ABSTRACT

Title of dissertation: PERFORMANCE EVALUATION OF WIRELESS AD-HOC NETWORKS AND THE PRESENCE OF HEAVY-TAILS AND LRD

Kaustubh Jain, Doctor of Philosophy, 2015

Dissertation directed by: Professor John S. Baras
Department of Electrical & Computer Engineering

Multi-hop wireless networks lack widespread deployment due to their performance issues. There are very few models which can aid detailed analysis of performance bottlenecks and design improvements. Our objective is to develop hybrid (analytical and numerical) models which can efficiently approximate the performance of a wireless network. We focus on modeling the MAC and routing protocol performance, and their impact on network performance. For MAC layer modeling, we improve a fixed-point loss network model to evaluate throughput and packet loss for wireless ad-hoc networks based on 802.11 MAC protocol. The methodology is based on development of implicit equations that model interaction and dependency of network parameters, and using a fixed-point iterative method for finding a consistent solution. Here, we assume pre-specified / static routes. Given the topology and input offered load on multiple connections, the model finds the average carried load at various links, along with other MAC performance metrics like frame drop probability and service times. Next, we focus our attention on the routing

protocol, which significantly impact network performance especially for multi-hop connections. Towards this, we use a previously developed OLSR protocol model and modify it to combine with the 802.11 MAC model developed above. Unlike any work before, we give a combined model of a MAC and Routing Protocol which captures the cross-layer interaction.

The above models for MAC and routing focus on average performance like throughput and losses. The models assume smooth or exponential input traffic models in the usual Markovian framework. However, it is known that traffic sources can have heavy-tails and the traffic arrival pattern bursty even at large timescales. Self-similarity and the related phenomena of long-range dependence (LRD) characterizes the unexpected burstiness and correlation of the network traffic at large-time scales. LRD nature of the traffic has significant impact on network performance through higher queue-occupancy, service times, packet losses, etc. Hence, we study traffic LRD in three stages, which form the main contributions of this work. Firstly, we investigate the presence (or absence) of traffic LRD with models similar to the ones studied for wireline networks. Secondly, we investigate the role of wireless medium and protocols in creating traffic LRD in new ways deemed unlikely in wireline networks. Even though, our overall focus in on performance modeling, the first two contributions are of significant interest by themselves. Lastly, given the existence of traffic LRD in wireless networks, we study the impact of traffic LRD on network performance.

The high variability in traffic sources, characterized by heavy-tail distributions, has been attributed as one of the main causes of traffic LRD. While the existence

of LRD has been widely studied in wire-line networks, there are very few studies in wireless networks. In the absence of any large time-scale traffic traces for wireless multi-hop networks, we rely on simulation-based study for determining traffic LRD in different network scenarios. We confirm the presence of traffic LRD in scenarios with heavy-tailed traffic sources using TCP protocols. Like the wireline networks, we see little traffic LRD in UDP traffic. However, our primary interest is on the different role of wireless network protocol in creating traffic LRD. Previous protocol studies, for both wired and wireless networks, have argued that the network protocols dynamics affect only the small-time scale properties, and that they cannot create traffic LRD by themselves. However, recent work by Jelenkovic et al. also show that ALOHA protocol can lead to heavy-tailed delays in certain situations. With that as a motivation, we investigate the role of more complex 802.11 MAC protocols in shaping the delays and traffic burstiness. We show that service times for 802.11 MAC, can indeed have power-law (or truncated heavy-tail) distributions. However, even with complex topologies with hidden-nodes, the maximum MAC service times are bounded at a few seconds. Hence, they do not create traffic LRD.

More significantly, due to variable link conditions in wireless networks, the routing protocol can also cause changes in multi-hop routes even in a static topology. The link 'failure' detection mechanisms are different for wireless networks; a certain number of consecutive packet drops is usually used as indicator of link failure, even though these packet drops can be due to fading or congestion. Thus, the wireless medium can introduce spurious route changes which can potentially affect traffic burstiness at large timescales. Hence, we re-examine the impact of protocols

on the traffic LRD, in wireless multi-hop networks. We show, that contrary to conventional wisdom, wireless multi-hop network protocols can play a significant role in causing traffic burstiness at large time-scales. In particular, routing control packet losses at MAC due to congestion can lead to route oscillations in link-state routing protocols like OLSR. These route oscillations can create psuedo traffic LRD. We modify our previous MAC and OLSR models to predict which scenarios will lead to route oscillations. We also show, using the models, how we can tune the protocol design parameters to avoid route oscillations and traffic LRD.

Lastly, we investigate the impact of traffic LRD on network performance. Service times of 802.11 MAC frames depends significantly on the congestion perceived in the wireless channel. Bursty traffic can also lead to increased retries or losses during periods to high load. We analyze the average achievable throughput at the MAC layer and the impact of traffic LRD. We show that traffic LRD can significantly impact network performance. However, contrary to our intuition, bursty traffic can lead to higher throughput in some scenarios. We give some models to explain and predict such behavior in simple 4 node topologies with two links. We use these models to characterize the 'capacity region' defined in terms of the MAC protocol functioning for these 2 link topologies. In particular, given bursty or smooth traffic on one link, we characterize what is the maximum achievable throughput ('capacity') on the other link as well as the sum capacity. The models show that the achievable / maximum throughput depends on the traffic burstiness at some load levels. Thus, it is important to use appropriate input traffic models for any good performance analysis.

PERFORMANCE EVALUATION OF WIRELESS AD HOC NETWORKS AND THE PRESENCE OF HEAVY-TAILS & LRD

by

Kaustubh Jain

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2015

Advisory Committee:
Professor John S. Baras, Chair/Advisor
Professor Armand M. Makowski
Professor Richard J. La
Dr. Vahid Tabatabaee
Professor Subramanian Raghavan, Dean's Representative

Dedication

To my grandparents, parents, sister and my lovely wife.

Acknowledgments

I am deeply grateful to my advisor, Dr. John S. Baras for the years of support and guidance. Besides exposing me to broad set of research areas, he was instrumental in keeping me focused and not letting me go astray. He gave me plenty of freedom and opportunities to define my own problems, while at the same time giving new high-level ideas to explore new directions whenever I was stuck. He has taught me a lot of valuable lessons, including the importance of being independent from an early stage. There are many things that I still need to learn from him, and I hope I continue to do so.

I am really grateful to Dr. Vahid Tabatabaee. He was the one who introduced me to the field of component-based performance modeling. His guidance, especially in the initial days of my research, allowed me to get a foot-hold in this area. Every time I was stuck, I would go to him for guidance and he would patiently sit with me and provide great ideas to work out the details. His help in the first two contributions of the thesis - MAC and routing models, ultimately went beyond those chapters. With slight modifications to our models, I was able to re-use those to explain completely new phenomena of traffic LRD and route oscillations.

I am thankful to Prof. Armand Makoswki for teaching me a lot about advanced networks, queueing theory, etc. He was instrumental in keeping some of the models and definitions to be mathematically sound. Given the experimental bent of the thesis, his feedback was important to back it with appropriate theory and make the results more useful. I would also like to thank Prof. Prakash Narayan, for

providing useful directions during the research proposal stages to focus on some of the peculiarities of the wireless networks and their differences from the wired networks. This led me to focus on the cross-layer interaction between the MAC and the routing layers and the resulting dynamics specific to wireless multi-hop networks. I am also grateful to Prof. Richard La for suggesting good additions to the results to evaluate the performance trade-offs, especially when looking the the impact of bursty traffic. Thanks are due to Prof. Subramanian Raghavan for taking time to serve on my committee and reviewing the thesis.

I am deeply grateful to my mentors and colleagues at AT&T Labs - Dr. Vaidyanathan Ramaswami, Dr. Rittwik Jana and Dr. Vaneet Aggarwal. I am also grateful to AT&T Labs themselves for allowing to extend the collaboration beyond a summer internship. I am especially indebted to Dr. Ramaswami for constantly encouraging me and believing in my abilities. More importantly, I am grateful to him for teaching me invaluable lessons about conducting good theoretically sound research and being thorough in experiments as well as succint in exposition. He introduced me to the logPH models and gave me the opportunity to study their utility, for more accurate performance evaluation. His expertise in Phase-type (PH) and logPH models gave me a deeper understanding of the advantages and limitations of the Markovian queueing models. For example, the limitations of Markovian MAC and routing models in capturing the observed heavy-tails. I learned a great deal of experimental skills from Dr. Jana as well. His range of expertise and skills are a constant motivation for me. Discussions with them and other distinguished researchers, at the AT&T Labs, increased my understanding about statistical modeling, impor-

tance of data sanitization and some of the limitations of existing conformity test for heavy-tailed distributions.

I cannot thank my dear friend, Kiran Kumar Somasundaram, enough. Having worked closely with him during our working on improving the OLSR protocols, as well as the OLSR and MAC models, I learned a lot. The depth of this theoretical knowledge as well as expertise with experimental and hands-on skills provided an excellent role model. Not only has he been a source of inspiration, his support and guidance all these years was irreplaceable. I will be forever thankful to him for his guidance, and motivation, especially during the initial and middle stages of my struggles.

I am thankful to Christophoros Somarakis for help with the theoretical background and tools for studying LRD and heavy-tails. I would also like to thank Senni Perumal, Punyaslok Purkayastha, Shalabh Jain, Johnny Ta, Ayan Roy-Chowdhary, Brian Wang, and others colleagues for many useful collaborations and insightful discussions. I am really grateful to Nimish Sane, Kapil Anand, Nitesh Shroff and Harshavardhan Agashe for their help and support with other aspects of PhD life.

I would like to express my gratitude to the wonderful staff of the ISR and the ECE departments. A big thanks to Kimberly Edwards for her efficient handling of all the administrative tasks. All other staff members were always welcoming, friendly and helpful. I would like to especially thank Melanie Prange, Alexis Jenkins, Gwen Flasinski and Carlos Luceno.

I would like to express by deep gratitude to Dr. Lin-Nan Lee, from Hughes Network Systems, for being a wonderful manager and being patient with me. He

gave me a lot of freedom and time to wrap-up my thesis, while working full-time at Hughes. I also also truly grateful to my colleagues at Hughes - especially Chi-Jiun Su, Se Gi Hong, Udaya Bhaskara, Stan Kay, and others. I learned a lot from them, which not only was instrumental in my progress at Hughes, but also helped me in my PhD research.

I am also grateful to my under-graduate thesis advisor, Dr. Vishal Sharma. Not only did he teach me a lot during the undergraduate studies, he has been an wonderful mentor all these years. He has guided me at all the major junctions of my PhD life.

A big thanks to my room-mates and pseudo-roommates, with whom I would spend most of my weekends with. Their constant encouragement and the wonderful memories helped me stay strong, refreshed and motivated. The time spent with them and the wonderful memories will keep growing and will be cherished forever.

Lastly, all this would not have been possible without the love, and support of my lovely family. My deepest gratitude to my parents, who taught me the important lessons of life and provided the springboard for my success. They have always given me plenty of freedom and always encouraged me to achieve my true potential. A big thanks to my sister, and mother-, father- and sister-in-law for their constant support and encouragement. And the biggest thanks to my lovely wife, Nidhi. This simply would not have been possible without her love.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Performance Models for Wireless Ad-Hoc Networks

Multi-hop wireless networks lack widespread deployment due to their performance issues. There are very few models which can aid detailed analysis of performance bottlenecks and design improvements. Apart from the usual complexity due to network of queues, as in the wired networks, wireless networks add more complexities of their own. One of the main reasons in the difficulty in characterizing wireless 'link' capacity. Collisions and interference from neighboring transmissions make the capacity difficult to be determined. The problem is exacerbated by fading channel conditions and node mobility. Thus, the analysis, design and optimization of such systems are daunting tasks. Although use of packet level simulation tools based on PHY and MAC layer models can provide accurate performance measures, they are too complex and time consuming for analyzing the performance bottlenecks. Our objective is to develop hybrid (analytical and numerical) models which can efficiently *approximate* the performance of a wireless network. The performance of wireless multi-hop network is dominated by the shared nature of wireless channels. Cross-layer interaction of routing protocols with MAC and PHY further complicate the analysis. In this work, work towards modeling the MAC and routing protocol

performance, and their impact on network performance.

For wireless ad-hoc networks, contention-based or CSMA/CA protocols are preferred over scheduling-based MAC protocols, because of their ease of use. For high data-rate scenarios, the preferred MAC protocol is 802.11 MAC. Hence, given their popularity and ubiquity, there are many bodies of work [1–5] for modeling the 802.11 MAC. However, many of the models are restricted to some simple topologies with single-hop or disjoint-path connections [4, 5]. [6, 7] are two models which can work on any arbitrary multi-hop topologies. We improve the fixed-point loss-network model given in [6] to give more accurate performance metrics validated across a wider set of topologies. The methodology is based on development of implicit equations that model interaction and dependency of network parameters and using a fixed-point iterative method for finding a consistent solution. To focus our attention on MAC, we assume pre-specified routes. Given the topology, average PHY conditions and input offered load on multiple connections, the model find the average carried load at various links, along with other MAC performance metrics like frame drop probability, transmission attempt rates, and service times for MAC frames on each of the links. The numerical model enable us not only to estimate the average performance, but also the ability to analyze the impact of protocol parameters. In particular, we can study the impact of design parameters on performance metrics.

Secondly, a large number of routing protocols have been proposed for multi-hop wireless networks. Routing protocols are commonly implemented as large monolithic software, which are very difficult to adapt to varying scenarios. Identifying the performance bottlenecks by simulations is very time consuming, since we have to

run a large number of scenarios to pin-point the bottleneck. Because of the complex nature of the simulations, they provide little insight on design parameter sensitivities or on how we can improve performance and adaptivity of these routing protocols. In [8] model the working of OLSR - which is one of the popular routing protocol for ad-hoc networks. However, their model assume fixed and known MAC layer loss probabilities. Performance of any routing protocol depends a lot on the underlying MAC protocol. Since the process of gathering topology information depends on topology packet flooding, performance evaluation of a routing protocol cannot be done correctly without considering the underlying MAC. Similarly, the performance of a MAC protocol depends on the traffic load offered and the routing used to handle the traffic. Hence, to model the wireless network performance correctly, cross-layer interaction between the routing and MAC protocols has to be considered. Towards this, we use the OLSR model from [8], and modify it to combine with the 802.11 MAC model developed above. We are able to demonstrate that the combined model captures the cross-layer interaction. We also re-use parts of the OLSR and 802.11 MAC models later to model and explain some interesting new phenomenon observed using network simulations. Thus, we show the utility and power of such models in obtaining not just the usual performance metrics like throughput and losses, but also in explaining other important performance artifacts.

1.2 Heavy-Tails and Long Range Dependence

The above models for MAC and routing focus on average performance like throughput and losses. However, the models assume smooth or exponential input traffic models in the usual Markovian or Poisson framework. It is known that traffic sources can have heavy-tails and the arrival pattern bursty even at large timescales [9–13]. These deviations can lead to performance degradation [14–16] and need to be accounted for while doing performance estimation for network dimensioning. We focus on a particular aspect of large time-scale traffic burstiness. Self-similarity and the related phenomena of long-range dependence (LRD) characterizes the unexpected burstiness and correlation of the network traffic at large-time scales. LRD nature of the traffic has significant impact on network performance [15, 16] through queue-occupancy, service times, packet losses, etc. Hence, we study traffic LRD in three stages first, we investigate the presence (or absence) of traffic LRD with models similar to the ones studied for wireline networks. Secondly, we investigate the role of wireless medium and protocols in creating traffic LRD in new ways deemed unlikely in wireline networks. Lastly, given the existence of traffic LRD in wireless networks, we study the impact of traffic LRD on network performance. More details are given below.

The high variability in traffic sources, characterized by heavy-tail distributions, has been attributed as one of the main causes of traffic LRD [10–13]. These observations and other protocol studies [15,17,18] have led to the belief that the network protocols themselves play little part in the large time-scale behavior of traffic, and

that protocol dynamics affect only the small-time scale properties [14, 19]. While the existence of LRD has been widely studied in wire-line networks, there are very few studies in wireless networks [19–26]. Given the belief that network protocols won't affect large time-scale behavior, it is generally expected that wireless networks will yield similar results for traffic LRD. However, we believe, the wireless network protocols are expected to have more pronounced impact in shaping traffic at large time-scales. Recent work by Jelenkovic et al. also show that ALOHA protocol can lead to heavy-tailed delays in certain situations. With that as a motivation, we investigate the role of more complex 802.11 MAC protocols in shaping the delays and traffic burstiness. We show that service times for 802.11 MAC, can indeed have power-law (truncated heavy-tailed) distributions. Secondly, due to variable link conditions in wireless networks, the routing protocol can also cause changes in multi-hop routes even in a static topology. The link 'failure' detection mechanisms are different for wireless networks; a certain number of consecutive packet drops is usually used as indicator of link failure, even though these packet drops can be due to fading or congestion. Thus, the wireless medium can introduce route changes which can potentially affect traffic burstiness at large timescales. Hence, we re-examine the impact of protocols on the traffic LRD, in wireless multi-hop networks. We show, that contrary to conventional wisdom, wireless multi-hop network protocols can play a significant role in causing traffic burstiness at large time-scales and traffic LRD. In particular, routing control packet losses at MAC due to congestion can lead to route oscillations in link-state routing protocols like OLSR. These route oscillations can create traffic LRD. We also modify our previously developed 802.11

MAC and OLSR models to predict which scenarios will lead to route oscillations. We also show, using our models, how we can tune the protocol design parameters to avoid route oscillations and traffic LRD.

