region. For instance, a node: 1) Signals when it detects high packet loss, requesting other nodes in the region to join the network to relay messages. 2) Reduces duty cycle if it detects high packet losses due to collisions. 3) Probes the local communication environment and does not join the multihop routing infrastructure if it is not helpful to do so. The experimental results show that this scheme has the potential for greatly reducing packet loss while improving energy efficiency.

A large portion of the literature discusses the architecture of self-configuring systems - the components that form this system and their roles in the self-configuring process. Subramanian and H.Katz [6] discuss one such architecture of self-configuring systems. This paper identifies the following components to be a part of self-configuring systems:

1. Specialized nodes: Nodes that perform the application-specific functions. Examples include camera sensors.
2. Routing nodes: Each specialized node is connected to one routing node that is responsible for data dissemination.
3. Aggregator nodes: These nodes aggregate the information obtained from several specialized nodes.
4. Sink Nodes: These are powerful computing systems that store large amounts of data and connect the sensor network to the Internet or to other large networks.

The paper discusses a self-configuring algorithm which helps in self-organizing a set of sensor nodes randomly scattered in an area. The router nodes self-configure themselves using this algorithm and the specialized sensors only keep track of the nearest router node that is alive. The algorithm consists of four phases:

1. Discovery phase: Each node independently discovers its neighbors in the network.
2. Organizational phase: During this phase the following operations are performed:

   (a) Nodes aggregate themselves into groups and groups are aggregated into larger groups.
   (b) Each node is allocated an address based on its position in the hierarchy
   (c) A routing table of $O(logn)$ is computed for every node in the network.
   (d) A broadcast tree and a broadcast graph spanning all nodes in the graph is constructed.

3. Maintenance phase: In this phase, the nodes send keep alive messages to their neighbors periodically and constantly update their routing table about the next hop in the least power consuming path. Nodes also inform their neighbors about their routing tables and energy levels to neighboring nodes.

4. Self-Reorganization phase: In this phase nodes detect group partitions or node failures and change their routing table based on the topology. The algorithm mainly targets power constraints and attempts to minimize the power consumed at various stages of the algorithm. The paths and the tree structures are made fault tolerant by constantly making failing nodes as leaf nodes.

Malatras et al. [51] discuss a Self-Configuring and Optimizing Mobile Ad-hoc network (SCOMANET) to exploit context awareness and cross-layer design principles to achieve self-configuration and self-optimization. In this scheme, each mobile node collects its own low-level context and processes it to higher level context information that has an impact on the management plane of the ad-hoc network. A set of nodes in the MANET form a management body that identifies the need for possible configuration changes based on this information and pre-defined policies. The management entities of the MANET are present on the connected domain set of the graph that describes MANET connectivity.

2.1.2 Self-Configuring in Wired IP networks

The growing heterogeneity and complexity of Internet services has complicated the management of network devices to a great extent. Autonomic networking is being used to control this complex environment in an efficient and automatic manner. It aims to enhance network elements with capabilities that allow them to choose their own behavior for achieving high-level directives. Lehtihet et al. [52] attempt to build a self-configured and self-optimized IP network that automatically supports the QoS requirements of heterogeneous applications without external intervention. The paper proposes a new goal-based management approach for the management of communication systems.

The goal-based framework is organized into two layers:

1. Goal specification layer: This layer defines the high-level objectives (goals) in terms of explicit behavior expected from various autonomic network elements. The goals are defined in the form of state machines representing different possible network element behaviors as well as the transition between states. The Goal Server is responsible for performing this function.

2. Goal execution layer: This layer consists of the autonomic network elements that work in an autonomic manner to achieve the goals. These elements are self-managed and capable of adapting their behavior to the environment.

In addition to proposing mechanisms that make a network a self-configuring system, a lot of the literature on self-configuring in wired networks also uses the already existing self-configuring properties of peer to peer systems to build more complex systems to achieve different functionalities like Grid Computing and so on. Ganguly et al. [7] discuss a network virtualization technique based on IP tunneling over peer to peer (P2P) networks for grid computing. Current network virtualization techniques for Grid computing require an administrator to setup overlay routing tables. Hence, the process of adding, configuring and managing clients and servers that route traffic within the overlay is not scalable. This paper proposes a protocol that supports seamless, self-configured addition of nodes to a virtual IP network. It exploits the self-configuring property of P2P networks that allow user mobility and is scalable, providing extremely robust service.

2.1.3 Self-Deploying Networks

The literature on self-deploying networks explores the possibility of mobility of base stations, allowing adjustment of the network to changes in user-traffic. While base station mobility seems futuristic for commercial wireless applications, the concept has near-term applications in the field of emergency and military communications where fast network deployment is required in high-risk areas. Claussen [53] discuss the autonomous self deployment of networks in an airport environment. In environments with large and dynamic user densities, a high over-dimensioning of wireless networks is needed in order to support the large amount of user traffic. The paper discusses self-deployment as an alternative to this. Distributed algorithms are proposed which autonomously identify the need for change in position and configuration and adapt the wireless network nodes to the environment. It proves that a self-deploying network greatly reduces the number of base stations required as compared to a statically deployed network. Claussen et al. [8] explore the concept of robotic, wireless base stations and discuss their behavior subject to principles inspired by Asimov's Law of Robotics. This paper introduces the concept of robotic base stations which would automatically sense their environment based on different types of measurements and use these to adapt various parameters and possibly even their position. The backhaul for the robotic base stations would be provided via wireless links to an anchor point acting as a gateway to an IP network.

Self-deployment is more common in the case of wireless sensor networks since, significant efforts are involved in trying to manually position a large number of sensors over a large geographic area. Huang [54] discusses one such method called Ion-6. The authors model sensors as individual ions and the links between them as ionic bonds. With this model, sensors can calculate their direction of movement and distances independently and the problem of self-deployment is converted to a problem of building ionic bonds. Through experiments the authors prove that the resulting sensor positions adhere to the near-optimal hexagon topology and do not move back and forth too much between regions either. On similar lines, [55] models the received signal strength at a sensor as the repulsion force used to calculate the sensor's move distance and direction. The authors propose the use of a hybrid approach to save the energy spent on mobility and communication during self-deployment.

Self-configuring is thus, the need of the hour for the following reasons: 1) To manage the large network complexity due to the variety of applications with a diverse range of QoS and bandwidth requirements, 2) To reduce operating expenditure of wireless networks on deployment and maintenance as the user density grows necessitating the need for smaller cells and a larger number of base stations, 3) To reduce the wireless operators capital expenditure spent on engineering and installation services, 4) To ensure the wireless networks of the future, are able to dynamically adapt to the changing environment without the need for external intervention.

2.2 Self-Optimizing Networks

In this section, we look at self-optimizing properties of different types of wireline and wireless networks. Although self-optimizing in networks is prevalent now because of the advent of 4G networks, it has also been used to some extent in wired IP networks. However, this forms a very small portion of the literature, so we do not discuss it here. The literature on self-optimizing networks can be divided into three categories:

1. Self-optimizing in LTE networks - This is the predominant usage of self-optimization in communication networks.
2. Self-optimizing in Ad-hoc and Sensor Networks - Due to the nature of these networks in terms of absence of centralized control and the need to optimize resource usage, self-optimization is essential in these networks.
3. Frameworks for self-optimization - Literature in this category proposes how self-optimizing systems must be built and the essential components of such systems.

2.2.1 Self-Optimizing in LTE networks

Important use cases for self-optimization in LTE include coverage and capacity optimization, energy savings and mobility load balancing optimization. However, a large portion of the literature on self-optimizing in LTE networks discusses self-optimization as applied to load balancing in these networks.

Viering et al. [10] discuss that the self-optimizing algorithms need to be sluggish in order to avoid collisions with radio resource management functionality. Due to this requirement, most system level simulators do not qualify for research on these algorithms because they track underlying variations in the order of seconds rather than in minutes, hours or weeks as is required by the SON algorithms. The authors propose a mathematical framework that can be used for this purpose and can be used as the basic for simulation tools. The basis of this mathematical framework is the definition of the target functions, i.e., the metrics that will be used to evaluate functions. The authors propose two classes of target functions: a) capacity-based targets which optimize throughput related figures and b) energy-based targets which optimize power consumption of base stations and UEs. The capacity- based targets are based on the signal-to-interference and noise ratio of every user and the different throughput parameters are defined as a function of the SINR calculated per UE. The SINR itelf is defined as a function of transmit power, antenna down-tilt and connection function, which are important optimization parameters. The energy based targets are calculated based on the overall power consumption of each cell. Based on these, the authors provide an approximation and iterative solution defined different target functions such as the number of unsatisfied users. In order to demonstrate the usage of this framework, the authors use it to evaluate a simple-load balancing algorithm that they propose. The load-balancing algorithm adjusts the handover offsets for cells until the point that all users are found to be happy or all cells have reached a minimum handover offset. The basic idea of the algorithm is to decrease the served area of the overloaded cells by decreasing the handover offset for these and to increase the served area of the underloaded cells. Wei and Peng [11] discuss a similar mobile load balancing algorithm for hybrid architecture self-organizing networks that takes the neighbor cell call load status into account in order to avoid the ping-pong effect between source and target cell after a handover. This algorithm requires the assistance of the Operations and Maintenance (OAM) entity in the network.

Lobinger et al. [12] run a more detailed simulation of the algorithm proposed in [10] and refine the algorithm further. In order to determine the target cell for handover, each cell first prepares a list of candidate cells that could be possible targets. It then estimates the load condition of these cells and the impact of changing the handover parameters on the load of the cell. Load estimation is done based on SINR prediction and utilizes UE measurements. The authors prove through simulation that the average number of satisfied users can be improved with the load-balancing algorithm and the proposed metric for load estimation. Jansen et al. [56] discuss the use of certain handover performance indicators (HPI) in order to steer handover parameters that drive various load balancing algorithms. The HPIs indicate the current handover performance of the network. The prominent HPIs discussed are the radio link failures due to too low SINR conditions, the handover failures detected as too low SINR conditions during handover execution and the ping-pong handovers detected as handovers returning to the originating cell within a short time. The algorithm observes the individual HPIs per cell and computes a weighted sum in order to evaluate the handover performance, giving radio link failures the highest priority.

Lobinger et al. [57] present simulation results of a self-optimizing LTE network which uses the algorithms proposed by [12] and [56]. The authors discuss that the two algorithms change different parameters but control the same decisions related to handover. Through simulations, the authors demonstrate the interaction between the two algorithms, which each capable of reducing the impact of the other. One proposed solution to avoid this is to combine the two algorithms into a single function, thus eliminating the need for them to explicitly interact and co-ordinate with each other. However, this cannot be done for all SON functions and the need is for a more generic approach to handle interactions between different SON algorithms. The authors propose one such approach of introducing a SON function which looks at the KPI metrics and uses these to control the two algorithms by limiting their optimization actions. The algorithms adapt their control parameters based on the KPI metrics if necessary. Simulation results show system performance comparable to the individual performance of the two algorithms and in some cases better than that of the individual algorithms. The advantage of this solution is that the individual SON algorithms stay unchanged and a new module called the coordinator takes control on top of them.

Nihtila et al. [58] propose another algorithm that builds on top of [12]. The authors use a combination of algorithms of [12] and [59]. The authors contend that in the uplink (UL) direction, the load of a cell is not a sufficient trigger for load-balancing because the UE may also be limited by other aspects such as power or the number of control channels. The authors define UE's power limitation as the condition in which the UE is not transmitting enough to be happy but because of being in poor channel conditions, it cannot increase its transmission power and data rate. The UL scheduler may also run out of Physical Downlink Control Channel (PDCCH) resources, used to inform the users of upcoming scheduling grants. This may happen when several users are ready to transmit but a lot of these may become unhappy users because of the lack of PDCCH resources even though lots of free PRB resources are present. In these cases, the load of the cell may be in normal range and may indicate no need for any sort of load-balancing. Hence, the algorithm used for the downlink needs to be enhanced to take care of these limitations. This enhancement comes in the form of LAPC (Load

Adaptive Power Control) of [59] to minimize UL interference by letting users transmit with lower power in low load situations. This decreases the UL interference implying a decrease in power limitation. But this also means a smaller SINR for the UE, meaning a larger bandwidth would be needed to reach the same committed bit rate. The authors demonstrate through simulations that the plain load balancing is most effective with low bit rate UL services. Higher bit rate UL service users suffer from power limitation and are benefitted most by LAPC. This is because plain load balancing algorithms force users to connect to worse cells making power limitation worse. Thus, load balancing used without LAPC in UL may actually hurt UL performance.

Amirijoo et al. [9] assess the impacts of antenna tilt and transmit power settings adaptation on cell outage management. The authors assess these impacts by taking into account the site density and traffic load in the networks, using metrics such as the number of satisfied users and the uplink/downlink throughput as deciding factors. The numerical results indicate that the transmit power and antenna tilt management are most effective in improving user throughput and coverage in the face of cell outages.

2.2.2 Self-Optimization in Ad-Hoc and Sensor Networks

Mobile ad hoc networks (MANET) are collection of geographically distributed nodes that can be self-configured to form a network without predetermined topology. Since, there is no centralized control, these networks, like LTE are the main reason for the research on self-optimization and other self-* properties. The main problems in these networks are those of routing and energy-efficiency. Since, the nodes are not fixed, they can move in and out of the network requiring all other nodes to adjust their routing entries. The routing algorithms need to thus be extremely adaptable to changing conditions and must be able to inform nodes of the change in routing without too much delay. However, increased communication between nodes means increase in energy usage. This is something that ad-hoc and sensor networks cannot afford because of the energy limitations of individual nodes. As a result most optimization algorithms seek to achieve their functionality in a way that always optimizes the energy usage in these networks. In several cases, special algorithms need to be developed to just take care of the energy efficiency in such networks.

As discussed above, the main problem in MANETs is that of routing. In MANETs, it is important to maintain correct routes to the destination in a changing topology resulting from node failure or mobility. Several algorithms have been proposed, chief among them being Ad-hoc On-demand Distance Vector(AODV), the Dynamic Source Routing (DSR), the Destination Sequenced Distance Vector (DSDV) and the Temporally-Ordered Routing Algorithm (TORA). All of these however, use the shortest hop to compute the best route and do not take into account the load on individual links. Coucheney et al. [13] propose an algorithm that takes into account the load of each in flow in a MANET before deciding in the best route. The algorithm is built on top of AODV but provides an advantage over AODV, since AODV does not take the load into account. The authors provide a fully distributed algorithm and provide a theoretical validation of the algorithm based on three fields: game theory, Gibbs sampling and distributed algorithms. The authors prove through simulations that using Gibbs sampling to choose the best route among a small number of candidate routes for every flow independently provides an optimal configuration and perfoms better than AODV. Shi et al. [60] also use game theory to achieve self-optimization in wireless sensor networks, using it to achieve optimization of different aspects such as QoS control, topology optimization and routing optimizations.

Saleh et al. [14] discuss the energy limitation problem in wireless sensor networks (WSN). The authors present a self-optimization scheme for WSNs which is able to utilize and optimize the sensor nodes' batteries to achieve balanced energy consumption across all sensor nodes. This energy balancing method is decentralized and is made as generic as possible to allow any WSN application that requires reduction in energy consumption to be able to use it. The authors also develop a routing algorithm whose main consideration is the balancing of energy consumption in order to maximize sensor network lifetime. The authors use the Ant Colony Optimization (ACO) metaheuristic to enhance the paths with the best quality function. The assessment of this function depends on multi-criteria metrics such as the minimal residual battery power, hop count and average energy of both route and network. In order to reduce energy usage, the algorithm also distributes the traffic load of sensor nodes throughout the network leading to reduced energy usage, extended network life time and reduced packet loss. The authors prove through simulations that the proposed routing algorithm functions much better compared to other prominent WSN algorithms in terms of total energy usage, energy efficiency and energy balancing. Camilo et al. [61] discuss the Energy Efficient Ant-Based Routing (EEABR) algorithm that attempts to extend the lifetime of WSNs by reducing the communication overhead in the discovery phase of routing protocols. As the name indicates, this algorithm is also based on the ACO meta-heuristic and is very similar to the previous algorithm except for the assessment functions used and the table updation criteria.

2.2.3 Frameworks for Self-Optimization

As the research in self-optimization gathers momentum, researchers have discovered that regardless of the network in which it is applied, all self-optimizing systems have several common components. These can be integrated into a common framework to allow a homogeneous architecture of self-optimizing systems regardless of the application or the network.

Soldani et al. [15] present an autonomic framework that can be used for many autonomic tasks, but the focus is on self-optimization. The main components of this framework include:

1. Policy Matrix: This comprises of two types of policies : a) Monitoring policies - These consist solely of a condition determined in terms of network measurements or parameters and b) Analysis policies: These consist of conditions in terms of network measurements and a function consisting of several actions.
2. Autonomic Manager: This forms the main component, comprising of the monitoring, analyzing and execution sub-components. The monitoring sub-component evaluates all the monitoring policies periodically and evaluates them based on the measurements and other relevant parameters in the Repository. If any of the conditions evaluates to true, the Analyze and Plan process is activated. This process goes through all the analysis policies contained within the policy matrix and initiates the functions of all those policies whose condition evaluates to true. The data returned by these functions is given to the Execute process which checks and resolves conflicts and executes the actions that remain. The Autonomic Manager

achieves its functionality using various sub-agents so as to avoid embedding domain functionality in the Autonomic Manager itself. This also allows for a degree of generality and scalability.

The authors use this framework to build an autonomic system that solves two of LTE's prominent SON problems: neighbor list prune and cell removal and redeployment. Using these, the authors attempt to assess the system adaptability, time to react, stability and sensitivity. The authors demonstrate that the autonomic manager has the capability to react and adapt to changes in the environment and to stabilize within the constraints set by input policies. The stability of the system is dependent on the combination of policies and sub-agents. This paper is the most prominent work in the field of frameworks for self-optimization and several other frameworks have been developed that are a variation of this one.

Another interesting approach to self-optimizing frameworks is proposed by [62] which aims to build a self-optimizing mobile network using agents called fireflies. The fireflies are built on the concept of swarm-technology [63] and perform the task of distributed agents working together to achieve auto-tuning of mobile network parameters for optimized operation. These fireflies do not require any centralized or de-centralized control and are completely autonomous.

As the literature proves, the current focus of self-optimizing networks is clearly load optimization in LTE networks and energy optimization in ad-hoc and sensor networks. These networks thrive on the presence of self-optimizing algorithms and their success is dependent on the effectiveness of these algorithms.

2.3 Self-Healing in Wireless Networks

In this section, we focus on the property of self-healing and survivability of communication networks, where survivability is defined as the ability to provide uninterrupted services to applications amidst unanticipated network element failures in communication networks [27].

Self-healing in communication networks has a variety of applications, ranging from improving performance and availability of wired systems, to being the foundation principle for ad-hoc networks in providing capabilities like improving the QoS. Kant and Chen [27] define survivability as the ability to provide uninterrupted services to applications amidst unanticipated network element failures in communication networks. The authors specify two main reasons for requiring self-healing and survivability: a) the dynamic and unpredictable nature of wireless networks wherein network resource availability can diminish sporadically, and b) wireless resources are expensive. Since wireless networks are increasingly used for applications like video-conferencing where delays affect the end-user experience, self-healing and survivability mechanisms must take into account criticality of applications in order to prove most effective. Psaier and Dustdar [28] survey various self-healing systems and define self-healing as the ability of the system to recover from abnormal state to normal state, to function in the same fashion as it did before the failure.

The literature in the area of self-healing and survivability can be divided into three main categories:

1. Literature that proposes a framework for self-healing: This category comprises a major portion of the literature on self-healing. Generic frameworks and architectures are proposed to facilitate self-healing in different types of networks.

2. Self-healing in the context of ad-hoc networks: Self-healing forms the basis of ad-hoc networks and hence, there is a large amount of literature that discusses the use of self-healing mechanisms to address the various types of problems in these networks.

3. Self-healing using biological foundations: While this category is similar to the first category that proposes a framework for self-healing, it requires a special mention due to the amount of research dedicated to it. This category of literature attempts to base the self-healing of networks on the way self-healing is accomplished in the human body. Various artificial intelligence mechanisms attempt to emulate the human body and accomplish self-healing in communication networks.

We look at the literature in these areas in more detail in the remaining sections.

2.3.1 Frameworks for Self-Healing

The self-healing mechanisms and frameworks discussed in the literature can in turn be divided into three main categories:

1. Policy-based mechanisms: This is the most prevalent scheme for achieving automated management of any system. The operator defines policies during the startup of a system. These policies define a) the parameters to be monitored b) the threshold values for these parameters c) actions to be taken when the values of these parameters change. Some of these policies can be tuned by the operator during the operation of the system.

2. Adaptive policy-based mechanisms: These schemes are an improvement over the policy-based mechanisms. In the adaptive policy-based mechanisms, the system while acting on certain conditions also learns from them and records this information in the form of a new policy or as a modification of an already existing policy.

3. Biological foundation mechanisms: The essence of these mechanisms are the schemes that allow a biological system to scale, adapt and survive. Applying biological analogies has led to some efficient techniques for solving self-healing problems. The most prevalent examples of applying biological analogies include artificial neural networks, genetic algorithms, molecular computing and swarm intelligence.

The literature on survivability primarily consists of two types of schemes to make the networks more fault-tolerant and to allow them to operate even in adverse conditions. The most popular scheme is the Automatic Protection Switching (APS) scheme. This scheme is based on the principle of redundancy. Some network components are dedicated for redundancy and only come into operation when the corresponding network component fails. While, this ensures a high degree of availability, it is expensive in wireless networks. Wireless networks have limited resources and having some parts of the network dedicated only for redundancy is wasteful. The APS scheme is also more suitable for hard and deterministic failures and not the soft and non-deterministic failures seen in wireless networks. The second survivability scheme uses the APS redundancy mechanism only for time-critical, high priority applications. Restoration of services for other applications is done in the higher layers of the protocol stack such as the network layer or the transport layer.

2.3.2 Self-Healing Overview

Regardless of the type of scheme used to achieve self-healing, the basic conceptual architecture of a self-healing system remains the same. Autonomic elements are the building blocks of self-managing systems. Agoulmine et al. [29] propose an architecture is composed of a number of functional modules that enable autonomic behavior by performing several autonomic operations. The autonomic operations are achieved using a self-adjusting control loop that consists of status signals from the component being controlled along with policy-driven management rules.

The various modules in the figure are described as follows:

1. The Sensors and Effectors together form the Control Module. The sensors provide mechanisms to collect events from the environment while the effectors allow the configuration of its managed resources.
2. The Monitoring Module provides the Autonomic Element different mechanisms to collect, aggregate, filter and manage information collected from the environment.
3. The Analyze module performs diagnosis of the Monitoring results and detects any disruptions in the network or system resources.
4. The Planning Module specifies the actions to be performed depending on the events.
5. The Execution Module performs the actions specified by the Planning Module.

In essence, the process of self-healing and autonomic management consists of :

1. Monitoring the environment to determine the operating conditions.
2. Analyzing the data from Step 1 to check for abnormal situations or situations that require adaptation by the system.
3. Based on the results from Step 2, determining the actions to be performed.
4. Executing the actions from Step 3.

2.3.3 Biological Schemes

Agoulmine et al. [29] describe some biological analogy based schemes used in self-management and the associated challenges. These schemes map biological principles to local rules within the autonomic elements. In order to be successful, these schemes require a de-centralized approach to local decision making. Some important biological principles that are mapped to autonomic elements are discussed in this paper. These include:

Neural System: In organisms, the neural system gives the organism the ability to receive and act upon stimuli. The units of the neural system that receive and process the input stimuli and produce an output are called neurons. The neurons operate in networks. Based on the patterns of these networked neurons in the brain tissue, the brain learns from the experience.

The neural system mapping to an autonomic element allows it to learn from past experiences. In the context of self-healing, this is very important as a learning system means that there would be no need of operator intervention and the fine-tuning of policies would be done automatically by the system. For example, routers can use past experience to make a decision on the packets to be dropped in heavily loaded conditions.

Immune System: The immune system is the bodys defense mechanism against external invaders. In the event of an attack by foreign organisms (called pathogens), the immune system attempts to counter them. However, there are times when the strength of the pathogens is greater than that of the immune system. In this case, the immune system resorts to the adaptive immune system. The cells that form the adaptive immune system produce a large number of clones of themselves. Antibodies that combat the antigens produced by pathogens are produced as a part of this cloning process. The combination of the clones and antibodies succeeds in getting rid of the pathogens. In addition, the immune system cells also possess memory of antigens that have already been seen, causing them to respond faster upon repeated exposure to the same pathogen.

The immune system mapping to an autonomic element allows it to protect itself and recover from malicious attacks. Similar to the cells used by the adaptive immune system to overcome pathogens, the autonomic element can detect malicious attacks and respond to them. The immune system memory enables faster detection of attacks in future, allowing the system to react quickly. The ability of the autonomic element to function as an immune system means that there is no need for someone to constantly monitor the system for attacks since, the system can detect and heal itself from these attacks.

2.3.4 Policy-Based And Adaptive Policy-Based Mechanisms

Soldani et al. [15] propose a policy based framework for self-management. The behavior of this framework is governed by a policy matrix. The policy matrix consists of analysis policies and monitoring policies. The monitoring policy specifies the parameters that are to be monitored and the analysis policy uses these measurements to determine if any action is needed and the particular action. An analysis policy thus, consists of a condition in terms of network parameters and a function, which may include several actions.

Shen et al. [64] use the policy based mechanism to manage hybrid network behavior. They use three types of policies: action policies, goal policies and unity policies. Action policies are of the form IF (condition) THEN (action), which means the specified action must be taken when the condition is satisfied. Goal policies are at a higher level than action policies and specify the conditions to be achieved but not how to achieve them. The goal policies have a higher influence than the action policies, because an autonomic element can request other autonomic elements without detailed knowledge. Utility policies automatically determine the most important goal in any situation and hence, exercise the most influence when compared to the other policies. In each of the three policies, automated decisions are made by using machine learning approaches.

All policy based mechanisms can be extended to adaptive policy-based mechanisms by providing the autonomic elements with memory to remember conditions and intelligence to use these in future decisions.

2.3.5 Survivability

Self-healing and survivability are closely related concepts, dealing with the ability of the network to continue operation in adverse conditions. As discussed in the earlier sections, most systems attempt to achieve survivability by using APS based mechanisms which are rather expensive for wireless networks. Kant and Chen [27] attempt to achieve survivability using multi-layer self-healing. They propose to use priority and criticality of applications to determine the layer at which the healing will be formed. Only the most critical of applications will be restored using the APS mechanisms.

Kant and Chen [27] discuss the disadvantages of APS based mechanisms for wireless networks. In addition to being expensive, the APS mechanisms do not take the priority of the applications into account. Since, each application has its own service survivability requirements, this approach of switching all application en masse is unacceptable. This work proposes a multi-layer mechanism to provide service restoration. They propose the use of SCTP as the transport protocol instead of TCP. SCTP is preferred because of its support for multi-homing. The fault management module of the system determines the active interfaces and constantly monitors their load. In the event of an interface failure or congestion on a link associated with an interface, the affected applications are switched to an alternate interface.

Self-healing at the L3 layer is discussed in [65], [66]. At L3, self-healing is achieved by re-routing around the failed network element. The affected services will therefore, be restored by performing a priority-based on-demand re-routing around the failed network elements.

Kant [66] discusses some self-healing mechanisms of overcoming failures in a GPRS-based wireless network. This paper discusses restoration schemes for the different types of elements that may fail in the wireless network. Some of these include:

1. Failure of the link between two MSCs: In this case, the restoration mechanism will determine the services that have been affected by the failure of this link. These services will then be re-mapped and restored using other available links and MSCs in the network.

2. Failure of one of the RF channels of the BTS: In this case, the MSC is contacted to determine the services that are affected by this failure. Once that is determined, these services are then re-allocated to alternate channels in a manner similar to a handover.

3. Failure of the link between the BSC and the MSC: The connectivity of the BSC to the MSC represents a single point of failure and failure of this link will result in several BTSs being disconnected from the network. Hence, restoration of this

requires a redundant link to which all the services will immediately be remapped. The paper presents results obtained by simulating the above failures. The restoration time for services in the event of a failure in the Core Network was of the order of 26 msec which is not very significant.

Sterbenz et al. [67] discuss the three main thrusts that have the potential to drastically increase the survivability of wireless networks. These include establishing and maintaining network connectivity in adverse conditions, expecting and exploiting mobility and exploiting technology advances such as adaptive protocols and satellites to enhance connectivity.

To establish and maintain connectivity in the network is the first main goal. Establishing connectivity involves auto-configuration of nodes. Maintaining this connectivity needs dynamic adjustment of transmit power of nodes and having bi-connected topologies to survive the loss of individual links. A bi-connected topology is one in which loss of a single link still leaves the network connected. Use of adaptive nodes and protocols allows the dynamic selection of not only MAC and network layer parameters but also complete protocols based on application requirements and the environment. On the other hand, satellites can play an important role in mitigating the effects of weakly connected channels and node mobility. The high altitude of a satellite enables it to have a very large terrestrial footprint within which any ground node can receive communication. This coupled with the inherent broadcast capability of the satellite allows it to communicate better with any ground node.

2.3.6 Self-Healing in Ad-hoc Networks

Ad-hoc networks have brought a new paradigm in communication making pervasive computing a reality. Due to the lack of infrastructure and constant mobility of nodes in ad-hoc networks, all nodes must possess self-management capabilities to enable the network to be formed and sustained. Nodes joining an ad-hoc network need to be able to configure themselves based on the properties of the current network. Nodes leaving the network may result in fragmentation of the network and hence, the other nodes in the network will have to employ self-healing to ensure seamless communication in the presence of these departing nodes. Aly et al. [30] discuss the problem of single link failures in ad-hoc networks and develop a scheme called network protection codes (NPC) in order to recover from these failures. This scheme is also extended to recover from multiple link failures which are most common. This scheme tends to reduce the network capacity but the authors prove through experiments that this is only a fraction. In addition the authors prove that this scheme is better than 1+1 redundancy of links in terms of efficiency. On similar lines, Conti et al. [31] look at the detection of node replication attacks in wireless sensor networks. The authors propose a randomized, efficient and distributed protocol for detection of these attacks and prove that this protocol is more efficient than others in the literature in terms of its use of computation, memory and communication resources.

Due to the high mobility in ad-hoc networks, developing routing algorithms for them is a challenge. The routing algorithms must be able to detect and respond quickly to the loss of links, which is a common occurrence in ad-hoc networks. Automated network discovery through link and route discovery and evaluation lies at the heart of the self-healing network algorithms. Through discovery, a path is established between the originator and recipient of a message. Through evaluation, networks detect route failures and trigger route discovery for alternate routes. However, ad-hoc networks have limited network capacity and the nodes involved are constrained by the available power. Hence, any algorithms designed must take the minimal resources into account in order to achieve good network performance.

Poor et al. [68] discuss two main types of route discovery algorithms:

1. Proactive Discovery: These algorithms configure and reconfigure constantly and assume that link breakages and performance changes are continuously happening. Hence, they are structured to continuously discover and reinforce optimal linkages. Proactive discovery occurs when nodes assume all routes are possible and attempt to discover every one of them. This strategy of route discovery is an “always on” strategy resulting in significant network resource usage even in low traffic conditions, which is unacceptable in ad-hoc networks.