The above observations are significant in their own right. Moreover, our models are used to predict such interesting phenomena and thus show their utility. However, the presence of heavy-tails and traffic LRD is important for purposes of improving the models as well. Impact of traffic LRD needs to be studied for proper network performance analysis and network provisioning and dimensioning. Presence of heavy-tails can lead to high variability in traffic in the network. Traffic LRD can lead to performance degradation due to higher queuing delays, more packet losses, etc. Since our focus is on developing good performance models, we study the direct impact of bursty traffic on them. Service times of contention-based and CSMA/CA MAC (like 802.11) frames depends significantly on the congestion perceived in the wireless channel. If a sender perceives the channel as busy or experiences a lot of collisions, it may significantly increase the backoff time and take longer time to serve the packet. With this in mind, bursty traffic from the higher layers can lead to significant performance degradation, not just due to queuing but variation in service times. During the peaks of the bursty traffic, there may be increased contention, collisions and losses. This will increase the service time, and may also have some spill-over effect on the subsequent period of low load. The bursty traffic can also lead to increased losses due to retry attempt limit and finite buffer. We analyze the average achievable throughput at the MAC layer and the impact of traffic LRD. We show that traffic LRD can significantly impact network performance. And, contrary

to our intuition, bursty traffic leads to higher throughput in some scenarios. Thus, it is important to use appropriate input traffic models for any good performance analysis. We modify our previous 802.11 MAC model to deal with bursty input traffic. Using the modified model, we explain and predict such behavior in simple 4 node topologies with two links. We use these models to characterize the 'capacity region' defined in terms of the MAC protocol functioning for these 2 link topologies. In particular, given bursty or smooth traffic on one link, we characterize what is the maximum achievable throughput ('capacity') on the other link as well as the sum capacity. The models show that the achievable / maximum throughput depends on the traffic burstiness at some load levels.

1.3 Contributions

1.3.1 Performance Modeling of 802.11 MAC

We improve a loss-network model for multi-hop wireless networks based on IEEE 802.11 MAC and compute the network throughput and packet loss parameters. For the 802.11 MAC layer modeling, modeling link layer losses involves correct modeling of transmission attempt and collision probabilities. These, in turn, need modeling of backoff evolution, packet service times and simultaneous transmission. Even for a relatively simple single-cell network, modeling of the backoff evolution and transmission attempt rates at a node is non-trivial due to the dependence on the neighboring nodes. For multi-hop scenarios, its much more complicated due to hidden nodes and different views of channels at various nodes. There have been

good models for single-cell scenarios where there are no multi-hop paths and each node has the same view of the shared channel. Previous work to extend these models have many limitations in terms of the topology they handle. Good models for any general topology, with arbitrary load levels (non-saturated queues) etc., are lacking. One of the most general model was introduced in [6]. They developed a loss model for multi-hop wireless networks based on IEEE 802.11 MAC. Given a multi-hop network topology, connection demands and routes, it models the working of 802.11 MAC in DCF mode to find good approximations to average MAC layer losses, service times and carried load. The model is defined as an implicit function amongst the variables in the model and solved using a fixed point approach. In this work, we improve the model to make it converge faster and give better performance estimates. We observed that the previous fixed point model [6] did not converge in a few network scenarios and gave unrealistic throughput estimates in certain others. We address those issues in this work and give an improved model. In particular, we use ensure convergence to realistic values using nested iterations. We also modify some of the equations modeling the interdependence of parameters by using better approximations. We also use the model for performance optimization. As an illustration, we use sensitivity analysis to optimize the overall network throughput. We validate the model and the performance improvements using network simulations.

1.3.2 Performance Modeling with OLSR Routing Protocol

The interaction and overall performance of the MAC and routing protocols in an ad-hoc network can be modeled by dividing the various parts of the protocols into components. The first step towards modeling the complex OLSR routing protocol was introduced in [8]. In this approach a proactive routing protocol is divided into three main components - (i) Neighbor Discovery Component (NDC), (ii) Selector of Information for Dissemination Component (STIDC), and (iii) Route Selection Component (RSC). In this work, we modify the routing components to capture their interdependence with the MAC layer. Simultaneously, we modify the 802.11 MAC model described previously. While [8] considers fixed MAC loss probabilities for the links, the 802.11 MAC model, described above, assumes pre-specified static routes. In this part, we specify appropriate modifications to the components and close the interdependence loop, taking cross-layer effects into account. We then describe a fixed point algorithm to find a consistent set of solution [27]. The combined component models *approximate* the performance of routing and MAC protocols, but enable us to evaluate and analyze the cross-layer interaction under various network topologies. For specified static network topology, traffic demands and physical layer losses, the models evaluate the performance metrics of the various components and find out network performance metrics - average throughput and average end-to-end delay for each connection. By analyzing the performances under varying network scenarios, we can identify the sources of performance degradation and try to improve the corresponding components. We can also study the trade-offs and choose the

component design parameters appropriately. To illustrate how to identify bottleneck components and design them for better performance, we study the performance of a few network topologies under varying design parameters for the Neighbor Discovery Component.

1.3.3 Role of MAC Protocols in Traffic LRD

An important aspect of accurate performance evaluation is the role of network traffic characteristic. High variability and burstiness of traffic is one such aspect. Before studying the role of traffic burstiness on network performance, we aim to characterize the traffic burstiness and the role of protocols in shaping traffic at large time-scales. We carry out a simulation-based study to find the relation between traffic burstiness and the impact of protocols in multi-hop wireless networks. In particular, we focus on the role of routing protocols and contention-based MAC in inducing or preventing traffic burstiness.

We look at the role of MAC retransmissions in introducing heavy-tailed delays. Motivated by Jelenkovic and Tan's work [28], we explore the service time and delays in the MAC layer and look for evidence of heavy-tails. In their work, Tan et al. showed MAC retransmissions, under some conditions, can lead to delays with power-law distribution. However, in their theoretical analysis, they had to make simplifying assumptions including the absence of any queue (queue length of 1). To perform similar theoretical analysis for more realistic network scenarios with more complex MAC protocols based on CSMA/CA is difficult. Here, we use

simulation-based approach to investigate similar phenomenon under more realistic network scenarios with 802.11 MAC protocol. We find some evidence of power-laws in MAC service time. In particular, the MAC service times can show power-law distribution. However, the simulations show the service times are not pure heavy-tails and have a finite support which does not stretch beyond a few seconds. Furthermore, even though the service times have power-laws, the delay (including the queueing delay) do not show any power laws. Also, since the power-laws in service times are also bounded, MAC protocol does not impact traffic burstiness above few tens of seconds. Certainly, we found no evidence of traffic LRD due to MAC protocol dynamics alone. However, the power-laws in MAC service time indeed lead to some traffic scaling upto few orders of magnitude (tens of seconds) above the mean MAC service time (few milliseconds).

Similar to previous studies for wired networks [14, 29], we carry out a detailed study of traffic LRD in the presence of heavy-tailed traffic sources. We wish to see if wireless MAC protocols and the shared nature of the wireless medium give rise to some interesting phenomenon. We observe heavy-tails in TCP traffic lead to traffic LRD, consistent with the previous wired network observations. However for UDP traffic, finite buffers and losses at the MAC layer, lead to truncation of heavy-tails and the absence of any traffic LRD. We study the impact of various network and protocol configurations - like buffer size, PHY data rate, MAC retry limit, etc. on the level and extent of traffic scaling.

1.3.4 Role of Routing Protocols in Creating Traffic LRD

Similar to the role of MAC protocol, we study the role of routing protocol in creating or sustaining traffic LRD. We study two of the most popular routing protocols - OLSR and AODV. We observe that losses of OLSR routing control packets, at the MAC layer, can lead to route changes even in static networks. Such route changes, in turn, result in bursty traffic at intermediate nodes. Even though, the typical timescale of routing protocol packets is of the order of few seconds, we see that traffic burstiness is observed even at time-scales of up to several hours. Thus, unlike the wired networks studied before, the varying nature of wireless 'links' and their impact on routing protocols, result in traffic brustiness at much larger timescales.

For the OLSR scenario, we also model the observed traffic burstiness phenomena using models in two steps. At first, using a simple On/Off packet train model, we explain how route oscillations can lead to traffic burstiness, and how a Markovian framework for route oscillations is not sufficient to explain the large-timescales burstiness. Using a simple 4-node scenario with a single connection, we provide evidence for sub-exponential On and Off durations, providing some evidence for protocol-induced 'heavy-tails'. In the second part, we use the previously developed models for the 802.11 MAC and OLSR protocols to show how input traffic load levels can impact protocol dynamics and lead to topology changes even in a static network. We use approximate models to predict the presence of traffic burstiness at large timescales. In particular, we use our 802.11 MAC and OLSR models to

find the probability of topology changes due to congestion and collisions; thereby, leading to traffic burstiness in at least some of the intermediate nodes. We also show, while protocols can lead to traffic burstiness at large-timescales, appropriate tuning of protocol parameters can mitigate the effect. We use our models to correctly predict what choices of protocol parameters will results in traffic burstiness and what choices will not. Thus, we use the models not just to explain the observed phenomenon, but also to aid in network design. Both the contributions - new origins of traffic burstiness (due to routing protocol) and modeling the network dynamics to mitigate them, are significant.

1.3.5 Impact of Traffic Burstiness on 802.11 MAC and Network Performance

Next, we turn our attention to impact of traffic burstiness on the 802.11 MAC performance. We look at how traffic burstiness at large timescales affects MAC's performance. As a first part, we show that high variability in input traffic can impact network performance. Performance of contention-based MAC protocols like 802.11 depends significantly on the congestion perceived in the wireless channel. If a sender perceives the channel as busy or experiences a lot of collisions, it may significantly increase the backoff time and take longer time to serve the packet. With this in mind, bursty traffic from the higher layers can lead to significant performance degradation. During the peaks of the bursty traffic, there may be increased contention, collisions and losses. This will increase the service time, and may also have some spill-over

effect on the subsequent period of low load. The bursty traffic can also lead to increased losses due to retry attempt limit and finite buffer. Apart from simulation-based analysis, modeling the MAC service times in the presence of bursty load, can help us in developing better MAC performance model and suggest improvements to previous MAC models.

We carry out simulation-based study to study the performance impact. While traffic burstiness leads to performance degradation in many scenarios, we made some interesting and unexpected observations. Simulations show that for bursty traffic can lead to higher throughput in some scenarios. This is un-intuitive and unexpected. This is because, for bursty input load, the retransmissions and losses are bursty. While these leads to slightly higher frame drops during periods of high burstiness, the overall MAC service times are lower. This leads to higher throughput.

However, only in extreme cases there is a clear increase in MAC throughput. When the traffic bursts are (mis)aligned at neighboring nodes, such that there is little contention, only then throughput increases. We give some models to explain and predict such behavior in simple 4 node topologies with two links. We use these models to characterize the 'capacity region' defined in terms of the MAC protocol functioning for these 2 link topologies. In particular, given bursty or smooth traffic on one link, we characterize what is the maximum achievable throughput ('capacity') on the other link as well as the sum capacity. The models show that the achievable / maximum throughput depends on the traffic burstiness at some load levels. Moreover, the model correctly predicts (validated via OPNET simulations) scenarios and load levels when we can expect the role of traffic burstiness.

1.3.6 Importance of Modeling Heavy-Tails in Traffic Sources

Stepping aside from the impact of heavy-tails in traffic sources on protocol performance, we focus our attention on the heavy-tailed traffic sources themselves. In order to get a better understanding of the network performance, we need to model not only the protocol dynamics but also accurately characterize the traffic sources. Given the significant impact exerted by the tail, much of the current literature has been focused on the tail. Most of the heavy-tailed traffic sources are characterized using probability distributions like Pareto, Weibull and Lognormal. But we believe that the head of the distribution, which contain the bulk of the data, also has significant impact on the network performance. For example, the slow-start phase of TCP will result in poor performance due to small file-sizes or connection durations. Hence, we feel the need to focus on the head as well as the tail of the distributions. While there have been some attempts to characterize the entire distribution of heavy-tailed traffic sources [30], they seem to be ad-hoc fixes. We need a more general framework for modeling the heavy-tailed traffic sources. With this motivation, we turn to using *Phase-type (PH) distributions* [31]. PH distributions form a dense family on $[0, \infty)$, and are defined over a Markov chain with given transition matrix. However, a difficulty with this class in modeling heavy tailed random variables is that phase type distributions always have an exponentially decaying tail. Hence, [32] introduced the idea of using logPH distributions to model the heavy-tailed distributions. The main idea is to describe the distribution as an exponent of a PH random variable. We extend that idea to characterize different

traffic sources. We use the logPH distribution to describe various source traffic measurements and evaluate the goodness-of-fit. We also compare the logPH fits with other conventional models to demonstrate the reliability of the logPH family.

Our interest in logPH as a model class is from its ability to model heavy tailed data in its entire range. But there is a fundamental question as to if it even matters whether the entire distribution is matched or not. Hence, we use network simulations to study network performance - throughput and queue lengths. In particular, we use different set of models (Pareto, Weibull, logPH etc.) for file sizes and compare the simulation performance metrics for each of those models with one when we use the actual trace [12]. We show that logPH model predicts performance closest to the actual.

1.4 Organization of the Thesis

This thesis is organized as follows. In Ch. 2, we model the 802.11 MAC and give performance estimates for multi-hop wireless networks with static routes. In Ch. 3, we combine the MAC model with a model for OLSR protocol for routing. This enables us to perform cross-layer analysis of the two layers and we study it's impact on network performance. We introduce the concepts on heavy-tails and LRD in Ch. 4. There, we also study the presence of traffic LRD in wireless networks, and focus on the role of MAC protocol in creating heavy-tails and shaping traffic burstiness. Ch. 5 deals with the role of routing protocols in creating traffic burstiness via route oscillations. We also modify our MAC and OLSR models to predict scenarios which

have traffic LRD. We also use the model to study appropriate parameter tuning to avoid such undesirable route oscillations and traffic burstiness. Having shown the presence of high traffic variability, we return our focus on performance modeling of the MAC protocol, especially in the presence of bursty traffic, in Ch. 6. We show that even bursty traffic at timescales of a few tens of seconds can impact the MAC 'capacity' and hence should be accounted for in performance modeling as well. Ch. 7 deals with the importance of modeling the entire distribution, instead of focusing on the mean or the tail. We use a family of distribution logPH to model heavy-tailed traffic sources and show that the model comes closest in performance analysis when compared to the actual trace. We conclude the thesis with future work in Ch. 8.

Chapter 2: Performance Modeling of 802.11 MAC

Wireless ad-hoc networks were made popular because of their supposed ease of deployment in diverse conditions. Given the various aspects such as channel fading and node mobility, most of the use-cases assume a distributed network architecture without any centralized controller. Because of this, the underlying MAC protocols based on CSMA/CA such as 802.11 or Zigbee are popular. Even though scheduling-based MAC protocols have been proposed, they are not widely studied because of the supposed control overhead in implementing centralized or distributed scheduling-based algorithms. Given the shared nature of the wireless medium, MAC protocol efficiency significantly impact the network performance.

Towards this end, we focus on the MAC layer modeling and develop a fixed point loss model to evaluate connection and network throughput and packet loss for a wireless network based on 802.11 MAC protocol. These models assist us in performance analysis, design and parameter tuning procedures for wireless networks. The methodology is based on development of equations that model interaction and dependency of network parameters and using a fixed point iterative method for finding a consistent solution. Then, we use Automatic Differentiation (AD) [33] for sensitivity analysis and optimize the network performance. We assume we know

the exogenous traffic rate for each connection (source-destination pair), and the set of paths (with multiple paths per connections). We then optimally split the traffic of each connection amongst it's multiple paths to improve the overall network throughput.

We apply our methodology on IEEE 802.11 wireless networks and compute the network throughput, packet loss and delay parameters. For the 802.11 MAC layer modeling, modeling link layer losses involves correct modeling of transmission attempt and collision probabilities. These, in turn, need modeling of backoff evolution, packet service times and simultaneous transmission. Even for a relatively simple single-cell network, modeling of the backoff evolution and transmission attempt rates at a node is non-trivial due to the dependence on the neighboring nodes. For multi-hop scenarios, its much more complicated due to hidden nodes and different views of channels at various nodes. Starting with the seminal work Bianchi [1], and extensions by Kumar et al. [2, 3], for single-cell networks, [4, 5] extended the models for multi-hop scenarios. However, they had many restrictions on the topologies and connections considered and could not be used to analyze realistic scenarios. Baras et al. in [6] extended the model from [4] to arbitrary multi-hop topology. Their model obtained the loss parameters and throughput depending upon the topology and offered load. Subsequently, Jindal et al. [7] extended models from [5] to obtain achievable rate region in arbitrary multi-hop networks. In this work, we improve the model presented in [6] to make them converge faster and give better performance estimates. We observed that the previous fixed point model [6] did not converge in a few network scenarios and gave unrealistic throughput estimates in certain others.

We address those issues in this work and give an improved model. In particular, we use ensure convergence to realistic values using nested iterations. We also modify some of the equations modeling the interdependence of parameters by using better approximations. We also use the model for performance optimization and validate the performance improvements through network simulations. While Jindal's work models the different channel activities in great detail, our model make certain simplifying independence assumptions at the cost of some accuracy. Through simulations, we show, the loss in accuracy due to our simplifying assumptions is not significant and does not affect the overall goal of performance improvement. More importantly, we use sensitivity analysis to optimize the overall network throughput. We validate the performance improvements via network simulations.

Rest of the chapter is organized as follows. In Section 2.1, we describe the network model and give an overview of our fixed point approach. We give the detailed equations relating various MAC parameters in Section 2.2. We give the sensitivity analysis and optimization framework in Section 2.3. In Section 2.4, we first validate the throughput estimates from our model with the empirical throughput from network simulations. Then we use the optimization framework to demonstrate the performance improvements.

2.1 Preliminaries

2.1.1 Network Model and Assumptions

We study static multi-hop network topology represented by a graph $G = (V, E)$, where V and E are the set of nodes and edges. There is an edge between two nodes iff they are in the communication range of each other. A collision happens at a node if two nodes in its communication range transmit simultaneously. The network consists of a specified path set P that is used to forward traffic between the source destination pairs. The exogenous traffic arrival rate for each connection and fraction of traffic over each path is also specified. In general, we use indexes i and j for nodes and p for paths. The next and previous hops of node i in path p are shown by $h_{i,p}$ and $h^-_{i,p}$ respectively. P_i denotes the set of paths p passing through node i. C_i denotes the set of nodes that are in the communication range of node i (neighbors of i in graph G) and C_i^+ is C_i plus node i, C_i^- is the set of the nodes not in C_i^+.

For the MAC layer, we consider a slotted time system, where a time slot equals one backoff time slot of the IEEE 802.11 protocol. For simplicity, we assume that the data packets have equal length and all nodes use the same data rate. In the 802.11 protocol with RTS/CTS exchange there are two stages for packet transmission: 1) the RTS and CTS are sent between two nodes; and 2) the data packet and the ACK are sent. Different transmission failures from node i to node j or from node i over path p are represented as follows: $\beta_{i,p}$ is the probability of

PHY or MAC layer transmission failure during stage 1 or 2, $\epsilon_{i,p}$ is the probability of PHY layer transmission failure during stage 2, and $l_{i,j}$ is the probability of PHY layer transmission failure at stage 1 or 2 from node i to node j. PHY layer losses, ϵ and l, are part of the input, while β is part of the model's output parameters. The other important output parameters are the probability of transmission from node i neighbors that are hidden from node j, $\theta_{i,j}$, the arrival rate of the path p packets at node i, $\lambda_{i,p}$, and the service times, $T_{i,p}$. From various $\lambda_{i,p}$, we can obtain the connection and network throughput. In obtaining the above parameters, we assume the topology does not change for till the system reaches steady-state and time averages converge. We also assume that the packet errors are independent and not bursty. In modeling the interdependence between parameters, we use some independence assumptions on the collision events so that mean of one parameter can be determined using mean of other. For example, mean transmission duration and service times can be used to determine the mean attempt rate and collision probability. We explain these independence assumption in more detail when we define the model equations.

Thus, for a given set of inputs, which include the topology graph $G = (V, E)$, the set of paths P, the exogenous traffic arrival rate for connections, fraction of traffic that is transmitted over each path and the PHY layer error probabilities, we find the average system performance. In particular, we find MAC layer packet transmission attempt rate, service times, throughput and loss rate.

2.1.2 Model Structure

Here, we give a brief overview of the model structure that approximate the average behavior of a multi-hop wireless network based on the IEEE 802.11 MAC protocol. However, we will present the detailed set of equations in the next section. The model structure enables us to find a consistent solution for the given set of non-linear equations. The derived equations are used iteratively in this structure to find a consistent solution for all parameters. Let $\underline{Y}$ be the set of parameters and $\underline{F}(\underline{Y})$ be the set of equations that express the average system behavior by approximating each parameter in terms of the other parameters. One common way to find a solution to the given set of equations is to use fixed point iterations,

$$\underline{Y}^{new} = \underline{F}(\underline{Y}) \tag{2.1}$$

This approach was used in [6, 34]. However, on further investigation, we observed that for large networks when the arrival rate is close or above the network capacity, the parameters may not converge. In some other scenarios, the fixed point algorithm converges to give unrealistic values. A single-loop fixed point algorithm oscillate or converges to unrealistic values because of the implicit dependence of losses at a node and service times of its neighboring nodes.Hence, we use a two-step algorithm to find link losses and service times. First, we fix β and θ, and run a fixed point algorithm to find the service times T, and other parameters. Then, using these parameters, we update β and θ. Hence, we update the parameters iteratively in two nested loops as follows:

Algorithm 1 FPA Model

Initialize: $(\underline{\beta}, \underline{X}, \underline{\theta})$

while $\underline{\beta}, \underline{\theta}$ are not converged **do**

  while $\underline{T}$ is not converged **do**

    $\underline{X}^{new} = F_{\underline{X}}(\underline{X}, \underline{\beta}, \underline{\theta})$

    $\underline{X} = \underline{X}^{new}$

  end while

  $\underline{\theta}^{temp} = F_{\underline{\theta}}(\underline{X})$

  $\underline{\theta}^{new} = \epsilon\underline{\theta}^{temp} + (1-\epsilon)\underline{\theta}$

  $\underline{\beta}^{temp} = F_{\underline{\beta}}(\underline{X}, \underline{\theta}^{new})$

  $\underline{\beta}^{new} = \epsilon\underline{\beta}^{temp} + (1-\epsilon)\underline{\beta}$

  $\underline{\theta} = \underline{\theta}^{new}$

  $\underline{\beta} = \underline{\beta}^{new}$

end while

Let $\underline{X}$ be the set of parameters excluding $\underline{\theta}$ and $\underline{\beta}$ and $F_{\underline{X}}(\underline{X}, \underline{\beta}, \underline{\theta})$, $F_{\underline{\theta}}(\underline{X})$ and $F_{\underline{\beta}}(\underline{X}, \underline{\theta})$ be the set of equations that are derived to approximate $\underline{X}, \underline{\theta}$, and $\underline{\beta}$ respectively. The iterative algorithm is described in the Algorithm 1 pseudo code. The inner while loop updates $\underline{X}$ parameters using a fixed point iteration. The outer loop updates $\underline{\theta}$ using a convex combination (weighted average) of previous and new value, where ϵ specifies the weight. The updated values of $\underline{\theta}$ and $\underline{X}$ are then used to update $\underline{\beta}$. The convergence criteria in both cases is based on the maximum difference in old and new computed values of the specified parameters, which are $\underline{\theta}$ and $\underline{\beta}$ for the outer loop and $\underline{T}$ for the inner loop.

We need to update θ and β with memory, otherwise the *FPA* algorithm does not converge when the network load is high. As an example, suppose we start with low initial values for θ. This results in low estimates of the hidden nodes effect and low losses. Hence, the estimated traffic rates over links will be high. In the next iteration, the high traffic rates, results in high θ values, which in turn results in high losses and low traffic rates. The low traffic rate estimates, results in low θ values in the next iteration. Therefore, fixed point equations, oscillate between high θ, low data rate, and low θ high rate regimes. Hence, we have to update θ parameters in small step sizes. We tested *FPA* model with many scenarios. In all the scenarios, the algorithm converged to give realistic solution.

2.2 The MAC layer Equations

In this section we provide the set of equations that we use to approximate the wireless links average loss parameters and service times. We first present the set of equations that are used in the fixed point iteration (inner loop) and then the outer loop equations for updating the θ and β parameters. These equations are similar to the one introduced in [6], however, we change some of the equations and the underlying independence assumptions to make them more accurate.

2.2.1 Inner loop (fixed point) equations:

The set of equations that are numbered are used in the fixed point iterations and the estimated parameter in the left hand side of the equations are specified with

superscript *'new'*. Hence, the following equation specifies that it is used in the fixed point iteration to update $y_{i,p}$ variables.

$$y_{i,p}^{new} = f(\underline{X}) \tag{2.2}$$

The unnumbered equations are auxiliary and used in computation of fixed point equations. These computations are carried out for every path p and node i in the network.

2.2.1.1 Link Loss and Hidden Nodes Modeling:

To obtain the transmission failure probability, we first need to characterize the probability of a node accessing the channel, assuming that this node is scheduled to serve a packet on the path p. Let W and M be respectively the minimum and maximum sizes of 802.11 back-off windows respectively. Then L, the number of back-off stages which the window size reaches its maximum value, is $L = \log_2 \frac{M}{W}$. Let m be the maximum retry limit. Suppose that node i is scheduled to serve a packet on path p. Assuming that node accesses the channel with a fixed probability $\alpha''_{i,p}$, and setting $\beta = \beta_{i,p}$, we can use the following relation derived from Eq. (1)

of [3]:

$$\alpha''^{new}_{i,p} = \begin{cases} \frac{2(1-2\beta)(1-\beta^{m+1})}{(1-2\beta)(1-\beta^{m+1})+W(1-\beta-\beta(2\beta)^{L}(1+\beta^{m-L}(1-2\beta)))}, \\ \\ \text{if } L < m \\ \\ \frac{2(1-2\beta)(1-\beta^{m+1})}{(1-2\beta)(1-\beta^{m+1})+W(1-\beta)(1-(2\beta)^{m+1})}, \\ \\ \text{if } L \geq m \end{cases} \tag{2.3}$$

The scheduler behavior is, then, specified by the scheduler coefficient $k_{i,p}$, which is the average serving rate of the path p packets at node i and is given by the following equations:

$$k^{new}_{i,p} = \begin{cases} \lambda_{i,p} & \text{if } \sum\limits_{p' \in P_i} \lambda_{i,p'} T_{i,p'} \leq 1 \\ \\ \frac{\lambda_{i,p}}{\sum\limits_{p' \in P_i} \lambda_{i,p'} T_{i,p'}} & \text{otherwise} \end{cases} \tag{2.4}$$

The total average throughput $\bar{\rho}_i$, of node i, is $\bar{\rho}_i = \sum\limits_{p \in P_i} k_{i,p} T_{i,p}$. Note that if $\sum\limits_{p' \in P_i} \lambda_{i,p'} T_{i,p'}$, which is the required utilization at node i to serve total incoming traffic, is less than one then we can serve all packets and the scheduling rate equals the arrival rate. On the other hand if this condition is not satisfied the scheduling rate should be adjusted to make sure that the utilization does not exceed 1. Here, we have assumed that the scheduling rate is proportional to the arrival rate for each path. This assumption is reasonable and valid for scheduling policies such as FCFS scheduling policy.

Let, $v_{i,p}$ be the channel holding time by node i when it attempts to transmit

path p packets to node $h_{i,p}$. There are two different components in $v_{i,p}$: (i) the average holding time τ_P when the attempted transmission is successful and (ii) the average holding time $f_{i,p}$ when attempted transmission fails. We have,

$$f_{i,p}^{new} = \frac{\epsilon_{i,p}}{\beta_{i,p}}\tau_P + (1 - \frac{\epsilon_{i,p}}{\beta_{i,p}})\tau_H \tag{2.5}$$

The first and second terms are the average channel holding times when there are transmission failures in the data packet/ACK and RTS/CTS failure respectively:

$$\begin{aligned}\tau_P &= T_{\text{RTS}} + \text{SIFS} + T_{\text{CTS}} + \text{SIFS} \\ &\quad + T_{\text{P}} + \text{SIFS} + T_{\text{ACK}} + \text{DIFS} \\ \tau_H &= T_{\text{RTS}} + \text{SIFS} + T_{\text{CTS}} + \text{SIFS}\end{aligned}$$

where SIFS and DIFS are IEEE 802.11 parameters and T_{RTS}, T_{CTS}, T_{P} and T_{ACK} are the transmission times for RTS, CTS, data and ACK on the corresponding connection respectively. We have neglected the propagation delay between the two nodes for simplicity. Now, we can compute the average holding time $v_{i,p}$,

$$v_{i,p}^{new} = (1 - \beta_{i,p}^m)\tau_P + \frac{1 - \beta_{i,p}^m}{1 - \beta_{i,p}}\beta_{i,p} f_{i,p} \tag{2.6}$$

The average loss factor for path p transmission at node i is $\beta_{i,p}^m$ since after m transmission failures a packet is discarded. Hence, the arrival rate of path p packets to node $h_{i,p}$ is,

$$\lambda_{h_{i,p},p}^{new} = k_{i,p}(1 - \beta_{i,p}^m) \text{ for all } i, p \in \Pi_i \tag{2.7}$$

The arrival rate for the first-hop of the path is determined by the exogenous arrival rate and the routing policy and from the fixed point perspective is fixed and given

by:

$$\lambda_{i,p} = \text{Exogenous arrival rate for path } p$$
$$\text{if } i \text{ is the first node of } p \tag{2.8}$$

The average throughput of path p at node i is the fraction of time that node i is busy serving transmission of path p packets,

$$\rho_{i,p}^{new} = k_{i,p} T_{i,p} \tag{2.9}$$

The attempt rate of node i for path p, $\alpha''_{i,p,j}$, as given in (2.3) is conditional to node i being scheduled to serve path p packets. The unconditional attempt rate is,

$$\alpha_{i,p,i}^{new} = \rho_{i,p} \alpha''_{i,p} \tag{2.10}$$

Consider a node j that is neighbor of a node i. The access (attempt) rate of node i for path p packets as observed by node j is different from the access rate of node i given in (2.10), since node j does not know about transmission from i neighbors that are hidden from it. Hence, $\alpha_{i,p,j}$, the attempt rate of path p packets at node i as observed by node j is,

$$\alpha_{i,p,j}^{new} = \rho_{i,p}(1 - \theta_{i,j}) \alpha''_{i,p} \text{ for all } j \in C_i \quad i \neq j \tag{2.11}$$

where θ_{ij} is the probability of hidden node transmissions.

2.2.1.2 The Serving Time Components

The average serving time of a path p packet at node i, $T_{i,p}$, consists of 4 components as follows:

- $s_{i,p}$: average time for successful transmission of path p packets at node i.
- $u_{i,p}$: average time for successful transmission to and from node i neighbors.
- $b_{i,p}$: average back-off time of node i.
- $c_{i,p}$: average time of unsuccessful transmission due to PHY layer errors and collisions at the MAC layer in the neighborhood of node i.

The probability of successful transmission of a packet of path p at node i is $(1 - \beta_{i,p}^m)$. And successful transmission time for a packet is τ_P. Hence, the first component is,

$$s_{i,p}^{new} = (1 - \beta_{i,p}^m)\tau_P \tag{2.12}$$

Let CW_n be the back-off window size at stage n, then $W_n = CW_n/2$ is the average back-off time at stage n. The average back-off time is,

$$b_{i,p}^{new} = \sum_{n=0}^{m} W_n \beta_{i,p}^n \tag{2.13}$$

Now, we compute $u_{i,p}$, the average time of successful transmission of node i neighbors. The probability of successful transmission of node i, when it is scheduled to transmit path p packets is,

$$q_{i,p}^{new} = (1 - \beta_{i,p})\alpha''_{i,p}$$

Assuming that successful transmission of neighbors are independent events, the probability of successful transmission to and from neighborhood of i, which is scheduled to transmit path p packets is,

$$r_{i,p} = 1 - (1 - q_{i,p}) \prod_{j \in C_i} \left(1 - (S_j^1 + S_j^2)(1 - \theta_{ji})\right)$$

where

$$S_j^1 = \sum_{p' \in P_j} q_{j,p'} \rho_{j,p'} \quad \text{and}$$

$$S_j^2 = \sum_{p' \in P_j, h_{j,p'}^{-1} \in C_i^-} \left(q_{h_{j,p'}^{-1},p'} \rho_{h_{j,p'}^{-1}} (1 - \theta_{h_{j,p'}^{-1},i}) \right)$$

S_j^1 is for transmission from neighbor j and S_j^2 is for transmission to neighbor j from nodes that are not i neighbors. Note that node i is aware of transmissions to its neighbors from the CTS and ACK messages.

The probability that the next successful transmission is from node i given that there has been a successful transmission in its neighborhood is,

$$\gamma_{i,p} = \frac{q_{i,p}}{r_{i,p}}$$

Let $Q_{i,p}$ be the average number of successful transmissions in the neighborhood of i per successful transmission from node i,

$$Q_{i,p} = \frac{1 - \gamma_{i,p}}{\gamma_{i,p}}$$

Then, $u_{i,p}$ is,

$$u_{i,p}^{new} = Q_{i,p} \tau_P \tag{2.14}$$

For $c_{i,p}$ we need to compute: (1) $x_{i,p}$ the probability of successful transmission of node i given that there is at least one transmission in its neighborhood and (ii) $y_{i,p}$ the probability that a transmission fails in the neighborhood of i given there has been a transmission. Let $z_{i,p}$ be the probability of at least one transmission in the neighborhood of node i, when it is scheduled to transmit path p packets:

$$z_{i,p} = 1 - (1 - \alpha''_{i,p}) \prod_{j \in C_i} \left(1 - (\sum_{p' \in P_j} \alpha''_{j,p'} \rho_{j,p'})(1 - \theta_{ji}) \right)$$

The probabilities $x_{i,p}$ and $y_{i,p}$ can be computed using conditional probability rules,

$$x_{i,p} = \frac{q_{i,p}}{z_{i,p}} \quad \text{and} \quad y_{i,p} = 1 - \frac{r_{i,p}}{z_{i,p}}$$

Then, the average number of collisions during transmission time is $y_{i,p}/x_{i,p}$ and the average collision time is

$$c_{i,p}^{new} = \frac{y_{i,p}}{x_{i,p}} w_{i,p} \tag{2.15}$$

where $w_{i,p}$ is the average time consumed for failure transmissions in the neighborhood of i:

$$w_{i,p} = \frac{\alpha''_{i,p}\beta_{i,p}f_{i,p} + \sum\limits_{j \in C_i} S_j^3(1-\theta_{j,i})f_{j,p'}}{\alpha''_{i,p}\beta_{i,p} + \sum\limits_{j \in C_i} S_j^3(1-\theta_{j,i})}$$

where $S_j^3 = \sum\limits_{p' \in P_j} \alpha''_{j,p'}\beta_{j,p'}\rho_{j,p'}$

Now, we have computed all 4 components of the average transmission time and can update it:

$$T_{i,p}^{new} = s_{i,p}^{new} + u_{i,p}^{new} + b_{i,p}^{new} + c_{i,p}^{new} \tag{2.16}$$

In the fixed point iteration, after computing the new variables, we calculate the maximum difference between the new and old computed average transmission times:

$$\Delta_{\text{inner loop}} = \max_{i,p} \left| T_{i,p}^{new} - T_{i,p} \right|$$

If $\Delta_{\text{inner loop}} < 1$, we consider that the fixed point iteration is converged, otherwise the inner loop is repeated.