2. On-Demand Discovery: This scheme, in contrast, only establishes routes that are called for by the applications. This preserves network resources. However, during times of link degradation, these algorithms can take a longer time to react and re-configure. Ad-hoc On-Demand Distance Vector Routing Protocol and Dynamic Source Routing Protocol are examples of on-demand discovery. Maintenance of

the established routes can also be proactive or on-demand. The actual routing of packets can similarly follow two strategies:

(a) Single Path Routing: In this case, a single route is selected for a source-destination pair. The entire end-to-end route may be predetermined or just the next hop may be predetermined.

(b) Dynamic Routing: This strategy takes advantage of the broadcast nature of the wireless medium and the packets are broadcast to all neighbors. The packets are forwarded from the neighbors based on a "cost-to-destination" scheme.

Trehan [69] discusses how to keep ad-hoc networks connected and robust in the face of adverserial attacks. The author discusses three self-healing algorithms to maintain network connectivity in the face of adverserial attacks to the network that lead to deletion of random nodes in the network. The algorithms concentrate on maintaining connectivity, ensuring low degree increase for all nodes and ensuring a low diameter increase of the network during network reconstruction. The algorithms, namely, Degree Assisted Self Healing, Forgiving Tree and Forgiving Graph differ primarily in the degree of increase.

Another prominent application of self-healing in ad-hoc networks is that of key distribution for secure communication amongst nodes. Tian et al. [32] and Wang [33] survey the various schemes available for key distribution in ad-hoc and wireless networks. Wang [33] compare various schemes and find that almost none of the available schemes are suitable for resource-constricted wireless sensor networks and there still exists an urgent need for a secure yet efficient scheme to achieve key distribution. Tian et al. [32] studies the characteristics of each scheme and uses this to arrive at the common architecture of key distribution schemes while also exposing the common vulnerabilities amongst these.

Self-healing makes the ad-hoc networks robust to link failures, helping them cope with unreliable wireless links. However, this dynamic nature of ad-hoc networks makes the network and the corresponding protocols difficult to test. As is proved by the broad variety of literature on self-healing networks, this is one of the first self-management paradigms to have been used extensively in all types of networks. All systems need mechanisms that allow them to recover from failures without manual intervention - hence, the research in this area successfully borrows ideas from even the biological field to build robust communication networks.

2.4 TCP Performance Improvements through Self-Optimization and Self-Healing

To enable a seamless movement from the wired to the wireless access domain, it is important to solve impending problems that reduce the achievable throughput and the efficiency of wireless networks. One such problem is the reduction in throughput observed due to TCP's congestion control mechanisms that come into operation during periods of mobility or fading of a wireless link.

While this problem needs to be addressed immediately, it is important to keep in mind that most of the current Internet infrastructure is TCP-based and hence, any change proposed must not entail large changes to the Internet infrastructure. Massive changes to the operation of the protocols may introduce the risk of destabilizing the entire infrastructure while also incurring large overheads in terms of effort and cost. The literature in this area has proposed several mechanisms, some requiring changes to only the mobile and some others requiring changes to the mobile as well as the base station. This section surveys the various categories of solutions proposed and compares various aspects of these mechanisms, such as the extent of change required, the performance and so on.

The literature in this area can be categorized into two broad categories - reactive schemes versus proactive schemes. Most of the research in the area of communication networks involves finding different ways of tackling a problem once it has occurred. We call these mechanisms as reactive schemes. An alternative is, for the elements in computer networks to be able to forecast the onset of a problem and take preventive action. We call these mechanisms as proactive schemes. In the proactive scheme, a node may learn from previous "experiences" of the start of an adverse condition. The node may then take preventive action, which may be able to reduce the adverse effect of this condition. A node may also use some pre-configured parameters to detect such conditions. For example, a base station may be configured to predict an impending disconnection, if the strength of the signal from the mobile falls below a certain threshold. Section 2.4.1 discusses further categorization of the various TCP performance improvement mechanisms. Section 2.4.1 discusses the various proposed mechanisms to improve TCP throughput in wireless networks. Sections 2.4.2- 2.4.5 describe these mechanisms in further detail and discuss their advantages and disadvantages.

2.4.1 Proposed Solution Categories

The various mechanisms proposed can be divided into three main categories, as proposed by [70] and [71] :

1. Link-Layer Schemes
2. End-to-End Schemes
3. Split-connection Schemes

Link Layer Schemes: This category of solutions involves link layer retransmissions (local retransmissions) to keep the TCP layer transparent to losses, to prevent triggering of the congestion control mechanisms. Another technique used in link layer schemes is error correction using mechanisms such as forward error correction (FEC) and so on. The most well known solution in this category is the Snoop protocol [36] discussed in Section 2.4.2.

End to End Solutions: End to end solutions comprise mechanisms like Selective Acknowledgements (SACK), which is an enhancement of the fast recovery scheme in TCP. In the SACK scheme, the duplicate ACK contains the sequence number of the packet that triggered the duplicate ACK. Another famous end-to-end scheme is the Explicit Loss Notification (ELN) scheme. If a packet is lost due to lossy links, a bit is set in the ACK, to indicate to the sender that congestion control mechanisms need not be invoked. Freeze-TCP [40] is another end-to-end mechanism that involves the receiver, advertising a window of zero, when it detects an impending disconnection. This causes the sender to freeze in its current state until the advertised window size is increased. These mechanisms are discussed in detail in Section 2.4.3.

Split-connection schemes: These schemes attempt to solve the problem by splitting the TCP connection between the fixed host and mobile host into two separate connections - one from the fixed host to the base station and the second from the base station to the mobile host. The main aim of this technique is to keep the fixed host independent of the problems of wireless connectivity. Special mechanisms or protocols can be used on the wireless part of the connection. Some of the problems with split connection techniques include loss of end to end semantics of TCP, since the ACKs may reach the fixed host even before the packet reaches the mobile host. In addition, each packet undergoes TCP processing four times as opposed to zero times in an end to end connection. Indirect TCP [41] discussed in the Split-Connection section is an example of a split connection scheme.

Balakrishnan et al. [70] analyse the performance of each of these techniques and conclude that TCP-aware link layer protocols provide a better throughput as compared to the other mechanisms and the current TCP implementations. The commonality among all the above schemes, is however, that an attempt has been made to make the smallest possible changes to the existing infrastructure.

Besides the above categories of solutions, there are two other solution types that attempt to improve TCP performance in wireless networks. These are:

1. Cross-Layer Schemes
2. ACK pacing Schemes

The literature on TCP performance improvement consists of many cross-layer schemes, which require TCP to obtain input from the link layer and network layers before it makes a decision on interpreting the packet losses and timeouts as congestion [72–75]. TCP may either request for input from these layers when it detects a packet loss, or the lower layers may send indications to TCP when they detect a problem in the link conditions. In either case, it has been seen that this cross-layer interaction involving the exchange of small and controlled amounts of information across layers can lead to large performance benefits without negating the advantages of layering. One such scheme is the Inter-layer Collaboration Protocol (ILC) discussed in Chinta et al. [76]. This protocol introduces an additional entity called the State Manager in the protocol stack which maintains information that TCP can use to determine if packet losses may be due to congestion or not.

Another technique to improve TCP performance is the concept of ACK pacing. Cho et al. [77] propose a mechanism to reduce the losses due to large blackout periods for a mobile on a wireless network. They use the ACK pacing mechanism to prevent bursty data delivery to the old path during a handoff and also triggers a faster route update. This algorithm paces the ACKs at constant intervals of 100ms.

2.4.2 Link Layer Schemes

As discussed in Section 2.4, the link layer schemes use techniques like forward error correction and link layer retransmissions to achieve improvement in TCP throughput. In this case the link layer detects packet loss and attempts to retransmit the lost packets while keeping TCP transparent to the loss. The most important solution in this category is the Snoop protocol [36]. The Snoop protocol uses TCP duplicate acknowledgements to detect packet losses and initiate a link-level local retransmission. The base station runs no transport layer. It caches every packet that has been sent to the mobile and keeps track of TCP acknowledgements to detect packet loss. If a duplicate ACK is received, it retransmits the lost packet. Hence, by not propagating the duplicate ACK, it prevents the sender from invoking congestion control.

The Snoop protocol introduces a snoop agent at the base station that is responsible for monitoring all packets on the TCP connection. It caches all packets that are unacknowledged by the receiver. The Snoop agent detects packet losses when it receives a small number of duplicate acknowledgements. On receiving these acknowledgements, Snoop retransmits the packet without forwarding the duplicate ACK to the TCP layer. Another mechanism used for packet loss detection is a local timeout. The Snoop agent only triggers a timeout after the first retransmission of a packet caused by the arrival of a duplicate acknowledgement. A retransmission may also be triggered after 200 ms of no activity from the sender or the receiver. To fulfill this functionality, the Snoop agent maintains two timers:

1. The Round-trip timer: This is based on the estimate of the smoothed round-trip time of the last link.
2. The Persist timer: This is the timer that triggers retransmission after 200 ms of inactivity.

These timers increase the number of transmission attempts during periods of high loss and hence, increase the chances of a packet reaching the destination sooner. The snoop agent also keeps track of the number of local retransmissions for each packet. It resets this counter once the packet is retransmitted from the fixed host.

Another link-layer scheme called delayed DupAcks [37] is based on Snoop, but is TCP-unaware. As in Snoop, the base station implements a link layer retransmission scheme. But in this case, the link layer does not need to understand and read TCP headers. Instead of using TCP duplicate ACKs to detect packet loss, this mechanism uses link layer ACKs. This scheme also serves to reduce the interference between TCP retransmissions and link level retransmissions. It does so by requiring the TCP receiver to delay the third and subsequent dupACK for some interval d. When the receiver receives out of order packets, it responds to the first two by sending out duplicate ACKs but it delays the third by an interval of d. If the in-order packet is received during this interval, the dupACK is not sent, otherwise the packet is assumed to be lost and the dupACK is sent.

The delayed dupACK scheme works better than Snoop in the presence of IPSec and other encryption mechanisms in which the TCP header is not readable at the link-layer. However, link layer protocols such as delayed dupACKs which are TCP-unaware may cause performance degradation instead of improvement because of the lack of synchronization between the link layer and TCP timers. Both Snoop and delayed dupACKs perform badly when the round-trip time on the wireless link is large. In this case, TCP on the fixed host cannot be prevented from timing out. Balakrishnan et al. [70] show that a TCP-aware link layer mechanism like Snoop achieves a 10-30% higher throughput than TCP-unaware schemes that do not attempt to deliver packets in order. Augmentation of the TCP-aware scheme with selective acknowledgements further improves the performance of this category of schemes.

2.4.3 End-To-End Schemes

In this section, we discuss the three main solutions in this category. The first is the Explicit Loss Notification scheme [78]. In this scheme, the Explicit Loss Notification field in the TCP ACK is set by the receiver to indicate to the sender if a packet loss was due to congestion or due to lossy wireless links. When a packet is lost on the wireless link, further cumulative acknowledgements corresponding to the lost packet are marked to indicate to the sender that the loss was not due to congestion. When the sender receives these acknowledgements it retransmits the lost packets but does not invoke its congestion control procedures. The main disadvantage of this scheme is the difficulty in identifying what packets are lost due to errors on lossy links.

The other solution in this category is the Selective Acknowledgements (SACKs) scheme. RFC 1072 [79] introduced SACKs as an option in TCP. SACKs are an attempt to provide the TCP sender with more information on the packets lost. The RFC proposes that each ACK must contain information of upto three non-contiguous blocks of data received successfully by the receiver. This provides the sender with enough information to determine what packets are lost when there may be multiple losses in a single window.

Another end-to-end solution is Freeze-TCP [40]. Freeze-TCP is a true end-to-end solution which belongs to the proactive solution category. In this scheme, the mobile detects an impending disconnection by measuring the strength of the signal. The mobile then advertises a window size of zero in any subsequent ACKs that it sends. This zero window size freezes the sender in its current state. The sender then continues to probe the receiver to determine if the window size has increased but no congestion control mechanisms are invoked. This is a true end-to-end scheme as it completely retains the TCP end-to-end semantics without requiring any intervention from the intermediate base station.

Ho [80] and Chen et al. [81] discuss another end-to-end scheme based on network coding retransmissions. Network coding in TCP serves to mask packet losses from the congestion control algorithm. The coding and decoding are performed at the end hosts only and hence, retains TCP end-to-end semantics. Chen et al. [81] argue that the existing network coding schemes are not practical to be implemented in real networks because of their ignorance of the decoding delays. Hence, the authors propose a new feedback based network coding scheme which substantially reduces the encoding and decoding delays and can handle not only random losses but also bursty losses.

Wang et al. [82] discuss an end-to-end approach to solve the TCP throughput problem during vertical handoff in wireless networks. This scheme updates the slow-start threshold and the congestion window size at the sender to improve TCP performance during vertical handoff in hybrid mobile networks. Khurshid et al. [83] discuss another scheme that requires a change in TCP at the sender's side. The authors use counters to track the occurrence of timeouts and 3-dupacks and then use the ratio between these in order to distinguish between a congestion and a non-congestion event.

The advantage of end-to-end solutions is that these retain the end-to-end semantics of TCP. These schemes do not require any added functionality at any intermediate base stations to improve TCP performance. Shah et al. [84] compare the performance of the end to end schemes. Overall, experimental results show that link-layer schemes perform better than these end-to-end schemes. Mechanisms like SACK only provide additional information in the presence of a loss but nothing to prevent the congestion control mechanisms from being invoked.

2.4.4 Split Connection Schemes

This category of schemes is characterized by the splitting of the TCP connection from the fixed host to the mobile into two parts - the wired connection from the fixed host to the base station and the wireless connection from the base station to the mobile. The most important solution in this category is Indirect-TCP (I-TCP) [41].

I-TCP uses the current base station as the center point of the connection. When there is a handoff, the center point of the connection moves to the new base station. At the start of a connection, the mobile sends a request to the base station to establish a connection with the fixed host on its behalf. The mobile communicates with the base station using any protocol that is tuned for the wireless environment. The fixed host always only sees an image of its peer mobile which in fact resides on the base station. When a handoff occurs, this image moves to a new base station.

The advantage of split connection schemes is that the fixed host is completely isolated from wireless losses. All retransmissions are managed by the base station. The disadvantage is the loss of end-to-end semantics. The fixed host may receive an ACK for the packet even before it reaches its final destination, that is, the mobile. In addition, the packet traverses the protocol stack till the TCP layer four times - twice at the base station. This may cause large packet delays due to the protocol processing that needs to be carried out multiple times. Balakrishnan et al. [70] show that these schemes do not achieve a major performance advantage due to the above disadvantages.

2.4.5 Cross Layer Mechanisms - Inter-Layer Collaboration Protocol

As discussed in Section 2.4, the Inter-Layer Collaboration Protocol attempts to improve TCP performance by using cross-layer techniques. In this scheme, a new entity called the State Manager is introduced in the protocol stack. The State Manager is responsible for providing information to TCP regarding the possible reasons for a packet loss.

The link-layer provides an indication to the State Manager when the wireless link is disconnected and when the connection is re-established. The network layer provides an indication to the State Manager when the mobile is not in its home network and is not yet registered with any of the foreign networks. When a packet loss occurs at TCP, it queries the State Manager to determine the next course of action. If during this period, either the link-layer has indicated a disconnection or the mobile is not registered in any of the networks, the State Manager informs TCP to not invoke the congestion control procedures. If the link layer and network layer indicate "healthy connections" to the State Manager, then TCP invokes its congestion control procedures in the event of a packet loss.

The advantage of this scheme is that TCP receives information from the other layers regarding the network conditions. This information allows TCP to make a more informed decision on whether to invoke the congestion control procedures or not. The main problem with this scheme is that it violates the layered paradigm by requiring decisions to be based on information from other lower layers of the protocol stack. However, experiments show that exchange of controlled amounts of information between layers improves performance and is indeed needed in the wireless environment.

2.4.6 Comparison of Schemes

Balakrishnan et al. [70] capture the comparison between the various schemes based on the throughput achieved by each scheme in the presence of an average error rate of one every 65536 bytes of data. They show that the TCP-aware link layer scheme (LL-TCP-Aware) achieves the best throughput as compared to the Link-Layer (LL), end-to-end (E2E) and Split-connection (SPLIT) schemes. There are however many disadvantages of the link-layer schemes that prevent them from being adopted by all base stations. As the wireless access technologies evolve, and more and more Internet and data access is carried out over 3G and 4G networks, it is imperative that the problem of throughput degradation due to TCP congestion control mechanisms be solved, since most of the data is accessed over TCP. A scheme is needed that entails minimal changes to the existing infrastructure but results in maximum benefit in terms of achieved throughput.

2.5 Use of Machine-Learning Based Prediction Techniques in Wireless Networks

In this section, we look at the applications of machine learning techniques to solve the self-management problem in wireless networks. As operators try and make the networks more and more self-managed, the need for building intelligent networks gains prominence. Training networks to monitor their surroundings and react accordingly is a crucial step in making them self-sufficient. Machine-learning is one such technique that allows systems to learn from historical situations and react accordingly. In this section, we look at some of the problems that have been solved using machine-learning techniques.

2.5.1 Wireless Link Status Prediction

One of the main problems in wireless networks is the unpredictable signal quality. The signal strength at any point in a wireless network is impacted by several factors - topology of the area, presence of buildings, interference from different networks and appliances operating at similar frequencies and so on. Since, the networks or the neighboring devices can keep changing, having a static algorithm to attack this problem will not work. Instead, having wireless network elements sense interference and respond to it appropriately will allow adaptation to changing environments. A key area of research here is in wireless link status prediction. Being able to predict when a link's strength will drop below threshold levels and for how long, will allow applications to take corrective action in advance thus ensuring minimal service disruption.

Liu and Cerpa [85] propose a scheme called 4C for link quality prediction and link estimation. The algorithm consists of three steps: 1. Data Collection: In this step, link quality data is collected and analyzed. The output of this step is a recommendation on the data to be collected for incorporating this scheme in running systems. 2. Offline Modeling: Using the collected data, machine-learning models are constructed to predict the link quality. The authors use three different models: Naive Bayes classifier, logistic regression and artificial neural networks. These models are constructed based on a combination of packet reception rate (PRR) for link estimation and Received Signal Strength Indicator (RSSI), Signal to Noise Ratio (SNR) and Link Quality Input (LQI). The output of each model is the success probability of delivering each packet. 3. Online Prediction: After the models are trained, they are deployed for online prediction. With each prediction, the error is fed back to the models to continue the process of learning. The authors compare the prediction accuracy of each of these models against a Bernoulli process whose success probability is set to the packet reception rate. Experimental results show that all three models have a greater prediction accuracy than the Bernoulli process with the Logistic regression model having the best accuracy at very low computational cost. The authors also compare the 4C algorithm against other similar estimator algorithms like [86] and Short Term Link Estimator (STLE) [87] and find a 20% to 30% difference in accuracy with 4C performing the best. In order to predict wireless network connectivity, that is, the signal to noise ratio for a mobile station, [88] proposes the use of a new Taylor Kriging model, which is basically the Kriging model with third order Taylor expansion for prediction. The prediction accuracy is compared against that of a predictor built using the Ordinary Kriging model [89] and an artificial neural network based predictor [90]. The prediction accuracy of the Taylor Kriging model is significantly higher than both these models, but the prediction error is still substantially high.

2.5.2 Prediction in Routing

As discussed in the earlier sections, packet routing is one of the main challenges in ad-hoc networks and wireless sensor networks. Since, these networks work based on different mobile stations serving as intermediate hops, changes in topology and thus routing paths are very frequent. Routing algorithms that adapt to these changing topologies are an important requirement in these networks. Having a routing algorithm that consumes minimal energy in retrieving and disseminating network information, while still adapting quickly to the changing network is the ideal requirement in these networks.This section looks at how prediction techniques can be used to overcome some of the routing problems in ad-hoc and sensor networks.

Liang et al. [91] proposes a Secure and Reliable Routing framework for Wireless Body Area Networks (WBANs). Wireless Body Area Networks consist of a network of sensors to monitor bodily functions. The data from these sensors is then aggregated and sent over the Internet to a central monitoring entity. Although these networks are small in size and extremely localized, bodily movements subject them to frequently changing topologies and correspondingly the routing algorithms must adjust to this quickly to allow reliable data transfer of critical data. The authors propose a framework in which each node measures the link quality of all its neighbors. Using past link quality measurements, the nodes predict the incidental quality of a link. When the routing algorithm has to select among a set of candidate nodes, it uses the one with the best link quality. Each node uses an auto-regression model to predict the link quality. Auto-regression models are used for modeling of time-series data and to predict the value at a particular instant of time. When a node sends a data packet, all nodes listen to the transmission even though it is not destined to them. These nodes respond back with an ACK to the sender node, the ACK packet containing the received signal strength. The sender node uses this to update the link-quality measurement that it maintains. A node which does not respond with an ACK is marked as unreachable by the sender node.Through simulations, the authors prove that the proposed technique improves the routing reliability because one of the best links is chosen for transmission at each point, and the probability of sending a packet to a dead or a non-existent node is minimal.

Cognitive radio is a technology devised to overcome the spectrum shortage problem, by allowing unlicensed users (also called cognitive users - CU) to use the unused parts of the spectrum originally allotted to primary users (PU). Towards this end, individual nodes sense their environment and adjust their transmission parameters to minimize interference with primary users. While this allows extremely efficient spectrum utilization, it makes routing a challenge in these networks, where the chances of interference with primary users are much higher and thus the links are unreliable and available for shorter durations. The application of prediction-based algorithms to the problem of topology control and reliable routing in Cognitive Radio networks is discussed in [92]. The authors argue that the routing algorithms must take the link availability into account when choosing next hops. This paper proposes a distributed prediction-based cognitive topology control scheme to provide this capability to the routing layer. In order to provide a minimal risk solution, this scheme is built into a separate layer between the cognitive radio layer and the routing layer to avoid making changes to well-established routing algorithms like AODV, DSR and distance-vector routing. This layer is referred to as the cognitive topology control layer and works to establish a reliable topology for routing protocols to operate on. This topology is constructed by using a new link reliability metric, determined based on: 1. Link availability time, predicted using the scheme proposed in [93] 2. Period of time spent in re-routing 3. Link data rate This reliability metric is used to determine the weight of a link and the weight of a path. The algorithm then proceeds to construct the topology, by identifying all neighbors, estimating the path weights from initial node to unvisited nodes and then constructing the complete topology using the paths with maximum weight. This ensures that re-routings are minimized because using the path weight equation, links with high data rate and low availability time and links with low data rate and high availability time are avoided as far as possible. This allows the routing algorithms to indirectly take into account the mobility of CUs as well as the interference from PUs in routing decisions. Through simulations, the authors prove that the resulting routes are more reliable and lead to lesser re-routings. The algorithm proposed in [94] performs routing based on link lifetime and coverage area. In this case, the energy drain rate is used to predict the link energy and the movement is calculated by relative motion estimation. Using this information, the packet can be routed on a path with a longer lifetime and lesser chances of packet loss.

2.5.3 Prediction in Intrusion Detection

The emergence of ad-hoc and wireless sensor networks has brought in several advantages like efficient utilization of the spectrum, reduction in capital expenditure (CAPEX) due to the absence of a fixed infrastructure, reduction in operating expenditure (OPEX) because of their self-configuring nature and allowing much better monitoring of military areas and making wireless body area networks and ad-hoc vehicular communication a reality. However, greater flexibility also makes these networks more vulnerable to security threats and attacks. Hence, different authentication mechanisms, attack detection and attack prevention mechanisms have been studied extensively. However, increase in the computing power allows large equations and passwords to be broken in a matter of seconds and hence, security algorithms need to keep evolving to keep finding and fixing newer and newer vulnerabilities. In all cases of network security, detection of an attack or a security threat is the biggest challenge. In this section we look at the use of machine learning techniques to detect intrusion and denial- of-service attacks in ad-hoc and wireless sensor networks. The algorithm presented in [95] uses Support Vector machines to detect a denial-of-service (DoS) attack in mobile ad-hoc networks. Support Vector Machine (SVM) is a machine-learning technique used for regression as well as classification. In this case, the authors use the SVM as a classifier, to classify packets as normal or attack. The packets classified as attack packets are dropped by the network. Two datasets - one of normal traffic and one of an abnormal attack are used to train the SVM classifier. Since, attacks are continuously evolving, their detection requires the algorithm to learn like the human brain does as detailed in [96]. The authors propose a three-layer hierarchical Brain-Like Learning algorithm for intrusion detection and prediction in Wireless Sensor Networks. The authors contend that intrusion detection schemes cannot contain the damage that intrusions cause, because by the time they detect the intrusion, it has mostly passed. So they propose a technique that predicts and detects intrusion. The scheme uses three layers and four agents to detect and predict intrusion: 1) The supervised learning layer resides in the individual sensors. A decision tree is used as a classifier to perform this supervised learning. The decision tree is contained within the detection agent. This agent uses a set of rules to drive the classification process of the tree. The results of the classification are further used to update the rules. If an attack that is unknown to the sensors occurs, it is sent to the sink node. 2) Unsupervised learning performed at the sink. A decision tree is used to perform clustering at the base station. If an attack unknown to the sink occurs, it is propagated to the base station. 3) Reinforcement learning performed at the base station: Reinforcement learning is used to predict intrusion in advance. The authors use a convergent temporal-difference learning scheme [97] in this layer to predict an intrusion. Whenever an attack is detected by the sensor or sink, it is reported to the base station which uses this to further build its prediction system. The prediction system consists of the input layer, the output layer, the hidden layer and the stochastic layer. Temporal-difference learning is used to update the weights. The prediction agent contains the logic to implement intrusion prediction based on reinforcement learning. The database agent logs all events and attacks and provides an interface for querying by the detection and prediction agents. The communication agent facilitates communication between the sensors, sinks and the base station. The authors evaluate the time overhead, memory consumption and communication overhead of this scheme in addition to its prediction accuracy. The algorithm is found to have the lowest time overhead as compared to the schemes proposed in [98] and [99]. The energy consumption in detecting an attack is obviously greater than the energy consumption in a normal operating sensor, but is only marginally greater for an unknown attack versus a known attack. The prediction rate is found to be 12 percentages higher than that of the SGA based scheme proposed in [100]. A scheme to detect malicious nodes based on energy prediction is proposed in [101]. Most schemes use node interactions or traffic profiles to detect an intrusion. However, this scheme uses the energy consumption of a node to detect an attack in cluster-based wireless sensor networks. The sensors consume energy in one of 4 states: 1) Transmitting 2) Receiving 3) Sensing 4) Calculating. The authors propose a means for predicting the energy consumed by a sensor by having the cluster head calculate the probability that each sensor node will move from one state to the other in a given set of time slots. Using this probability, the energy dissipation for those set of time slots is predicted. At the start of each time period, the cluster head predicts the energy dissipation of each sensor. At the end of each time period it determines the remaining energy levels at each sensor. If the actual energy consumed deviates largely from the predicted dissipation, it is classified as an attack and the node is blacklisted. A blacklisted node is removed from all routing tables and is thus isolated from the network. The authors also provide a means for determining the type of attack the malicious node is involved in by characterizing the energy dissipation deviation for five different types of attacks, namely: 1) Selective forwarding attack: In this case, the energy dissipation is lower than the predicted value. 2) Hello flood attack: Substantially higher energy dissipation than predicted. 3) Sybil attack: Difference in energy consumption is larger than a preset threshold. 4) Wormhole attack: Double the predicted energy is consumed. 5) Sinkhole attack: Difference in energy dissipation increases gradually. Simulation results show that the scheme is more efficient than existing ones in that it has high prediction accuracy and does not require any monitoring at individual sensor nodes. As a result, it can detect attacks with the least energy consumption which is ideal for limited-resource networks like wireless sensor networks.

The research gaps in the current literature have been discussed in Section 1.1.7. In the remainder of the thesis, we work to solve the identified research gaps, namely, the problem of throughput degradation due to TCP's congestion control mechanisms, proactive detection of wireless disconnections and the self-optimizing problem in P2PTV networks. The survey performed in this section has earlier appeared in [102].

Chapter 3

A Near-Optimal Scheme for TCP ACK Pacing to Maintain Throughput in Wireless Networks

With wireless devices becoming an integral part of everybody's life, there is a great emphasis on improving the offered quality of service and data rates, while increasing the range of applications that can be accessed using these devices. The third generation of wireless networks has provided a big boost to the popularity of wireless devices, with their emphasis on data and Internet access in addition to the support for voice calls. However, to enable a seamless movement from the wired to the wireless access domain, it is important to solve impending problems that reduce the achievable throughput and the efficiency of these networks. One such problem is the reduction in throughput observed due to TCP's congestion control mechanisms that come into operation during periods of mobility or fading of a wireless link.

While this problem needs to be addressed immediately, it is important to keep in mind that most of the current Internet infrastructure is TCP-based and hence, any change proposed must not entail large changes to the Internet infrastructure. Massive changes to the operation of the protocols may introduce the risk of destabilizing the entire infrastructure while also incurring large overheads in terms of effort and cost. The literature in this area has proposed several mechanisms, some requiring changes to only the mobile and some others requiring changes to the mobile as well as the base station.

This chapter addresses this requirement, by only requiring changes to the TCP protocol and to the link layer of the base station. The changes made to the TCP protocol are only invoked when the base station detects that the signal from the mobile is fading. We use a proactive scheme to protect the mobile from having its throughput reduced due to a short duration of bad signal quality. It makes use of a cross-layer scheme to provide the TCP layer with updated information about the network conditions in the event of bad quality links. The ACK holding scheme discussed in this chapter is a hybrid of a cross-layer mechanism coupled with a proactive scheme to prevent the wired host from taking extreme measures when the mobile experiences fading.

We use Monte-Carlo simulations to quantify the improvement brought about by introduction of this scheme. Network simulator-2 (ns2) [43] is an open-source software that allows for simulation of wireless networks and the different conditions in these networks, such as, fading, interference, mobility and so on. ns-2 has different TCP implementations such as TCP-NewReno [44] and TCP-Westwood [45]. The TCP ACK Pacing scheme is implemented as part of the TCP-Reno protocol and used to run experiments in which we simulate disconnections in wireless networks and quantify the advantages of the scheme.

3.1 Overview of ACK Holding Scheme

The ACK holding scheme is a hybrid of a cross-layer scheme and a proactive method of operation. This scheme requires changes to be made to the link-layer and the TCP layer at the base station. The mobile and the fixed host remain unchanged.

The link layer constantly monitors the strength of the signal received from the mobile, when TCP data is being exchanged between the fixed host and the mobile host. When the strength of the signal falls below a certain threshold, the link layer assumes the effects of fading or interference and concludes that the mobile may be temporarily unreachable within a certain time period. It also calculates the approximate duration for which the mobile may be unreachable. One method for doing this is proposed in [103]. Further discussion on the link layer mechanisms to achieve this forecasting is beyond the scope of this study. In addition to this, the link layer must also continuously calculate the Round Trip Time (RTT) to the mobile. Once, the link layer has detected the weakening of the signal, it sends the TCP layer an indication message that specifies the TCP connection(s) corresponding to the weak link, the approximate time for which the link may be down and the mean RTT to the mobile.

Upon receipt of this message, the TCP layer knows that the link layer will now send all ACKs corresponding to this TCP connection to it and it now needs to use the ACK holding algorithm to determine the time at which to send out the next ACK for this TCP connection. In addition, after receipt of this indication, TCP is also required to cache the packets intended for this mobile. TCP needs to cache these packets to avoid packet loss and subsequent timeouts. Once, the link is further weakened and the link layer detects that any more data on this link may be lost, it sends an indication to TCP to specify that the link is now down. This indicates to TCP that it will receive no more ACKs and the ACKs currently in its cache are the ones that need to be scheduled. Keeping this behavior in mind, the ACK holding algorithm should meet these constraints:

1. Pace ACKs so that they are being sent over the entire estimated duration of mobile being unreachable

2. Pace ACKs so as to avoid timeouts on the fixed host for any packet for which the base station has received an ACK from the mobile

3. Pace ACKs so that RTT and retransmission timeout (RTO) on the fixed host do not increase uncontrollably

4. Use a maximum of two duplicates of each ACK, if necessary to fulfill the above requirements

In order to meet the above requirements, the base station requires TCP on the fixed host to implement the timestamps option defined in RFC 1323 [104]. These timestamps are needed for the base station TCP to be able to calculate the RTT from the fixed host to the base station.

The ACK holding algorithm operates as follows:

Step 1 Based on

1. Current estimate of RTT to the fixed host (R)

2. Current estimate of RTT to the mobile (r)

Calculate the approximate retransmission timeout on the fixed host for the first packet whose ACK it is holding. Let us call this t.

Step 2 Based on

1. Number of ACKs present (n)
2. Approximate duration for which the mobile is going to be unreachable (d)
3. The estimated current retransmission timeout at the fixed host (t)

Calculate the time at which the ACKs are to be sent out, while optimizing the RTO at the fixed host.

TCP at the base station then sends out the ACKs at the time intervals determined. Meanwhile, if there are packets arriving for the mobile, TCP uses the timestamps in these packets in order to update its estimate of RTT, RTO and eventually the time at which the ACK is to be sent to the fixed host.

All incoming packets for the mobile need to be cached by the base station. The base station reserves space that is equal to the maximum advertised window size in the ACKs that it holds. When the mobile is accessible, and the link layer detects that the signal strength is above the threshold, it sends an indication to the TCP layer to stop holding ACKs and send out any remaining ACKs immediately. TCP will also transmit all cached packets to the mobile and move to the normal mode of operation. Figure 3.1 shows the complete operation of this ACK holding scheme.

3.2 Implementation of the ACK Holding Scheme

In this section, we discuss the implementation of the ACK holding scheme. Since, the base station is not an end host, it does not possess a TCP layer that actively participates in the user data packet exchange. As part of the ACK pacing scheme, we introduce the TCP layer at the base station, to receive link-layer indications and to sniff and cache TCP packets and their ACKs accordingly. For the cross layer interaction between TCP and the link-layer, the following new messages will be introduced:

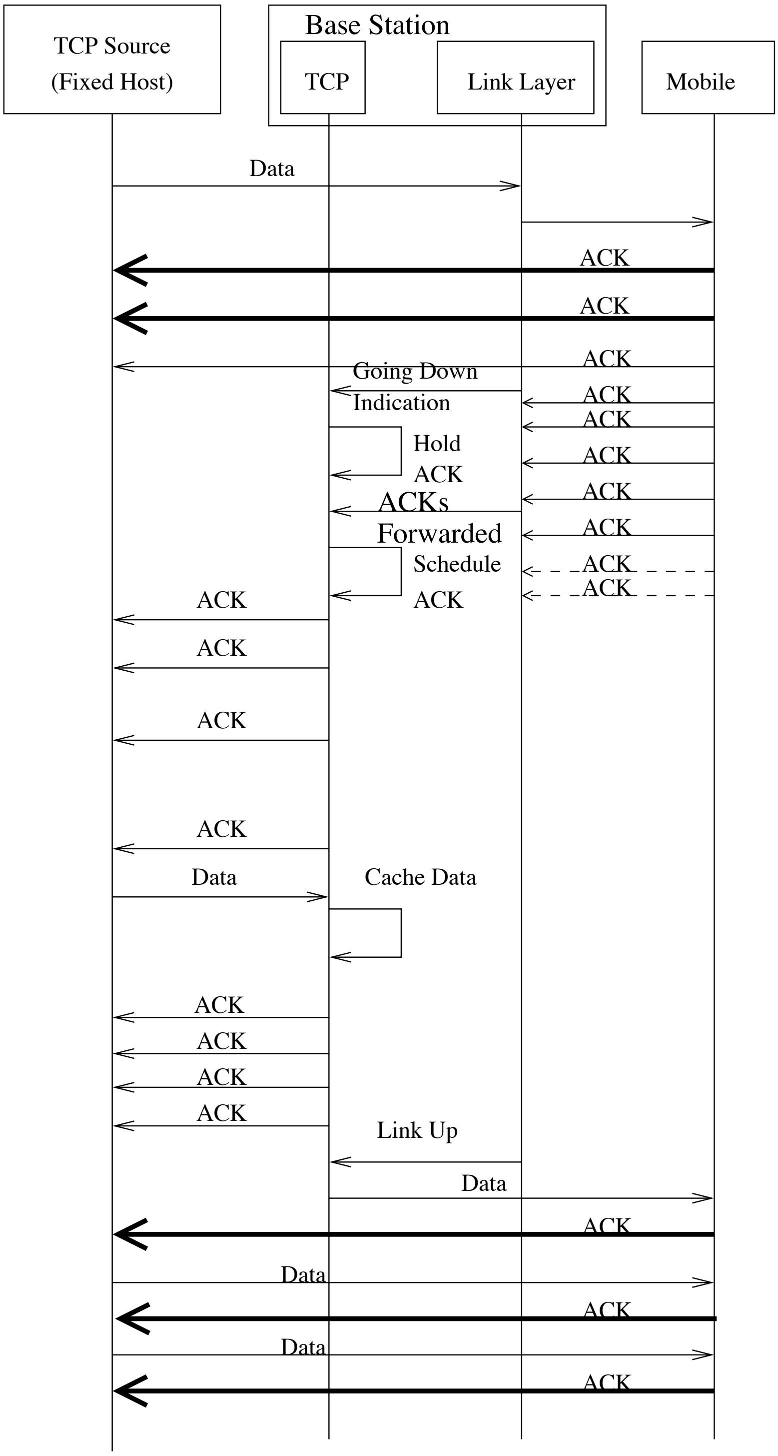

FIGURE 3.1: ACK Holding Algorithm operation

1. Link Going Down Indication: Indicates that TCP must start holding ACKs. The contents of this message are:
   (a) Current RTT to mobile
   (b) Approximate duration for which the link is expected to be down.
2. Link Gone Indication: Indicates to TCP that no more ACKs are expected. TCP can now start calculating scheduling times for the ACKs that are present.
3. Link Up Indication: Indicates to TCP that the link is up and TCP must now send out all the ACKs to the fixed host and all the cached packets to the mobile.

The TCP layer in the base station will be augmented with the ACK holder module which is responsible for the implementation of this scheme. The working of the ACK holder algorithm is as follows:

1. Receive message from the link layer
2. Based on the message type of the received message, either spawn the ACK holding thread, or send an indication to the existing ACK holding thread.
   (a) On a `LINK_GOING_DOWN_IND` message, spawn the ACK holding thread.
   (b) On a `LINK_GONE_IND` message, send a pacing indication to the ACK holding thread.
   (c) On a `LINK_UP_IND` message, send an ACK flush indication to the ACK holding thread.

The algorithm of the ACK holding thread is as follows:

1. Receive indications or TCP ACKs from the main algorithm thread.
2. Based on the type of the received message, perform the following actions:
   (a) Store TCP ACK messages or TCP packets.
   (b) If a pacing indication is received, determine the maximum window size (w) in the ACKs and reserve space for w bytes of data from the fixed host. Invoke the `ScheduleAlgorithm()` procedure to determine the time at which the ACKs are to be sent out.

(c) If a flush indication is received, send stored ACKs to the fixed hosts and cached packets to the mobile.

The `ScheduleAlgorithm()` procedure must determine the time at which to send out each ACK based on the constraints discussed in Section 3.1. The mechanism for doing this is explained in Section 3.3.

3.3 Scheduling of ACKs

This section presents the the algorithm that is used to schedule the ACKs to meet the constraints described in Section 3.2. In order to demonstrate how the algorithm satisfies all the constraints, we describe each step used to arrive at the final equation. As described in [34], the iterative mechanism for calculating retransmission time outs by TCP is governed by the following equations:

$$\delta_i = x_i - \mu_{i-1} \tag{3.1}$$

$$\mu_i = \mu_{i-1} + g \cdot \delta_i \tag{3.2}$$

$$\sigma_i = \sigma_{i-1} + h \cdot (|\delta_i| - \sigma_{i-1}) \tag{3.3}$$

$$\mathrm{RTO}_i = \mu_i + 4\sigma_i \tag{3.4}$$

where x_i denotes the delay round-trip time associated with packet i. μ and σ are estimates of the mean and standard deviations in the round-trip times observed on the channel, as was proposed by [105]. The work also suggested use of $g = \frac{1}{8}$ and $h = \frac{1}{4}$ for TCP implementations, as these are near optimal values for appropriate traffic shaping and also render the calculations easy as the operations involved are simple right shift bit operations.

In order to prevent the fixed host from invoking its congestion control procedures when the mobile goes out of the service area, the proposed scheme divides the accumulated ACKs into two subsets. It utilizes the first subset of ACKs for increasing the RTO calculations at the fixed host, so that each ACK can be used to make the source wait longer. After having covered a large portion of the total time with the first subset of ACKs, the second subset of ACKs are then sent back to the source at a constant and smaller rate. This ensures that the calculated value of RTO that was made to increase by use of ACKs in the first subset, is returned to smaller manageable numbers by the time the mobile comes back into service.

We denote the total number of ACKs stored by the base station as N. The inter-ACK arrival time, as observed by the TCP source, for the i^{th} ACK is denoted by x_i. T denotes the amount of time after which the mobile is estimated to come back up and n denotes the number of ACKs made to wait as long as possible right at the start of the scheme. Note that we use the next set of $(N-n)$ packets to bring the RTO (and consequently μ and σ) values within acceptable bounds for a normal connection to operate properly when the mobile comes back up at the estimated time. This is done by sending these $(N-n)$ ACKs with a small fixed amount of time between them, denoted by θ.

As the first n ACKs are made to wait as long as possible without causing a retransmission from the TCP source, we set $x_i = \text{RTO}_{i-1}; \forall i \leq n$. Hence $\text{RTO}_0 = \mu_0 + 4\sigma_0 = x_1$, $\mu_1 = \mu_0 + \frac{\sigma_0}{2}$ and $\sigma_1 = \frac{7}{4}\sigma_0$. Similarly, $\text{RTO}_1 = \mu_0 + \frac{15}{2}\sigma_0 = x_2$, $\mu_2 = \mu_0 + \frac{11}{8}\sigma_0$ and $\sigma_2 = \left(\frac{7}{4}\right)^2 \sigma_0$, thereby leading to $\text{RTO}_2 = \mu_0 + \frac{109}{8}\sigma_0$. Expanding similarly, we find that for $i \leq n$, $\sigma_i = \left(\frac{7}{4}\right)^i \sigma_0$ and $\mu_i = \mu_0 + \left(\frac{\sigma_{i-1}}{2} + \frac{\sigma_{i-2}}{2} + \frac{\sigma_{i-3}}{2} + \cdots\right)$. Therefore, the values of σ, μ and RTO after the first n ACKs have been sent out (denoted as σ_n, μ_n and RTO_n respectively) can be determined as in Equations (3.5), (3.6) and (3.7).

$$\sigma_n = \left(\frac{7}{4}\right)^n \sigma_0 \tag{3.5}$$

$$\mu_n = \mu_0 + \frac{2}{3}\sigma_0 \left[\left(\frac{7}{4}\right)^n - 1\right] \tag{3.6}$$

$$\text{RTO}_n = \mu_0 + \sigma_0 \left[\frac{14}{3}\left(\frac{7}{4}\right)^n - \frac{2}{3}\right] \tag{3.7}$$

We denote by $S(n)$ the amount of time the TCP source has been made to wait using the first n ACKs alone. Note that this would equal the sum of inter-ACK times for the first n packets, thereby making $S(n) = \sum_{i=1}^{n} x_i$. The value of $S(n)$ is calculated in Equation (3.8).

$$S(n) = \sum_{i=0}^{n-1} \text{RTO}_i = n\mu_0 + \sigma_0 \left\{\frac{56}{9}\left[\left(\frac{7}{4}\right)^n - 1\right] - \frac{2n}{3}\right\} \tag{3.8}$$

To determine the final values of σ, μ and RTO after all N ACKs have reached the TCP source, denoted by $\sigma_N(n)$, $\mu_N(n)$ and $\mathrm{RTO}_N(n)$ respectively, we set $x_i = \frac{(T-S(n))}{(N-n)} \equiv \theta; \forall i \in \{n+1, \ldots, N\}$. Note that all ACKs from $(n+1)^{\text{th}}$ ACK till the N^{th} ACK are spaced out equally, with all of their inter-ACK arrival times being set equal to θ. We find that $\delta_{n+1} = (\theta - \mu_n)$, $\mu_{n+1} = \frac{7}{8}\mu_n + \frac{1}{8}\theta$ and $\sigma_{n+1} = \frac{3}{4}\sigma_n + \frac{1}{4} \cdot |\theta - \mu_n|$. Similarly, $\delta_{n+2} = \frac{7}{8}(\theta - \mu_n)$, $\mu_{n+2} = \left(\frac{7}{8}\right)^2(\mu_n - \theta) + \theta$ and $\sigma_{n+2} = \left(\frac{3}{4}\right)^2\sigma_n + \frac{1}{4}\left(\frac{3}{4} + \frac{7}{8}\right) \cdot |\theta - \mu_n|$. Expanding similarly, we find that $\delta_{n+i} = \left(\frac{7}{8}\right)^{i-1}(\theta - \mu_n)$, $\mu_{n+i} = \left(\frac{7}{8}\right)^{i}(\mu_n - \theta) + \theta$ and $\sigma_{n+i} = \left(\frac{3}{4}\right)^{i} + 2|\theta - \mu_n| \cdot \left[\left(\frac{7}{8}\right)^{i} - \left(\frac{3}{4}\right)^{i}\right]$. Hence expressions for $\mu_N(n)$ and $\sigma_N(n)$ can be written as

$$\mu_N(n) = \left(\frac{7}{8}\right)^{(N-n)} \mu_n + \left[1 - \left(\frac{7}{8}\right)^{(N-n)}\right] \theta \tag{3.9}$$

$$\sigma_N(n) = \left(\frac{3}{4}\right)^{(N-n)} \sigma_n + 2\left[\left(\frac{7}{8}\right)^{(N-n)} - \left(\frac{3}{4}\right)^{(N-n)}\right] |\mu_n - \theta| \tag{3.10}$$

RTO_N can now be calculated as per Equation (3.4) as

$$\begin{aligned}\mathrm{RTO}_N(n) = \left(\frac{7}{8}\right)^{(N-n)} \mu_n + 4\left(\frac{3}{4}\right)^{(N-n)} \sigma_n \\ + \left[1 - \left(\frac{7}{8}\right)^{(N-n)}\right] \theta(n) \\ + 8\left[\left(\frac{7}{8}\right)^{(N-n)} - \left(\frac{3}{4}\right)^{(N-n)}\right] |\mu_n - \theta(n)|\end{aligned} \tag{3.11}$$

However, following the traditional method of setting the partial derivative of $\mathrm{RTO}_N(n)$ with respect to n to zero for finding the minima in Equation (3.11) does not yield a closed form solution. For given values of T, N, μ_0 and σ_0, the typical variation in RTO_N as n is varied is shown in Figure 3.2. Note that while the function derivative of RTO_N appears to have multiple solutions, it has only one minima. Hence, the value of n that minimizes RTO_N can be calculated using a single step forward algorithm shown below.

```
Initialize:
  Set RTO_prev := RTO_curr := RTO_N(0);
  Set RTO_next := RTO_N(1);
```

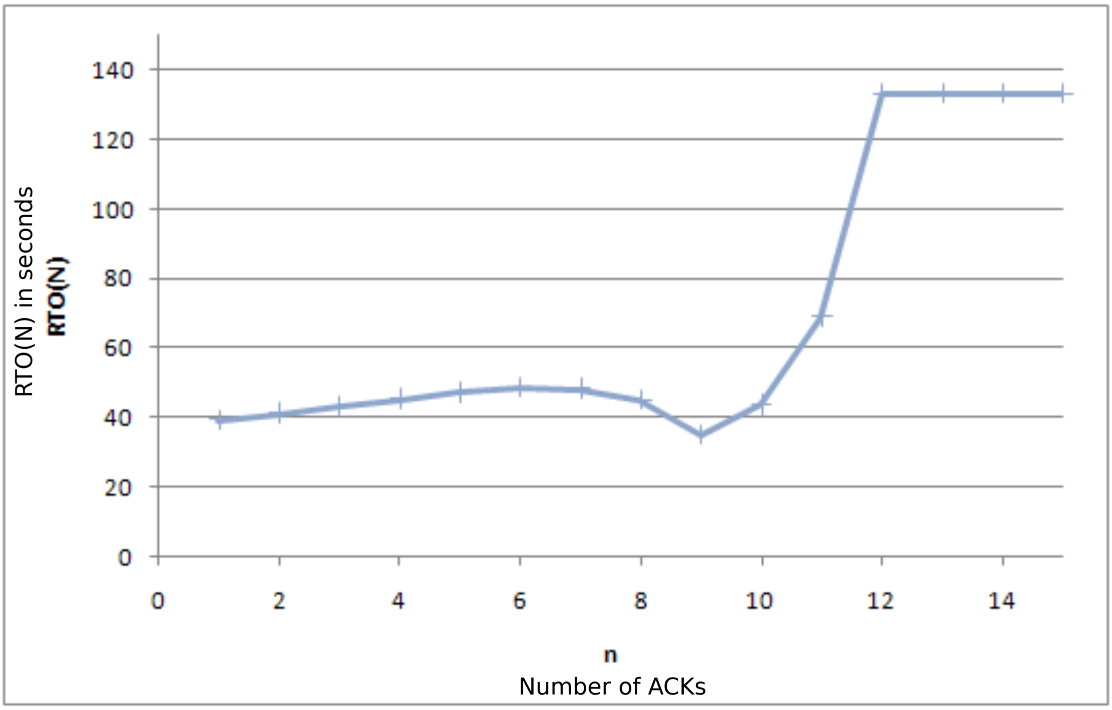

FIGURE 3.2: RTO_N as a function of n for $T = 1000s, N = 30, \mu_0 = 1, \sigma_0 = 0.3$.

```
  Iterate Over N:
    If ((RTO_curr <= RTO_prev) and
        (RTO_curr <= RTO_next))
      Stop;
    Else
      Increment n;
      Set RTO_prev := RTO_curr;
      Set RTO_curr := RTO_next;
      Set RTO_next := RTO_N(n+1);
    EndIf
  EndIterate
End
```

3.4 Simulation Results

The ACK holding scheme was implemented in ns-2 and its throughput compared to that of TCP without the ACK holding scheme, in the presence of lossy links. Figure 3.3

Disconnection Intervals	ACK Holding Scheme	TCP Reno	Throughput Improvement(%)
Single intervals of 10s	7688	6361	120.9%
Two intervals of 10s each	3981	503	791.5%

TABLE 3.1: Comparison of Throughput

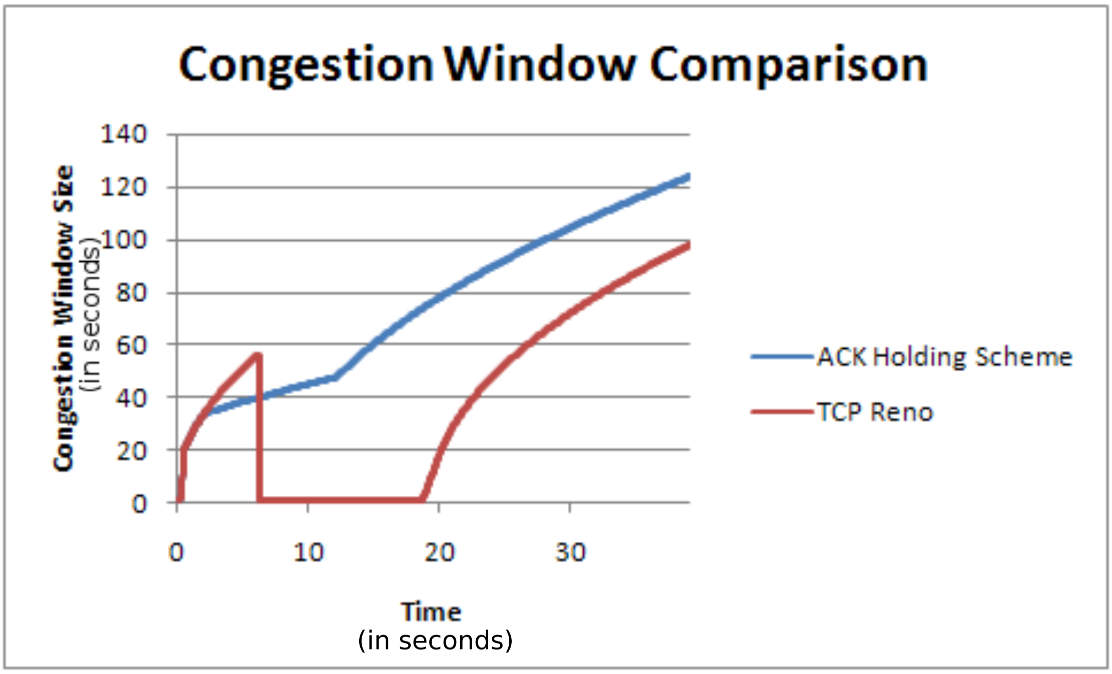

FIGURE 3.3: Congestion window comparison for single interval of fading

compares the congestion window size of TCP-Reno with that of TCP augmented with the ACK holding scheme, when the wireless link was down for 10 seconds during a 40 second data transfer interval. This figure clearly shows the positive effect of TCP's ACK holding scheme as it prevents any timeouts and hence, the reduction of the congestion window to 1. Figure 3.4 compares the congestion window size of TCP-Reno and the ACK holding scheme when there are two 10 second disconnections during a 40 second data transfer. This figure proves that the reason the ACK holding scheme performs better than the current TCP implementations is that the TCP at the sender requires no time to recover after the disconnection period.

Table 3.1 shows the throughput comparison of TCP-Reno and the ACK holding scheme for the above two conditions. It can be seen, that the ACK holding scheme, achieves throughput over 7.5 times that of the current TCP. This is due to the fact that after slow start has been invoked TCP takes some amount of time to recover and bring its data rate to what it was before the link went down. However, in the case of the ACK

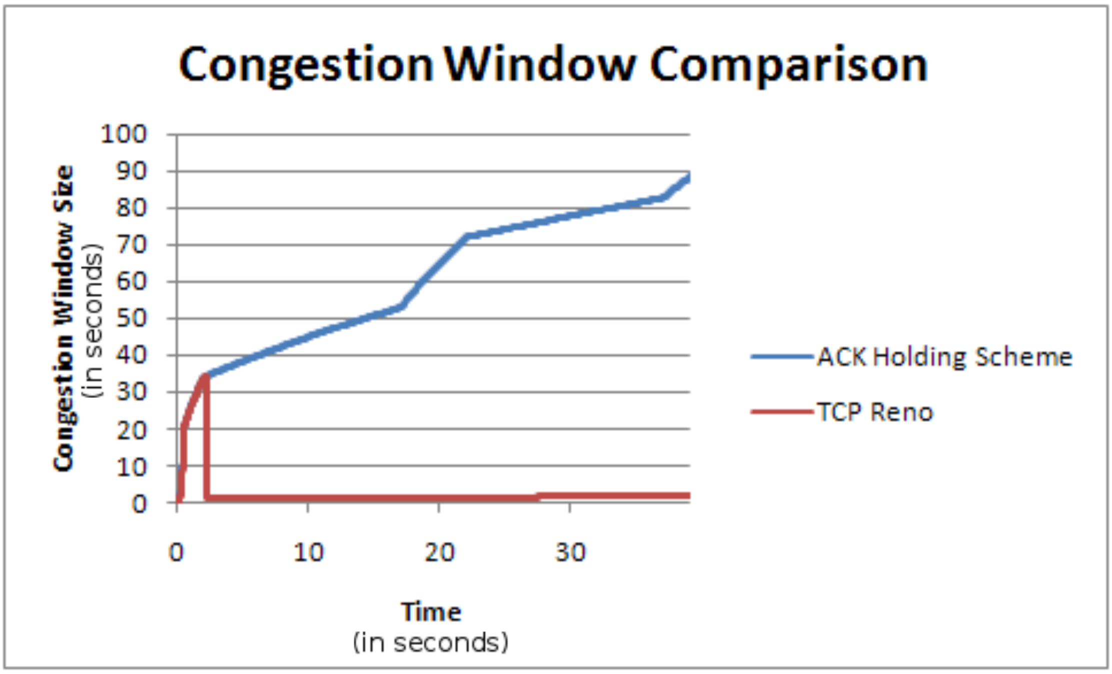

FIGURE 3.4: Congestion window comparison when fading takes place for two successive time intervals

holding scheme, there is no such recovery needed. After the wireless link is restored, the only component that needs to recover in the presence of the ACK holding scheme is the retransmission timeout estimate. Since, the ACKs are delayed by specific intervals, the RTT estimate at the sender is disturbed, making the RTO larger.

Figure 3.5 shows the change in the inter-ACK time using the equations in Section 3.3, when the base station has 20 ACKs to pace over a period of 500 seconds. The inter-ACK time initially increases for the first 5 ACKs. The inter-ACK time then decreases until it eventually stabilizes to a small constant value that will be instrumental in reducing the final RTO after the ACK holding algorithm terminates.

Figure 3.6 shows the variation in RTO value for the above case. In keeping with the increase in inter-ACK time, the RTO value calculated at the source first increases and then stabilizes at a higher value before decreasing and stabilizing at a lower value. Figure 3.7 shows the inter-ACK time variation using the ScheduleAlgorithm of Section 3.2 to a set of 20 ACKs paced over a period of 250 seconds. Figure 3.8 shows the RTO variation for the same case.

Figures 3.6 and 3.8 show that final values of RTO do not vary considerably even for large changes in the values of T. Hence, the scheme does not require that the predictor

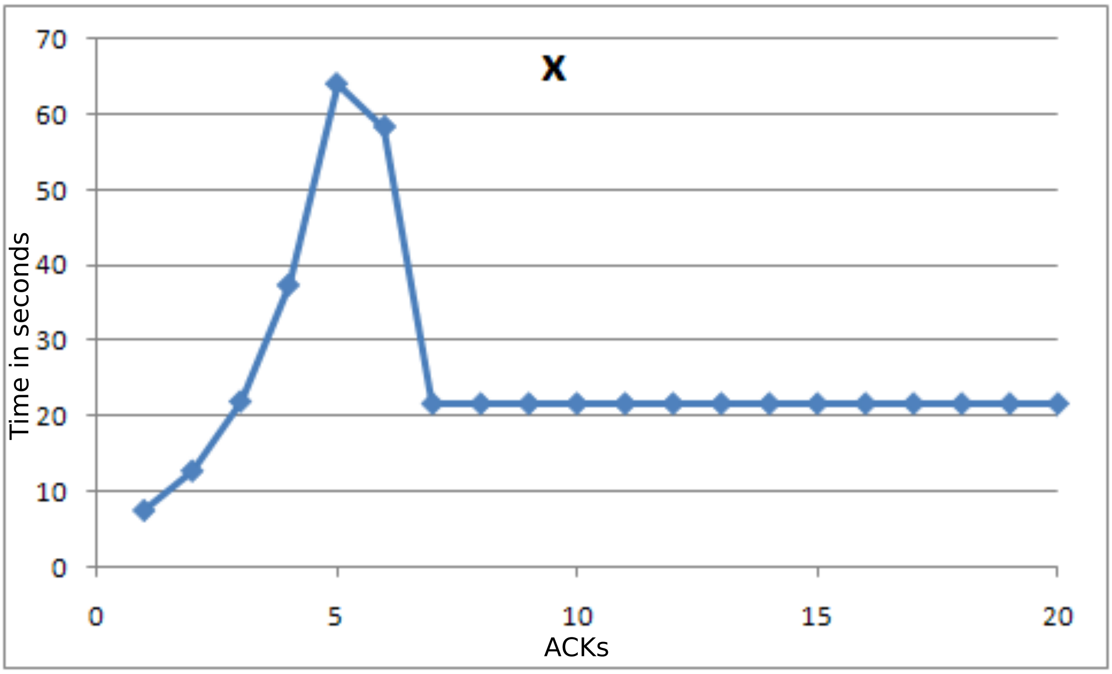

FIGURE 3.5: Optimal x for a wait of 500 seconds

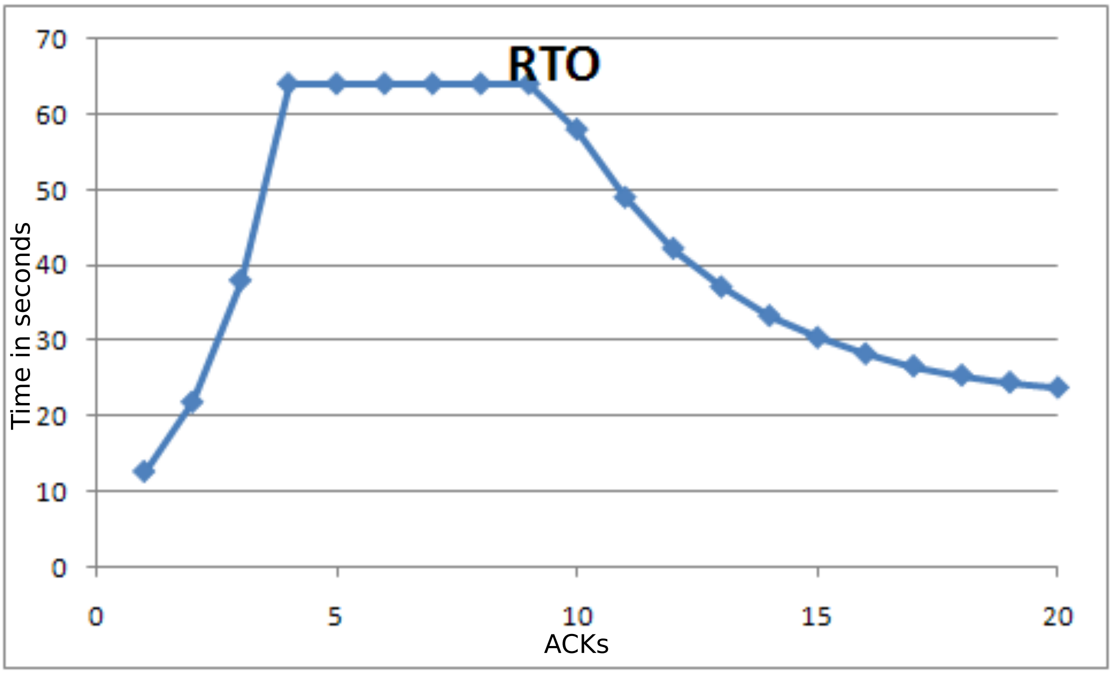

FIGURE 3.6: Variation of RTO with optimal x for a wait of 500 seconds

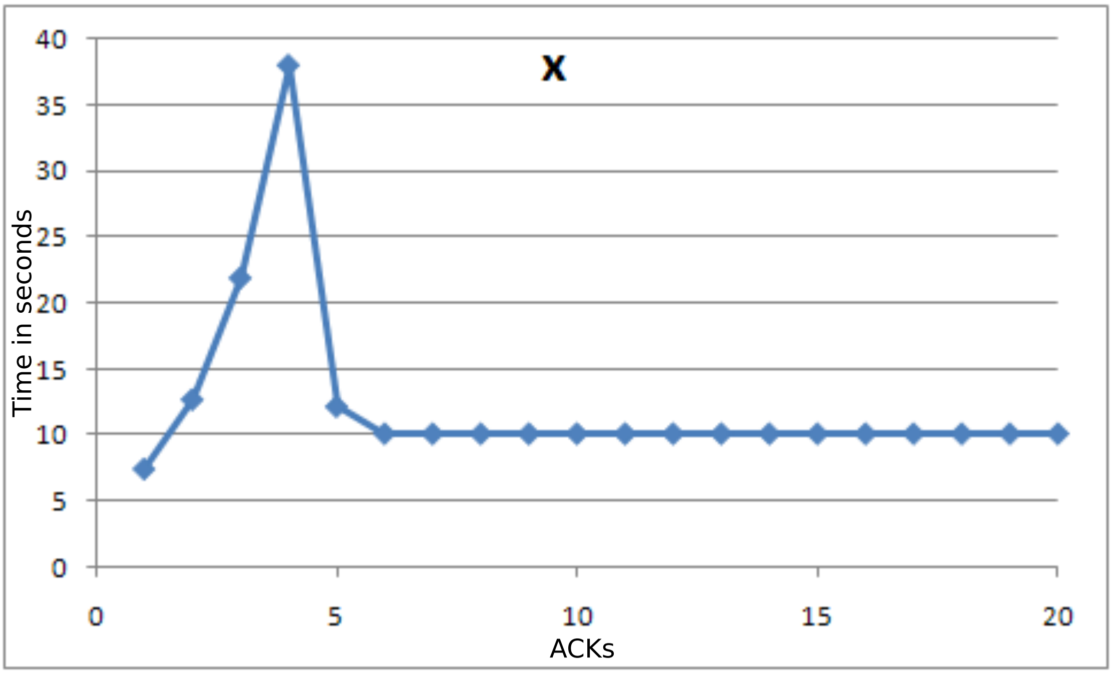

FIGURE 3.7: Optimal x for a wait of 250 seconds

make a very accurate prediction of the time for which the mobile will stay disconnected. A predictor that provides an over-estimate of the duration would work well for the TCP ACK Pacing scheme.