2.2.2 Outer loop Computations:

After convergence of the fixed point parameters in the inner-loop, we use them to update the θ and β parameters in the outer loop. Recall that $h_{n,p}$ and $h_{n,p}^{-}$ are the

next and previous hops of node n in path p respectively. Then, $\theta_{i,j}$, probability of transmission from neighbors of i that are hidden from j (when there is no detected transmission in the neighborhood of j), is:

$$\theta_{i,j}^{temp} = 1 - \prod_{n \in C_i \bigcap C_j^-} \left(1 - \frac{S_n^4 + S_n^5}{1 - S_n^6}\right) \tag{2.17}$$

where

$$S_n^4 = \sum_{p' \in P_n, h_{n,p'} \in C_j^-} (\frac{\upsilon_{n,p'}}{T_{n,p'}} \rho_{n,p'})$$

$$S_n^5 = \sum_{p' \in P_n, h_{n,p'}^- \in C_i^- \bigcap C_j^-} (\frac{\upsilon_{n,p'}}{T_{n,p'}} \rho_{n,p'})$$

$$S_n^6 = \sum_{p' \in P_n, h_{n,p'} \in C_j^+} (\frac{\upsilon_{n,p'}}{T_{n,p'}} \rho_{n,p'})$$

S_n^4 is the probability of transmission from hidden node n to another node not in the vicinity of node j. S_n^5 is the transmission probability to hidden node n which also keeps the channel busy (due to CTS). S_n^6 is the probability of transmission from hidden node n to neighbor of node j, of which the node j is aware-of.

As mentioned in Section 2.1.2, we update θ with some memory to avoid oscillations. To that end, at each iteration, the new value of θ is convex combination of the old value and the *temp* value derived in (2.17),

$$\theta_{i,j}^{new} = \epsilon \theta_{i,j}^{temp} + (1 - \epsilon)\theta_{i,j} \tag{2.18}$$

In our model, we set $\epsilon = 0.1$, so that the θ parameter changes are gradual and the parameters converge. We use the computed θ^{new} parameters to update the link

loss parameters β in a similar way,

$$1-\beta_{i,j}^{temp} = (1-l_{i,h_{i,p}})(1-\theta_{h_{i,p},i}^{new}) \times \prod_{j\in C_{h_{i,p}}^{+}\bigcap C_i}\left(1-\sum_{p'\in P_j}\alpha_{j,p',h_{i,p}}\right) \times \prod_{j\in C_{h_{i,p}}^{+}\bigcap C_i^{-}}\left(1-\sum_{p'\in P_j}\alpha_{j,p',h_{i,p}}\right)^{V_{i,p}} \tag{2.19}$$

The first product term is the probability of no new transmission from those neighbors of $h_{i,p}$ that are not hidden from i, In the second product term, $V_{i,p} = T_{\text{RTS}}(i,p)$+SIFS is the vulnerable period during which those neighbors of $h_{i,p}$ that are hidden from i are not aware of the ongoing transmission and may cause collision. Therefore, the second product term is the probability of no new transmission from those neighbors of $h_{i,p}$ that are hidden from i during the vulnerable period. Then we update β with memory,

$$\beta_{i,p}^{new} = \epsilon\beta_{i,p}^{temp} + (1-\epsilon)\beta_{i,p} \tag{2.20}$$

After computation of new θ and β variables in the outer loop, we compute the maximum difference between the new and old values,

$$\Delta_{\text{outer loop}} = \max_{i,j,p}\left((\theta_{i,j}^{new}-\theta_{i,j}),(\beta_{i,p}^{new}-\beta_{i,p})\right) \tag{2.21}$$

The iterative algorithm is continued until $\Delta_{\text{outer loop}} < 0.01$. At this stage the FPA model is considered converged and the parameters are used to estimate system performance. In particular, offered and carried load for each path can be obtained from $\lambda_{src,p}$ and $\lambda_{dest,p}$, respectively, where src and $dest$ are the source and destination for path p. Denoting by P_c the set of paths used in connection c, the throughput of

connection c between src and $dest$ is obtained as

$$T_c = \frac{\sum_{p \in P_c} \lambda_{dest,p}}{\sum_{p \in P_c} \lambda_{src,p}} \tag{2.22}$$

2.3 Sensitivity Analysis and Network Optimization

Estimating the system performance using the solution to the system of equations is not sufficient for systematic design and trade-off analysis of systems. For the latter we also need to quantify reliability and robustness of the solution using sensitivity and perturbation analysis. However, due to the complexity of relations of the component models, it is not possible to compute the derivatives analytically. Numerical methods such as Automatic Differentiation (AD) can be used to obtain the derivatives and sensitivity parameters. AD [33] provides the partial derivative of the performance metric (e.g. throughput) with respect to the input parameters (i.e., design variables or parameters). We can use the computed derivatives in gradient-based optimization methods to improve the performance. We use optimal routing design as an example to illustrate our proposed design methodology. But trade-off analysis and optimization can be carried out with many different objectives. For example, throughput sensitivity to backoff window size, retry limit, etc. can be obtained using AD. Using these sensitivity and gradient values, we can find optimal backoff window size. This setup is similar to the one introduced in [6], but we summarize the method.

Given the network topology and traffic demands, we find multiple paths and optimally split traffic along these multiple paths so as to achieve optimal network

throughput. We implement the Dreyfus K-shortest path algorithm [35] for path selection. We use the gradient projection method to find the optimal values for the routing probabilities to maximize the network throughput. The gradient projection method requires iterative computation of the throughput gradient. The fixed point method provides a computational scheme that, after convergence, describes the performance metric (i.e. throughput) as an implicit function of the design parameters (i.e. routing parameters). Since we do not have analytic expressions of the performance metric, we use ADOL-C [36] for obtaining the parameter sensitivities using AD.

Now we present the methodology to optimize the overall throughput in the network by changing the path probability distribution of each connection on the network. The total network throughput T is:

$$T = \frac{\sum_{c \in C} (\sum_{p \in P_c} \lambda_{last,p})}{\sum_{c \in C} (\sum_{p \in P_c} \lambda_{first,p})} \tag{2.23}$$

Then, assuming there are $m = |C|$ active connections in the network, n_c paths used in the connection c and denoting by $\pi_{i,c}$ the probability associated with using path i in connection c, we know that the total throughput is a function of these input probabilities, namely: $T = T(\pi_{1,c_1}, \cdots, \pi_{n_{c_1},c_1}, \cdots, \pi_{n_{c_m},c_m})$. Thus, the optimization problem is:

$$\begin{aligned} max\ T &= T(\pi_{1,c_1}, \cdots, \pi_{n_{c_1},c_1}, \cdots, \pi_{n_{c_m},c_m}) \\ \text{subject to} \quad & \sum_{i \in P_c} \pi_{i,c} = 1, \quad \forall c \in C \\ & \pi_{i,c} \geqslant 0, \quad \forall (i,c) \in P_c \times C \end{aligned} \tag{2.24}$$

We solve the optimization problem using gradient projection method. Denoting by $\overline{\nabla}_c$ the connection c average gradient, and by $\eta > 0$ the step size, the route probabilities are iteratively updated, to find the optimal solution, as follows:

$$\pi_{i,c_k} = max\ (0, \pi_{i,c_k} + \eta(\frac{\delta T}{\delta \pi_{i,c_k}} - \overline{\nabla}_{c_k})), \forall k \in \{0, \cdots, m\} \tag{2.25}$$

2.4 Results

We run two sets of simulation studies. We first validate the model by comparing the FPA throughput estimates for different network scenarios with those obtained through network simulations. We use OPNET Modeler [37] for network simulation. For each scenario under study, we increase the offered load on each connection proportionally, and see how much of the load is carried to the destination. In the second set, we use AD to optimize the overall network throughput. In particular, we find the optimal routing probability / traffic splits between different routes so that the overall network throughput is maximized. We also validate the performance improvements from optimal route splitting through network simulations.

In each of the scenarios, the network topology is specified along with the traffic demands (offered load) along with the multihop paths for each of the connections. The nodes positions are fixed for each scenario. In order to focus on the MAC, we assume unit disc model for communication range with zero PHY losses within the disc. Hence the links can be specified using an adjacency matrix. The traffic demands are specified by setting up UDP traffic at the source nodes. The traffic source generates packets of fixed length (8384 bits) and with constant inter-arrival

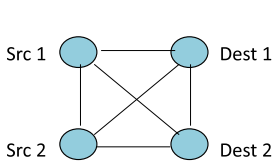
time, the interval depending upon the desired load. We modify the MANET routing protocol to use pre-specified static routes instead of standard routing protocols like OLSR or AODV. We use the 802.11b MAC with 11 Mbps data rate. However, the control messages (RTS, CTS, ACK) are sent at 1Mbps. The minimum and maximum contention window size is 32 and 1024, and the maximum retry limit is 4. The RTS threshold is set to a low value so that the RTS-CTS mode is always used. We use the free-space propagation model for wireless channel. Zero PHY layers losses (unit disc model) are achieved by modifying the BER curves. If transmission of two packets overlaps, they are declared as lost.

2.4.1 Validation of Throughput Approximation Models

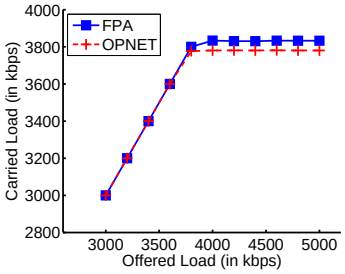
At first, we use simple scenarios to validate the 802.11 MAC model. We look at different two-edge topologies as shown in Figs. 2.1(a),2.2(a),2.3(a),2.4(a). The edges in the topology graphs show the communication and interference links. These four scenarios cover the different possibilities of interference between two simultaneous transmissions and can be seen as building blocks for more complicated scenarios [7]. In each scenario, we assign equal traffic on both the sources and increase the traffic proportionally till the load is high. We do not study very high load scenarios where the traffic demand exceeds the network capacity and system goes into unstable region. For the coordinated nodes scenario, Fig. 2.1 shows how the FPA model throughput matches closely with the simulated one. Connection 1 and 2 should have same average throughput due to the symmetry. But due to the limitations of

the simulator, connection 2 achieved slightly lower throughput, which explains the larger difference in Fig 2.1(d). However, the overall (and the average) throughput is approximated very well by the model (Fig. 2.1(b)). Figs. 2.2, 2.3 and 2.4, show the corresponding results for the other three topologies. In all the scenarios, the FPA throughput gives good approximations to the throughput obtained from network simulations. The lower throughput for connection 1 in Fig. 2.2(c) can be attributed to higher estimates for hidden node transmission probability (θ). At the worst, the FPA throughput shows an error of 20 %. However, in all the scenarios, the FPA model follows the trends of increasing (or decreasing) traffic demands. They give good approximation to capacity (or saturation) throughput in those scenarios. This is an improvement from the previous model of [6], which in some scenarios gave very high throughput. For example, in Far Hidden Node scenario, the FPA would converge to give negligible link losses. However, the newer model does not suffer from such problems.

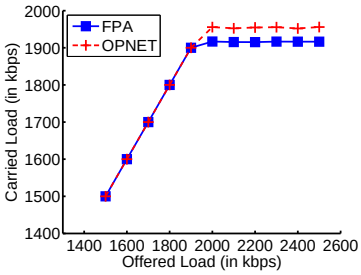
Next, we study a 30-node multi-hop scenario shown in Fig. 2.5. The overall carried load in FPA model matches well with OPNET. Table 2.1 shows the various connections, and Figs. 2.6, 2.7 and 2.8 show carried load for some of the connections. Single-hop connections achieve 100 % throughput for both FPA and OPNET (plots not shown). In this scenario, offered load on individual connections are not necessarily equal. The plots show that the carried load for individual connections is approximated reasonably well by the FPA. Some connections show higher approximation error due to the independence assumptions used in the model. However, the capacity for the overall network and longer connections is approximated well.



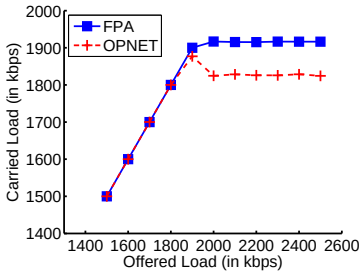
(a) Topology



(b) Overall Traffic



(c) Connection 1



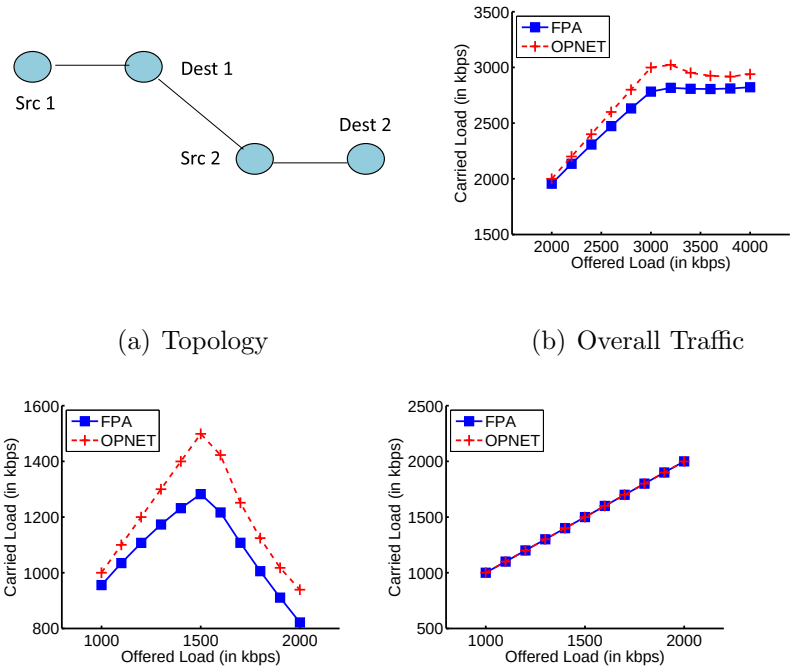
(d) Connection 2

Figure 2.1: Coordinated Nodes Scenario

Connections	**1-hop**	**2-hop**	**3-hop**	**5-hop**
(Src, Dest) pairs	(2,4), (5,7), (17,12), (18,19), (24,29), (24,26)	(3,10), (8,6), (11,2), (15,18), (20,13) (21,12), (21,30), (22,23)	(2,19), (22,11)	(21,1)

Table 2.1: Connections for the 30 node scenario

We validate the FPA model in many other scenarios. In all the scenarios, the fixed point algorithm converged quickly to give good approximation to connection and network throughput.



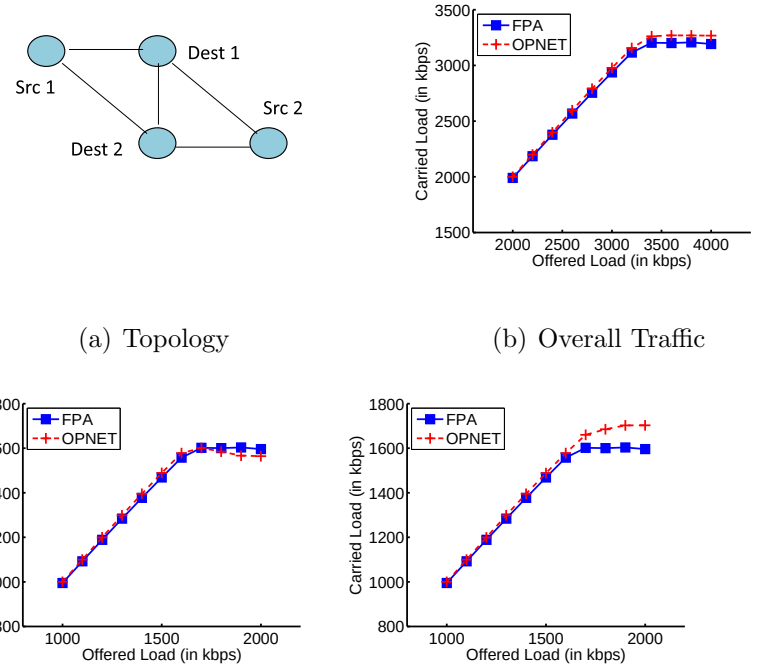
(a) Topology (b) Overall Traffic

(c) Connection 1 (d) Connection 2

Figure 2.2: Asymmetric Nodes Scenario

2.4.2 Throughput Optimization Using AD

Now we set multiple paths per connections in the 30-node scenario, and use the optimization framework (Sec. 2.3) to demonstrate the utility of the FPA model. At first we assign equal traffic splits between multiple paths and compare the FPA and OPNET traffic. Then we run optimization algorithm using AD in the FPA model to find optimal traffic splits to maximize the network throughput. Using these optimized traffic splits, we run the OPNET simulations to obtain the newer throughput. As can be seen from Fig. 2.5, the carried load in FPA approximates well the actual carried load. Moreover, despite the small error in the throughput approximation, route-split optimization leads to improved performance in network



(a) Topology

(b) Overall Traffic

(c) Connection 1

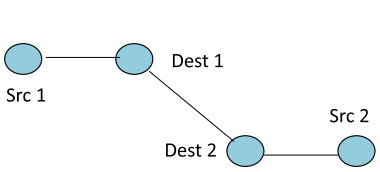
(d) Connection 2

Figure 2.3: Near Hidden Nodes Scenario

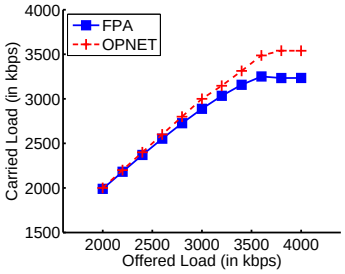
simulations. Again, similar experiments with different topologies yielded the same results: predicted increase in throughput due to route-split optimization in FPA model are validated through network simulations.