3.5 Conclusions

This chapter proposed a proactive scheme that used cross-layer techniques at the base station to inform TCP of a temporary loss of connectivity to a mobile. The base station TCP then used the estimate of the link down period to schedule the ACKs received from the mobile in such a way that the fixed host was kept transparent to the loss of connectivity. In Section 3.4, we saw that this improves the throughput by almost 7.5 times as compared to the current TCP implementation. This proves that the ACK holding schemes provides large improvements in scenarios where there are multiple disconnections in a single data transfer session.

In Section 3.3, we have assumed that accurate predictions are available of duration for which the mobile would remain disconnected. It may therefore seem that the scheme is dependent on availability of such an accurate predictor. However, note from Figures 3.6 and 3.8 that the final values of RTO do not vary considerably even for large changes in

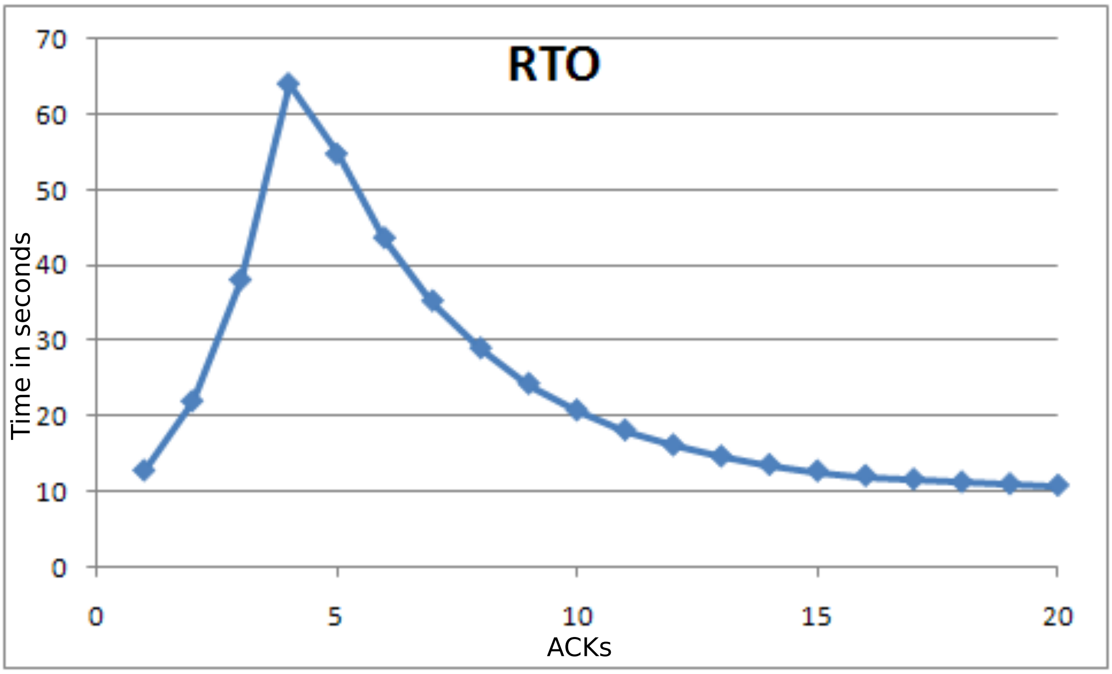

FIGURE 3.8: Variation of RTO with optimal x for a wait of 250 seconds

values of T. Hence, in the absence of an accurate prediction of T, we suggest that this scheme should be implemented with over-estimates of T rather than under-estimates.

The results were obtained by modifying ns-2 to hold and schedule ACKs in order to simulate the ACK holding scheme and drop ACKs and packets to simulate the working of normal TCP in the event of a disconnection. Using ns-2 we simulated a wireless network and the different disconnection conditions in it. Through each disconnection, ns-2 was made to use the ACK Pacing scheme to schedule the ACKs along with an over-estimated disconnection prediction duration.

In later chapters we look at schemes for 1) Predicting disconnection in wireless networks 2) Predicting the duration of disconnection in wireless networks. TCP ACK Pacing needs to be combined with these schemes in order to achieve maintenance of throughput in light of disconnections in wireless networks.

Chapter 4

Mobility Models for Disconnection Prediction in Cellular Networks

Mobility management and disconnection prediction schemes are important as they allow us to take various design decisions regarding the setup of a wireless network. As India gears towards adoption of 3G mobile networks, we are going to see one of the largest deployments of an advanced mobile technology. However, there are many issues that need attention for such a large deployment. Base stations support limited number of physical connections. When a mobile moves from one base station to another, if the destination base station is already servicing the maximum possible number of mobiles, the call would be dropped. However, prediction of the mobility and the approximate time of being under the coverage area of the target base station will allow resource allocation in advance, thus preventing the call drop.

In single-hop wireless networks like GSM and UMTS, prediction of disconnection can allow the base station to cache TCP data and acknowledgements for packet connections to the mobile. This helps prevent loss of throughput when the connection to the mobile is restored. Goff et al. [40] and Bhutani [106] discuss the use of disconnection prediction schemes to prevent TCP's congestion control mechanisms from kicking in and causing a drastic reduction in throughput. Goff et al. [40] allow the mobile to indicate its incapability to receive data to the content server just prior to the disconnection so that no data is lost during the period of the disconnection. In Chapter 3 and [106], we discussed a scheme that allowed the base station to space out the TCP acknowledgement that are sent to the server based on the prediction information, thus, keeping the server entirely transparent to the disconnection.

In this chapter, we attempt to solve the problem of predicting the disconnection duration of a mobile in single-hop wireless networks, such as cellular networks, under two mobility models: the Gauss-Markov mobility model [107] and the city-section mobility model [108]. The Gauss-Markov model has the ability to more accurately model various mobility patterns, like stop-start nature of mobility, and is more suited for adhoc networks in military applications. The Gauss-Markov model is able to model for the temporal dependency between past and present values of the mobile's velocity and direction. On the other hand, while the city-section model is much simpler, and is well suited to model traffic conditions in a metropolitan city, it accounts for the geographic correlation of position of various mobiles due to restrictions placed by artifacts, like roads in a city. For both mobility models, we mathematically derive the disconnection duration expression. For the Gauss-Markov mobility model, we realize that such a computation leads us to a mathematical expression that may be difficult to calculate in field applications. Hence, we turn around the original Gauss-Markov model in our search for a more tractable expression for disconnection duration period. We thus formulate the modified Gauss-Markov mobility model, wherein, unlike the original model, location varies in discrete steps whereas time varies as a continuous variable.

4.1 Predictive Mobility Management

Liang and Haas [107] present a mobile tracking scheme that exploits predictability of user mobility patterns in a wireless network. This paper discusses the use of the Gauss Markov model where the mobile's velocity is correlated in time to a various degree. The mobile's current location is predicted based on its previous location and velocity. When a call is made, the mobile is paged around the predicted location. The network shares the location prediction with the mobile, which in turn sends a location update when it reaches a threshold distance away from this predicted location. This paper discusses the advantages of using a Gauss-Markov model instead of the random-walk mobility model. In the real world users generally move with a particular destination in mind and hence, their location in the future is a function of their current location and velocity. This information is better represented by using a Gauss Markov model as compared to the random-walk model which is memory-less. Simulations performed show that this scheme provides a performance improvement ranging from unity to a factor of 10 in comparison to the regular non-predictive distance based schemes.

4.2 Disconnection Duration Prediction for a Mobile

In order to achieve the various advantages of seamless communication in the presence of mobility, it is thus, important to be able to predict not only the location of the mobile, but also an impending disconnection and its duration. In this section, we look at a disconnection prediction scheme that uses the Gauss-Markov mobility model discussed in [107]. We arrive at an expression that can be used by the base station to predict the disconnection duration of a mobile.

For developing the disconnection prediction model in this chapter, we make assumptions similar to those made by [107]. First, we assume that time is divided into discrete intervals, denoted by subscripts $1, 2, 3 \ldots n \ldots$ on various variables. Mobile user-equipments have the capability to measure their location (S) and velocity (v) in respect to the closeby base-stations. Further, we assume that the velocity v of the mobile user-equipment follows a stationary Gauss-Markov process. Liang and Haas [107] note that this is a reasonable assumption, and holds true in various practical scenarios. Accordingly, the velocity of the mobile at time n can be written as

$$v_n = \alpha \cdot v_{n-1} + (1-\alpha) \cdot \mu + \sqrt{1-\alpha^2} \cdot x_{n-1} \tag{4.1}$$

where μ is the asymptotic mean of v_n as $n \to \infty$. Let σ^2 denote the variance of v_n as $n \to \infty$; x's denote independent, uncorrelated and stationary Gaussian processes with mean $\mu_x = 0$ and $\sigma_x = \sigma$. For our current analysis, we restrict ourselves to the single-dimensional case. The more realistic two-dimensional case can be derived in a similar fashion. We also restrict ourselves to a single base-station case for the purpose of analysis in this report. We let the origin denote the location of the base-station. The mobile is assumed to have a starting location of S_0 and initial velocity of v_0. The range of the base-station is assumed to be between $-d$ and $+d$. The mobile is assumed to be disconnected outside this range.

The following can be inferred from Equation (4.1)

$$v_1 = \alpha \cdot v_0 + (1-\alpha) \cdot \mu + \sqrt{1-\alpha^2} \cdot x_0 \tag{4.2}$$

$$v_2 = \alpha \cdot v_1 + (1-\alpha) \cdot \mu + \sqrt{1-\alpha^2} \cdot x_1 \tag{4.3}$$

$$= \alpha \left[\alpha \cdot v_0 + (1-\alpha) \cdot \mu + \sqrt{1-\alpha^2} \cdot x_0\right] + (1-\alpha) \cdot \mu + \sqrt{1-\alpha^2} \cdot x_1 \tag{4.4}$$

$$= \alpha^2 v_0 + (1-\alpha^2) \cdot \mu + \sqrt{1-\alpha^2} \cdot (x_1 + \alpha \cdot x_0) \tag{4.5}$$

$$v_n = \alpha^n v_0 + (1-\alpha^n) \cdot \mu + \sqrt{1-\alpha^2} \cdot \sum_{i=0}^{n-1} \alpha^{n-i-1} x_i \tag{4.6}$$

The location of the mobile after n time periods can be defined as

$$S_n = S_0 + \sum_{i=0}^{n-1} v_i \tag{4.7}$$

$$= S_0 + v_0 + \sum_{i=1}^{n-1} \left[\alpha^i v_0 + (1-\alpha^i)\mu + \sqrt{1-\alpha^2} \sum_{j=0}^{i-1} \alpha^{i-j-1} x_j\right] \tag{4.8}$$

$$= S_0 + v_0 \left[1 + \sum_{i=1}^{n-1} \alpha^i\right] + \mu \left[\sum_{i=1}^{n-1} (1-\alpha^i)\right] + \sqrt{1-\alpha^2} \left[\sum_{i=1}^{n-1} \sum_{j=0}^{i-1} \alpha^{i-j-1} x_j\right] \tag{4.9}$$

Let us define

$$\mathcal{A} \equiv 1 + \sum_{i=1}^{n-1} \alpha^i \tag{4.10}$$

$$= \frac{(1-\alpha^n)}{(1-\alpha)} \tag{4.11}$$

$$\mathcal{B} \equiv \sum_{i=1}^{n-1} (1-\alpha^i) \tag{4.12}$$

$$= (n-1) - \sum_{i=1}^{n-1} \alpha^i \tag{4.13}$$

$$= (n-1) - \frac{\alpha \cdot (1-\alpha^{n-1})}{(1-\alpha)} \tag{4.14}$$

$$\mathcal{C} = \sum_{i=1}^{n-1} \sum_{j=0}^{i-1} \alpha^{i-j-1} x_j \tag{4.15}$$

So that

$$S_n = S_0 + v_0 \cdot \mathcal{A} + \mu \cdot \mathcal{B} + \sqrt{1-\alpha^2} \cdot \mathcal{C} \tag{4.16}$$

The terms of summation in calculation of $\mathcal{C}$ can be swapped. The nested summation over j in Equation (4.15) can be swapped out so that

$$\mathcal{C} = \sum_{j=0}^{n-2} \frac{x_j}{\alpha^j} \cdot \sum_{i=j+1}^{n-1} \alpha^{i-1} \tag{4.17}$$

$$= \sum_{j=0}^{n-2} \frac{x_j}{\alpha^j} \cdot \sum_{i=j}^{n-2} \alpha^{i} \tag{4.18}$$

$$= \sum_{j=0}^{n-2} \frac{x_j}{\alpha^j} \cdot \alpha^j \cdot \sum_{i=0}^{n-2-j} \alpha^{i} \tag{4.19}$$

$$= \sum_{j=0}^{n-2} x_j \cdot \frac{(1-\alpha^{n-1-j})}{(1-\alpha)} \tag{4.20}$$

$$= \frac{1}{(1-\alpha)} \sum_{j=0}^{n-2} x_j \cdot (1-\alpha^{n-1-j}) \tag{4.21}$$

Hence, location of the mobile after n time periods, S_n can be written as

$$S_n = S_0 + \frac{(1-\alpha^n)}{(1-\alpha)} v_0 + \left[(n-1) - \frac{\alpha(1-\alpha^{n-1})}{(1-\alpha)} \right] \mu + \frac{\sqrt{1-\alpha^2}}{(1-\alpha)} \sum_{j=0}^{n-2} x_j (1-\alpha^{n-1-j}) \tag{4.22}$$

We now define $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$ as

$$\mathcal{C}_1 = \sum_{j=0}^{n-2} x_j (1-\alpha^{n-1-j}) \tag{4.23}$$

$$= \sum_{j=0}^{n-2} x_j - \sum_{j=0}^{n-2} \alpha^{n-1-j} x_j \tag{4.24}$$

$$= \mathcal{C}_2 + \mathcal{C}_3 \tag{4.25}$$

Since x_j's are independent, uncorrelated and stationary gaussian processes with $\mu_x = 0$ and $\sigma_x = \sigma$, $\mathcal{C}_2$ can be shown to be a stationary gaussian process with mean $\mu_{\mathcal{C}_2} = 0$ and variance $\sigma_{\mathcal{C}_2} = (n-2)\sigma^2$. Mean and variance of $\mathcal{C}_3$ can be expressed as

$$\mu_{\mathcal{C}_3} = \sum_{j=0}^{n-2} \alpha^{n-1-j} \mu_{x_j} = 0 \tag{4.26}$$

$$\sigma^2_{\mathcal{C}_3} = \sum_{j=0}^{n-2} \left(\alpha^{n-1-j}\right)^2 \cdot \sigma^2_{x_j} \tag{4.27}$$

$$= \sigma^2 \sum_{j=0}^{n-2} \left(\alpha^{n-1-j}\right)^2 \tag{4.28}$$

$$= \alpha^2\sigma^2 \left[\frac{1-\left(\alpha^2\right)^{n-1}}{1-\alpha^2}\right] \tag{4.29}$$

Mean and variance of $\mathcal{C}_1$ can be calculated as

$$\mu_{\mathcal{C}_1} = \mu_{\mathcal{C}_2} + \mu_{\mathcal{C}_3} \tag{4.30}$$

$$\sigma^2_{\mathcal{C}_1} = \sigma^2_{\mathcal{C}_2} + \sigma^2_{\mathcal{C}_3} \tag{4.31}$$

$$= (n-2)\sigma^2 + \frac{\sigma^2\alpha^2}{(1-\alpha^2)}\left[1-\alpha^{2(n-1)}\right] \tag{4.32}$$

We denote the mean and variance of S_n by η and τ^2 respectively. Then

$$\eta = S_0 + \frac{(1-\alpha^n)v_0}{(1-\alpha)} + \left[(n-1) - \frac{\alpha\left(1-\alpha^{n-1}\right)}{(1-\alpha)}\right]\mu \tag{4.33}$$

and

$$\tau^2 = \frac{(1-\alpha^2)}{(1-\alpha)^2}\left[(n-2)\sigma^2 + \frac{\sigma^2\alpha^2}{(1-\alpha^2)}\left(1-\alpha^{2(n-1)}\right)\right] \tag{4.34}$$

$$= \frac{1}{(1-\alpha)^2}\left[(n-2)\sigma^2(1-\alpha^2) + \sigma^2\alpha^2\left(1-\alpha^{2(n-1)}\right)\right] \tag{4.35}$$

$$= \frac{\sigma^2}{(1-\alpha)^2}\left[(n-2)(1-\alpha^2) + \alpha^2 - \alpha^{2n}\right] \tag{4.36}$$

Note that the mobile is connected at time n if $-d \leq S_n < +d$. We denote this event of being connected by $\mathcal{S}(n)$, and that of being disconnected by $\mathcal{F}(n)$. Then

$$\text{Prob.}\left(\mathcal{S}(n)\right) = \text{Prob.}\left(-d \leq S_n < +d\right) \tag{4.37}$$

$$= \Phi\left(\frac{(d-\eta)}{\tau}\right) - \Phi\left(\frac{(-d-\eta)}{\tau}\right) \tag{4.38}$$

where Φ denotes the cumulative distribution function of the standard normal distribution.

$$\text{Prob.}\left(\mathcal{F}(n)\right) = 1 - \text{Prob.}\left(\mathcal{S}(n)\right) \tag{4.39}$$

Let us denote the random variable for disconnection duration using the symbol Δ. Let $\mathcal{G}(\zeta, \delta)$ denote the probability that the mobile stays connected till time ζ, and then stays disconnected for exactly δ period of time. In terms of Bernoulli trials, this equates to having $(\zeta - 1)$ successes $(\mathcal{S})$, followed by δ failures $(\mathcal{F})$, followed by a success $(\mathcal{S})$. Then

$$\text{Prob.}(\Delta = \delta) = \sum_{\forall \zeta} \mathcal{G}(\zeta, \delta) \tag{4.40}$$

and

$$\begin{aligned}\mathcal{G}(\zeta, \delta) = \prod_{i=1}^{\zeta-1} &\left\{ \Phi\left(\frac{d - \eta(i)}{\tau(i)}\right) - \Phi\left(\frac{-d - \eta(i)}{\tau(i)}\right) \right\} \\ &\times \prod_{i=\zeta}^{\zeta+\delta-1} \left\{ 1 - \left[\Phi\left(\frac{d - \eta(i)}{\tau(i)}\right) - \Phi\left(\frac{-d - \eta(i)}{\tau(i)}\right) \right] \right\} \\ &\times \left\{ \Phi\left(\frac{d - \eta(\zeta+\delta)}{\tau(\zeta+\delta)}\right) - \Phi\left(\frac{-d - \eta(\zeta+\delta)}{\tau(\zeta+\delta)}\right) \right\}\end{aligned} \tag{4.41}$$

Correspondingly, the probability of disconnection duration for the mobile taking a value of δ can be stated as

$$\begin{aligned}\text{Prob.}(\Delta = \delta) = \sum_{\forall \zeta} \prod_{i=1}^{\zeta-1} &\left\{ \Phi\left(\frac{d - \eta(i)}{\tau(i)}\right) - \Phi\left(\frac{-d - \eta(i)}{\tau(i)}\right) \right\} \\ &\times \prod_{i=\zeta}^{\zeta+\delta-1} \left\{ 1 - \left[\Phi\left(\frac{d - \eta(i)}{\tau(i)}\right) - \Phi\left(\frac{-d - \eta(i)}{\tau(i)}\right) \right] \right\} \\ &\times \left\{ \Phi\left(\frac{d - \eta(\zeta+\delta)}{\tau(\zeta+\delta)}\right) - \Phi\left(\frac{-d - \eta(\zeta+\delta)}{\tau(\zeta+\delta)}\right) \right\}\end{aligned} \tag{4.42}$$

Note that while Equation (4.42) gives us a method to find the probability of disconnection duration taking on a particular value, it is difficult to calculate in practice as the summations being carried out are on possible values of ζ, the starting point of the disconnection duration. To overcome this difficulty, we approach the problem through a different tack in the next section.

4.3 Changing the interpretation of the Gauss-Markov Process

The Gauss-Markov mobility model, and even the one proposed by [107], assumes that time increments in terms of discrete periods, and the decisions about the velocity and the direction of motion are made at the transition of these time periods. While such a model is useful for other purposes, its usage complicates analysis when trying to model the path of a mobile as it moves from within-range to out-of-range and then comes back within-range. In particular, the problem posed is that the mobile can make different sized distance steps in different time-periods, thereby making the computation of the time when it is going to land back within-range convoluted. Therefore, henceforth in this analysis we assume that location varies as a discrete variable taking on only unit positive and unit negative values based on the direction of motion, while the time taken between two location steps varies continuously based on the magnitude of velocity. Also, changes need to be done to Equation (4.1), where we model the inverse of velocity rather than velocity itself as a Gauss-Markov process.

$$\nu_n = \alpha \cdot \nu_{n-1} + (1-\alpha) \cdot \mu + \sqrt{1-\alpha^2} \cdot \xi_{n-1} \tag{4.43}$$

Based on such a model, we now try and count the number of ways in which the mobile can make forward and backward movements once it has reached the range boundary. For the mobile to take $(2n+2)$ steps to return back to within-range, it must make $(n+1)$ forward and $(n+1)$ backward steps. These $(n+1)$ forward and $(n+1)$ backward steps can be arranged in any order, as long as the mobile does not land up within-range at any intermediate time during those $(2n+2)$ steps. For setting up the calculation, note that the mobile must first make a forward 1F step, and at the end make a backward 1B step. In the meanwhile, it can at best return back to within 1-step of the starting point, but no further. Hence, for our calculations, we assume that the 1-step away point is origin, and that the mobile makes 1F step to reach origin at the start of the sequence, makes a series of intermediate steps and returns back to the origin at the end of $2n$ steps, and then makes 1B step to return to within range. We let $\mathbf{\{n\}}$ denote the sequence of intermediate steps that the mobile makes, where it at best returns back to origin and no further. Note that in this setup, the minimum number of times the mobile can visit

Number of visits to origin	Sequence of Steps
0	(3F-3B)(2F-1B-1F-2B)
1	(1F-1B-2F-2B)(2F-2B-1F-1B)
2	(1F-1B-1F-1B-1F-1B)

TABLE 4.1: **{3}**: Series of forward and backward steps to return within-range in $(2n + 2) = 8$ steps

Number of visits to origin	Sequence of Steps
0	(1F→**{3}**→1B)
1	(1F-1B-1F→**{2}**→1B)(1F→**{1}**→1B-1F→**{1}**→1B)
	(1F→**{2}**→1B-1F-1B)
2	(1F-1B-1F-1B-1F→**{1}**→1B)(1F-1B-1F→**{1}**→1B-1F-1B)
	(1F→**{1}**→1B-1F-1B-1F-1B)
3	(1F-1B-1F-1B-1F-1B-1F-1B)

TABLE 4.2: **{4}**: Series of forward and backward steps to return within-range in $(2n + 2) = 10$ steps

Number of visits to origin	Sequence of Steps
0	(1F→**{4}**→1B)
1	(1F-1B-1F→**{3}**→1B)(1F→**{1}**→1B-1F→**{2}**→1B)
	(1F→**{2}**→1B-1F→**{1}**→1B)(1F→**{3}**→1B-1F-1B)
2	(1F-1B-1F-1B-1F→**{2}**→1B)(1F-1B-1F→**{1}**→1B-1F→**{1}**→1B)
	(1F-1B-1F→**{2}**→1B-1F-1B)(1F→**{1}**→1B-1F-1B-1F→**{1}**→1B)
	(1F→**{1}**→1B-1F→**{1}**→1B-1F-1B)(1F→**{2}**→1B-1F-1B-1F-1B)
3	(1F-1B-1F-1B-1F-1B-1F→**{1}**→1B)(1F-1B-1F-1B-1F→**{1}**→1B-1F-1B)
	(1F-1B-1F→**{1}**→1B-1F-1B-1F-1B)(1F→**{1}**→1B-1F-1B-1F-1B-1F-1B)
4	(1F-1B-1F-1B-1F-1B-1F-1B-1F-1B)

TABLE 4.3: **{5}**: Series of forward and backward steps to return within-range in $(2n + 2) = 12$ steps

the origin in the meanwhile is 0, while the maximum number of times it can do so is $(n - 1)$. We work out examples of **{n}** with $n = (3, 4, 5)$ here in Tables 4.1, 4.2 and 4.3.

A close examination of the examples worked out in Tables 4.1, 4.2 and 4.3 leads to the discovery of a structure. For defining this structure, we define the following operators:

Definition 4.1. We define unary operator $||\cdot||$, such that given a sequence of steps A, $||A||$ denotes the number of sequences contained in A.

Definition 4.2. We define binary operator $\rightarrow$, such that given two sequence of steps A and B, $A \rightarrow B$ indicates the set of steps in A followed by set of steps in B.

Definition 4.3. We define **{n,k}** as sequences involving n forward and n backward steps where the mobile returns back to origin exactly k times

Theorem 4.4. *The application of operator $\rightarrow$ generates sequences such that $||A \rightarrow B|| = ||A|| \cdot ||B||$.*

Proof. As per Definition 4.2, $A \rightarrow B$ comprises of all sequences of A followed by all sequences of B. Hence,

$$A \rightarrow B \equiv A \times B \tag{4.44}$$

where $A \times B$ denotes the set of sequences produced as a results of taking a set cartesian product of sequences of A with those of B. Hence,

$$||A \rightarrow B|| = ||A \times B|| = ||A|| \cdot ||B|| \tag{4.45}$$

$\square$

Corollary 4.5. *Let i denote the first position at which the mobile returns back to origin in a* $\{\boldsymbol{n,k}\}$ *sequence, $i \leq k$. Then* $\{\boldsymbol{n,k}\}\equiv$*(1F*$\rightarrow\{i-1\}\rightarrow$*1B*$\rightarrow\{n-i, k-1\}$*). This is so as the mobile returns back to the origin exactly after first i steps, which is the same as our original problem; and having returned back to origin after i steps, the mobile needs to visit the origin $k-1$ times in the rest $n-i$ steps.*

The above corollary helps us to come up with the following recursive formulation of $\{\mathbf{n}\}$:

$$\{\mathbf{n}\} = \bigcup_{k=0}^{n} \{\mathbf{n,k}\} \tag{4.46}$$

$$\{\mathbf{n,k}\} = \bigcup_{i=1}^{n-k} \{\mathbf{i\text{-}1}\} \rightarrow \{\mathbf{n\text{-}i,k\text{-}1}\} \tag{4.47}$$

$$\{\mathbf{1}\} = \text{1F-1B} \tag{4.48}$$

$$\{\mathbf{0}\} = \phi \tag{4.49}$$

where k denotes the number of times the mobile visits origin and i denotes the first position at which the mobile returns back to origin for a given k. Correspondingly, the algorithm for determining $||\{\mathbf{n}\}||$ can be formulated as

input : The number of steps the mobile spends out of range, $(2n + 2)$
output: The number of ways in which the mobile can take these steps, $||\{\mathbf{n}\}||$

if n=0 **then**
    NumberOfSteps $\leftarrow$ 1;
end
for $k \leftarrow 0$ **to** n **do**
    this $\leftarrow$ *NumberOfStepsWithJumps*(n, k);
    NumberOfSteps $\leftarrow$ *NumberOfSteps* + *this*;
end

Algorithm 1: NumberOfSteps

input : Number of steps, and number of times the mobile returns to origin in doing so (n, k).
output: The number of ways in which the mobile can make such steps, $||\{\mathbf{n,k}\}||$

for $i \leftarrow 1$ **to** $(n - k)$ **do**
    this $\leftarrow$ *NumberOfSteps*$(i - 1)$;
    right $\leftarrow$ *NumberOfStepsWithJumps*$(n - i, k - 1)$;
    NumberOfStepsWithJumps $\leftarrow$ *NumberOfStepsWithJumps* + *this* * *right*;
end

Algorithm 2: NumberOfStepsWithJumps

input : The number of steps the mobile spends out of range, $(2n + 2)$
output: The total number of ways for the mobile to make such steps, accounting for double-counting

if $n \leq 0$ **then**
    TotalNumberOfSteps $\leftarrow$ 4;
end
else
    TotalNumberOfSteps $\leftarrow$
    (*TotalNumberOfSteps*$(n - 1)-$*NumberOfSteps*$(n - 1)) \times 4$;
end

Algorithm 3: TotalNumberOfSteps

Based on Algorithms 1-3, we numerically find the value of *NumberOfWays* and *Total-NumberOfWays* for different number of steps the mobile makes. These are summarized in Table 4.4.

TABLE 4.4: Calculating *NumberOfSteps* and *TotalNumberOfSteps* for different values of n

n	$\|\|\{\mathbf{n}\}\|\|$	**Total # of ways**	**% of total ways**
0	1	4	0.25
1	1	12	0.083333
2	2	44	0.045455
3	5	168	0.029762
4	14	652	0.021472
5	42	2552	0.016458
6	132	10040	0.013147
7	429	39632	0.010825
8	1430	156812	0.009119
9	4862	621528	0.007823
10	16796	2466664	0.006809
11	58786	9799472	0.005999
12	208012	38962744	0.005339
13	742900	155018928	0.004792
14	2674440	617104112	0.004334
15	9694845	2457718688	0.003945
16	35357670	9792095372	0.003611
17	129644790	39026950808	0.003322
18	477638700	155589224072	0.00307
19	1767263190	620446341488	0.002848
20	6564120420	2474716313192	0.002652
21	24466267020	9872608771088	0.002478
22	91482563640	39392570016272	0.002322
23	343059613650	157204349810528	0.002182
24	1289904147324	627445160787512	0.002056
25	4861946401452	2504621026560750	0.001941
26	18367353072152	9999036320637200	0.001837
27	69533550916004	39922675870260100	0.001742
28	263747951750360	159412569277376000	0.001654
29	1002242216651360	636595285302505000	0.001574
30	3814986502092300	2542372172343410000	0.001501

Equation 4.43 means that the time T_n taken to travel a unit distance at instance n would be distributed as a Gaussian, and that T_i and T_k would be independent and identically distributed (i.i.d.). Hence, in the 1-D single base-station case, time taken for the mobile to return back to within-range can be written as:

$$T_{\text{disconnect}} = \left(2 * \frac{1}{4} + 4 * \frac{1}{12} + 6 * \frac{2}{44} + 8 * \frac{5}{168} + 10 * \frac{14}{652} + 12 * \frac{42}{2552} + \cdots\right) * T_n \tag{4.50}$$

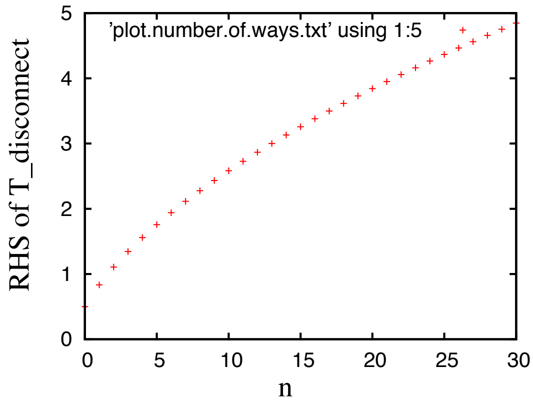

FIGURE 4.1: Plot of the r.h.s summation series of Equation 4.50, showing asymptotically convergent behavior

The right-hand side of the equation above is plotted in Figure 4.1. Note that the coefficients in Equation 4.50 can not be reduced to a closed form, and hence need to be calculated numerically. However, since their values are not dependent on mobility patterns of individual mobile users, these can be easily pre-computed and stored for future use. Also, the time spent by the mobile in carrying out individual steps T_n is also i.i.d., thereby implying that the time spent in n steps would be simply be n times the time spent in an individual step.

4.4 Disconnection Prediction using a Modified City Section Model

In the previous sections, we looked at disconnection prediction using the Gauss Markov and modified Gauss Markov as our reference mobility models. Mobility models are often studied in the literature on wireless networks from the point of view of designing benchmarks for testing various wireless network protocols. Reader is referred to [108] for a survey on the work carried out in the area of mobility modeling in ad-hoc networks. Interest in designing such studies is not to be able to model and forecast the mobility patterns of a particular mobile node, but to be able to simulate an ensemble of various mobile nodes and their mobility. Also, these mobility models are postulated by keeping in mind that most popular use of ad hoc networks is in military applications. In spite of the differences in assumptions needed for modeling mobility in cellular networks in cities and ad hoc networks in difficult terrain, various studies on design of protocols and algorithms for cellular networks [40, 106, 107] borrow mobility models from the ad hoc

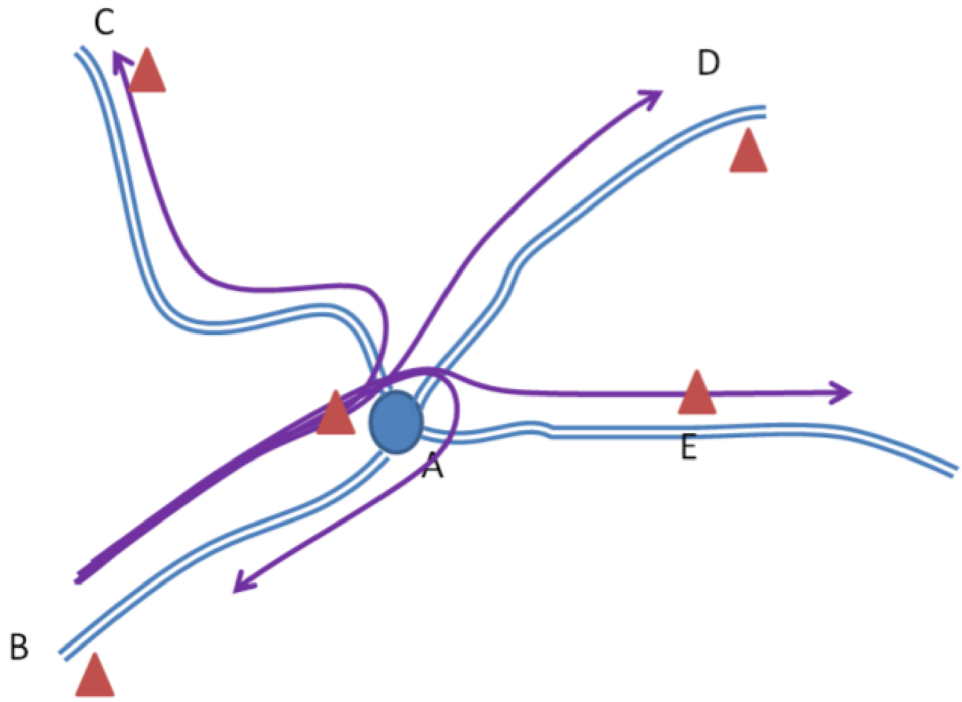

FIGURE 4.2: Typical traffic pattern in a city

networks literature. In this section, we use the city-section mobility model, which is based on a metropolitan city traffic conditions, and derive expressions for disconnection duration assuming such a model of mobility.

4.5 Formulation of the City-Section Model

Figure 4.2 shows a typical traffic pattern in a metropolitan city. Point A denotes a busy road intersection, while points B, C, D and E denote alternative routes that originate from A. Good network design of the cellular network around point A would typically have base-stations situated at each of the points A to E, denoted by triangles in Figure 4.2. Without loss of generality, let us consider only traffic inbound from point B towards point A. There are four alternatives for the traffic to follow, as shown by the four arrows in Figure 4.2. Let p_B, p_C, p_D, p_E denote the probability of a mobile node travelling towards points B, C, D and E with average speeds s_B, s_C, s_D, s_E respectively.

Figure 4.3 shows expected signal strength patterns around the setup of Figure 4.2. Signal interference would ensure that signals from individual base-stations are not spread equally in all directions. Let τ denote the signal to noise ratio threshold below which the

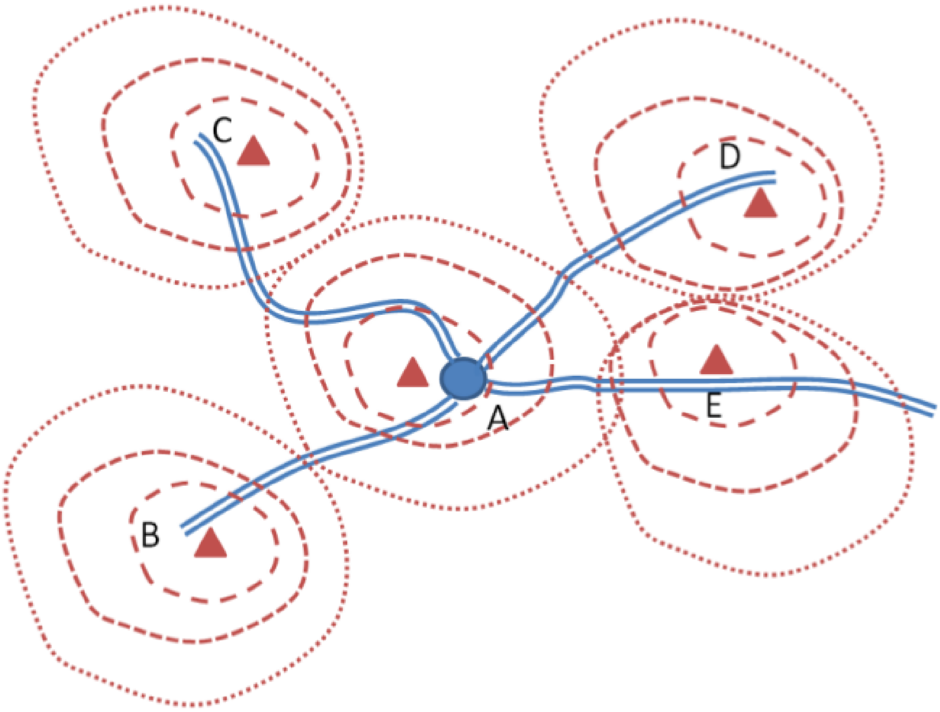

FIGURE 4.3: Typical signal strength patterns for traffic flow of Figure 4.2

mobile node does not remain connected to the nearest base-station. In the rectangular coordinate system of Figure 4.4, this indicates that for a well-configured cellular network, the distance between two nearest base-stations would be 2τ. Note that a pair of base-stations which are nearest to each other would either be located on the same vertical axis or the same horizontal axis of the rectangular coordinate system.

During any particular time, when the base station, due to various network traffic and weather-related signal strength fluctuation reasons, can only connect to a mobile node a unit distance away, the amount of time the mobile node would stay disconnected when travelling from point A to point $i \in \{B, C, D, E\}$ would be

$$t_i = \frac{(2\tau - 2)}{s_i} \tag{4.51}$$

As the mobile node travels to point i with probability p_i and speed s_i, the point estimate of the disconnection duration can be written as

$$t = \sum_{i \in \{B,C,D,E\}} p_i t_i = \sum_{i \in \{B,C,D,E\}} p_i \cdot \frac{(2\tau - 2)}{s_i} \tag{4.52}$$

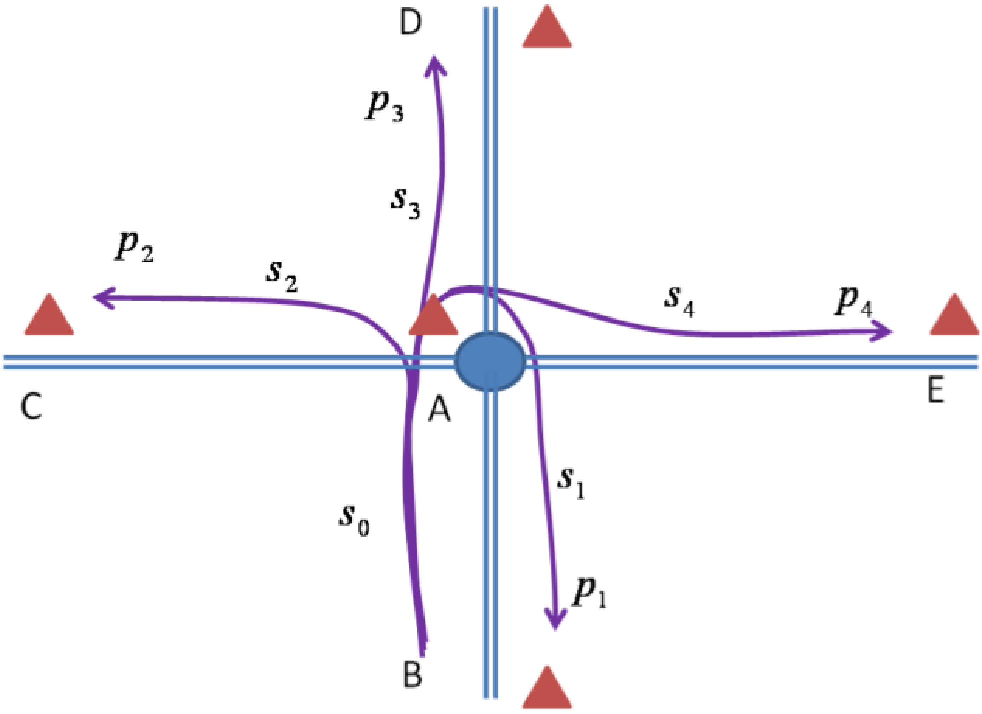

FIGURE 4.4: Transformation of traffic patterns and signal strengths to a rectangular coordinate system

4.6 Analysis of the model

One downside of the proposed scheme is that it requires a number of parameters values to be estimated by the base-station A. It needs to calculate values of probability that a mobile node would take a particular route p_i, and also the average speeds s_i with which they would travel on these routes. p_i values can be estimated by keeping a count of the handovers from each of the base-stations. So, in the current case the following should hold.

$$p_B = \frac{\text{\#Handovers from B that were handed back to B}}{\text{Total \# of handovers from B}} \tag{4.53}$$

$$p_i = \frac{\text{\#Handovers from B that are forwarded to } i}{\text{Total \# of handovers from B}}, \forall i \in \{C, D, E\} \tag{4.54}$$

Estimation of speeds s_B, s_C, s_D, s_E may seem more complicated as their values can not be estimated by just looking at the records of handover requests originating from base-station B. However, if we consider all requests for handovers, these values can be calculated in the following manner. Let $\mu_{xy}, \forall (x, y) \in (A \times \{B, C, D, E\}) \cup (\{B, C, D, E\} \times A)$ denote the speeds observed by mobile nodes on each route (x, y) links and $t_{kl}, \forall k, l \in \{B, C, D, E\}$ denote the average times between successive handovers $k \to A, A \to l$. Dependencies between μ and t can be written as follows.

$$t_{kl} = \frac{1}{\mu_{kA}} + \frac{1}{\mu_{Al}}, \forall k, l \in \{B, C, D, E\} \tag{4.55}$$

This would lead to a system of equations to be solved with 16 equations (one for each (k, l) pair) and 8 unknowns (all values of μ's. Speeds s_B, s_C, s_D, s_E can now be written as

$$s_B = \frac{\mu_{BA} + \mu_{AB}}{2} \tag{4.56}$$

$$s_C = \frac{\mu_{BA} + \mu_{AC}}{2} \tag{4.57}$$

$$s_D = \frac{\mu_{BA} + \mu_{AD}}{2} \tag{4.58}$$

$$s_E = \frac{\mu_{BA} + \mu_{AE}}{2} \tag{4.59}$$

Values of probability estimates p obtained in Equations (3) and (4), and speed estimates obtained in Equations (6)- (9) can now be plugged back in Equation (2) to find an estimate of the disconnection duration. Such estimates can then be used as input to the ACK Pacing algorithm discussed in Chapter 3.

4.7 Conclusions

In this Chapter, we looked at disconnection prediction using the Gauss-Markov mobility modeling scheme. This led us to an expression that would be extremely performance intensive for real-world systems. We then developed an alternate mobility model based on Gauss-Markov process. Table 4.4 represents the number of distinct sequences following which the mobile can land up within range in exactly $(2n + 2)$ steps. These values combined with the distribution of the amount of time the mobile spends between each step of the mobility model helped us get the distribution of the total amount of time spent by the mobile outside range. Such a formulation can then be utilized to serve as the mobility prediction module for various applications and it is not as performance intensive as the one that used the Gauss-Markov mobility model directly.

The computations shown in Table 4.4 are compute-intensive, and are time-consuming to compute on the fly. However, per Equation 4.50, the summation series is independent of the nature of movement of a mobile, and hence can be pre-computed and tabulated. Also, the time spent by the mobile in carrying out individual steps T_n is also i.i.d., thereby implying that the time spent in n steps would be simply be n times the time spent in an individual step.

In the second part of the chapter, we outlined a mobility model based on a metropolitan city traffic conditions. Based on the described model, only point estimates for disconnection duration were developed. As we have pointed out in Chapter 3, the TCP ACK Pacing algorithm needs a over-estimate of the disconnection duration. Finding the statistical distribution for the estimated disconnection prediction would help us develop performance guarantee bounds for the ACK Pacing algorithm. While implementing the suggested scheme in a real-world setup, we envisage that the mobile nodes (and associated users) would not exhibit one homogeneous behavior. For example, the speed of a user between any two points would depend upon his/her choice of mode of transport. But, for a similar choice of mode of transport, users can be expected to spend nearly equal times for transit with nearly equal speeds. Hence, we foresee that it may be necessary to develop a class-based mechanism that estimates the parameters of the model based on the estimated class that the user belongs to. Such a classification-based model may be a better approximation of the real-world phenomena.

The Modified Gauss Markov Model for disconnection prediction discussed in Section 4.3 was published in [109].

Chapter 5

Machine Learning methods for predicting disconnection durations

In earlier chapters, we have developed mechanisms for predicting disconnection duration of a mobile node whose movements are modeled using different mobility models. However, in the real world, mobile nodes change their patterns of movement and we cannot generalize all the movements with a single mobility model. The efficiency of disconnection prediction will also be improved if the predicting entity could adapt its predictions based on changes in network, changes in mobile node behavior and so on. This is achieved much better with the use of machine learning techniques that learn and adapt to changing conditions. In this chapter, we first take a brief look at the various machine learning techniques. We then use these techniques to predict disconnection by training and validating them using samples generated from an entity synthetic mobility model.

In order to compare the quantify and compare the different machine-learning models in this context, we use Random Walk and Random Direction mobility models in order to generate mobile movement data and data to indicate if the mobile is disconnected at each point of movement. We then construct the different machine-learning models such as artificial neural networks, Radial Basis Function Networks and Random Forests using Weka [46]. We use 90% of this data as the training set for the machine learning models and 10% of it as the test data set. In each case, we tune the parameters of the model until we reach a combination that provides us the best accuracy of prediction for the given data set.

5.1 An Overview of Machine Learning Techniques

Machine learning is a sub-field of artificial intelligence concerned with algorithms that allow computers to learn. This means that given a set of data, an algorithm infers information about the properties of the data. This information allows it to make predictions about other data it may see in the future. The main focus of machine learning is the design of algorithms that recognize complex patterns and make intelligent decisions based on input data, like humans do. However, it is not without its problems. The main problem is in interpreting "new" patterns that do not conform to any pattern in the input data. These patterns are more or less always misinterpreted. The other main problem is over-generalizing of data. Machine learning has found uses in areas like biotechnology, fraud detection, wireless networks, stock market analysis and national security.

Machine learning algorithms can be organized into a taxonomy based on the desired outcome of the algorithm.

- Supervised learning generates a function that maps inputs to desired outputs (also called labels, because they are often provided by human experts labeling the training examples). For example, in a classification problem, the learner approximates a function mapping a vector into classes by looking at input-output examples of the function.
- Unsupervised learning models a set of inputs, like clustering. Data mining and knowledge discovery are other examples.
- Semi-supervised learning combines both labeled and unlabeled examples to generate an appropriate function or classifier.
- Reinforcement learning learns how to act given an observation of the world. Every action has some impact in the environment, and the environment provides feedback in the form of rewards that guides the learning algorithm.

- Transduction, or transductive inference, tries to predict new outputs on specific and fixed (test) cases from observed, specific (training) cases.
- "Learning to learn" learns its own inductive bias based on previous experience.

In this section, we focus on some of the most popular supervised learning techniques and discuss the algorithm used in each of them and their pros and cons.

5.1.1 Neural Networks

Artificial Neural networks (ANNs) come under the general class of learning mechanisms that are widely used in prediction problems today. ANNs have the capability to learn the particulars of a problem at hand based on the samples that are provided to it. ANNs could undergo a supervised learning, where the ANN learns the relationship between input and output variables based on samples (or training set) that has both input values and output values specified. ANNs can be designed based on reinforcement learning, especially for classification problem, where a critic is available, which for every given set of input values critics the ANN regarding whether its output on those set of input values was correct or incorrect. ANNs can also be designed to operate completely in an unsupervised scenario, where the ANN is supposed to learn the in-built statistical properties existing in the input variables.

Perceptrons are the most basic of ANNs that can be used to carry out classification tasks. A Perceptron can be specified in the following functional form:

$$f(x) = \begin{cases} 1 \text{ if } (w.x + b) > 0 \\ 0 \text{ otherwise.} \end{cases} \tag{5.1}$$

where x is the set of inputs available, $f(x)$ is the output classification being learnt, w is the set of internal weights that the Perceptron learns to attach to individual inputs based on its training set, and b is a constant whose value is also learnt as part of training.

Note that the Perceptron specified above operates on a threshold. If the value of $(w.x+b)$ is above 0 the output is 1, and 0 otherwise. In this case, the Perceptron is termed to have a threshold activation function. As part of a neural network design, activation functions are applied to the output of the neural network and their primary function is to restrict the values of the ANN output to desired values, rather than the unrestricted values that the weighted sum of inputs tend to take. Other activation functions that are frequently used are the linear, semi-linear and sigmoid functions. Artificial Neural networks (ANNs) come under the general class of learning mechanisms that are widely used in prediction problems today. ANNs have the capability to learn the particulars of a problem at hand based on the samples that are provided to it. ANNs could undergo a supervised learning, where the ANN learns the relationship between input and output variables based on samples (or training set) that has both input values and output values specified. ANNs can be designed based on reinforcement learning, especially for classification problem, where a critic is available, which for every given set of input values critics the ANN regarding whether its output on those set of input values was correct or incorrect. ANNs can also be designed to operate completely in an unsupervised scenario, where the ANN is supposed to learn the in-built statistical properties existing in the input variables.

Back-propagation networks or backprops are one of the most popular types of ANNs. A backprop is made up of many neurons. Each neuron takes a weighted sum of its inputs and passes the result through an activation function to yield the output. In a backprop network, many such neurons are interconnected. These neurons are arranged in layers known as the input layer, the hidden layers, and the output layer. Backprop networks undergo supervised training, where for a set of inputs of the training set, the correct set of outputs is made available to the network. The set dk of correct output values represent the target values for the neurons in the output layer, but no such targets are available for neurons in the hidden layers. An important feature of backprops is that errors in the output layer are propagated backwards towards the hidden layers. In this way the hidden layers learn the correct set of weights. If there are L neurons in the output layer, the error for the output layer is

$$E(W) = \frac{1}{2}\sum_{k=1}^{L}(d_k - y_k)^2 \tag{5.2}$$

where W represents the set of all the weights of the backprop network and y_k is the output of the backprop for the given set of inputs $x_1, x_2 \ldots x_N$ and weights W. Under the error backpropagation learning rule, the weights of the connections to the output units are modified in the negative direction of the error gradient, thereby reducing the error in the next iteration. Note that Figure 1 shows a particular architecture of a backprop network where the neural network as N inputs, 2 outputs, an input layer, 2 hidden layers of J hidden units each and an output layer. The activation function of the nodes in the hidden layer is a sigmoid, whereas the output layer has the linear activation function. As part of design of the neural network, the designer has to have one input and one output layer, but has the choice of choosing zero, one or multiple hidden layers. The designer also chooses the number of hidden units that are part of each of these hidden layers, along-with their activation functions.

Learning through backpropagation in a neural network is equivalent to the problem of global optimization through use of local search methods like gradient descent. The learnt set of weights correspond to the local minimum found in the proximity of starting weight solution, and is found out by using the gradient descent procedure described earlier. While the structure of the neural networks is such that usually the set of weights corresponding to a local minimum perform well in the prediction and classification task but an ANN designer would do well to carry out a multi-start approach during the training phase. As part of a multi-start approach, the neural network is tasked to begin its training from different set of starting weights. These weights can be chosen either randomly, or in a methodical manner based on the knowledge of the structure of the problem at hand.

Neural networks and many other learning mechanisms require large training sets to be able to learn the functional existing between the inputs and the outputs of a particular prediction / classification problem. However, many problems do not possess a large set of training data available, for which the designer is forced to use a bootstrapping mechanism for generating a larger set of training data based on the available smaller set. One has to be careful while doing this, as neural networks, especially the ones that are designed with many hidden layer units, can approximate very complex functionals. In such cases, the neural network ends up memorizing the set of input-output value pairs and is unable to generalize for input values that it may not have seen as part of training. A popular approach for avoiding this pitfall is to look at the objective of minimizing the validation error rather than the training error. Also, the designer is advised to follow the Occam's razor rule of designing the simplest neural network that satisfactorily performs the prediction / classification task for the problem at hand.

5.1.2 RBF Networks

A Radial Basis function is an artificial neural network that uses radial basis functions as the activation functions. It is a linear combination of radial basis functions. These are used in function approximation, time series prediction and control.

A radial basis function (RBF) is a real-valued function whose value depends only on the distance from the origin, so that $\phi(x) = \phi(||x||)$; or alternatively on the distance from some other point c, called a center, so that $\phi(x, c) = \phi(||x - c||)$. Any function ϕ that satisfies the property $\phi(x) = \phi(||x||)$ is a radial function. The norm is usually chosen as the Euclidean distance, although other distance functions are also possible. For example by using Lukaszyk-Karmowski metric, it is possible for some radial functions to avoid problems with ill-conditioning of the matrix solved to determine coefficients w_i (see below), since the $||x||$ is always greater than zero [110]. Sums of radial basis functions are typically used to approximate given functions. This approximation process can also be interpreted as a simple kind of neural network.

Radial basis function (RBF) networks typically have three layers: an input layer, a hidden layer with a non-linear RBF activation function and a linear output layer. The norm is typically taken to be the Euclidean distance (though Mahalanobis distance appears to perform better in general) and the basis function is taken to be Gaussian. The Gaussian basis functions are local in the sense that changing parameters of one neuron has only a small effect for input values that are far away from the center of that neuron.

RBF networks are universal approximators on a compact subset of $\mathcal{R}^n$. This means that a RBF network with enough hidden neurons can approximate any continuous function with arbitrary precision.

5.1.3 Decision Trees

Decision trees are a completely transparent method of classifying observations, which, after training look like a series of if-then statements arranged into a tree. The most popular algorithm for building a decision tree is called CART Classification And Regression Trees. The algorithm works by first creating a root node using the best variable to divide up the data. In order to determine the best variable, each variable is used to divide the data into 2 sets, and different metrics are used to determine how mixed the resultant sets are. Two such metrics are described below:

1. Gini impurity: This is the expected error rate if one of the results from a set is randomly applied to one of the items in the set. If every item in the set is in the same category, the guess will always be correct, so the error rate is 0. If there are four possible results evenly divided in the group, theres a 75 percent chance that the guess would be incorrect, so the error rate is 0.75. Gini impurity calculates the probability of each possible outcome by dividing the number of times that outcome occurs by the total number of rows in the set. It then adds up the products of all these probabilities. This gives the overall chance that a row of data would be randomly assigned to the wrong outcome. The higher this probability, the worse the split.

2. Entropy: In information theory, this is the amount of disorder in a set basically, how mixed a set is. Entropy calculates the frequency of each item, and applies these formulae:

$$P(i) = \mathit{frequency}(\text{outcome}) = \mathit{count}(\text{outcome})/\mathit{count}(\text{total rows}) \tag{5.3}$$

$$\text{Entropy} = \sum P(i) * \mathit{log}(P(i)) \forall i \in \text{outcomes} \tag{5.4}$$

The more mixed the groups are, the higher is the entropy. Our goal in dividing the data into two new groups is to reduce the entropy. Decision trees are built recursively, wherein the information gain is calculated to find the best attribute (the one with the highest information gain). This attribute is then used as the root node with two branches those that meet the condition, and those that do not. For each branch, the algorithm then determines the next variable to use. This process continues and stops only when the information gain from splitting an attribute is not more than a pre-specified threshold.

The main problem with decision trees is that they can become overfitted that is, they become too specific to the training data. An overfitted tree may give an answer as being more certain than it really is by creating branches that decrease entropy slightly for the training set, but whose conditions are actually completely arbitrary. In order to avoid this, a strategy called tree pruning is used to eliminate superfluous nodes. Pruning involves checking pairs of nodes that have the same parent to see if merging them would increase the entropy by less than a specified threshold. If so, the leaves are merged into a single node with all the possible outcomes. This helps avoid overfitting.

5.1.4 Random Forest - Specialization of decision trees

Random forest is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the classes output by individual trees. Each tree is constructed using the following algorithm:

1. Let the number of training cases be N, and the number of variables in the classifier be M.

2. We are told the number m of input variables to be used to determine the decision at a node of the tree; $m << M$.

3. Choose a training set for this tree by choosing n times with replacement from all N available training cases (i.e. take a bootstrap sample). Use the rest of the cases to estimate the error of the tree, by predicting their classes.

4. For each node of the tree, randomly choose m variables on which to base the decision at that node. Calculate the best split based on these m variables in the training set.

5. Each tree is fully grown and not pruned (as may be done in constructing a normal tree classifier).

For prediction a new sample is pushed down the tree. It is assigned the label of the training sample in the terminal node it ends up in. This procedure is iterated over all trees in the ensemble, and the mode vote of all trees is reported as random forest prediction.

The main advantages of random forest are that it runs efficiently on large databases and is one of the most accurate learning algorithms. In addition, it can handle a large number of input variables and give estimates of what variables are important in the classification. However, random forests have been observed to overfit for some datasets and are biased in favor of attributes with more levels in datasets with categorical variables.

5.1.5 Evaluation and Comparison

Tan and Gilbert [111] perform an empirical comparison of rule-based systems (decision trees), statistical learning systems (Naive Bayes, neural networks) and ensemble methods (stacking, bagging and boosting). The authors use predictive accuracy (the percentage of correctly classified instances) and sensitivity (the percentage of actual positive samples that are correctly classified) as the basis for comparing the different techniques. Experiments indicate that combination approaches perform better than any of the individual learning techniques. Only Naive Bayes performed better than the combination approach because it is capable of classifying instances based on simple prior probabilistic knowledge. Based on results, the authors chalk out a set of rules for determining the approach to be used for a new problem. Some of these rules include:

1. Better and more robust classifiers are built if the data set contains an equal number of true positives and true negatives.

2. The nature of the attributes also contributes to the decision on the type of technique. Statistical techniques perform much better over multiple dimensions and continuous attributes. Rule-based systems work better in discrete / categorical attributes.

3. The right technique is also chosen based on what the user wants to discover from the data. If the intention is to generate a set of understandable hypotheses, then a rule-based system must be used.

4. Combined approaches work better than a single approach. This is because the combined approach averages out the different hypotheses, producing a good approximation to the true hypotheses.

Vanneschi et al. [112] compare machine learning techniques for survival prediction in cancer. Current cancer therapies have serious side effects: ideally type and dosage of the therapy should be matched to each individual patient based on his/her risk of relapse. Therefore the classification of cancer patients into risk classes is a very active field of research, with direct clinical applications. Until recently patient classification was based on a series of clinical and histological parameters. The advent of high-throughput techniques to measure gene expression led in the last decade to a large body of research on gene expression in cancer, and in particular on the possibility of using gene expression data to improve patient classification. A gene signature is a set of genes whose levels of expression can be used to predict a biological state [113]: in the case of cancer, gene signatures have been developed both to distinguish cancerous from non- cancerous conditions and to classify cancer patients based on the aggressiveness of the tumor, as measured for example by the probability of relapsing within a given time.

While many studies have been devoted to the identification of gene signatures in various types of cancer, the question of the algorithms to be used to maximize the predictive power of a gene signature has received less attention. To investigate this issue systematically, the authors consider one of the best established gene signatures, the 70-gene signature for breast cancer, and compare the performance of four different machine learning algorithms (Genetic Programming, Support Vector Machines, Multilayered Perceptrons and Random Forests) in using this signature to predict the survival of a cohort of breast cancer patients.