2.5 Summary

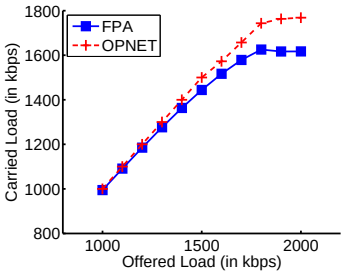
We presented a loss model for design and optimization of multi-hop wireless networks. Given any static multi-hop topology and offered load on various connections, the model finds average throughput, link losses and other MAC parameters through a fixed point approach. The FPA algorithm converges to give good performance estimates. We validated our model by comparing the throughput estimates with the empirical throughput from network simulations. We also demonstrated



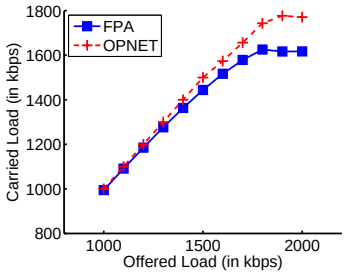
(a) Topology



(b) Overall Traffic

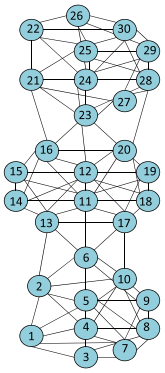


(c) Connection 1

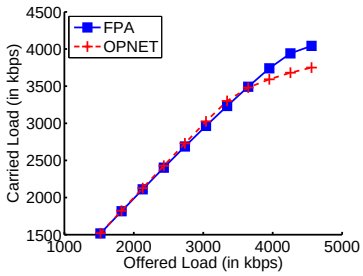


(d) Connection 2

Figure 2.4: Far Hidden Nodes Scenario



(a) Topology



(b) Overall Carried Load

Figure 2.5: 30-node scenario

the utility of the model in network performance optimization, by performing sensitivity analysis based on model parameters. Using AD, we saw how the network throughput varies by rerouting some traffic along multiple paths. We demonstrated

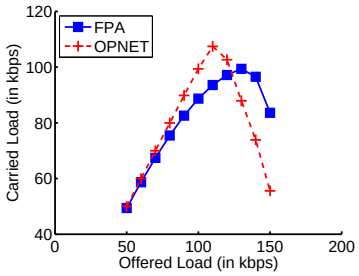
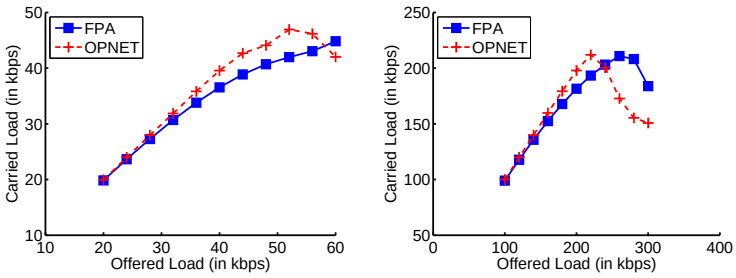


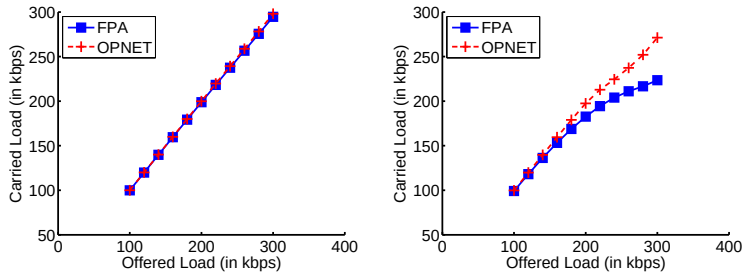
Figure 2.6: 5-hop connection in 30-node scenario



(a) Connection (2,19) (b) Connection (22,11)

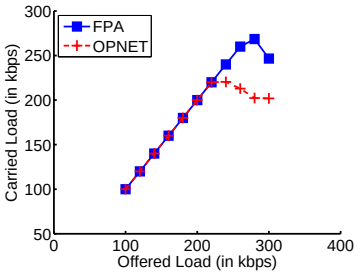
Figure 2.7: 3-hop connections in 30-node scenario

that by optimizing the traffic splits using the FPA model, the network throughput can be increased. We also validated the performance improvements using network simulation results.

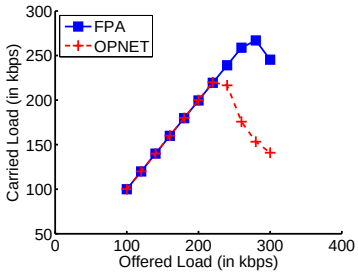


(a) Connection (3,10)

(b) Connection (8,6)



(c) Connection (21,30)



(d) Connection (22,23)

Figure 2.8: 2-hop connections in 30-node scenario

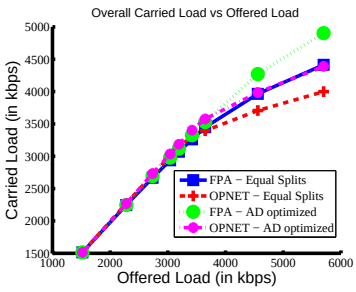


Figure 2.9: Optimized Carried Load in 30-node scenario

Chapter 3: Component Based Modeling for Cross-layer Analysis of 802.11 MAC and OLSR Routing Protocols in Ad-hoc Networks

A large number of routing protocols have been proposed for MANET. However, to date most of performance studies are based on discrete event simulations, with very few exceptions. Routing protocols are commonly implemented as large monolithic software, which are very difficult to adapt to varying conditions and mission scenarios. Identifying the performance bottlenecks by these simulations is very time consuming, since we have to run a large number of scenarios to pin-point the bottleneck. Because of the complex nature of the simulations, they provide little insight on design parameter sensitivities or on how we can improve performance and adaptivity of these routing protocols. Furthermore, performance of routing protocol depends a lot on the MAC protocol. Since the process of gathering topology information depends on topology packet flooding, performance evaluation of a routing protocol cannot be done correctly without considering the underlying MAC. Similarly, the performance of a MAC protocol depends on the traffic load offered and the routing used to handle the traffic. Hence, to model the wireless network performance correctly, cross-layer interaction between the routing and MAC protocols

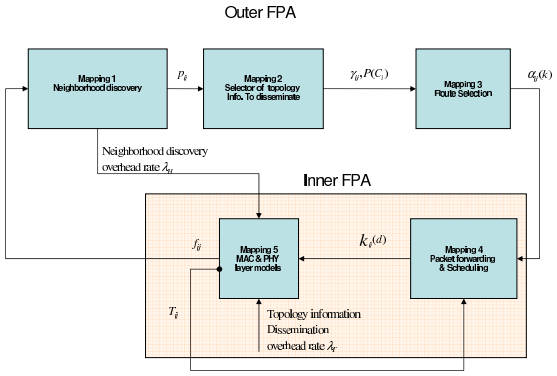


Figure 3.1: Components for topology driven performance modeling

has to be considered.

The interaction and overall performance of the MAC and routing protocols in an ad-hoc network can be modeled by dividing the various parts of the protocols into components. In this approach a proactive routing protocol is divided into three main components - (i) Neighbor Discovery Component (NDC), (ii) Selector of Information for Dissemination Component (STIDC), and (iii) Route Selection Component (RSC). These routing components are coupled with the Scheduler and the MAC components. Figure 3.1 illustrates the interdependence of the various components. While the performance of the components are coupled, each of them can be independently evaluated and designed for better performance.

In [8], Baras et. al. describe and develop the routing components - namely NDC, STIDC and RSC, based on the OLSR protocol [38]. This approach of component based routing was first introduced in [39]. However, their model assume fixed and known MAC layer loss probabilities. On the other hand, MAC performance models (including our 802.11 MAC model discussed previously), do not study the impact of routing protocol. Performance of any routing protocol depends a lot on

the underlying MAC protocol. Since the process of gathering topology information depends on topology packet flooding, performance evaluation of a routing protocol cannot be done correctly without considering the underlying MAC. Similarly, the performance of a MAC protocol depends on the traffic load offered and the routing used to handle the traffic. Hence, to model the wireless network performance correctly, cross-layer interaction between the routing and MAC protocols has to be considered. Hence, we jointly consider the OLSR and 802.11 MAC protocols and focus on the cross-layer interaction. While OLSR model in [8] considers fixed MAC loss probabilities for the links, our 802.11 MAC model from previous chapter assumes pre-specified static routes. Here, we specify appropriate modifications to the components for both the models and close the loop, taking cross-layer effects into account. We then describe a fixed point algorithm to find a consistent set of solution.

The combined component models *approximate* the performance of routing and MAC protocols, but enable us to evaluate and analyze the cross-layer interaction under various network topologies. For specified static network topology, traffic demands and physical layer losses, the models evaluate the performance metrics of the various components and find out network performance metrics - average throughput and average end-to-end delay for each connection. By analyzing the performances under varying network scenarios, we can identify the sources of performance degradation and try to improve the corresponding components. We can also study the trade-offs and choose the component design parameters appropriately. To illustrate, we study the performance of a few network topologies under varying design param-

eters for NDC.

The chapter is organized as follows: In Section 3.1, we describe the routing components based on OLSR and their extensions. In Section 3.2, we describe the modifications to 802.11 MAC models introduced earlier. Section 3.3 gives the details about the fixed point algorithm to integrate the MAC and routing performance models. In Section 3.4, we use the component models to evaluate network performance and study the effects of certain design parameters.

3.1 The Routing Components

A proactive routing protocol in MANET can be divided into three tasks - (i) gather local neighborhood information, (ii) flood a pruned version of the local information, and (iii) select routes based on the available information. These parts are abstracted into - (i) Neighbor Discovery Component (NDC), (ii) Selector of Information for Dissemination Component (STIDC), and (iii) Route Selection Component (RSC), respectively.

In this section, we describe the NDC performance model, how it depends on outputs from the MAC model, and describe the design parameters for the NDC. We refer the reader to [8] for a detailed exposition of modeling approaches to STIDC and RSC components, but give a short description of their interplay with the MAC model.

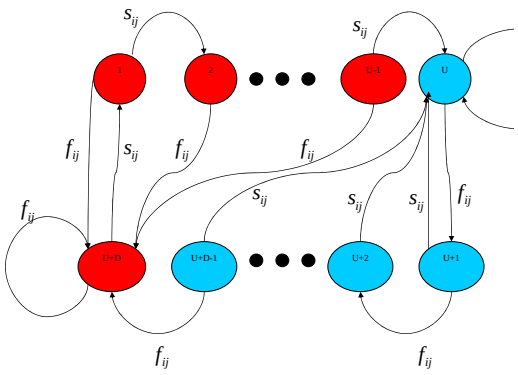


Figure 3.2: FSM of neighbor detection mechanism

3.1.1 NDC

The neighbor discovery methods in ad hoc networks are usually driven by proactive HELLO packet broadcasts. Based on the reception of these HELLO packets, node identify their neighbors. Since these packets are susceptible to channel and contention losses, the local neighborhoods as seen by the nodes depends upon the underlying link loss probabilities.

Based on the link detection criteria and link loss probabilities, a state space model for these neighbor discovery methods has been introduced in [8]. A node declares a new unidirectional link as *up* if it receives U consecutive HELLO messages and declares a unidirectional link as *down* if it does not hear D consecutive HELLO messages. In OLSR [38], the values of U and D will correspond to 1 and 3 respectively.

The Finite State Machine (FSM) for the neighbor discovery algorithm is shown in Figure 3.2. Each station i in the network runs the FSM for every station j in its radio range. Whenever this station receives a HELLO packet from j it corresponds to

a decision edge $s_{i,j}$ and when the HELLO packet is lost in transmission it corresponds to a decision edge $f_{i,j}$. Station i declares a unidirectional link $i \rightarrow j$ if it is in any one of states from U to $U+D-1$. Otherwise it declares a unidirectional link failure. This FSM then forms the *executable model* for the NDC in system design.

The corresponding *performance model* which captures the steady state behavior of the FSM can be obtained using Markov chain analysis. The inputs to this model are the probabilities of success ($s_{i,j}$) and failure ($f_{i,j} = 1 - s_{i,j}$) in transmission of HELLO messages. These inputs, which represent the link loss probabilities, come from the MAC model.

Let $\pi_k(f_{i,j})$ be the steady state probability that the NDC is in state k of the Markov chain, defined as a function of $f_{i,j}$. We can use the generalized global balance equations to derive the steady state probabilities. One of the main NDC performance metrics is the probability of detecting a directional link to node j at node i and is given by:

$$q_{i,j} = \sum_{k=U}^{U+D-1} \pi_k(f_{i,j}) \tag{3.1}$$

and if we assume that the probability of successful transmission from i to j and from j to i are independent from each other, then the probability of a bidirectional link detection is:

$$p_{i,j} = q_{i,j} \cdot q_{j,i} \tag{3.2}$$

Thus, even for given link loss probabilities, the performance metric - probability of detecting a bidirectional link, depends on the the choice of U and D. Hence

they can be set to achieve the desired performance and form the design (or control) parameters for NDC.

3.1.2 STIDC and RSC

The STIDC specifies the topology graph that is presented to the network nodes. For each node, STIDC determines a set of important links which are broadcast in the network. Because of losses in HELLO packets, the local neighborhood information changes even in a static topology and so do the links that are broadcast by STIDC. RSC then uses this subset of links to find out routes to the various destination nodes. Again, because of changing topology information, the routes keep changing.

In terms of the performance models, the probabilities of bidirectional link detection from NDC form the input to STIDC. The output of the STIDC is the probability mass function $P(C_i)$ for the state vectors C_i determining the set of important links for node i. Outputs from STIDC and NDC are used by the RSC to find routing probabilities at the nodes for various destinations. The routing probability $\alpha_{i,j}(k)$ specifies the probability of node i choosing node j as the next hop to destination k. In a way, $\alpha_{i,j}(k)$ is the fraction of time node i sees a graph in which j is the next hop on the shortest path to k. Depending on the outputs of NDC and STIDC, a node i may not find any path to the destination k. This amounts to $\sum_{\forall j} \alpha_{i,j}(k) \neq 1$, and the difference $1 - \sum_{\forall j} \alpha_{i,j}(k)$ representing the fraction of time, there is no path from i to k. Similarly, the NDC output may be such that there is

no path from i to k at any time, making $\alpha_{i,j}(k) = 0,\ \forall j$. We refer the reader to [8] for details about these component models.

The performance models of STIDC and RSC do not depend directly on MAC, but are affected indirectly through NDC. Depending upon the MAC layer losses, the probability of detecting bidirectional links at NDC will change, thereby changing the performance of STIDC and RSC. For example, if all the links have high MAC layer losses, the network as seen after NDC and STIDC will seem disconnected most of the time, and the routing probabilities will be low.

The cross-layer effect takes place in the reverse direction through the routing probabilities. The output from RSC is an input to the Scheduler and MAC models, and hence directly affect their performance. In particular, MAC efficiency or throughput for CSMA/CA or 802.11 MAC depends on the load on the wireless channel. Thus, MAC throughput for one node depends on the load on it's neighbors' as well. Hence, the routes chosed by the routing protocol for multi-hop connections impact the MAC performance itself. We describe the Scheduler and MAC models in the next section.

3.2 The MAC components

The performance of any MAC protocol can be broadly divided into (i) Scheduler modeling, which determines the amount of traffic demand on each link, and (ii) MAC modeling which determines the actual traffic flow, link delay and link losses, etc. Detailed Scheduler and MAC models for 802.11 MAC are developed in Ch. 2.

However, they assume pre-specified static routes. Since in our model, the routes are indirectly specified in-terms of routing probabilities, we need to modify the scheduler and MAC equations. In this section we briefly describe the models focusing on the relevant part, while referring the reader to the originals for more details.

3.2.1 Scheduler Modeling

Given the input traffic demand for various source-destination pairs (identified by unique connection IDs), and the routing probabilities for each destination (from RSC), the scheduler model finds out the average traffic at each node and the fraction of time spent by the MAC serving a particular connection traffic.

The average rate of packets that are served at source node i for connection d using node j as the next hop is denoted by $k_{i,j}(d)$ and is given by the following relationship:

$$k_{i,j}(d) = \begin{cases} \lambda_{i,j}(d) & \text{if } \sum_{\forall j} \sum_{\forall d} k_{i,j}(d) \cdot E(T_{i,j}) \leq 1 \\ \frac{\lambda_{i,j}(d)}{\sum_{\forall j} \sum_{\forall d} k_{i,j}(d) \cdot E(T_{i,j})} & \text{otherwise} \end{cases} \tag{3.3}$$

where $\lambda_{i,j}(d)$ denotes the arrival rate of packets at node i for connection d that will use node j as the next hop. $E(T_{i,j})$, is the average service time to send a packet from node i to node j. We have two possibilities when scheduling packets at node i: Either the utilization of node i is less than 1, and we can serve all incoming packets (as is described in the first line of Equation (3.3)) or we have to normalize the scheduler coefficients by the utilization of the node (as is described in the second line of the equation). Here we have assumed that all outgoing traffic receive proportional link resources. Then, the total rate of packets served at source i for next hop j (for all

connections) is given by:

$$k_{i,j} = \sum_{\forall d} k_{i,j}(d) \tag{3.4}$$

The net arrival rate of packets at node i for connection d is given by:

$$\lambda_i(d) = \begin{cases} input\ rate & \text{if } i \text{ is the source for connection } d \\ \sum_{\forall l} k_{l,i}(d) \cdot (1 - \beta_{l,i}^m) & \text{otherwise} \end{cases} \tag{3.5}$$

where $\beta_{i,j}$ is the probability of transmission failure at the MAC layer for the link $i \rightarrow j$. m represents the maximum number of retries at the MAC before the packet is dropped at the MAC. Each term in the second line of Eq. 3.5 gives the traffic received at node i from node l, which is the scheduler rate at node l times the probability of a successful transmission over the link (l, i). Summing over all such nodes l, we get the total traffic at node i. Now, depending on the routing probabilities, node i will route the packets through node j. Thus, the arrival rate of packets at node i meant for next hop j is given by:

$$\lambda_{i,j}(d) = \alpha_{i,j}(dest(d)) \cdot \lambda_i(d) \tag{3.6}$$

where as described in section 3.1.2, $\alpha_{i,j}(dest(d))$ is the probability of node i choosing node j as the next hop for destination $dest$ of connection d.

The fraction of time that a node i is serving a packet going through node j for connection d is:

$$\rho_{i,j}(d) = k_{i,j}(d) \cdot E(T_{i,j}) \tag{3.7}$$

And, the fraction of time node i is serving packets for next hop j:

$$\rho_{i,j} = \sum_{\forall d} \rho_{i,j}(d) \tag{3.8}$$

3.2.2 MAC Modeling

Depending upon the traffic on each link (given by $k_{i,j}$ and $\rho_{i,j}$) the MAC equations find out the probability of MAC layer transmission failure $\beta_{i,j}$, and packet service time $E(T_{i,j})$. The MAC equations model the actual traffic on the link by taking into account different parameters like contention for channel, transmission by hidden nodes, back-offs on collision, etc. We use the same equations as in Ch. 2, except for a slight change in notation. While Ch. 2 indexes each of the link variables by current node i and path p, we index the link variables by the directional link (i, j). This is required due to the modified routing model - the routes are no longer specified by pre-defined paths, but defined through the routing probabilities from RSC. The MAC component currently does not take contention and collision, due to topology control and HELLO packets, into account. We assume these control packets do not cause any significant increase in contention and collision. This assumption is justified as they cause insignificant amount of traffic for high capacity MAC and PHY models such as 802.11.

3.2.3 MAC FPA

The various MAC variables are defined implicitly. The traffic demand on each link $k_{i,j}(d)$ in the scheduler component, depends on the link metrics $\beta_{i,j}$ and $E(T_{i,j})$, while the link metrics themselves depend on traffic demand on the links. To find a consistent solution for the MAC variables, we use a fixed point algorithm. We call this the *MAC FPA* (which itself has two loops - inner loop and outer loop).

Given the input traffic demand $\lambda_i(d)$ for the source nodes for each connection d and the routing probabilities $\alpha_{i,j}(k)$, we initialize the MAC variables appropriately and iterate over the MAC model equations till we reach a fixed point. for more details.

3.2.4 Performance metrics

$\lambda_i(d)$, $\beta_{i,j}$ and $E(T_{i,j})$ form the main performance metrics of the MAC model. We use $\lambda_i(d)$ and $E(T_{i,j})$ to find out network performance metrics - throughput and delay (described later in Section 3.3). The link loss probability $\beta_{i,j}$, although determined for data packets, gives a good approximation for the link loss probability for HELLO packets because they both use the same MAC protocol. Hence in our model, we currently assume $f_{i,j} = \beta_{i,j}$, that is, the probability of failure of HELLO packet transmission over a link is the same as the probability of MAC layer transmission failure. Thus, the performance metric of MAC, $\beta_{i,j}$, forms the input $f_{i,j}$ to NDC and closes the loop (Fig. 3.1).

3.3 Integration of Routing and MAC components

In the previous sections, we described how outputs from one component are fed to others (Fig. 3.1). In particular, we described how the routing components take inputs from the MAC components and the MAC components in turn, take inputs from routing components. To determine a set of consistent solutions for the various component outputs, we run another fixed point algorithm on this outer loop. We call this *Outer FPA*.

We start the loop assuming zero link loss probabilities $\beta_{i,j}$ for nodes within radio range. Given the link loss probabilities, we calculate the routing probabilities $\alpha_{i,j}(k)$ from the routing components. With input traffic demand and these routing probabilities, the MAC components (MAC FPA) calculate the new $\beta_{i,j}$, which form the input $f_{i,j}$ to the NDC. We iterate over this loop till we reach a fixed point. We say a fixed point is reached when link loss probabilities for all the links have converged. To avoid oscillations, we use some memory η on $\alpha_{i,j}(k)$ and $\beta_{i,j}$, and update the values in each iteration as:

$$\alpha_{i,j}^{new}(k) = \eta \cdot \alpha_{i,j}^{old}(k) + (1 - \eta) \cdot alpha_{i,j}^{current}(k) \tag{3.9}$$

$$\beta_{i,j}^{new} = \eta \cdot \beta_{i,j}^{old} + (1 - \eta) \cdot \beta_{i,j}^{current} \tag{3.10}$$

After reaching the fixed point for the Outer FPA, we get the desired performance metrics of the various components. Additionally, we obtain the throughput and end-to-end delay for each source-destination pair. Throughput is calculated as $\lambda_{dest}(d)/\lambda_{src}(d)$ where src is the source node and $dest$ is the destination node for connection d. To calculate the end-to-end delay, we first find average delays over each link $LD_{i,j}$ using the link layer statistics from the MAC component. (We assume an M/M/1 queuing model at each node, with arrival rate λ_i obtained from $\lambda_{i,j}(d)$ and service time obtained from $E(T_{i,j})$). Then for each connection, the end-to-end delay $D_{src,dest}$ is calculated using the equations :

$$D_{i,dest} = \sum_{\forall j} \alpha_{i,j}(dest) \cdot (D_{j,dest} + LD_{i,j}) \tag{3.11}$$

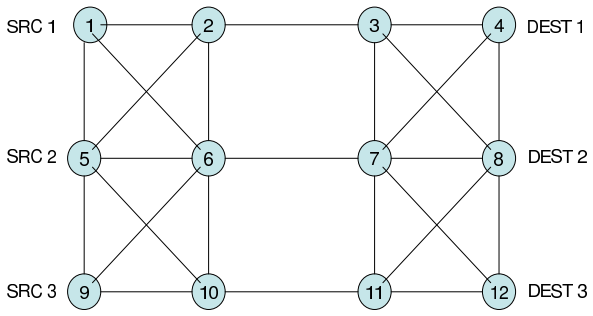


Figure 3.3: Physical layer topology of the 12 node network

3.4 Simulation Results

We run the FPA on two sample topologies for varying input parameters. The first set of simulations look at a simple network, but demonstrate the effect of NDC design parameters U and D. The second set of simulations look at a more complicated network and analyze the cross-layer effects.

At first, we consider a 12 node topology with 3 connections (Fig. 3.3). For varying NDC parameters U and D, we run the simulations for increasing load on each connection. Fig. 3.4 shows the overall throughput for the network for increasing load factors and parametrized by U and D. Fig. 3.5 shows the throughput for each connection and Fig.3.6 shows the corresponding end-to-end delays. In these figures, a load factor of k corresponds to each connection having a traffic demand of $k * 100\ kbps$. We see that as we go on increasing the offered load, the throughput for each connection goes on decreasing and delay keep increasing as expected. Furthermore, for some load factors, the network performance metrics are very sensitive to the

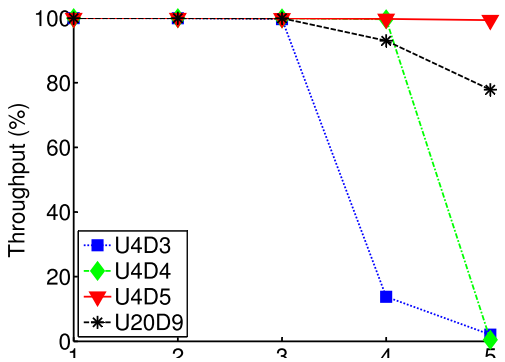
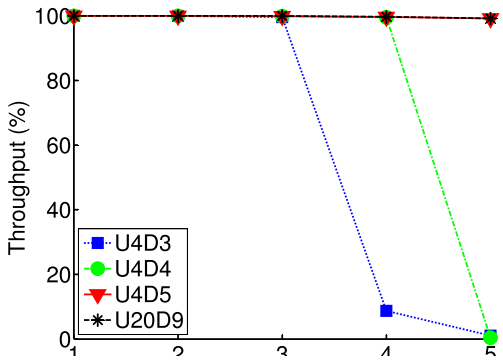


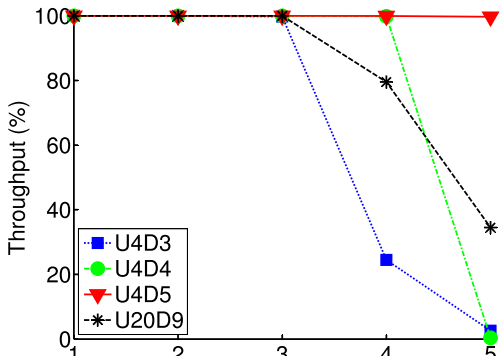
Figure 3.4: Total Throughput for the 12 node network

values of U and D.

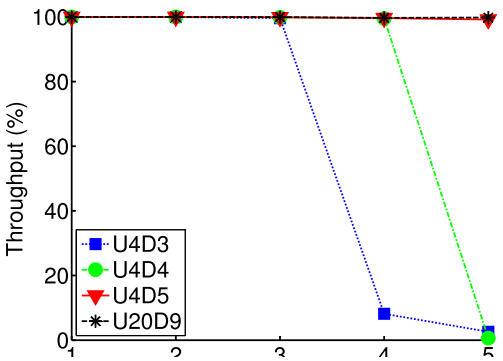
Focusing on the scenarios with NDC parameters $U = 4$ and $D = 3, 4, 5$, we recall that a low value of D corresponds to faster discarding of the existing bidirectional links, similarly a higher value of D corresponds to retaining the links for longer time. For a low value of D, even a slight increase in load will increase the MAC loss probabilities, and the existing links will be discarded quickly by NDC. This explains the faster drop in throughput for the low value of $D = 3$. Similarly, higher values of D will ensure that the links are retained for a longer time. Hence better throughput for high $D = 5$. [8] proposed setting $U = 20$ and $D = 9$, for detecting stable links and discarding unstable links with high probability. For $U = 20$ and $D = 9$, we find that the network is able to carry more load for connections 1 and connection 3, but the carried load for connection 2 drops significantly. This is because, the link loss probabilities from node SRC2 to its neighbors even though only slightly higher, are high enough for the links to be not discovered by its NDC. Performance degradation in end-to-end delays follows similar trends. As the offered



(a) Connection 1

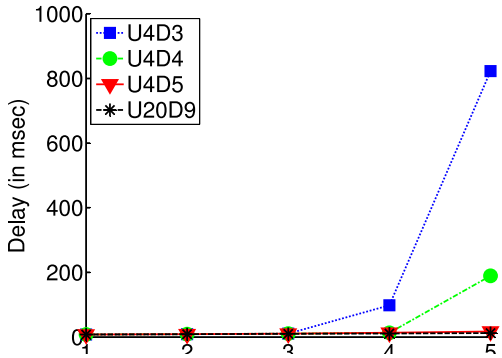


(b) Connection 2

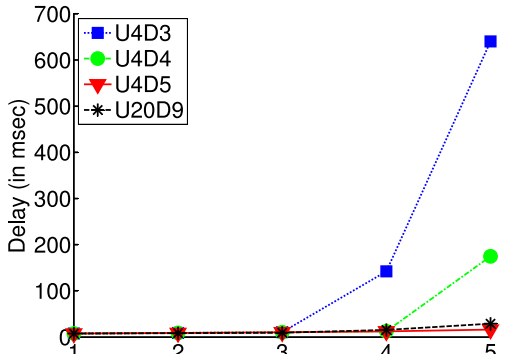


(c) Connection 3

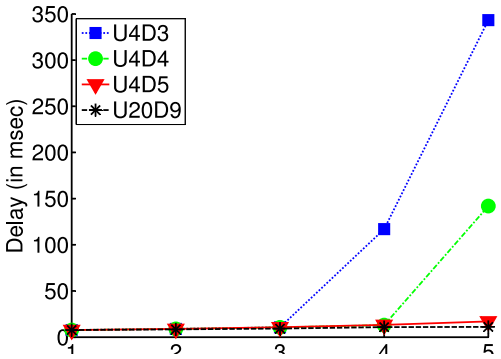
Figure 3.5: Connection Throughputs for the 12 node network



(a) Connection 1



(b) Connection 2



(c) Connection 3

Figure 3.6: Delays for the 12 node network

load increases, the increase in contention and collision, increases the delays. The drastic increase in end-to-end delay for low throughput scenarios is due to large number of retries and significant buffering.

Next, we run the simulations for a more complicated 20 node scenario (Fig. 3.7) with 10 connections. Fig. 3.8 shows the throughput for 2 of the 10 connections and the overall throughput. As before, the scale load factor is used to uniformly increase the traffic demand on each connection. The values of U and D are fixed to be 2. The two connections chosen in Fig 3.8 achieve the best and the worst throughput across the various scenarios. Connection 1, which experiences the worst throughput corresponds to one of the longest connections in the network. Similarly connection 9, which experiences the best throughput corresponds to one of the shortest connections. Table 3.1 gives the connection throughputs for the scenario with scale load factor of 1.25. From the table, we can infer the following: short connections (connections 4 and 9) or connections which can route from low traffic region (connection 3) achieve high throughput; short connections in the heavy traffic region or long connections (connections 2, 5, 7, 8, 10) have moderate throughput; and long connections which pass through heavy traffic region (connections 1 and 6) achieve the lowest throughputs. These results are consistent with the expected network performance taking cross-layer effects into account. Such insights in the network performance would not have been possible by using just the routing layer models, with fixed MAC parameters.

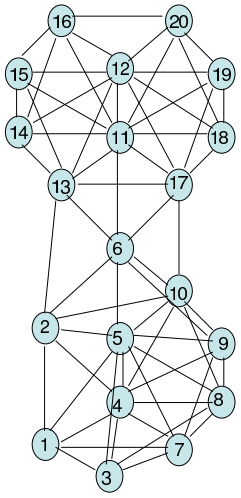


Figure 3.7: Physical layer topology of the 20 node network

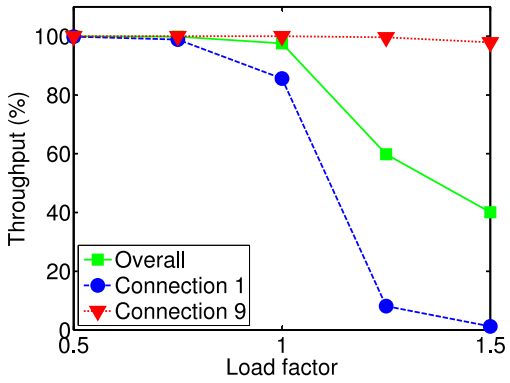


Figure 3.8: Throughputs for the 20 node network

Connection	**(Source, Destination)**	**Throughput (in %)**
1	(2,19)	8.1
2	(2,4)	69.9
3	(3,10)	90.5
4	(5,7)	95.8
5	(8,6)	57.1
6	(11,2)	12.1
7	(15,18)	37.3
8	(17,12)	43.3
9	(18,19)	99.6
10	(20,13)	42.8

Table 3.1: Connection Throughputs for the 20 node scenario with scale load 1.25

3.5 Summary

We presented a component based model for analysis and design of MAC and routing protocols for wireless networks. Taking cross-layer effects into account, we combined the performance models of 802.11 MAC [6] and generalized proactive routing protocol [8] based on OLSR. We proposed a fixed point algorithm to find a consistent solution for the interdependent component outputs and performance metrics. We studied the performance of the routing and MAC protocols under varying network settings and showed how the performance of one depends a lot on the other. Lastly, we demonstrated the effects of component design parameters - U and D for NDC on the network performance. We will revisit and reuse some of the component models developed here in Ch. 5. There we use the models to explain a new phenomenon observed due to cross-layer interaction of MAC and routing layer protocols.

Chapter 4: Role of MAC Protocol Dynamics in Traffic Scaling

Performance metrics in terms of average throughput, average delays or average losses, are the main focus of most studies regarding performance models. With decades of work on queuing theory, very good theoretical and tractable models have given significantly useful insights and enabled accurate performance analysis including second-order metrics and delay distributions. Even though there are some models for general systems, most of the closed form or analytical results are obtained for Markovian framework, where typically the input traffic is assumed to have Exponential distribution. However, one school of thought also attempts to build models for such systems with high input variability, especially since the observation of traffic self-similarity (defined later) in [9]. The related phenomenon of long-range dependence (LRD), which practically implies that the correlation (of traffic, in our case) does not die out even at large-timescales. Thus, LRD can break the Markovian assumption about independence and memoryless properties, leading to significantly higher queue occupancy, delays and losses. In some cases, the performance models can give incorrect estimates or insights because of traffic LRD.

Even the 802.11 MAC and OLSR models we developed before rely on the Markovian framework. Of course, our models have many simplifying assumptions,

but so far, we have shown them to be reasonably accurate under the simulation scenarios tested. However, our analysis and validation so far has not looked at the high-variability in traffic sources, and the impact of traffic LRD on network performance. We wish to evaluate their impact on our models. We hope to either validate the models in giving good estimates even in such scenarios, or else, we aim to appropriately modify out models to account for their (traffic LRD's) impact. But, before we analyze the impact of traffic LRD or traffic burstiness, we first need to study their existence in our networks of interest - wireless multi-hop networks.

Motivated by the previous studies in network traffic characteristics for wired networks, we strive to perform similar detailed analysis for wireless networks. The few studies on analysis of traffic in wireless networks [20–26] are not as detailed compared to previous studies in wired scenarios [10–13]. While the impact of IP, MAC and PHY layers in wired network traffic is less, wireless setting in expected to increase their impact. Shared wireless channel, contention and collisions due to hidden nodes, fading and mobility all can potentially change the traffic characteristics in a wireless setting. Also, the congestion control algorithms like TCP behave significantly differently because of packet losses due to mobility and fading. All these questions motivate us to analyze the traffic characteristics in wireless networks in detail. We start with simple settings for physical layer (minimal losses) and routing (static single hop topology), and focus on the MAC and transport layer. By using different source application demands, we analyze the traffic characteristics and study the impact of changing MAC and transport protocols and parameters.

Firstly, we characterize the role of MAC in introducing delays and attempt

to understand the maximum timescale of operations. Motivated by [28], we wish to find if MAC retransmissions can themselves introduce heavy-tails (defined later, but characterizes the high variability), and if MAC delays can really extend to large timescales of hundreds of seconds or large. In their work [28], Tan et al. show MAC retransmissions for ALOHA, under some conditions, can lead to delays with power-law distribution. However, in their theoretical analysis, they had to make simplifying assumptions including the absence of any queue (queue length of 1). To perform similar theoretical analysis for more realistic network scenarios with more complex MAC protocols based on CSMA/CA is difficult. In an older work [19], the authors characterized the inter-arrival time distribution for 802.11 MAC, which they show to be multi-modal. However, we believe their model made some simplifying assumptions and did not capture enough details to focus on the tail of the distribution, especially in light of Tan et al.'s work in [28]. For example, in Figs. 8 to 13 in [19], they focus on the body of the packet inter-arrival distribution, and do not look for tail characteristics. In Fig.14(a), they disregard the difference between the tail of the distribution and attribute the difference to higher-layer protocols. Hence, we revisit the problem of MAC retransmissions and service delays in 802.11 MAC. In this work, given the difficulty in coming up with good analytical models, we use simulation-based approach to investigate similar phenomenon under more realistic network scenarios with 802.11 MAC protocol. We find that under realistic operating conditions the MAC service times are bounded to a few seconds. Even a very large increase in MAC retry limit does not translate into larger service times. However, the MAC service times can show power-law distribution. Hence, even

though the service times are not pure heavy-tails, they do have power-law bodies, which introduces some traffic burstiness above few tens of seconds. One important aspect of this study was to see if heavy-tails in service times lead to traffic LRD. Heavy-tails in traffic sources have been attributed at the main reason for traffic LRD. Here, we wished to see if the network protocol themselves introduced heavy-tails and traffic LRD. Given that we did not see any true heavy-tails in MAC service times, we, certainly, found no evidence of traffic LRD due to MAC protocol dynamics alone. However, the power-laws in MAC service time indeed lead to some traffic scaling upto few orders of magnitude (tens of seconds) above the mean MAC service time (few milliseconds).