Results show that the best solutions were found by Genetic Programming and Multilayered Perceptrons and the best average result was found by Genetic Programming. Moreover, statistical analysis indicates that Genetic Programming consistently outperforms the other methods except Support Vector Machines using polynomial kernel with degree 2. In order to determine to what extent feature selection is responsible for the good performance of Genetic Programming, the authors identified the 10 features most often selected by Genetic Programming among the 70 initial features and ran again both GP and SVM with quadratic kernel using only these features. The performance of both methods significantly improved.

These results suggests on one hand, that the feature selection performed by GP has intrinsic value, not necessarily tied to the use of syntax trees, since the SVM can take advantage of the feature selection performed by GP to improve its performance. Second, that a recursive use of GP, in which a first run is used to select the best features to be used in a second run, might be a promising way of optimizing the method.

Abu-Nimeh et al. [114] compare machine learning techniques in predicting phishing. They compare the predictive accuracy of several machine learning methods including Logistic Regression (LR), Classification and Regression Trees (CART), Bayesian Additive Regression Trees (BART), Support Vector Machines (SVM), Random Forests (RF), and Neural Networks (NNet) for predicting phishing emails. During training and testing the authors used 10-fold- cross-validation and averaged the estimates of all 10 folds (sub-samples) to evaluate the mean error rate for all classifiers. The results showed that, when legitimate and phishing emails are weighted equally, RF outperforms all other classifiers followed by CART, LR, BART, SVM, and NNet respectively. NNet achieved the worst error rate. Although RF outperformed all classifiers, it achieved the worst false positive rate.LR had the minimum false positive rate. When applying cost-sensitive measures, i.e. penalizing false positives 9 times more than false negatives, LR outperformed all classifiers achieving the minimum weighted error rate followed by BART, NNet, CART, SVM, and RF respectively. However, RF had the worst weighted error rate.

All studies and comparisons of machine learning techniques show that there is no single metric for evaluating techniques. The percentage of correctly classified instances can never be solely used as the evaluation metric. Just like the best machine learning technique to be used depends on the characteristics of the data, so do the evaluation metrics.

5.2 Mobility Models Revisited

As we saw in Chapter 2, the literature on mobility models can be divided into entity and group mobility models. The entity mobility models simulate the movement of mobile nodes so that all mobile nodes move independent of each other. Some famous entity mobility models discussed in latter sections of this paper include the random walk, random direction, random waypoint and Gauss-Markov models. The group mobility models simulate the movement of mobile nodes with a dependency existing between the movements of all nodes. Some examples of the group mobility model are exponential correlated random mobility model, column mobility model and reference point group mobility model.

Bettstetter [115] provides a concise categorization of mobility models. Camp et al. [108] provide a survey and simulation-based comparison of a variety of mobility models. In addition, there is an increasing amount of literature on mobility models for ad-hoc networks that are more realistic in emulating mobile node movement so as to be able to evaluate the ad-hoc network protocols more accurately. Haas [116] discusses one such mobility model called the obstacle mobility model that models the movement of mobile nodes in terrains that resemble real world topographies. This model has been discussed in detail in the latter sections.

The Random Walk mobility model described in [117] has become the foundation of a number of mobility models. In this model, each node selects a direction to move in. The node chooses its speed based on a user-defined distribution of speeds and then moves in the chosen direction with the chosen speed. After some randomly chosen amount of time, each node halts and selects a new direction to move in. This is a memoryless mobility model where the direction and speed of the node at any point is independent of its speed and movement before this point. This characteristic generates unrealistic movements such as sudden stops and sharp turns.

The Random Direction Model in [118] is a variation of this, where instead of stopping after some amount of time, each node moves till it reaches the boundary of the simulation area and then chooses a new direction to move in. This model aims at maintaining a constant density of nodes throughout the simulation. An evaluation of this model shows that network partitions are more likely with this mobility model than others. Also, since, the nodes travel to and then pause at the end of the simulation area, the average hop count data packets in this mobility model is higher than the hop count of other mobility models. In [119], and [120], a different variation of this model is proposed. In this case, when the node reaches the boundary of the simulation area, it is reflected back into the simulation area while the velocity of the node is held constant.

Bettstetter [115], Bettstetter and Wagner [121], Resta and Santi [122] and Royer et al. [118] discuss characteristics of another similar model called the Random Waypoint model. In this model, each node selects a random point in the simulation area as its destination and a speed from an input range. The node then moves towards its destination. Once it reaches its destination, it pauses for some time and then selects a new destination and speed and resumes movement. It has been proved that due to the characteristics of the model, the density of nodes in the simulation area follows a cyclic pattern and the nodes tend to aggregate in the centre of the simulation area that is not realistic. Guerin [117] analyzes this model and finds that the nodes are initially randomly distributed in the simulation area. This initial random distribution is not representative of how the nodes distribute themselves when moving.

In this chapter, we use these three different mobility models, namely random walk, random direction and Gauss-Markov mobility model to generate synthetic mobility data for mobile nodes. We then benchmark various state-of-the-art machine learning techniques in their ability to predict disconnections for such mobility data.

5.3 Simulation Setup and Test Runs

For generating the mobility data using the three mobility models: random walk, random direction and gauss-markov mobility model, we choose the starting set of points (denoted by x and y coordinates) at random from a unit square, denoted as

$$x_0 \sim U[0,1]; y_0 \sim U[0,1] \tag{5.5}$$

indicating that both x_0 and y_0 are random draws from a uniform distribution $U[0,1]$ on the closed interval $[0,1]$. Let x_t and y_t denote the x-coordinate and y-coordinate of the location of the mobile node after t intervals of time. Then for the random walk model, the following relation holds

$$x_t = x_{(t-1)} + \nu \cdot \omega_{(t-1)} \tag{5.6}$$

$$y_t = y_{(t-1)} + \nu \cdot \phi_{(t-1)} \tag{5.7}$$

where ω and ϕ denote random numbers generated from a uniform $U[0,1]$ distribution.

For the random direction model, the following set of equations hold:

$$x_t = x_{(t-1)} + \nu \cdot cos(\rho_{(t-1)}) \tag{5.8}$$

$$y_t = y_{(t-1)} + \nu \cdot sin(\rho_{(t-1)}) \tag{5.9}$$

where ρ_t denotes the random direction chosen by the mobile node for time period t, and is a random number generated from a $U[-\pi, \pi]$ distribution.

For the gauss-markov mobility model, following up on Equation 4.1, the following set of equations hold:

$$v_{x,t} = \alpha \cdot v_{x,t-1} + (1-\alpha) \cdot \mu + \sqrt{1-\alpha^2} \cdot z_{t-1} \tag{5.10}$$

$$v_{y,t} = \alpha \cdot v_{y,t-1} + (1-\alpha) \cdot \mu + \sqrt{1-\alpha^2} \cdot z_{t-1} \tag{5.11}$$

$$x_t = x_{t-1} + v_{x,t-1} \tag{5.12}$$

$$y_t = y_{t-1} + v_{y,t-1} \tag{5.13}$$

As a target variable for the machine learning models, we also generate an indicator variable denoting if a disconnection took place from time $(t-1)$ to t. Note that a mobile node can undergo two kinds of disconnections in our setup: one where the node transitions from one cell to the next, and the other where it maintains its earlier cell but is at a distance far enough from the base-station to be disconnected from it.

For purposes of this work, we define the setup for the wireless network as shown in Figure 5.1. The base station is situated at location (0.5, 0.5), and is part of a rectangular lattice arrangement of base-stations. We simplify, and assume that the base-stations are so located as part of wireless network design that their signals do not overlap much, and hence are one unit distance away from the next base-station on the x and y-axis on the rectangular grid. The range of a base-station is assumed to be a circle of diameter 1 unit distance. Figure 5.1 shows the locations of a mobile node following a random walk mobility model at various time instances. Similar plots can be shown for the random direction and gauss-markov mobility models , which essentially show a similar behavior of the mobile node. The red circle denotes the range of the base- station located at the center of the region. Any mobile that continues its random walk outside the region under study is assumed to land up in a similar region that is replicated to the top, the right, the bottom and to the left of the region under study. Since we are conducting large-scale simulations, we assume that if a mobile were to transition to the region to the left of the region under study, a similar mobile node would enter from the region on the right. Such an assumption is on the same lines as the mean field assumption for simulation of large-scale systems [123], and can be summarized as the following set of

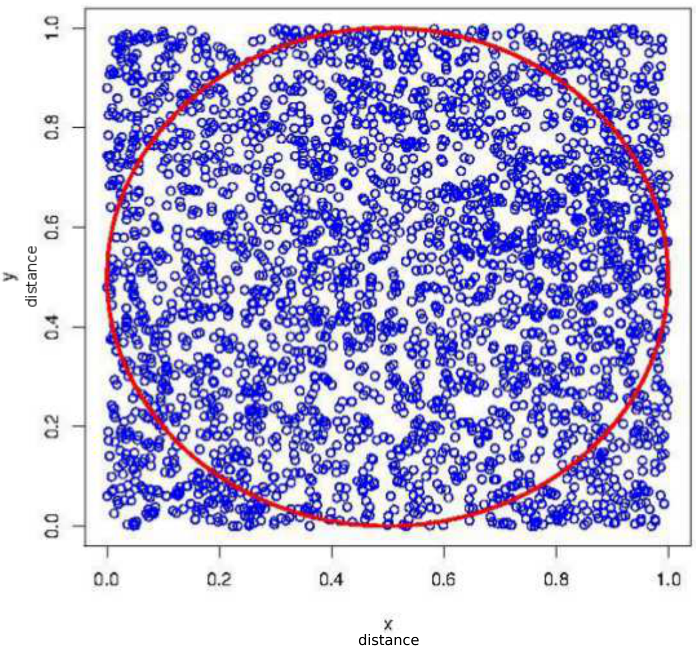

Figure 5.1: Definition of connectivity of a mobile node when the base station is situated at (0.5, 0.5)

equations:

$$x'_t = \begin{cases} 1 - (1 - x_t), x_t > 1 \\ x_t, 0 \leq x_t \leq 1 \\ -x_t, x_t < 0 \end{cases} \tag{5.14}$$

$$y'_t = \begin{cases} 1 - (1 - y_t), y_t > 1 \\ y_t, 0 \leq y_t \leq 1 \\ -y_t, y_t < 0 \end{cases} \tag{5.15}$$

Using the wireless network setup as described above, we generate mobility data using the Random Walk Mobility Model with different velocities. We then proceed to build the artificial neural network at each step varying the number of hidden layers to determine the optimal configuration. We follow a similar approach for building the RBF network and the Random Forest to determine the optimal configuration of the models. For mobility data corresponding to a certain velocity, 90% of the data set is used to train the models and 10% is used to test it and to compute the accuracy of the models. These steps are repeated for mobility data generated using the Random Direction Mobility model. The next section tabulates and discusses the results that we arrive at with these experiments.

5.4 Results Discussion

Table 5.1 shows detailed results computed for experiments related to estimating disconnection for random walk with a velocity of 0.1 distance units per time. The results show that various machine learning techniques across their various architectures have the following performance: ANN on average is able to classify 84.26% instances correctly, while RBF networks correctly classify 84.62% cases and Random Forests are able to classify 84.15% cases correctly.

Similar experiments were carried out with the Random Walk and Random Direction models with velocities ranging in the set (0.1, 0.3, 0.7) for each of them. The aggregated results are shown in Table 5.2. Based on these results, we can see that RBF networks work as well or better than ANNs and Random Forests in most cases. In addition, RBF networks are also favorable in terms of ease of implementation in network elements given the tight processor occupancy and real-time constraints of these systems. Thus, it seems prudent to implement an RBF network in the base station to predict disconnection. This RBF network will initially be trained with disconnection data during deployment. Thereon, it will adapt its parameters based on the real-world behavior. However, this network adaptation will be carried out only at specific times over a long-term period to prevent minor changes in behavior from misleading the RBF network. Once the RBF network is deployed, additional experimentation could be carried out to combine it with other machine learning techniques to check for better performance and results in terms of prediction.

5.5 Conclusion

While implementing a single equation to predict disconnection (as proposed in earlier chapters) is less computationally intensive, the accuracy of the technique is limited because the equation is constructed using a single mobility model that tries to come as close as possible to the real-world, but is not the real-world. In comparison, using an artificial intelligence technique ensures that the network element always works according to the real world conditions and data, even though it may be computationally a little more expensive. The real challenge in this case is then only to find an optimal implementation of the least expensive and most accurate technique to predict disconnection.

In this chapter, we explored the use of Artificial Neural Networks, Radial Basis Function Networks and Random Forests to predict disconnection. In order to train and test these models, we used the data generated from the Random Walk and Random Direction mobility models in our wireless network simulation using a rectangular lattice of base stations. Experiments were run by varying the different parameters of the machine learning models in order to measure the accuracy and the point at which over-fitting sets in. Based on the results of this chapter, we find that the RBF network satisfies our requirements of a reasonably accurate model that is also not extremely computationally intensive.

All the experiments in the current chapter were run in a simulation setup. The next step in this area of work would be to implement these models in a real base station and to measure the accuracy and overheads entailed by the use of machine-learning models. The experiment with a real base station is essential to quantify the tradeoffs involved in having machine-learning models running in real-time systems. In addition, this would allow us to devise algorithms for the thresholds and times at which the model uses feedback to re-tune its parameters, while also giving us feedback if additional parameters such as time-of-day, topology and so on need to be a part of the model.

Table 5.1: Detailed results obtained for Random Walk with a velocity of 0.1

Experiment	**Mean Absolute Error**	**Root Mean Squared Error**	**Relative Absolute Error**	**Root Relative Squared Error**	**Correctly Classified Instances**
ANN, Hidden Layers=1 #perceptrons in it=2	0.2432	0.3602	60.464	80.821	84.6%
ANN, Hidden Layers=2 #perceptrons in each=10	0.1882	0.3116	46.782	69.927	84.5%
ANN, Hidden Layers=2, #perceptrons in $1^{st} = 10$, $2^{nd} = 8$	0.1873	0.3116	46.559	69.786	84.7%
ANN, Hidden Layers=1 #perceptrons in each=5	0.1873	0.315	46.572	70.694	83.7%
ANN, Hidden Layers=1 #perceptrons in it=7	0.2004	0.3221	49.819	72.271	83.4%
ANN, Hidden Layers=1 #perceptrons in it=3	0.2039	0.3278	50.696	73.566	84.7%
ANN, Hidden Layers=1 #perceptrons in it=6	0.1972	0.3244	49.036	72.795	84.6%
ANN, Hidden Layers=1 #perceptrons in it=4	0.2006	0.3235	49.876	72.591	83.1%
ANN, Hidden Layers=1 #perceptrons in it=5	0.1988	0.3223	49.437	72.311	84.6%
ANN, Hidden Layers=1 #perceptrons in it=8	0.1936	0.3151	48.127	70.709	84.6%
ANN, Hidden Layers=1 #perceptrons in it=9	0.1947	0.3198	48.415	71.762	84.4%
RBF, #clusters=3	0.2522	0.3487	62.692	78.243	83.2%
RBF, #clusters=4	0.2294	0.3329	57.040	74.704	84.2%
RBF, #clusters=5	0.2254	0.3341	56.041	74.977	83.5%
RBF, #clusters=6	0.2066	0.3168	51.357	71.085	84.7%
RBF, #clusters=7	0.2107	0.3247	52.393	72.863	83.5%
RBF, #clusters=8	0.2003	0.3146	49.799	70.603	84.9%
RBF, #clusters=9	0.1995	0.3123	49.606	70.078	86.1%
RBF, #clusters=10	0.1923	0.3069	47.808	68.874	86.3%
RBF, #clusters=11	0.1965	0.3115	48.856	69.908	85.2%
RandomForest, #trees=2	0.1849	0.3068	45.971	68.835	84.0%
RandomForest, #trees=3	0.1842	0.3065	45.801	68.778	84.0%
RandomForest, #trees=4	0.1837	0.3060	45.668	68.664	84.0%
RandomForest, #trees=5	0.1836	0.3061	45.649	68.688	84.0%
RandomForest, #trees=6	0.1838	0.3058	45.706	68.626	84.0%
RandomForest, #trees=7	0.1837	0.3057	45.666	68.594	84.4%
RandomForest, #trees=8	0.1836	0.3056	45.638	68.583	84.4%
RandomForest, #trees=9	0.1835	0.3056	45.633	68.577	84.4%

Table 5.2: Aggregated results showing performance of ANN, RBF and Random Forests for Random Walk and Random Direction mobility models for different velocities

Mobility Model	**Velocity**	**ANN**	**RBF**	**Random Forest**
Random Walk	0.1	84.3%	84.6%	84.2%
	0.3	64.9%	63.4%	66.3%
	0.7	80.0%	80.0%	80.0%
Random Direction	0.1	74.5%	86.7%	83.6%
	0.3	–	86.9%	83.6%
	0.7	–	80.7%	72.3%

Chapter 6

Self-Configuration of P2PTV Networks through Incentivization Schemes

The past few years have witnessed a significant growth in the usage and utility of P2P networks. These networks have grown from earlier basic ones like Napster and Gnutella to more recent and sophisticated ones like BitTorrent. P2P networks provide an effective mechanism for resource sharing, allowing sharing of memory and processing power. This allows workload distribution and load balancing making computing systems more efficient. P2P networks also provide advantages such as scalability and reliability that are important characteristics of distributed systems. They make information sharing easy and make information pools easily accessible to a large group of users.

P2P networks originally emerged to provide extensive file and information sharing. These systems undoubtedly met their requirements and introduced a new trend in information sharing making it easier and widely accessible to a large group of users. With the emergence of more sophisticated P2P networks, their capabilities were used to add a new dimension to resource sharing. Systems on these networks can now share resources like storage space and processing power, thus providing a higher performance at lower costs. A related advantage is workload distribution enabling parallel processing among systems on the network, thereby leading to a more efficient, robust and reliable computing system.

The lack of central administration in P2P networks ensures that there is no need for users to trust a single authority or depend on one for their operation. An emerging trend is the harnessing of advantages of P2P networks in the television domain, commonly called P2PTV. Software applications supporting P2PTV are designed to redistribute video streams or files on a P2P network. They allow users to watch the content as it is being downloaded, and each user, while downloading the content also uploads it to other interested peers thereby contributing to the overall available bandwidth. Such behavior provides significant savings in bandwidth while also making downloading and sharing of live content cost-effective and simpler. By allowing content to be downloaded by peers from peers, P2PTV networks reduce the load on the central content generating server. Due to the server workload reduction and the ability to share TV channels among peers, P2PTV makes TV channels available to a larger community of users at reasonable costs.

However, the design of P2PTV systems must take into account selfish nodes and their refusal to cooperate and share content. In the absence of a reliable content dissemination scheme, it is impossible for nodes to access any real-time content in these networks. We look to enhance the reliability of these networks using incentivization schemes to ensure that nodes stay on in the network for the period of time that they are involved in content transfer. This chapter discusses a round-based pricing adjustment scheme to provide incentives to users as the TV channel content is downloaded and viewed by increasing number of users. It explores a mechanism that maximizes the revenue of the P2PTV service provider while providing incentives to users and encouraging them to share content and download from a closely situated peer. An earlier version of this work has appeared in [42].

In this chapter, we use Monte-Carlo simulations to quantify the profits it brings to the network provider. We devise a simulator for the P2PTV network that allows for configurable number of users, TV programmes and pricing rounds. This simulator is used to arrive at the optimal number of pricing rounds and to determine the network provider's profit. By establishing the monetary advantage gained by users and the service provider, we implicitly prove that this scheme improves the reliability of these networks by ensuring uninterrupted content download atleast for the duration of a programme.

6.1 Experimental Setup

6.1.1 Modeling the Network

For our experimental setup, there are two main characteristics that would play an important role in deciding the benefits users and the service provider observes from offering P2PTV on a particular network. One characteristic is the density and the interconnectivity existing in the network, and the other is the link costing prevailing in the network. Networks that are denser in structure than others are likely to provide their users with choice of connecting to a variety of peers, and can be expected to serve as a better "breeding ground" for Peer-to-Peer applications like P2PTV. In our work, we estimate the density of a network through the ratio of the number of interconnections to the number of users. Link costing in viewing a P2PTV program has important connotations in terms of the demand the network would see for the P2PTV programs being served on it. Larger link costs would serve as a deterrent for users in subscribing to various programs, as many of them would find that the link costs plus content costs of the programs to be larger than their willingness-to-pay. In such cases, such users would altogether abstain from watching any P2PTV program. Link costs can have a variety of cost components and pricing schemes - some of them are the setup costs, operational or maintenance costs, pay-per-usage costs and pay-for-connected-time costs. Some networks like an Ethernet-based LAN would have high setup costs but low operational and pay-per-usage costs, while others like Dialup-based connections would have low setup costs but high pay-per-usage and pay-per-connected-time costs. In our simulations, we assume that the link cost between two nodes of the network is a function of the distance between them. In real-life the function can be expected to have a step-wise linear form similar to the one shown in Figure 6.1. Note that users have a choice in the type of network they use to connect to their peers even for a given physical distance between them. For shorter distances, networks like WiFi and Ethernet-based LANs may be used, while for larger distances Dialup and DSL based connections may be used.

Figure 6.1 can be explained as follows. For networks with high setup costs and low pay-per-usage costs, the link cost can be assumed to a constant number that is representative of the amortized cost of installing and operating that network. Once peers are on such networks, the link cost does not depend upon the distance between them and is denoted

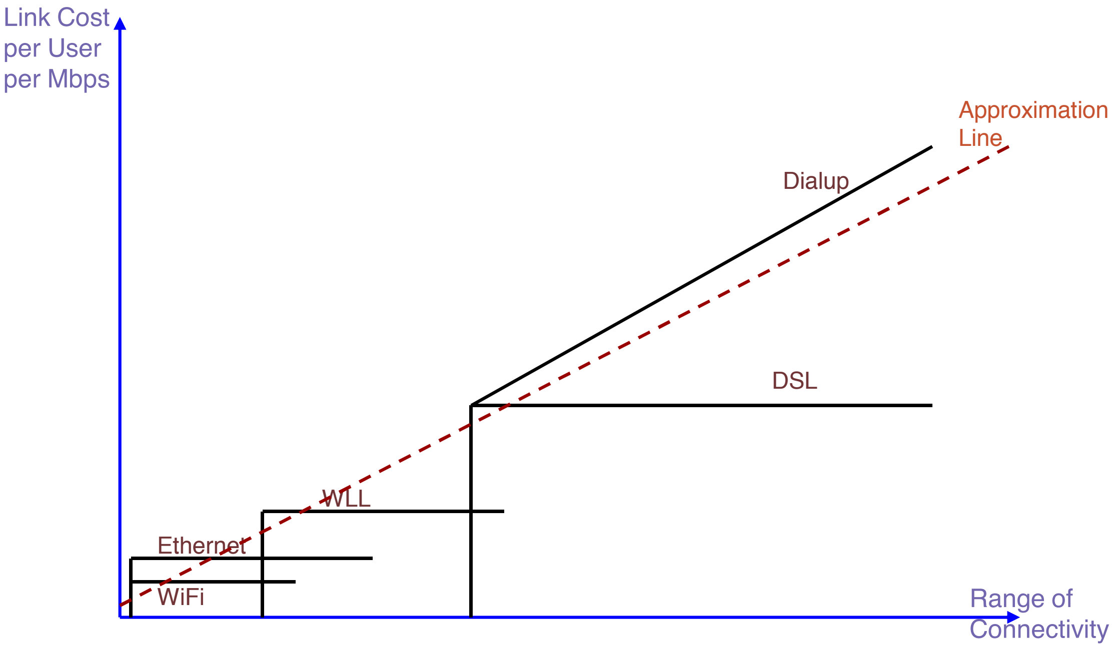

FIGURE 6.1: Characteristic Link Costs versus amount of data transferred for various types of networks and representative Approximation Line

by a constant value function like that depicted for Ethernet-based LANs. However, for networks with low setup costs and high pay-per-usage and pay-for-connected-time costs, the characteristics are different. For example, for Dialup based networks, the user pays for the length of time he is connected to the network. In such cases, as the distance between peers increases, the speed of connectivity between them reduces and the transfer of same amount of traffic takes longer thereby making link costs vary with time. This is shown in Figure 6.1 for the link cost function of Dialup connections. However, with many network connectivity options being available for a given pair of peers separated by some distance, and with different characteristics of each of these connectivity options, the combined overlapping behavior is complicated and makes it difficult for us to comprehend the implications of our results. Hence, we approximate the combined overlapping link cost function of various networks by a straight line as shown in Figure 6.1.

6.1.2 Modeling the Users

While simulating users, we assume that their willingness-to-pay is derived from a common, yet hidden mathematical model. In usual circumstances where users are not incentivized to reveal their willingness-to-pay (or utility from watching a P2PTV program), the service provider tries to estimate this hidden model of true demand, and prices the service accordingly. Such estimation is typically carried out over a period of time, where the service provider learns the right pricing through trial and error in the marketplace. Note that such experimentation with pricing in the marketplace through trial and error is costly, as it loses the service provider important revenue opportunities.

In a particular simulation run of the model, we can assume that the willingness-to-pay for each of the users is either systematically drawn from the underlying hidden model of true demand, or is a random draw from it. A set of systematically drawn willingness-to-pay numbers would look like the staircase approximation shown in Figures 6.2-6.5. Note that such systematic draws are based on step functions indicating the prices at which the demand increases by one unit as per the true demand function. On the other hand, the set of random draws would have many spikes in them, and would look similar to the underlying true demand model only when the number of users, and hence the number of draws is large in the statistical sense. The shape of the observed willingness-to-pay demand function with random draws with 10, 30, 50 and 100 draws is shown in Figures 6.2-6.5 respectively.

Note that while the stair-case approximation correctly depicts the true underlying demand function, and would allow us to compute the true revenue generated, it increases the computational complexity in calculating the various user scenarios, as depicted in Figure 6.6. Figures 6.2-6.5 denote the error generated in estimating a one-dimensional demand function with stair-case approximation and random estimation. Note that the number of dimensions indicates the number of alternative programs that are available for the users to view at a particular instant of time. Figure 6.6 denotes the number of cases that need to be evaluated in generating the stair-case approximation of a two-dimensional demand function.

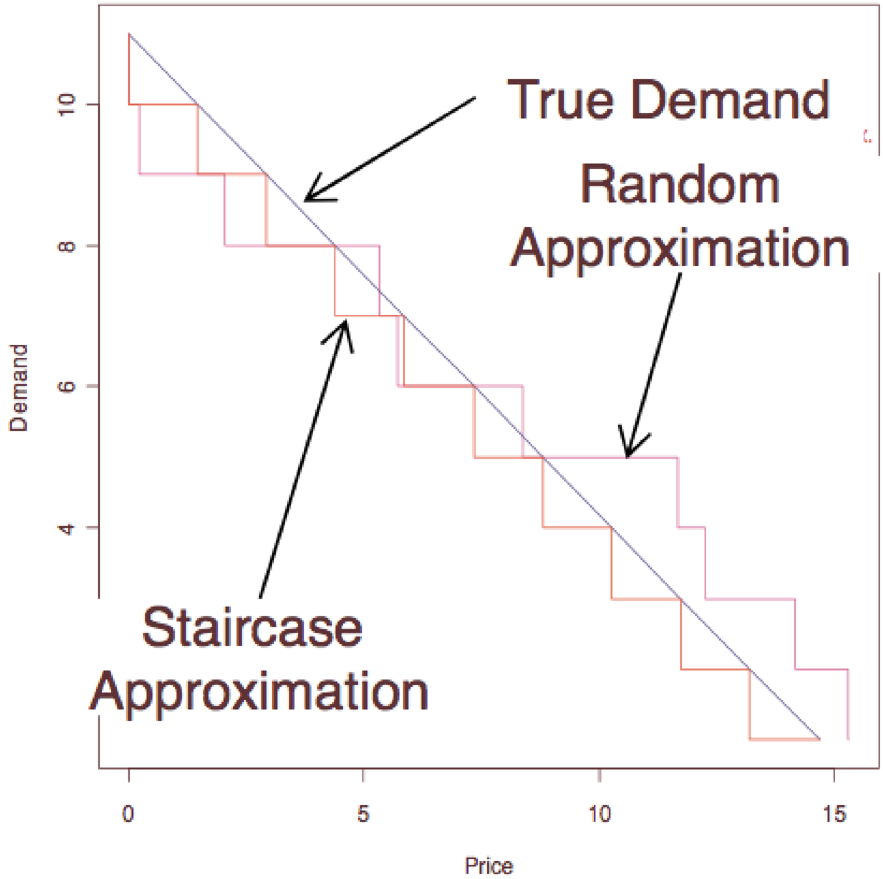

Figure 6.2: Estimation of true demand function through stair-case approximation and random draws for a single-dimensional demand with 10 samples

6.1.3 Modeling Content Requests

In this section, we develop the model for characterizing and simulating the content requests. Let us assume that there are n programs being telecast simultaneously, and that each of the users makes the choice of watching one and at most one of these programs at a given time. A user's choice of program depends upon his utility from watching the program (thereby leading to his willingness-to-pay), and the network and the content prices he has to pay for watching the program. Note that in our models, the service provider does not do any differential pricing of the content to individual users. However, the network cost being paid by a user would typically depend on his P2P network behavior, with users depicting cooperative behavior being incentivized appropriately. The amount of incentives given to individual users would depend on the incentivization mechanism being adopted by the service provider.