Then, having studied MAC separately, we investigate the combined role of MAC and transport layer in preserving traffic characteristics. In wired networks, it is well known that reliable transport like TCP preserves the heavy-tails in traffic sources (if present), and these heavy-tails in traffic sources translate into traffic LRD. However, unreliable transport UDP can truncate these heavy-tails and lead to absence of traffic LRD. We carefully study different wireless scenarios to understand the role of MAC and transport layers in destroying or preserving these results. We find that, reliability of transport protocols, load levels and MAC buffer sizes play an important role. Reliable TCP traffic is not affected by MAC buffers, and in such scenarios heavy-tails indeed lead to traffic LRD. However, finite buffer sizes in MAC for while carrying UDP traffic leads to losses, especially for heavy-tailed occurrences. This destroys the traffic LRD. By increasing the MAC buffer sizes, these losses are prevented and traffic LRD is preserved. We summarize our observations in Tab.

Scenario	Heavy-Tails in	Transport Protocol	MAC Buffer	LRD
1	Off Durations	UDP	Finite	Yes
2	Off Durations	TCP	Finite	Yes
3	File Sizes	TCP	Finite	Yes
4	File Sizes	UDP	Finite	No
6	File Sizes	UDP	Infinite	Yes

Table 4.1: Impact of MAC on Traffic LRD

4.1. To recap, there were some conjectures about the role of wireless networks and possibilities of new interesting phenomenons due to heavy-tails in MAC retransmissions. Our experiments confirm some of the conjectures about the limited role of wireless MAC, and also show evidence against the possibility of heavy-tails due to MAC retransmissions. At the same time, we give a more detailed understanding about the role of transport protocol, capacity of the wireless channels, load levels and the finite buffer. Unlike any previous study, by carefully varying the scenarios, we identify the impact of those parameters on traffic LRD. These insights help us better predict large time-scale traffic characteristics.

This chapter is organized as follows: In Sec. 4.1, we give the definitions of heavy-tails and long-range dependence. In Sec. 4.2, we give the model that explains how heavy-tails in traffic sources lead to LRD in traffic rate, and why some of those assumptions may not hold in wireless networks. Sec. 4.3 gives the detailed results from our simulation study. First, we show that MAC service delays are not always

heavy-tailed in a realistic scenario. This subsection also shows the limited role of MAC in inducing heavy-tails and traffic LRD. Then, we focus on the transport layer and it's interaction with MAC, especially the finite buffer. We show that the losses at the finite buffer of the MAC leads to truncation of heavy-tails in UDP traffic. Thus, heavy-tails in traffic sources need not lead to traffic LRD.

4.1 Background Theory

In this section, we give some of the mathematical definitions and theory behind heavy-tails and long-range dependence in data network traffic. We also list the statistical tools used to test the existence of, and estimate the relevant parameters for the heavy-tails and and LRD.

4.1.1 Definitions

4.1.1.1 Heavy Tails

A random variable Z has a heavy-tailed distribution if

$$Pr[Z > z] \sim cz^{-a}, \quad z \to \infty \tag{4.1}$$

where a is the shape parameter and c is a positive constant. That is, the tail of the distribution decays hyperbolically. A common heavy-tailed distribution is the Pareto distribution:

$$Pr[Z > z] = \left(\frac{b}{z}\right)^a, \quad z \geq b \tag{4.2}$$

where a is again the shape and $b > 0$ is the location parameter. Heavy-tailed distributions have infinite variance for $0 < a < 2$ and also infinite mean for $0 < a < 1$. We will be primarily interested in Pareto distributions with finite mean but infinite variance (i.e., $1 < a < 2$). We focus on the infinite variance case because many measurements of file sizes, session duration, etc. indeed show power-law exponent $1 < a < 2$. Moreover, infinite variance can significantly impact the performance like unbounded queuing delay (Eg: from Pollaczek-Khinchine formula for M

G

1 queues).

4.1.1.2 Self Similarity

A wide sense stationary process $\{X_k, k \in \mathbb{Z}\}$ is said to be *exactly second-order self-similar with Hurst parameter* H ($\frac{1}{2} < H < 1$) if the autocovariance function $\gamma(k)$ satisfies

$$\gamma(k) = \frac{\sigma_X^2}{2}\left[(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}\right] \tag{4.3}$$

for all $k \geq 1$. Under weaker conditions, the process is said to be *asymptotically second-order self-similar* if

$$\lim_{m\to\infty} \gamma^{(m)}(k) = \gamma(k) \tag{4.4}$$

where $\gamma^{(m)}(k)$ is the autocovariance function of the m- aggregated process $X_k^{(m)}$ given by

$$X_k^{(m)} = \frac{1}{m} \sum_{i=m(k-1)+1}^{mk} X_i$$

4.1.1.3 Long Range Dependence

A second-order stationary process $\{X_k,\, k \in \mathbb{Z}\}$ with autocorrelation function $r(k)$ is said to be *long range dependent (LRD)* if

$$\sum_{k=-\infty}^{+\infty} r(k) = \infty \tag{4.5}$$

Alternatively, a second-order stationary process $\{X_k,\ k \in \mathbb{Z}\}$ with PSD $S(\nu)$ is said to be long range depended if

$$S(\nu) \sim c_H |\nu|^{-\alpha}, \quad \nu \to 0^+ \tag{4.6}$$

where $c_H = \frac{\sigma_X^2}{2\pi} \sin(\pi H)\Gamma(2H+1)$, $0 < \alpha = 2H - 1 < 1$ and $\Gamma(\cdot)$ the Gamma function.

There is a strong connection between LRD and asymptotic self-similarity in stochastic processes, although they are, in principle, distinct. From Eq. 4.4 it can be easily verified that

$$r(k) \sim H(2H-1)k^{2(H-1)}, \quad k \to \infty \tag{4.7}$$

which satisfies Eq. 4.5 only if $\frac{1}{2} < H < 1$. This is the case where LRD and self-similarity are equivalent. Since we mostly deal with sequences with $\frac{1}{2} < H < 1$, we use the terms LRD and self-similarity interchangeably.

4.1.2 Statistical Analysis

Now, we introduce the statistical tools that we will use for the estimation of the main parameters that characterize heavy tails, LRD and self-similarity.

4.1.2.1 Estimation of Heavy-Tails

Eq. 4.1 suggests that observations of process of i.i.d. random variables that follow a power law, can be identified after some subtle manipulation of the observations. Among various techniques to reveal this power law relationship we will make use of the *Modified Quantile Quantile (QQ) Graph*, which is a nonparametric estimation technique which is proven to be asymptotically unbiased [40].

The main principle of the QQ graph is based on the following assumption: if $X_1 \geq \cdots \geq X_k$ are the samples from the process with the distribution function F and k is large enough, F at $x = X_j$ can be estimated from

$$Pr[x < X_j] = F(X_j) \approx 1 - \frac{j}{k+1} \tag{4.8}$$

This assumption allows the modified QQ graph to be defined in the following manner. Let $X_1 \geq \cdots \geq X_k = u$ be the order statistics of the distribution, which is approximately a Pareto distribution. Then the graph of

$$\left\{ \left[\log(X_j) - \log(u); -\log\left(j/(k+1)\right) \right], 1 \geq j \geq k \right\} \tag{4.9}$$

has the form of straight line with slope a, the shape parameter in Eq. 4.1.

4.1.2.2 Estimation of Self-Similarity and LRD

The level of LRD in a time series is indicated by the Hurst parameter $H, 1/2 < H < 1$. The closer H is to 1 the stronger the LRD. Several methods are available to estimate the Hurst parameter [15,16]. However, as per almost all of the recent works, we use the robust wavelet-based estimator introduced by Abry and Veitch [41].

The wavelet estimator is based on the Discrete Wavelet Transform (DWT) and the Multi Resolution Analysis (MRA) which consists of splitting a given sequence of observations $X = \{X_1 \dots X_n\}$ into the (low pass) *approximation* and the (high-pass) *details*. So the J-level transformation of the "signal" X is

$$X = \text{approx}_J + \sum_{j=1}^{J} \text{detail}_j(t) = \sum_{k=1}^{n/2^J} a_X(J,k)\phi_{J,k}(t) + \sum_{j=1}^{J}\sum_{k=1}^{n/2^J} d_X(j,k)\psi_{j,k}(t) \quad (4.10)$$

where the parameters

$$\begin{aligned} \phi_{j,k}(t) &= 2^{-j/2}\phi_0(2^{-j}t - k) \\ \psi_{j,k}(t) &= 2^{-j/2}\psi_0(2^{-j}t - k) \end{aligned}$$

are the set of shifted and dilated function of the scaling function ϕ_0 and the mother wavelet ψ_0. The DWT on the J^{th} level is the collection of coefficients $\{a_X(j,k),\ d_X(j,k),\ \ j = 1, \dots, J, k = 1, \dots, n/2^J\}$ called trend and wavelet coefficients respectively. They are defined by the following inner products:

$$\begin{aligned} d_X(j,k) &= \langle X, \psi_{j,k} \rangle \\ a_X(j,k) &= \langle X, \phi_{j,k} \rangle \end{aligned}$$

A commonly used mother wavelet is the Haar wavelet, given by $\psi_0(t) = 1$ if $t \in [0, 1/2), \psi_0(t) = -1$ if $t \in [0, 1/2)$, and $\psi_0(t) = 0$ otherwise. The corresponding scaling function is $\phi_0(t) = 1$ for $t \in [0, 1]$ and $\phi_0(t) = 0$ otherwise.

The quantity $|d_X(j,k)|^2$ measures the amount of energy in signal X about time $t_0 = 2^j k$ and about the frequency $2^{-j}\lambda_0$ where λ_0 is a reference frequency which depends on the choice of ψ_0. And the expectation of energy that lies within a

bandwidth 2^{-j} around the frequency $2^{-j}\lambda_0$, is denoted by $\mathrm{E}[E_j]$ and is approximated by $\frac{1}{n_j}\sum_{k=1}^{n_j}|d_X(j,k)^2|$, where n_j is the number of wavelet coefficients at scale j.

If X is LRD, then the wavelet coefficients process satisfies

$$y_j = \log_2\big(\frac{1}{n_j}\sum_{k=1}^{n_j}|d_X(j,k)^2|\big) \approx \log_2 \mathrm{E}[E_j] \sim (2H-1)j + c \tag{4.11}$$

where c is a constant. From Eq. 4.11, a linear regression fit to the plot of y_j against scale j gives an estimate of H. In this work, we use a more advanced version of the algorithm [42], the MATLAB code for which is available on the first author's website. Using the authors' notation, the plot of y_j against j is called a *logscale diagram* or *LD plot*. To get a good Hurst estimate the region of scaling in the LD plot needs to be carefully identified [29]. In particular, a good linear fit should be observed along the higher scales as the Eq 4.11 should hold as $j \to \infty$. If the slope is zero, then X is short-range dependent. If X is exactly self-similar, the Eq. 4.11 holds for every j. The wavelet plots can show multiple scaling regions if X is multifractal. While uniform scaling in higher scales indicates the asymptotic self-similarity, linearity at finer timescales also give an indication of burstiness at corresponding timescales.

4.2 Models for Heavy-Tails and LRD in Network Traffic

Heavy-tails in file sizes, connection duration, etc. has been attributed as one of the main cause of the observed LRD nature of traffic [10, 11]. The following mathematical results explain the relation between heavy-tails and LRD.

4.2.1 On/Off Model

In [10], Taqqu et al. described an *On/Off Model* for source traffic that gave the causal link between heavy-tails and LRD in an idealized setting. Using empirical evidence for both the traffic sources and aggregate traffic, they show the applicability of the model to explain the origins of asymptotic self-similarity and LRD in data network traffic. Since then, the model and it's variants have been widely used to explain the LRD nature of TCP traffic in many different wired networks [11, 13, 14]. The model is described below.

Consider M independent 0/1 reward renewal process denoted by $Z_i(t), t \in \mathbb{R}, i \in [1, M]$ as traffic sources. Denote by $S_M(t) = \sum_{i=1}^{M} Z_i(t)$ the aggregate traffic at time t (here, aggregated over sources), and by $Y_M(Tt) = \int_0^{Tt} S_M(s)ds$ the cumulative traffic upto time Tt. Further denote the i.i.d. 0 (off) periods by the random variable τ_{off} and i.i.d. 1 (on) periods by τ_{on}. If either τ_{on} or τ_{off} is heavy-tailed with shape $1 < a < 2$, then $Y_M(Tt)$ satisfies

$$\lim_{T\to\infty} \lim_{M\to\infty} \frac{Y_M(Tt) - \frac{E(\tau_{on})}{E(\tau_{on})+E(\tau_{off})} MTt}{C\sqrt{M}T^H} = B_H(t) \tag{4.12}$$

where $H = \frac{3-a}{2}$ and $B_H(t)$ is a *fractional Brownian Motion*. The increment process of $B_H(t)$ is *fractional Gaussian Noise* which is LRD with Hurst parameter H.

The on-periods corresponds to when a source transmits traffic at constant rate, while during the off-period, the source does not send any traffic. Thus, infinite aggregation of individual traffic sources with heavy-tailed on/off periods lead to LRD in traffic rate process. The related $M/G/\infty$ model [43] also explain the LRD

traffic rate due to heavy-tailed service times. In particular, Possion arrivals of jobs with heavy-tailed service durations in a infinite server queue results in long-range dependence in its departure rate. From real network measurements, file sizes, connection or flow durations, etc. have been found to be heavy-tailed in many networks like LAN, WAN. However, the heavy-tailed durations do not directly translate into heavy-tailed on-periods (of constant rate) because of the lower layer protocol dynamics (and the variable service rate for each connection). Despite the variable connection rate, the service rate can be assumed to be constant over the small-scales when viewed at larger time-scales. Thus, the protocol dynamics affect the traffic rate at small-time scales, but it's asymptotic LRD properties seem to hold in most network scenarios. Kurtz [44] give more general conditions on variable bitrate under which the asymptotic self-similarity is preserved. [44] allows for variable bitrate approximately mimicking the 'within-session' dynamics. Even though Kurtz's model can explain the observed large-time scaling behavior of WAN traffic, it cannot explain the richer scaling observed at smaller time scales.

4.2.2 Modeling the Aggregate UDP traffic

Based on the empirical evidence, existence of asymptotic self-similarity in traffic rate is generally attributed to the presence of heavy-tails in application traffic demands. The protocol hierarchy doesn't seem to impact the relation much in the wireline networks. Through the *On/Off* and $M/G/\infty$ models (described above), the network protocol dynamics are abstracted into a packet-train model. In essence, in-

stead of looking at the actual data rate, sources are assumed to send traffic at constant rate (with smaller average rate). While the model has successfully explained asymptotic self-similarity of TCP traffic, it does not seem to explain the 'diminished' self-similarity in UDP scenario [13]. Moreover, the model may not hold for the finite capacity wireless channel. We reexamine this model in detail to find conditions under which the model can be and cannot be applied.

4.2.3 Power Law Delays due to MAC Retransmissions

Contention-based MAC protocols like ALOHA and IEEE 802.11 can increase the variability in packet delays due to collisions and the backoff mechanisms. While some packets can get successfully transmitted immediately, some other may experience huge delays due to contention and collisions. While there have been some studies for delay distributions of various MAC protocols [45–47], one work that is relevant to this study is by Jelenkovic and Tan [28, 48]. In [28], the authors look at a simplified model for ALOHA and show that variable packet length can cause power law delays. They also show that variability in number of users, say, due to mobility, can also induce power law delays in slotted ALOHA even if the packet sizes are constant.

In order to prove the above results rigorously, Jelenkovic and Tan [28] use a simplified model for input load and packet arrival process. In particular, they look at a single cell scenario with M users sharing a common communication channel. Each user can store at most one packet in its queue and the packets are assumed to

be generated after an independent exponential time with mean $1/\lambda$ once the queue is empty. The packet size is given by a random variable L. Immediately upon a packet generation, the user attempts to transmit the packet. If there's a collision, the users backoffs for an independent exponential period with mean $1/\nu$ before retransmitting. Denoting by T the time between two successful transmissions, they prove the following: *If* $\mathrm{P}[L > x] \sim e^{-\mu x}$ *then* $\mathrm{P}[T > t] \sim t^{-\frac{M\mu}{(M-1)\nu}}$. In other words, if the packet size tail distribution is exponential, the tail distribution of T is a power law. In a related result, they look at slotted ALOHA with constant packet sizes, but variable number of user M and show that if the number of users have an exponential tail, the successful transmission times T have power-law tail.

In practice however, the packet sizes (or number of users) have a bounded support. In such scenarios, [28] through simulations also show that T has truncated power law distribution, which even though bounded, can give very long delays. They say that even though their model considers queue size of only one, the increased queuing in real systems will further increase the power law delays. However, more realistic models for ALOHA are missing and there are no such studies for more complicated protocols like IEEE 802.11. As mentioned earlier, the model for delay distribution obtained in [19], does not capture the tail of the distribution accurately enough. Given the Markovian model with independence assumptions, the delay distribution cannot capture the tail accurately enough. In another work, [49] do indeed improve the model, and show that for saturated-queues in single-cell network, MAC delays can indeed be heavy-tailed. However, it is non-trivial to extend the model for non-saturated queues as well as for multi-hop scenarios. Even our MAC

model introduced in Ch. 2, captures only the mean delay and does not give a distribution. Hence, we resort to simulation-based analysis. Also, to delineate the effect of traffic sources and transport (TCP or UDP) protocol dynamics, we focus on the service delays at the MAC layer. We do not focus on the queuing delay as it can be impacted by the traffic arrival pattern and load levels.