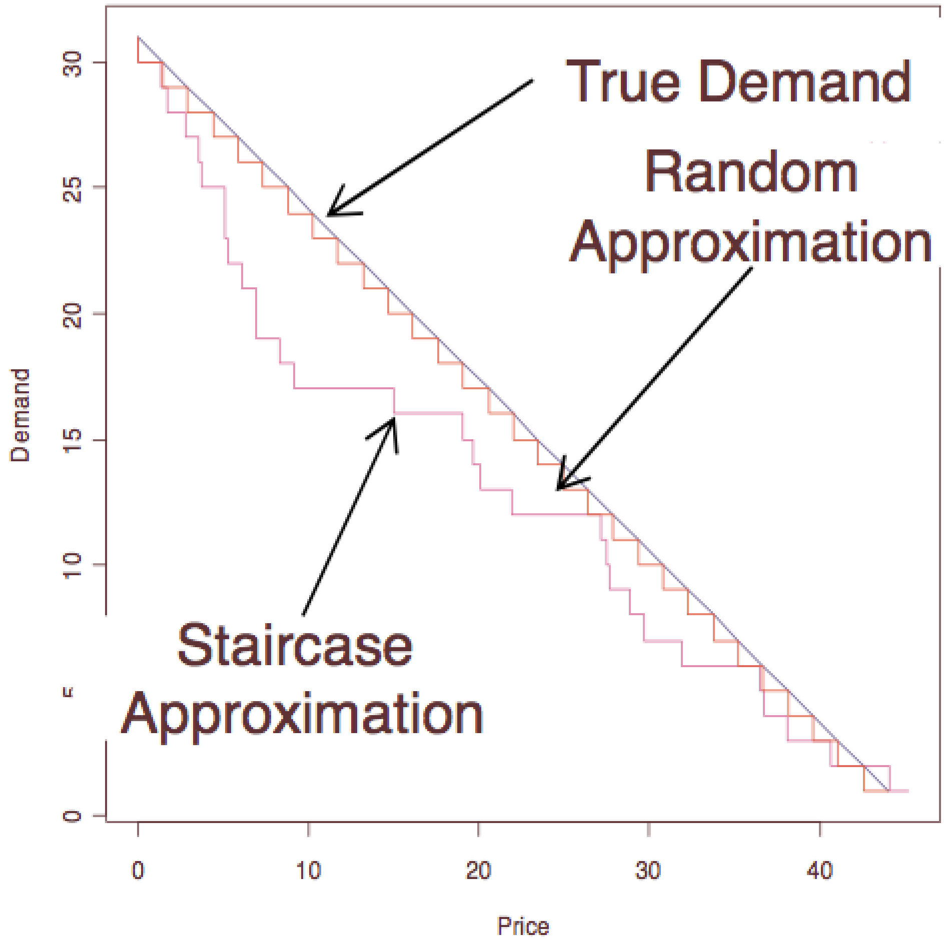

FIGURE 6.3: Estimation of true demand function through stair-case approximation and random draws for a single-dimensional demand with 30 samples

Let us assume further that the maximum demand to be expected of each of these n programs can be denoted by $D_1, D_2 \dots D_n$ respectively. The service provider in practice can obtain these values by conducting market surveys. Let $d_1, d_2, \dots d_n$ denote the actual demand for each of the programs when the content prices for each of them are $p_1, p_2, \dots p_n$ respectively. Note that the amount of viewership for a program not only depends upon its own price, but also on the price of all other programs. This indicates that we can expect to have a certain degree of correlation and cross-dependencies between the demands for various programs. We use Equation (6.1) to model the overall demand function of the various programs offered by the service provider, where $\mathcal{D}$ denotes the demand vector for all programs, $\mathcal{P}$ denotes the current pricing vector set by the service provider and $\mathcal{H}$ denotes the matrix of self and cross-elasticities associated with the

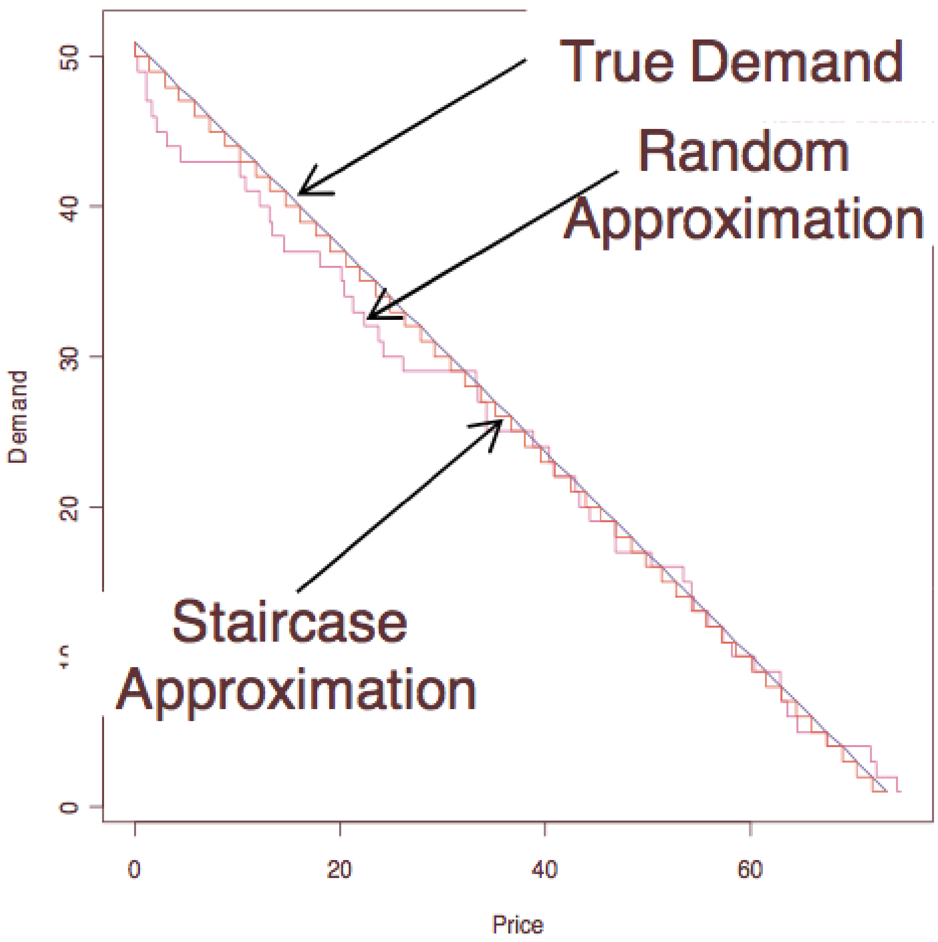

FIGURE 6.4: Estimation of true demand function through stair-case approximation and random draws for a single-dimensional demand with 50 samples

demand functions.

$$\mathcal{D} = \mathcal{H} \times \mathcal{P} \tag{6.1}$$

The service provider strives to learn the unknown $\mathcal{H}$ by changing the pricing vector $\mathcal{P}$ and observing the resultant demand vector $\mathcal{D}$. For a given set of M users in the system at a given point in time, following the random draws method of the previous section, the individual demands for each of the programs can be simulated as

$$d_1 \sim U[0, \min(M, D_1)] \tag{6.2}$$

$$d_2 \sim U\left[0, \max\left(0, \min\left(M - d_1, D_2\right)\right)\right] \tag{6.3}$$

$$\vdots \tag{6.4}$$

$$d_n \sim U\left[0, \max\left(0, \min\left(M - \sum_{i=1}^{n-1} d_i, D_n\right)\right)\right] \tag{6.5}$$

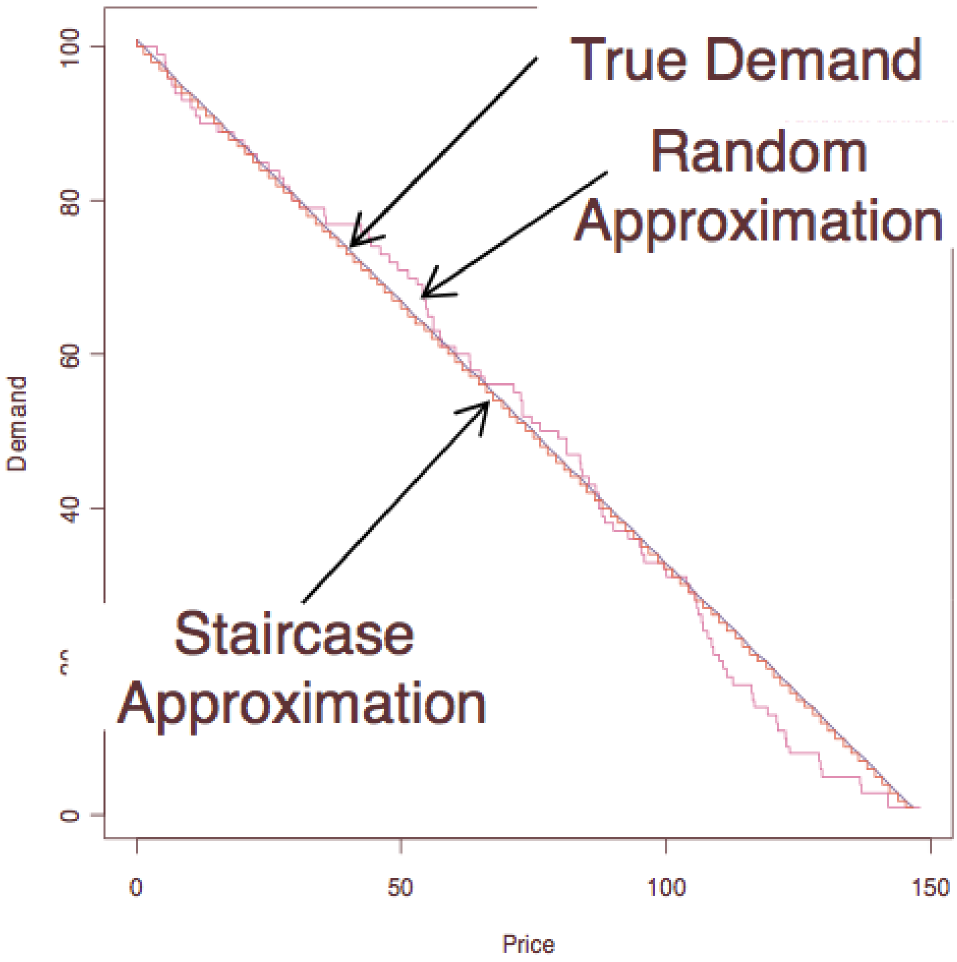

FIGURE 6.5: Estimation of true demand function through stair-case approximation and random draws for a single-dimensional demand with 100 samples

where $U[a, b]$ denotes a random draw from a uniform distribution on the range $[a, b]$. As far as the simulation is concerned, $\mathcal{H}$ can be assumed to be known and the prices corresponding to a sample set of individual demands can be obtained by inverting Equation (6.1).

6.1.4 Pricing Schemes and Learning Prices

As a base case for our experiments, we consider that the users are offered incentives based on the number of programs they serve to their peers. Such a scheme prompts serving users to serve whole programs, rather than break in the middle. We compare the revenue of the service provider who provides such an incentive scheme with that of a service provider relying explicitly on unicast transmissions from its servers. Also,

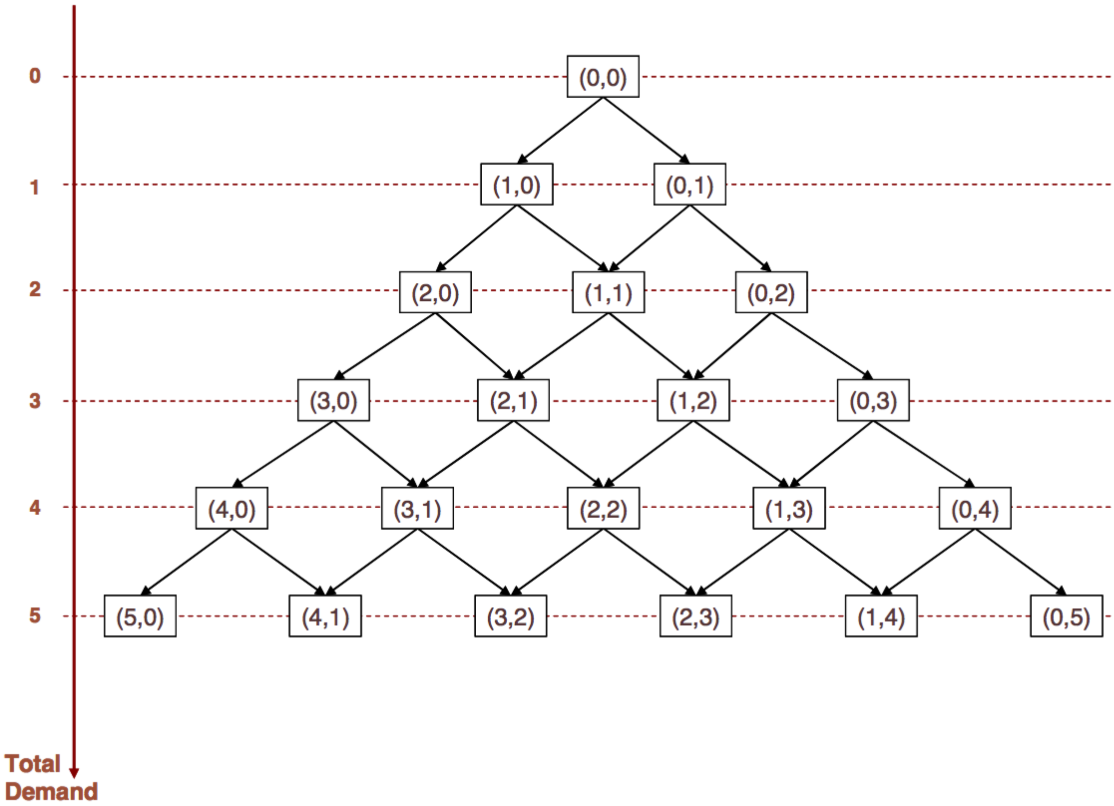

FIGURE 6.6: Computational Complexity associated with discrete approximation

we choose an incentive scheme based on the number of programs served, rather than a revenue-sharing based incentive model. Under a revenue-sharing model, there would be a tendency on the part of serving users to prefer serving high-value content to low-value content. Such a tendency would make serving the low-value content a high-cost venture for the service provider.

Incentive programs can also be characterized in terms of the times at which they require users to declare their intent. For our base case, we consider real-time sharing incentivization, where users declare their content requests in real-time, and the interested user servers declare their intent-to-share also in real-time. In contrast, in a know-ahead incentivization scheme, the users are the user servers declare their content requests and intents-to-share in advance. Such advance declaration of content requests and intents allows the service provider to exploit further revenue generating opportunities in the network. Any such revenue gains can be shared with users to create a win-win situation for both the users and the service provider.

Price discovery under uncertain demand is a difficult problem to solve [124, 125]. In this chapter, we use a simple directional descent based learning mechanism for price discovery. The process generating the price discovery can be described as: Users arrive randomly at the service providers "site", and check the current prices of all programs slotted for a particular time-slot. They choose the program that gives them the largest difference between their willingness-to-pay and the price being requested. As soon as a user selects a particular program, the service provider compares its current demand with its target popularity and adjusts the prices accordingly. Also, as the users expect prices to change, they come back to the site at a later point in time to re-evaluate their decisions. Each time a user comes back to re-evaluate his decision before the start time of the programs is termed as a round. For the learning mechanism, the service provider makes small-step price corrections within rounds, and large-step corrections between rounds.

6.2 Results

For the setup of the simulation experiments described, we carried out our experiments with 60 users and 15 programs. We first assumed that the users would re-visit the service provider's site for pricing 2-3 times, and hence varied the number of rounds for our experiments between 2 and 3. The willingness-to-pay function of various users was simulated. We then calculated the revenues collected by the service provider for the cases where he runs his network as a P2PTV network, and find the percentage gains with respect to the revenues he would have generated by running it as a unicast network. Figure 6.7 plots the histogram denoting the percentage benefits accrued by the service provider by running his network as a P2PTV network, as against a unicast network, and following the base case incentive policy described in the previous section.

In a real-world setup, a service provider would try to influence user behavior in terms of the number of times they re-visit the site. We also looked at the potential revenue benefits the service provider can expect by influencing such behavior. Table 6.1 tabulates the average profit and the associated standard deviation observed by the service provider in P2PTV over and above the unicast network case when the number of rounds are varied. Figure 6.8 depicts these results in the form of a graph. We notice that for a value of number of rounds = 5, the service provider observes the largest average revenue gain, with the smallest deviation around it.

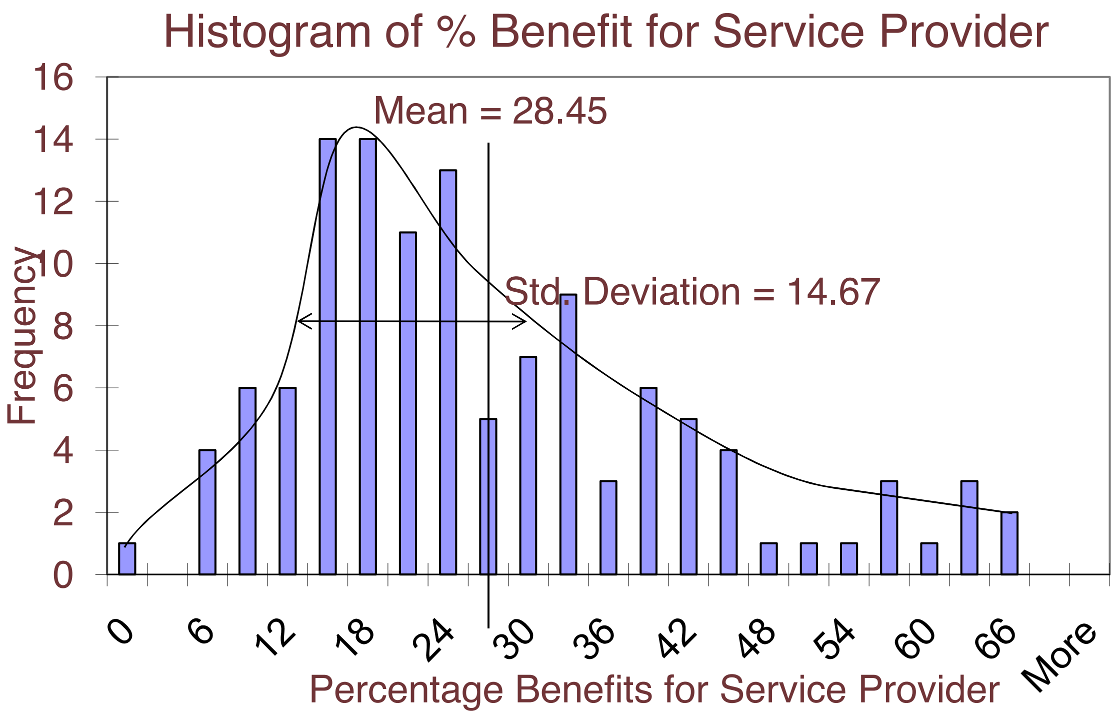

FIGURE 6.7: Percentage improvements in benefits in a P2PTV network over a unicast network

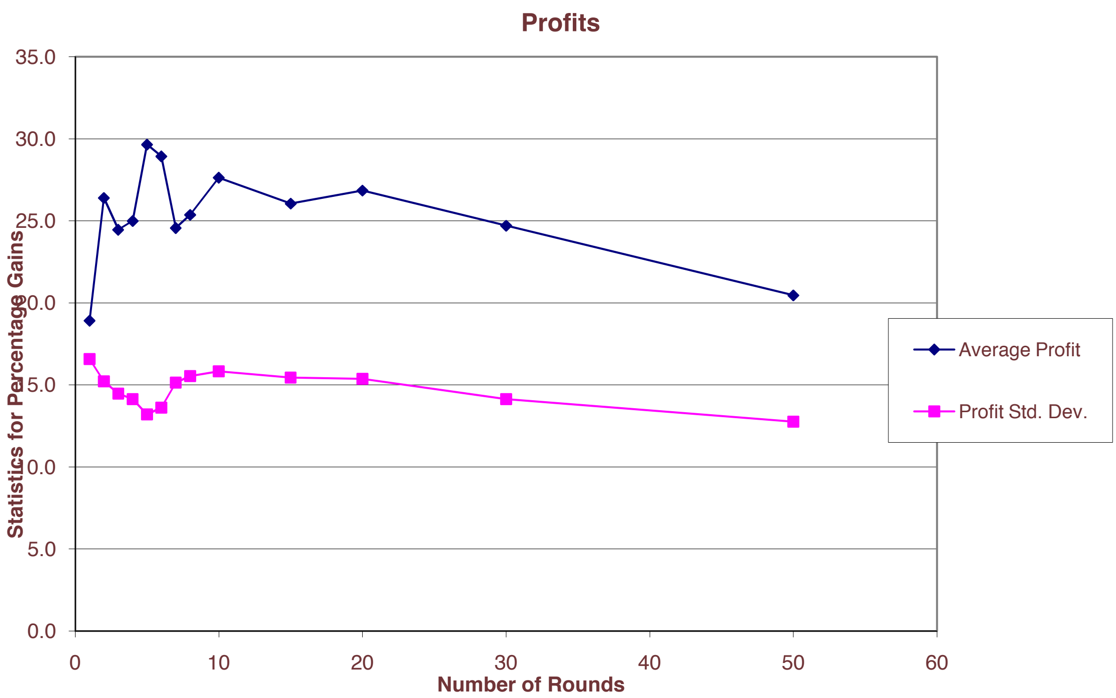

FIGURE 6.8: Service provider's average profit and associated standard deviation versus the number of rounds

TABLE 6.1: Service provider's average profit percentage over and above unicast network profits and associated standard deviations with variations in number of rounds

Rounds	**Average Profit**	**Profit Std. Dev.**
1	18.9	16.6
2	26.4	15.2
3	24.5	14.5
4	25.0	14.1
5	29.7	13.2
6	28.9	13.6
7	24.6	15.1
8	25.4	15.5
10	27.6	15.8
15	26.1	15.4
20	26.8	15.4
30	24.7	14.1
50	20.5	12.8

6.3 Conclusions

Pricing schemes and incentivization policies are an important instrument in the hands of a P2PTV network designer. Incentivization policies are necessary to ensure that the nodes stay co-operative at all times. Pricing schemes augment these policies to ensure that co-operating nodes and the service provide gain from the overall setup of the P2PTV network. In this chapter, we looked at an incentivization policy where users are given incentives based on the number of programmes that they share. This policy is augmented by a programme pricing policy where the prices of the programmes change based on the demand for these programmes. Since, the prices change with each request, users are allowed to revisit the programme offering before the scheduled time and re-consider their decisions. Each such "revisit" is called a round in our experiments.

In order to evaluate the scheme, a simulator was devised that allowed a configurable number of users, TV programmes and pricing rounds. Simulations were run using 60 users, 15 programmes and 2-3 rounds. With each simulation run, the revenue of the service provider is computed and compared to the revenue that would have been generated, had all these content been delivered via a unicast network. Through these simulations, we have proved that using this pricing and incentivization scheme, the network designer can provoke users to display socially favorable behavior, and earn as much as 30% more revenue on average than the case where he tries and builds a unicast network. In doing this, the users self-configure themselves to form a topology favorable to maximal users and also ensure that the network is more stable than it would be if the users did not apply this scheme. Users are more likely to stay in the network throughout the duration of a program rather than leave and re-join several times, thereby maintaining the stability of the network.

In addition, experiments were run to determine the optimal number of rounds over which the prices can be adjusted followed by users re-visiting their choices. As can be noticed in Figure 6.8, the network designer should try and influence users to re-visit the prices on an average 5 times for maximum revenue gains. This can be carried out by providing ample time between when the pricing activity starts and the scheduled start time of the programs. Having 5 rounds of price adjustments allow the service provider to maximize his average profit and minimize the standard deviation at the same time.

The next step in this area of work would be to look at dynamic models for user characterization in P2PTV networks. In P2PTV networks, users can join and leave the network at any time, and thus dynamic model would better characterize user behavior as compared to the static model used in this chapter. Secondly, tuning the incentivization policy by experimentation with the option of pre-declaring the intent to share may bring the setup closer to a real-world scenario where users do not change their programme choices till the very last minute based solely on monetary conditions. Finally, using advanced price discovery schemes to adjust the price of programmes will have to be explored to determine if these can perform better than the simple directional descent based learning mechanism proposed in this chapter.

.

Chapter 7

Conclusions

In this thesis, we have developed schemes for self-optimization and self-healing in wireless and overlay networks. In particular, we have looked at two main problems:

1. The problem of the loss of TCP throughput experienced in wireless environments as a side effect of transient disconnections of users
2. The problem of self-configuration of P2PTV overlay networks in a fashion that encourages users to participate in content sharing while providing substantial revenues to the service provider.

We noted that the TCP protocol, which is widely used across all networks can seriously impact the efficiency of wireless networks because of its inherent nature of treating all packet losses as indicators of network congestion. Since, this is not true for wireless networks in most cases of TCP packet loss, we developed a scheme that through mobile disconnection prediction and subsequent packet caching at the base station prevents TCP on the fixed host from triggering its slow start mechanism and hence, prevents a degradation in throughput due to transient wireless conditions. We implemented this scheme in ns-2 and tested it by simulating wireless link disruptions. With this scheme, we saw that the throughput improved by almost 7.5 times in the face of transient disconnections. We also proved that this scheme is not greatly impacted by the accuracy of the disconnection prediction. As long as an inaccurate algorithm provides over-estimates rather than under-estimates, the TCP ACK pacing scheme will work reasonably well.

We then moved on to find suitable disconnection predictors that can be implemented in the base station in order to complete the TCP ACK Pacing scheme. In Chapter 4 we used the Gauss-Markov mobility model and a modified version of it to derive a disconnection predictor that could be implemented in the base station to complete the TCP ACK Pacing scheme. The equation derived from the Gauss Markov mobility model was found to be too complex to implement on base stations, where processing resources are at a premium and there are stiff real-time constraints for all processing. The modified Gauss Markov mobility model led to a simpler and more practical equation for usage in disconnection prediction. In the second part of this chapter, we developed a mobility model based on metropolitan city traffic conditions. We then developed the disconnection predictor based on this model and found that this also led to a practical, implement-able equation for predicting disconnection. For each of these models, we mathematically derived the expression for disconnection duration prediction.

In Chapter 5, we looked at machine learning techniques to predict disconnection. For disconnection prediction to succeed in the real-world, it is important to make any assumptions about network conditions or user behavior or the schemes will have to be reworked several times as technology advances. The problem can be overcome by using machine learning techniques. After the initial training set is used to train the particular model, the model changes its parameters based on its observations of the real world, and as such does not rely on any particular mobility model. We looked at different techniques such as artificial neural networks, RBF networks and Random Forests. We modified different parameters of each model to find the one that gives us the best accuracy. We tested these models by using data generated from different mobility models such as Random Walk and Random Direction and found that the RBF network provided the best accuracy amongst all other predictors. In addition, theoretical studies show that RBF networks are the least performance intensive and hence, using these as predictors in the base stations seems a practical solution.

In Chapter 6, we looked at the prospect of constructing a self-optimizing and self-configuring overlay network such as a P2PTV network. We identified that the main problem in such networks is ensuring co-operation between nodes forming the network. These networks are hugely vulnerable to selfish nodes and there can be severe degradation brought about by even a small number of selfish nodes in such networks. We thus came up with a scheme that incentivizes the nodes to encourage cooperation and in the process leads to great benefits for the network operator as well. In order to experiment and quantify the revenue gains, we developed a simulator of a P2PTV network, that allowed for a configurable number of users, programmes and pricing rounds. We proved through experiments that the network operator can earn as much as 30% more revenue on average by implementing this scheme.

In summary, this research looked at making wireless networks self-managing by proposing solutions that allow the network to:

1. Detect a pending disconnection using mobility models and machine learning techniques. The use of machine learning makes the network completely self-sufficient and adaptive to whatever mobility pattern it may observe.

2. Control degradation in throughput observed by the user due to transient network conditions. The proposed TCP ACK Pacing mechanism uses inputs from the disconnection predictor to pace out the ACKs and to make the sender transparent to the wireless network condition. With this scheme, simulations showed a 7.5 times improvement in throughput observed by the user during degraded signal conditions as compared to when the ACK Pacing was not done.

3. Self-configure and self-optimize by solving the problem of non co-operative participants in overlay networks like P2PTV networks. The proposed incentivization scheme encourages users to not leave the network in the middle of content dissemination thereby, ensuring a stable network. Experiments show that implementing this scheme allows the network operator to earn as much as 30% more revenue on average.

7.1 Key Contributions of the Thesis

The key contributions of this thesis to the area of self-managing networks are the following:

1. A detailed review of the self-management schemes proposed for wireless networks over the past several years, together with an analysis of the schemes proposed to counteract the problem of wireless throughput degradation due to inter-operation of TCP and wireless networks.

2. Development of a TCP ACK Pacing scheme that serves to "heal" the after-effects of a transient wireless network condition, by restoring the throughput experienced by the user as quickly as possible. In contrast to most other schemes proposed in the literature, this scheme is proactive and serves to detect the problem in advance instead of reacting once the disconnection actually occurs. This ensures that the chances of a degradation are further reduced to minimum as compared to other schemes under which some degradation will still be observed during the time it takes it to react to the problem. This scheme provides for the self-healing aspect of wireless networks.

3. Development of a new mobility model - the modified Gauss-Markov mobility model. Currently existing mobility models in literature model mobility by discretizing the time dimension, and measuring variable location jumps the mobile makes in these equi-sized time slots. However, the modified Gauss-Morkov mobility model models mobility by discretizing locations, and measures variable time durations needed for transitioning amongst those locations.

4. Development of schemes based on mobility models to predict that the user would be out of signal range allowing measures for it to be taken in advance. This allows the optimization of the operating parameters for that user and reduces data loss to a minimum.

5. Development of schemes based on machine learning techniques to predict disconnection. The use of machine learning makes the network completely self-sufficient and adaptive to whatever mobility pattern it may observe.

6. Development of an incentivization scheme that allows for self-configuration and self-optimization of overlay networks such as P2PTV networks. The proposed incentivization scheme encourages users to not leave the network in the middle of content dissemination thereby, ensuring a stable network. Experiments show that implementing this scheme allows the network operator to earn a higher revenue, as well as ensure a higher quality of user experience.

7.2 Scope for Further Research

We have identified the following areas for further research:

1. Disconnection prediction algorithms based on real-world tests - By subjecting the different disconnection predictor equations to real-world tests, we need to determine if implementing the equations in base stations will indeed yield sufficient accuracy for us or we would need to base the equations on a combination of mobility models.

2. User classification for city-section model parameter estimation - Classifying users based on parameters like the mode of transport and so on, can lead to a better parameter estimation of the city-section model. The model right now expects users to exhibit one homogeneous behavior which is not practical in the real-world.

3. Dynamic models for user characterization in P2PTV networks - In Chapter 6, we developed the entire scheme based on static models of the network. However, in P2PTV networks, with users leaving and re-joining the network, dynamic models are more effective and may prove to be more accurate for further P2PTV studies.

4. Chapter 6 discussed a simple directional descent based learning mechanism for pricing adjustments. Usage of advanced price discovery schemes to adjust the programme prices will have to be explored to check if it gives better results than the current scheme.

Bibliography

[1] Salehie, M. and Tahvildari, L. Autonomic computing: Emerging trends and open problems. In *Proceedings of the 2005 Workshop on Design and Evolution of Autonomic Application Software*, pages 1–7, St. Louis, Missouri, USA, May 2005. ACM.

[2] Aliu, O. G., Imran, A., Imran, M. A., and Evans, B. A survey of self organisation in future cellular networks. *IEEE Communications Surveys and Tutorials*, 15(1): 336–361, 2013.

[3] Karnik, A. and Kumar, A. Distributed optimal self-organization in ad-hoc wireless sensor networks. *IEEE Transactions on Networks*, 15(5):1035–1045, Oct 2007.

[4] Sohrabi, K., Gao, J., Ailawadhi, V., and Pottie, G. J. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7 (5):16–27, 2000.

[5] Singh, S., Woo, M., and Raghavendra, C. S. Power-aware routing in mobile ad hoc networks. In *Proceedings of the 4th annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 181–190, Dallas, Texas, USA, Oct 1998. ACM.

[6] Subramanian, L. and H.Katz, R. An architecture for building self-configurable systems. In *Proceedings of the 1st Annual Workshop on Mobile and Ad-Hoc Networking and Computing*, pages 63–73, Boston, Massachusetts, USA, Aug 2000. IEEE.

[7] Ganguly, A., Agrawal, A., Boykin, P. O., and Figueiredo, R. IP over P2P: Enabling self-configuring virtual IP networks for grid computing. In *Proceedings of the*

20th International Parallel and Distributed Processing Symposium, Rhodes Island, Greece, Apr 2006. IEEE.

[8] Claussen, H., Ho, L. T. W., Karimi, H. R., Mullany, F. J., and Samuel, L. G. I, base station: Cognisant robots and future wireless access networks. In *Proceedings of the 3rd IEEE Consumer Communications and Networking Conference*, volume 1, pages 595–599, Las Vegas, Nevada, USA, Jan 2006. IEEE.

[9] Amirijoo, M., Jorguseski, L., Litjens, R., and Nascimento, R. do. Effectiveness of cell outage compensation in LTE networks. In *Proceedings of IEEE Consumer Communication and Networking Conference*, Las Vegas, Nevada, USA, Jan 2011.

[10] Viering, I., Dottling, M., and Lobinger, A. A mathematical perspective of self-optimizing wireless networks. In *Proceedings of the 2009 IEEE Conference on Communications*, pages 4320–4325, Dresden, Germany, Jun 2009.

[11] Wei, Y. and Peng, M. A mobility load optimization method for hybrid architecture in self-organizing networks. In *Proceedings of International Conference on Communication Technology and Application*, Beijing, China, Oct 2011.

[12] Lobinger, A., Stefanski, S., Jansen, T., and Balan, I. Load balancing in downlink LTE self-optimizing networks. In *Proceedings of the 71st Vehicular Technology Conference*, pages 1–5, Taipei, May 2010.

[13] Coucheney, P., Gaujal, B., and Touati, C. Self-optimizing routing in MANETs with multi-class flows. In *Proceedings of the 21st International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, pages 2751–2756, Istanbul, Turkey, Sep 2010.

[14] Saleh, A. M. S., Ali, B. H., Rashid, M. F. A., and Ismail, A. A self-optimizing scheme for energy balanced routing in wireless sensor networks using sensor ant. *Sensors*, 12:11307–11333, 2012.

[15] Soldani, D., Alford, G., Parodi, F., and Kylvaja, M. An autonomic framework for self-optimizing next generation mobile networks. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6, Espoo, Finland, Jun 2007. doi: 10.1109/WOWMOM.2007.4351738.

[16] Feldman, M., Papadimitriou, C., Stoica, I., and Chuang, J. Free-riding and whitewashing in peer-to-peer systems. In *Proceedings of the ACM SIGCOMM workshop on Practice and Theory of Incentives and Game Theory in Networked Systems*, pages 228–236, Portland, Oregon, USA, Sep 2004. ACM.

[17] Ma, R. T. B., Lee, S. M., Lui, J. C. S., and Yau, D. K. Y. An incentive mechanism for P2P networks. In *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)*. IEEE, 2004.

[18] Wierzbicki, A. Peer-to-peer direct sales. In *Proceedings of the Fifth IEEE International Conference on Peer-to-Peer Computing (P2P'05)*, 2005.

[19] Wongrujira, K., Hsin-Ting, T., and Seneviratne, A. Incentive service model for p2p. In *Computer Systems and Applications*, 2005.

[20] Camerer, C. F. *Behavioral Game Theory*. Princeton University Press, 2003.

[21] Golle, P., Leyton-Brown, K., Mironov, I., and Lillibridge, M. Incentives for sharing in peer-to-peer networks. In *Proceedings of the 3rd ACM Conference on Electronic Commerce*, volume 2232, pages 75–87, Tampa, Florida, USA, October 2001. Springer-Verlag.

[22] Andrade, N., Mowbray, M., Lima, A., Wagner, G., and Ripeanu, M. Influences on cooperation in bittorrent communities. In *Proceedings of the ACM SIGCOMM workshop on the Economics of Peer-to-Peer Systems*, pages 111–115, Philadelphia, USA, Aug 2005. ACM.

[23] Cohen, B. Incentives build robustness in bittorrent. In *Workshop on Economics of Peer-to-Peer Systems*, 2003.

[24] Huang, G., Hong, P., and Li, J. Optimal server selection and pricing in receiver-driven p2p streaming systems. In *ICON*, 2006.

[25] Gupta, R. and Somani, A. K. A pricing strategy for incentivizing selfish nodes to share resources in peer-to-peer (p2p) networks. In *ICON*, 2004.

[26] Moscibroda, T., Schmid, S., and Wattenhofer, R. On the topologies formed by selfish peers. In *ACM Symposium on Principles of Distributed Computing*, pages 133–142, 2006.

[27] Kant, L. and Chen, W. Service survivability in wireless networks via multi-layer self-healing. In *IEEE Wireless Communications and Networking Conference*, volume 4, pages 2446–2452, New Orleans, Louisiana, USA, Mar 2005. IEEE.

[28] Psaier, H. and Dustdar, S. A survey on self-healing systems: approaches and systems. *Computing*, 91(1):43–73, Jan 2011.

[29] Agoulmine, N., Balasubramaniam, S., Botvitch, D., Strassner, J., Lehtihet, E., and Donnelly, W. Challenges for autonomic network management. In *Proceedings of 1st Conference on Modeling Autonomic Communication Environment (MACE)*, Dublin, Ireland, Oct 2006.

[30] Aly, S. A., Kamal, A. E., and Al-Kofahi, O. M. Network protection codes: Providing self-healing in autonomic networks using network coding. *Computer Networks*, 56:99–111, 2012.

[31] Conti, M., Pietro, R. D., Mancini, L. V., and Mei, A. Distributed detection of clone attacks in wireless sensor networks. *IEEE Transactions on Dependable and Secure Computing*, 8(5):685–698, 2011.

[32] Tian, B., Han, S., Parvin, S., Hu, J., and Das, S. Self-healing key distribution schemes for wireless networks: A survey. *The Computer Journal*, 54‘(4):549–569, 2011.

[33] Wang, Q. Practicality analysis of the self-healing group key distribution schemes for resource-constricted wireless sensor networks. In *Proceedings of International Conference on Communications and Mobile Computing*, pages 37–40, Qingdao, China, Apr 2011.

[34] Stevens, W. R. *TCP/IP Illustrated*. Addison-Wesley, 1994.

[35] Allman, M., Paxson, V., and Planton, E. Tcp congestion control. RFC 5681, Available from: http://www.ietf.org/rfc/rfc5681.txt, 2009. The Internet Society.

[36] Balakrishnan, H., Seshan, S., Amir, E., and Katz, R. H. Improving TCP/IP performance over wireless networks. In *Proceedings of ACM Conference on Mobile Computing and Networking (MOBICOM)*, pages 2–11, Berkeley, California, USA, Nov 1995. ACM.

[37] Vaidya, N., Mehta, M., Perkins, C., and Montenegro, G. Delayed duplicate acknowledgements: A TCP-unaware approach to improve performance of TCP over wireless. *Wireless Communications and Mobile Computing*, 2(1):59–70, 2002.

[38] Mathis, M., Mahdavi, J., Floyd, S., and Romanow, A. TCP selective acknowledgement options. RFC 2018, Available from: http://www.ietf.org/rfc/rfc2018.txt, 1996. The Internet Society.

[39] Buchholcz, G., Gricser, A., Ziegler, T., and Do, T. V. Explicit loss notification to improve TCP performance over wireless networks. *Lecture Notes in Computer Science*, 2720:481–492, 2003. Springer, Berlin.

[40] Goff, T., Moronski, J., Phatak, D., and Gupta, V. Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments. In *Proceedings of IEEE Conference on Computer Communications INFOCOM*, volume 3, pages 1537–1545, Tel-Aviv, Israel, Mar 2000.

[41] Bakre, A. and Badrinath, B. R. I-TCP: Indirect TCP for mobile hosts. In *Proceedings of 15th International Conference on Distributed Computing Systems (ICDCS)*, pages 136–143, Vancouver, Canada, May 1995.

[42] Bhutani, G. A round-based pricing scheme for maximizing service provider's revenue in p2ptv networks. In *Proceedings of the 3rd International Conference on Internet Multimedia Services Architecture and Applications (IMSAA)*, pages 112–117, Bangalore, India, Dec 2009. IEEE.

[43] ns-2. http://nsnam.isi.edu/nsnam/index.php/Main_Page, 2011. Downloaded in Nov, 2011.

[44] Henderson, T., Floyd, S., Gurtov, A., and Nishida, Y. The newreno modification to tcp's fast recovery algorithm. RFC 6582, Available from: http://www.ietf.org/rfc/rfc6582.txt, 2012. The Internet Society.

[45] TCP westwood. http://en.wikipedia.org/wiki/TCP_Westwood, 2013. Downloaded in Dec, 2013.

[46] Weka. http://sourceforge.net/projects/weka, 2014. Downloaded in Jun, 2014.

[47] Singh, R., Singh, D. K., and Kumar, L. Performance evaluation of DSR and DSDV routing protocols for wireless ad hoc networks. *International Journal of Advanced Networking and Applications*, 2(4):732–737, 2011.

[48] Jeon, J., Lee, K., and Kim, C. Fast route recovery scheme for mobile ad-hoc networks. In *Proceedings of the International Conference on Information Networking*, pages 419–423, Barcelona, Spain, Jan 2011.

[49] Santi, P. Topology control in wireless ad-hoc and sensor networks. *Journal of ACM Computing Surveys*, 37(2):164–194, Jun 2005.

[50] Cerpa, A. and Estrin, D. ASCENT: Adaptive self-configuring sensor networks topologies. *IEEE Transactions on Mobile Computing*, 3(3):272–285, 2004.

[51] Malatras, A., Pavlou, G., Gouveris, S., and Sivavakeesar, S. Self-configuring and optimizing mobile ad-hoc networks. In *Proceedings of the 2nd International Conference on Autonomic Computing*, pages 372–373, Seattle, Washington, USA, Jun 2005. IEEE.

[52] Lehtihet, E., Derbel, H., Agoulmine, N., Ghamri-Doudane, Y., and Meer, S. van der. Initial approach toward self-configuration and self-optimization in IP networks. In Dalmau, J. and Hasegawa, G., editors, *Proceedings of the 8th International Conference on Management of Multimedia Networks and Services*, volume 3754 of *Lecture Notes in Computer Science*, pages 371–382. Springer-Verlag, 2005.

[53] Claussen, H. Autonomous self-deployment of wireless access networks in an airport environment. In *Proceedings of the 2nd International IFIP Conference on Autonomic Communication*, pages 86–98, Vouliagmeni-Athens, Greece, Oct 2005. Springer-Verlag.

[54] Huang, S. Ion-6: a positionless self-deploying method for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2011.

[55] Huang, S. Energy-aware, self-deploying approaches for wireless sensor networks. In *24th IEEE International Conference on Advanced Information Networking and Applications(AINA)*, pages 896–901, Perth, April 2010.

[56] Jansen, T., Balan, I., Turk, J., Moerman, I., and Kurner, T. Handover parameter optimization in LTE self-organizing networks. In *Proceedings of the 72nd Vehicular Technology Conference*, pages 1–5, Ottawa, Ontorio, Canada, Sep 2010.

[57] Lobinger, A., Stefanski, S., Jansen, T., and Balan, I. Coordinating handover parameter optimization and load balancing in LTE self-optimizing networks. In *Proceedings of the 73rd Vehicular Technology Conference*, pages 1–5, Yokohama, Japan, May 2011.

[58] Nihtila, T., Turkka, J., and Ingo, . Performance of LTE self-optimizing networks uplink load balancing. In *Proceedings of the 73rd Vehicular Technology Conference*, Yokohama, Japan, May 2011.

[59] Mullner, R., Ball, C. F., Boussif, M., Lienhart, J., Hric, P., Winkler, H., Kremnitzer, K., and Kronlachner, R. Enhancing upling performance in UTRAN LTE networks by load adaptive power control. *European Transactions on Telecommunications*, 21(5):458–468, Aug 2010.

[60] Shi, H.-Y., Wang, W.-L., Kwok, N.-M., and Chen, S.-Y. Game theory for wireless sensor networks: A survey. *Sensors*, 12:9055–9097, 2012.

[61] Camilo, T., Carreto, C., Silva, J. S., and Boavida, F. An energy-efficient ant-based routing algorithm for wireless sensor networks. In *Proceedings of the Ant Colony Optimization and Swarm Intelligence*, pages 49–59, Brussels, Belgium, Sep 2006.

[62] Bojic, I., Podobnik, V., Ljubi, I., Jezic, G., and Kusek, M. A self-optimizing mobile network: Auto-tuning the network with firefly-synchronized agents. *Information Sciences*, 182(1):77–82, 2010.

[63] Sunder, S. and Singh, A. A swarm intelligence approach to the quadratic minimum spanning tree problem. *Information Sciences*, 180(17):3182–3191, 2010.

[64] Shen, C., Pesch, D., and Irvine, J. A framework for self-management of hybrid wireless networks using autonomic computing principles. In *Proceedings of the 3rd Annual Communication Networks and Services Research Conference*, pages 261–266, Halifax, Novia Scotia, Canada, May 2005.

[65] Hsing, D., Kant, L., and Wu, T. Providing service survivability via multi-layer restoration strategies in broadband networks. In *Proceedings of Design of Reliable Communication Networks (DRCN)*, Burgge, Belgium, May 1998.

[66] Kant, L. Design and performance modeling & simulation of self-healing mechanisms for wireless communication networks. In *Proceedings of 35th Annual Simulation Symposium*, pages 35–42, San Diego, USA, Mar 2002. IEEE.

[67] Sterbenz, J. P., Krishnan, R., Hain, R. R., Jackson, A. W., Levin, D., Ramanathan, R., and Zao, J. Survivable mobile wireless networks: Issues, challenges, and research directions. In *Proceedings of the 1st ACM Conference on Wireless Security*, pages 31–40, Atlanta, Georgia, USA, Sep 2002.

[68] Poor, R., Bowman, C., and Auburn, C. B. Self-healing networks. *Queue*, 1(3): 52–59, 2003.

[69] Trehan, A. *Algorithms for Self-Healing Networks*. PhD thesis, University of New Mexico, Albuquerque, New Mexico, USA, May 2010. Downloaded from http://arxiv.org/abs/1305.4675 in Jun, 2013.

[70] Balakrishnan, H., Padmanabhan, V. N., Seshan, S., and Katz, R. H. A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM Transactions on Networking*, Dec:756–769, 1997.

[71] Lar, S.-U. and Liao, X. An initiative for a classified bibliography on TCP/IP congestion control. *Journal of Network and Computer Applications*, 36(1):126–133, 2013.

[72] Lin, X., Shroff, N. B., and Srikant, R. A tutorial on cross-layer optimization in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(8): 1452–1463, Aug 2006.

[73] Kliazovich, D. and Granelli, F. A cross-layer scheme for TCP performance improvement in wireless lans. In *Proc. IEEE GLOBECOM*, pages 840–844, 2004.

[74] Wang, Q. and Abu-Rgheff, M. A. Cross-layer signalling for next-generation wireless systems. In *IEEE*, 2003.

[75] Raisinghani, V. T., Singh, A. K., and Iyer, S. Improving TCP performance over mobile wireless environments using cross layer feedback. In *IEEE International Conference on Personal Wireless Communications*, 2002.

[76] Chinta, M., Helal, A. S., and Lee, C. ILC-TCP: An interlayer collaboration protocol for TCP performance improvement in mobile and wireless environments. In *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003*, volume 2, 2003.

[77] Cho, S., Woo, H., and Kim, C. TCP performance improvement with ack pacing during a long lasting loss period in wireless data networks. In *Lecture Notes in Computer Science*, pages 507–516. Springer-Verlag, 2003.

[78] Balakrishnan, H. and Katz, R. H. Explicit loss notification and wireless web performance. In *Proceedings of IEEE GLOBECOM'98*, Nov 1998.

[79] Jacobson, V. and Braden, R. TCP extensions for long delay paths. http://tools.ietf.org/html/rfc1072, 1988. URL http://www.faqs.org/rfcs/rfc1072.html. Accessed Jun, 2013.

[80] Ho, T. *Networking from a network coding perspective*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, May 2004. Downloaded from http://www.its.caltech.edu/~tho/ho-thesis.pdf in Jun, 2013.

[81] Chen, J., Liu, L., Hu, X., and Tan, W. Effective retransmission in network coding for TCP. *International Journal of Computers, Communications and Control*, VI (1):53–62, 2011.

[82] Wang, N.-C., Chen, Y.-L., Cheng, C.-H., Chiang, Y.-K., and Wang, Y.-Y. Improving tcp performance with fast adaptive congestion control during soft vertical handoff. *Telecommunication Systems*, 52(1):97–104, 2013.

[83] Khurshid, A., Kabir, M. H., and Prodhan, M. A. T. An improved tcp congestion control algorithm for wireless networks. In *Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pages 382–387, Victoria, British Columbia, Canada, Aug 2011.

[84] Shah, P. A., Yousaf, M., Qayyum, A., and Hasbullah, H. B. Performance comparison of end-to-end mobility management protocols for TCP. *Journal of Network and Computer Applications*, 35(6):1657–1673, 2012.

[85] Liu, T. and Cerpa, A. Foresee (4c): Wireless link prediction using link features. In *10th International Conference on Information Processing in Sensor Networks(IPSN)*, pages 294–305, Chicago, Apr 2011.

[86] Fonseca, R., Gnawali, O., Jamieson, K., and P., L. Four-bit wireless link estimation. In *Proceedings of the Sixth Workshop on Hot Topics in Networks (HotNets VI)*, Atlanta, November 2007.

[87] Alizai, M., Landsiedel, O., Link, J., Gotz, S., and Wehrle, K. Bursty traffic over bursty links. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, New York, November 2009.

[88] Liu, H., Al-Khafaji, S., and Smith, A. Prediction of wireless network connectivity using a taylor kriging approach. *International Journal of Advanced Intelligence Paradigms*, 3(2), 2011.

[89] Konak, A. A kriging approach to predicting coverage in wireless networks. *International Journal of Mobile Network Design and Innovation*, 3(2), 2009.

[90] Capka, J. and Boutaba, R. Mobility prediction in wireless networks using neural networks. In Marshall, A. and Agoulmine, N., editors, *Proceedings of the 6th International Conference on Management of Multimedia Networks and Services*, volume 2839 of *Lecture Notes in Computer Science*, pages 320–333. Springer-Verlag, 2004.

[91] Liang, X., Li, X., Shen, Q., Lu, R., Lin, X., Shen, X., and Zhuang, W. Exploiting prediction to enable secure and reliable routing in wireless body area networks. In *Proceedings of IEEE INFOCOM*, pages 388–396, 2012.

[92] Guan, Q., Yu, F., Jiang, S., and Wei, G. Prediction-based topology control and routing in cognitive radio mobile ad hoc networks. *IEEE Transactions on Vehicular Technology*, 59(9):4443–4452, 2010.

[93] Alavi, B. and Pahlavan, K. Modeling of the toa-based distance measurement error using uwb indoor radio measurements. *IEEE Communications Letters*, 10 (4):275–277, 2006.

[94] Ravi, R. and PonLakshmi, R. A new lifetime prediction algorithm based routing for vanets. *International Journal of Computer Science and Applications (TIJCSA)*, 1 (12), 2013.

[95] Sharma, A. and Parihar, P. An effective dos prevention system to analysis and prediction of network traffic using support vector machine learning. *International Journal of Application or Innovation in Engineering and Management*, 2(7), 2013.

[96] Wu, J., Liu, S., Zhou, Z., and Zhan, M. Toward intelligent intrusion prediction for wireless sensor networks using three-layer brain-like learning. *International Journal of Distributed Sensor Networks*, 2012.

[97] Maei, H., Szepesvari, C., Bhatnagar, S., Precup, D., Silver, D., and Sutton, R. Convergent temporal-difference learning with arbitrary smooth function approximation. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS 09)*, Vancouver, December 2009.

[98] Eik Loo, C., Yong Ng, M., Leckie, C., and Palaniswami, M. Intrusion detection for routing attacks in sensor networks. *International Journal of Distributed Sensor Networks*, 2(4), 2012.

[99] Chen, C., Ma, J., and Yu, K. Designing energy-efficient wireless sensor networks with mobile sinks. In *Proceeding of the 4th ACM Conference on Embedded Networked Sensor Systems (SenSys 2006)*, Colarado, November 2006.

[100] Yan, K., Wang, S., and Liu, C. A hybrid intrusion detection system of cluster-based wireless sensor networks. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Hong Kong, March 2009.

[101] Shen, W., Han, G., Shu, L., Rodrigues, J., and Chilamkurti, M. A new energy prediction approach for intrusion detection in cluster-based wireless sensor networks. *Green Communications and Networking*, 2012.

[102] Bhutani, G. Application of machine-learning based prediction techniques in wireless networks. *International Journal of Communications, Network and System Sciences*, 7(5), 2014.

[103] Linnartz, J. P. Multipath fade duration. http://wireless.per.nl/reference/chaptr03/fading/afd.htm, 2013. Part of reference website on Wireless Communication, downloaded Jun, 2013.

[104] Jacobson, V., Braden, R., and Borman, D. TCP extensions for high performance. RFC 1323, 1992.

[105] Jacobson, V. Congestion avoidance and control. In *Proc. ACM SIGCOMM'88*, pages 314–329, 1988.

[106] Bhutani, G. A near-optimal scheme for TCP ACK pacing to maintain throughput in wireless networks. In *Proceedings of the 2nd International Conference on Communication Systems and Networks*, pages 491–497, Bangalore, India, Jan 2010. IEEE.

[107] Liang, B. and Haas, Z. Predictive distance-based mobility management for PCS networks. In *Proceedings of IEEE INFOCOM*, pages 1377–1384, 1999.

[108] Camp, T., Boleng, J., and Davies, V. A survey of mobility models for ad hoc network research. *Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5): 483–502, 2002.

[109] Bhutani, G. An analytical framework for disconnection prediction in wireless networks. *International Journal of Communications, Network and System Sciences*, 7(6), 2014.

[110] Radial basis function. http://en.wikipedia.org/wiki/Radial_basis_function, 2013. Downloaded in Jun, 2013.

[111] Tan, A. and Gilbert, D. An empirical comparison of supervised machine learning techniques in bioinformatics. In *Proceedings of the Asia-Pacific Bioinformatics Conference*, volume 19, pages 219–222, Adelaide, Australia, 2003.

[112] Vanneschi, L., Farinaccio, A., Mauri, G., Antoniotti, M., Provero, P., and Giacobini, M. A comparison of machine learning techniques for survival prediction in breast cancer. *BioData Mining*, 4(12), 2011. doi: 10.1186/1756-0381-4-12.

[113] Nevins, J. and Potti, A. Mining gene expression profiles: expression signatures as cancer phenotypes. *National Review of Genetics*, 8(8):601–609, 2007.

[114] Abu-Nimeh, S., Nappa, D., Wang, X., and Nair, S. A comparison of machine learning techniques for phishing detection. In *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*, pages 60–69, Pittsburgh, Pennsylvania, 2007.

[115] Bettstetter, C. Smooth is better than sharp: a random mobility model for simulation of wireless networks. In *Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 19–27, Rome, Italy, 2001.

[116] Haas, Z. A new routing protocol for reconfigurable wireless networks. In *Proceedings of the IEEE International Conference on Universal Personal Communications*, volume 2, pages 562–566, San Diego, California, USA, Oct 1997.

[117] Guerin, R. A. Channel occupancy time distribution in a cellular radio system. *IEEE Transactions on Vehicular Technology*, 36(3):89–99, 1987.

[118] Royer, E. M., Melliar-Smith, P. M., and Moser, L. E. An analysis of the optimum node density for adhoc mobile networks. In *IEEE International Conference on Communications*, volume 3, pages 857–861, Helsinki, Finland, Jun 2001.

[119] Haas, Z. J. and Pearlman, M. R. The performance of query control schemes for the zone routing protocol. *IEEE/ACM Transactions on Networking*, 9(4):427–438, 2001.

[120] Pearlman, M. R., Haas, Z. J., Sholander, P., and Tabrizi, S. S. On the impact of alternate path routing for load balancing in mobile ad hoc networks. In *Proceedings of 1st Annual Workshop on Mobile and Ad Hoc Networking and Computing*, pages 3–10, Boston, Massachusetts, USA, 2000.

[121] Bettstetter, C. and Wagner, C. The spatial node distribution of the random waypoint mobility model. In *Proceedings of 1st German Workshop on Mobile Adhoc Networks*, pages 41–58, 2002.

[122] Resta, G. and Santi, P. An analysis of the node spatial distribution of the random waypoint mobility model for ad hoc networks. In *Proceedings of the second ACM international workshop on Principles of mobile computing*, POMC '02, pages 44–50, Toulouse, France, 2002. ISBN 1-58113-511-4.

[123] Mean field theory. http://en.wikipedia.org/wiki/Mean_field_theory, 2013. Downloaded in Jun, 2013.

[124] Phillips, R. *Pricing and Revenue Optimization*. Stanford University Press, Palo Alto, CA, 2005.

[125] Talluri, K. and Ryzin, G. J. van. *The Theory and Practice of Revenue Management*. Kluwer Academic, Dordrecht, 2004.

Appendix A

TCP ACK Pacing Algorithm

In Section 3.2, we looked at the TCP ACK Pacing algorithm. In order to implement this algorithm, the TCP layer in the base station is augmented with the ACK holder module which is responsible for implementation for this scheme. Here, we present the pseudo-code of the algorithm:

```
Begin:
  Wait on message(m) from link layer;
  Switch (m->type) {
    Case LINK_GOING_DOWN_IND:
      modeOfOperation = HOLD_ACK;
      Spawn ACK holding thread;

    Case LINK_GONE_IND:
      modeOfOperation = PACE_ACK;
      Send pacing indication to
        ACK holding thread;

    Case LINK_UP_IND:
      modeOfOperation = FLUSH_ACK;
      Send ACK flush indication
        to ACK holding thread;
  }
```

```
End
```

The pseudo code for the ACK holding thread is as follows:

```
Begin:
  Wait on message queue for TCP ACKs
    or indications from main thread
    of ACK holder (a);

  Switch(a->type) {
    Case TCP_ACK:

    Case TCP_PKT:
      Store(a);

    Case PACE_IND:
      Determine max window size (w)
        in ACKs;
      Reserve space for w bytes of
        data from fixed host;
      ScheduleAlgorithm();

    Case FLUSH_IND:
      Send ACKs to fixed host;
      Send cached packets to mobile;
  }
End
```

The `ScheduleAlgorithm()` procedure must determine the time at which to send out each ACK based on the constraints discussed in Section 3.1. The mechanism for doing this is explained in Section 3.3.

Appendix B

TCP ACK Pacing Algorithm Implementaion in ns-2

We used ns-2 in order to simulate disconnections and compare the performance of the TCP ACK Pacing scheme with TCP-Reno. Following are the simulation details:

1. Overall Setup The overall experimental setup consisted of two nodes designated the sink and source. An FTP data transfer was setup between the two nodes, with the sink serving as the file receiver and the source as the web-server. The FTP transfer duration was varied across experiments 40 seconds, 120 seconds, 600 seconds and so on. FTP is a file transfer application that runs on TCP/IP. In ns-2, the entire experimental setup was put together using an ex.tcl file that defined the nodes, the connection between the nodes, the application that exercised these links (FTP) the role of the nodes as source and sink and the duration of the file transfer. NS is a discrete event simulation, to which the .tcl file specified the time at which different events should occur start of file transfer, start of disconnection etc.

2. Simulate Disconnection The Linux kernel TCP implementation at the source (ns-allinone-3.16/nsc-0.5.3/linux-2.6.26/net/ipv4/tcp_input.c) was modified to drop ACKs after different intervals of data transfer in different experiments (10 seconds, 15 seconds and so on). Each time, the loss of ACKs would trigger TCP into re-transmitting data and finally entering the congestion control phase. The TCP

input and TCP output code was instrumented to dump the congestion window size and the retransmission timeout values for each packet transfer during this phase.

3. ACK Holding Scheme The Linux kernel TCP implementation at the sink (ns-allinone-3.16/nsc-0.5.3/linux-2.6.26/net/ipv4/tcp_output.c) was modified to start sending out ACKs paced out over a period of time, determined by the duration of disconnection for the experiment (10 seconds, 40 seconds). The duration of disconnection and the time of disconnection start were passed as parameters of the experiment. As the source started dropping ACKs and re-transmitting packets, the sink started to cache and send out delayed ACKs.

4. Throughput recovery At the end of the configured disconnection interval, the source was configured to resume normal operation without simulating dropped ACKs, but using the window size that it had arrived at by the end of the disconnection duration. The throughput of the connection was measured along with the time for the throughput to recover to the value it was at, at the start of the disconnection.

5. Performance comparison: The above experiments were performed with the ACK holding scheme at the sink and without, and in each case the values of the window sizes and number of bytes exchanged were compared to determine the performance improvement as part of the ACK holding scheme.

Papers from this Thesis

- Bhutani, G. A near-optimal scheme for TCP ACK pacing to maintain throughput in wireless networks. In *Proceedings of the 2nd International Conference on Communication Systems and Networks*, pages 491–497, Bangalore, India, Jan 2010. IEEE.

- Bhutani, G. A round-based pricing scheme for maximizing service provider's revenue in p2ptv networks. In *Proceedings of the 3rd International Conference on Internet Multimedia Services Architecture and Applications (IMSAA)*, pages 112–117, Bangalore, India, Dec 2009. IEEE.

- Bhutani, G. Application of machine-learning based prediction techniques in wireless networks. *International Journal of Communications, Network and System Sciences*, 7(5), 2014.

- Bhutani, G. An analytical framework for disconnection prediction in wireless networks. *International Journal of Communications, Network and System Sciences*, 7(6), 2014.

Bio Data of Candidate

Gitanjali Bhutani is currently working as Technical Specialist in Alcatel-Lucent, India. She has been working in Alcatel-Lucent for the past 9 years, working on software development of the platform layer of base stations and Radio Network Controllers. She has led several new feature developments in the transport area for these products. She currently also serves as the Scrum Master for a feature development and sustenance team called Skywalkers in Alcatel-Lucent.

On the academic front, Gitanjali completed her M.S. Software Systems from BITS, Pilani in 2008. Prior to that, she completed her B.E in Computer Science from University Visweswaraya College of Engineering in 2005. She won three gold medals for topping the university in the overall engineering batch as well as the computer science class.

Gitanjali's research interests include self-management in wireless networks, artificial intelligence and software algorithms. She has published two journal papers and two conference papers in the area of wireless networks and artificial intelligence.

Gitanjali's hobbies include reading and making little craft items with her two children.

Bio Data of Supervisor

Dr. Mruthyunjaya H. Kori retired as Technical Director from Alcatel-Lucent Technologies in 2010 and right now serves as an independent consultant. In his role as Technical Director, he led several teams such as the WCDMA architecture team and the Layer-1 team. He has been a pioneer of innovation in Alcatel-Lucent. Prior to this, he has worked in DSQ Software, Geosoft Technologies and C-DOT. He has also served as a faculty member in IIT Bombay. He has performed several roles right from being a research engineer to being a head of various wireless divisions. He also served as an independent consultant, providing consultation in the areas of technology identification, conceptualization of products and simulation design and development. He now continues his consultation work, while also keeping extremely busy providing talks and lectures in renowned conferences and institutes.

He attained his PhD from IIT Bombay in 1987. Prior to that, he completed his M.E from IIT Roorkee in 1979 and B.E from Mysore University in 1973. His research interests interests include wireless communications including Mobile Cellular, Satellite and Personal Communication Systems, Microelectronics and Microwaves.

He has also served as the chairman of the Institute of Electronics and Telecommunication Engineers (IETE), the International Microelectronics and Packaging Society (IMAPS) India Chapter. The IETE Bangalore center received the Best Center Award for two years under his Chairmanship. He has also served as Vice Chairman of the IEEE MTT & ED Society, India Chapter.

He is an avid reader, and pursues literature in his free time.