4.3 Simulation-Based Analysis

We run some simulations in OPNET Modeler [37], to analyze the traffic characteristics. By varying the traffic sources, and network protocols we analyze the traffic rate process to study the presence of heavy-tails, long-range dependence and self-similarity.

4.3.1 Simulation Setup

In this section, we describe the simulation setup. We use the setup to explore the relation between heavy-tails in traffic sources and LRD in traffic rate in a wireless network setting. We mainly study a 10-node clique scenario or WLAN scenario using 802.11 MAC with 11Mbps data rate. In this scenario, every node can listen to every other node and there are no hidden nodes or multi-hop connections. In this scenario, 9 of the nodes send traffic to the 10th node. However, in one particular scenario, we use a similar 20-node scenario to study the impact of increased number of sources. We also use a 11-node scenario for studying the MAC protocol impact in more detail. The 11-node scenario is similar to the 10-node clique scenario, except that the 11th

node is hidden from the first nine nodes.

4.3.1.1 Inputs for Different Scenarios

For different scenarios, we specify traffic demands at the application layer using the following parameters:

- **Inter-request Time (sec)** - This is the time between requests. It typically models the response time of a user.
- **Request File Size (bytes)** - This is the size of each file in the request. It models the size of files that are to be sent over the network.

All simulations are run with different probability distribution assigned to the above parameters. However, for each simulation scenario, all the connections use the same distribution for application traffic demands. We mainly use the UDP transport protocol, but also compare with a few TCP scenarios. For the TCP scenario, we use TCP-Reno.

We run multiple simulations with different scenarios, with varying distribution assigned to the inputs - application file sizes and inter-request times. In particular, we assign exponential or Pareto distribution to file sizes and inter-request times. We further subdivide the study into simulations with exponential file sizes, but heavy-tailed inter-request times, and simulations with heavy-tailed file sizes but exponential inter-request times. Even though these parameters are changed across simulation, all the connections use the same distribution for a particular simulation run. Unless otherwise mentioned, each of the simulation run uses UDP protocol at

the transport layer. Each simulation is run for several hours, so that each connection is able to generate about 100,000 to 1 million requests. Hence, we have enough data points to perform some meaningful statistical analysis. However, we note that for Pareto distribution with shape $a < 1.2$, even 1 million data points may not be sufficient for sample mean to converge. But such values have been commonly used in the past [13]. Nevertheless, for Pareto distribution with shape $a > 1.5$, 100,000 data points are sufficient for the distribution to reach its mean. We also check for stationarity of each of the observed statistics, by looking at trends in the sample means and variances. All the statistics shown subsequently have passed the stationarity tests.

Based on the WWW file size traces from [12], we use the mean file size of 13715 Bytes. We choose [12] file sizes, as it was one of the first set of work to give the evidence of LRD in traffic due to heavy-tails in file sizes. While we keep the mean file size fixed, we change other distribution parameters like exponential or Pareto with different shape, so as to study the impact of heavy-tails in file sizes. By changing the mean inter-request time, we vary the load on the network. For a request with file size 13715 Bytes, the service time is about 15 msec assuming an ideal link transmitting at 11 Mbps (but including the packet headers). The service time increases with load due to contention and backoff. For the 10 node clique scenario, 9 connection each generating traffic with mean inter-request time of 0.8 sec, will imply average inter-request time of 88.9 (= 800/9) msec for the aggregate traffic. Hence, it is a low load scenario (compare with service time of 15 msec). For the mean inter-request time of 0.4 sec, the average arrival rate for aggregate

traffic is 44.4 msec, resulting in medium load (and only negligible increase in service time). Similarly, connection inter-request time of 0.2 sec is high load scenario, and inter-request times of 0.1 sec imply load higher than capacity. We mainly use these parameters to analyze traffic characteristics at different load levels in the network.

4.3.1.2 Protocol Dynamics

Files are generated at the application layer according to the *request* parameters defined above. Each file is fragmented into packets of constant 1000 Bytes and passed to the transport layer. For UDP scenario, all the packets are passed down from the IP to the MAC layer without further fragmentation. Thus, each application level file consists of multiple packets and each packet corresponds to one MAC frame. Each MAC frame is queued in the MAC buffer of finite size. Because of the UDP protocol, all the packets in a file are instantly queued at the MAC buffer. The MAC frame, at the head, contends for the channel following the binary exponential back-off mechanism. A colliding frame is retransmitted until it is successfully transmitted or till MAC retry limit is exceeded (after which the frame is discarded). Routing layer forwards the packet to the destination in case of multi-hop scenario. There is no retransmission of packet in the event of failure and the packet is considered as lost. Packet drops also occur due to buffer overflow, especially when huge file size results in UDP dumping too many packets at the MAC.

Even in the TCP scenario, each MAC frame contains exactly one TCP-layer packet. However, unlike UDP, TCP protocol does not pass all the packets directly

to the MAC, but uses its congestion control algorithm to pass packets to the MAC. In particular TCP sends packets depending upon the congestion window, and sends new packets only after the previous ones are ACKed by the destination. Hence packets are retransmitted by TCP in the event of losses at the MAC. Thus, TCP not only ensures reliable packet transmissions, but also stretches the service time for each file transfer.

4.3.1.3 Collected Statistics

During the simulations the following IP layer statistics are collected at the destination of individual connection:

- Arrival time of a packet, T_n
- End-to-end delay of packet, δT_n
- Packet size, PS_n

From the IP statistics, we calculate a few other statistics of interest for individual connections as well as aggregate traffic (aggregated over all the connections):

- Interarrival time of packets, $\Delta T_n = T_{n-1} - T_n$
- Traffic packet rate - Number of packets that have arrived every τ seconds, N_k^τ
- Traffic bit rate - Bits per τ seconds, R_k^τ

During the simulations, the following IP layer statistics are collected at the destination of individual connection: Arrival time of a packet, T_n From the collected

statistics, we calculate a few other statistics of interest for individual connections as well as aggregate traffic (aggregated over all the connections): Traffic packet rate - Number of packets that have arrived every τ seconds, N_k^τ, and traffic bit rate - Bits per τ seconds, R_k^τ.

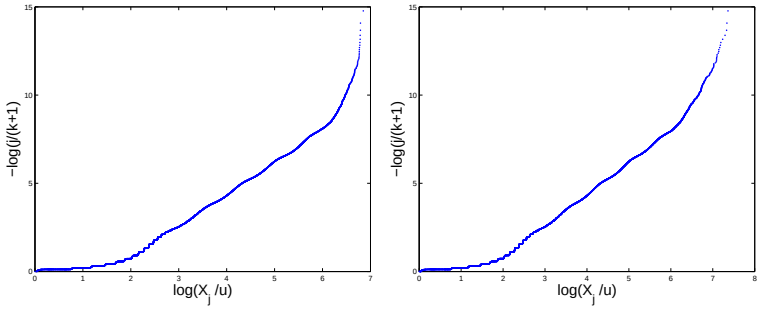
These statistics adequately characterize the traffic and the network response from the IP layer and above. In our statistical analysis, we look for LRD in traffic packet-rate and bit-rate at aggregate level. For the scaling analysis, the exact choice of τ does not matter, since it only determines the smallest scale in the wavelet analysis. Hence, we use $\tau = 0.1$ *sec* for using wavelet analysis to look at traffic scaling at time-scales above the packet service times.

4.3.2 MAC Retransmissions and Service Times

4.3.2.1 Analysis for Heavy-Tails

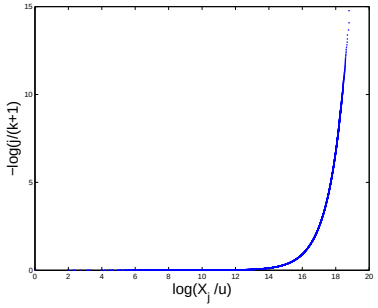
Motivated by [28], we analyze the service time for MAC frames for the presence of heavy-tails. We use the 10-node clique scenario with 9 connections using exponential inter-arrival times. Fig. 4.1 shows the QQ plot for MAC service times for the high load scenario (when the inter-arrival time is 0.1 sec). For both the scenario with a finite retry limit (of 7) and a MAC very high retry limit (of 255), clear linearity in the QQ plot indicates the presence of heavy-tails. However, given the finite support of observed traces, we cannot say if they are true heavy-tails. For both the samples, the shape of heavy-tailed distribution is obtained as 1.81 and 1.83 respectively. We also show the QQ plot for an exponential random variable for

comparison (which is clearly not linear).



(a) From Simulation Run

(b) From Simulation Run with High Retry Limit

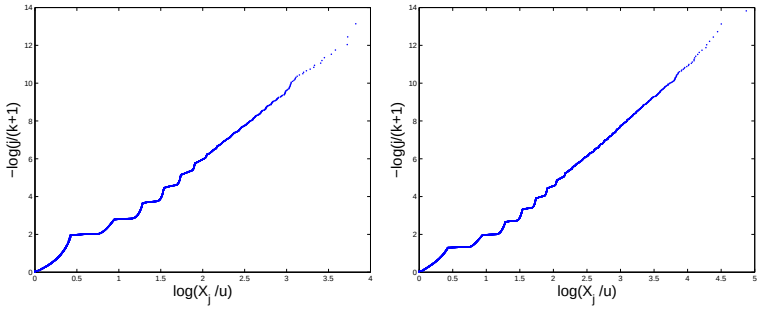


(c) Comparing with Exponential Random Variable

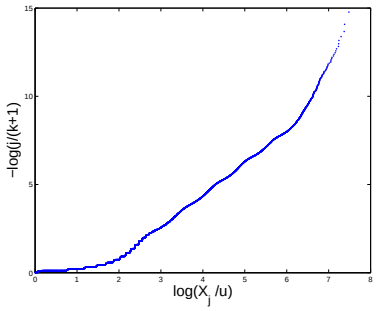
Figure 4.1: QQ Plot: Evidence of heavy-tails in MAC Service Times

Next we vary the load levels to see if we see similar evidence of heavy-tails in MAC service times. In Fig. 4.2, we see that even for medium and low load levels, the MAC service times were heavy-tailed. For the figure shows, we use high MAC retry limit and exponential inter-arrival times of 0.8, 0.4 and 0.1 seconds respectively for low, medium and high load levels. We obtained similar results for low MAC retry limit.

However, not all scenarios show heavy-tails in MAC service times. For exam-



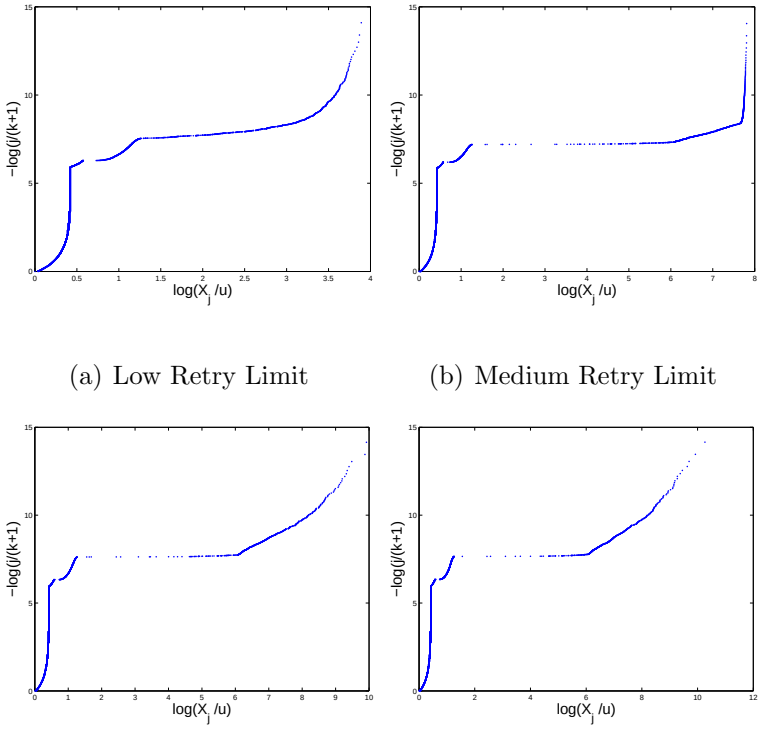
(a) Low Load (b) Medium Load



(c) High Load

Figure 4.2: QQ Plots of MAC Service Times with Varying Load

ple, for a 4-node-two-connection scenario (4-node asymmetric links scenario), Fig. 4.3 shows absence of heavy-tails in MAC service times. For a 30-node multi-hop topology, Fig. 4.4 shows some degree of linearity, indicating the presence of heavy-tails. However, the linearity is not as strong as in the 10-node clique scenario.



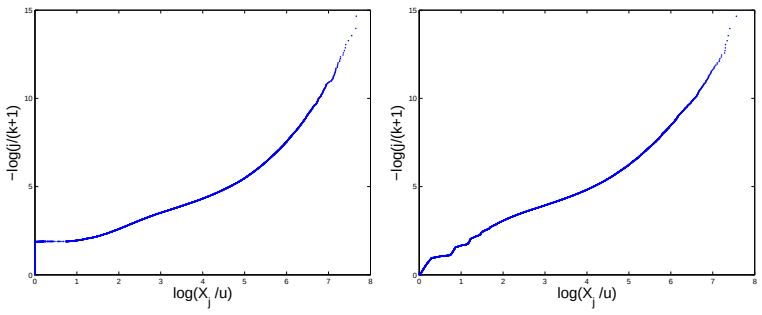
(a) Low Retry Limit (b) Medium Retry Limit

(c) High Retry Limit (d) Very High Retry Limit

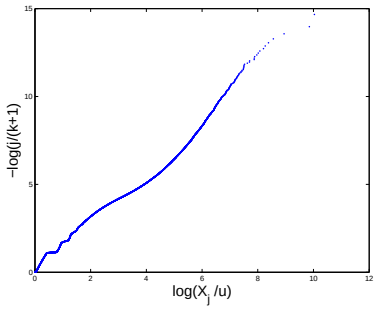
Figure 4.3: QQ Plots of MAC Service Times for 4-node Asymmetric Scenario

4.3.2.2 Finite Range for MAC Service Times

Despite the evidence of power-laws in MAC service times as seen above, the actual service times are finite. Finite MAC retry limit along with finite backoff windows, etc. restrict maximum MAC service time observed. Assuming a high retry limit of 255 (maximum allowed by standards), the average time spent in a backoff



(a) Medium Retry Limit (b) High Retry Limit



(c) Very High Retry Limit

Figure 4.4: QQ Plots of MAC Service Times for 30-node Scenario

Retry Limit	Maximum Attempts Made	Maximum Service Time (in Sec)
7	7	1.08
255	10	1.46
25555	10	2.24

Table 4.2: Maximum MAC Service Times with Retry Limits - Clique Scenario

can be calculated as follows: For each transmission attempt, average backoff is $CW/2 * SLOT\_DURATION$, where $SLOT\_DURATION = 20\mu s$. Approximating the exponential increase in backoff window CW, let's assume $CW = CW_{max} = 1024\ slots$ for each attempt. Then for a retry limit of 255, the total time in backoff is $255 * 1024/2 * 20\mu s$, which equals 2.5 seconds. However, the actual MAC service time will also include time when the channel is busy because of neighboring transmissions. In Tab. 4.2, we list the maximum MAC service time observed for various retry limits. We see that even for a very high retry limit of 25555 (which is higher than the limit of 255 as per the standards), the MAC service time is just a few seconds. We varied the load levels and parameters, but observed similar values for maximum MAC service times.

Thus, even though previously we saw that the MAC service times have power laws, they are no true heavy-tails. Hence, the impact of MAC on shaping the traffic cannot extent beyond the maximum MAC service times. Later, we will see that, indeed the maximum traffic scaling observed, purely due to MAC, is for up to a few

Retry Limit	Maximum Attempts Made	Maximum Service Time (in Sec)
7	7	0.62
255	255	3.14
2555	2555	26.04
25555	3163	36.68

Table 4.3: Maximum MAC Service Times with Retry Limits - 4-node Asymmetric Scenario

tens of seconds.

4.3.2.3 Role of Asymmetric Topology

In the above section, we looked at only a clique scenario. Since everyone can listen to everyone else, the maximum retransmission attempts even for a highly congested scenario is not much. To analyze the role of higher number of retransmissions, we look at a 4-node asymmetric scenario, where one connection suffers a lot of MAC transmission failures (and hence retransmissions) due to hidden-node collisions. Tab. 4.3 shows the maximum MAC service time obtained for a particular scenario.

We also performed similar analysis for a more realistic 30-node scenario. Again, even for very high retry limit, the maximum MAC service times were of the order of a couple of tens of seconds.

Thus, firstly, we show that MAC service times can indeed have power-law bodies (in some scenarios). However, in all the scenarios, the MAC service times are bounded. For the usual modes of operations, because of protocol parameters, the maximum service times are of the order of few tens of seconds. Thus, MAC retransmission by themselves cannot create traffic LRD or traffic burstiness lasting several hours or days.

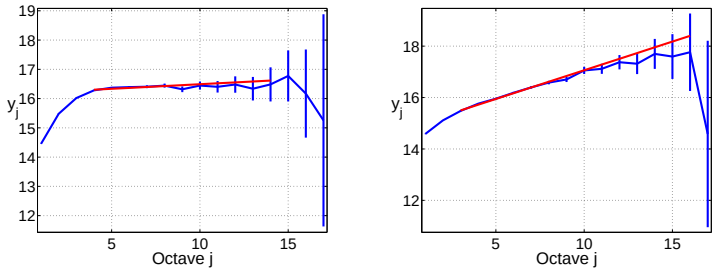
4.3.3 Heavy-Tails in Inter-Request Times

In this set of simulations, the inter-request time are heavy-tailed with different shapes or means. However, the file sizes are exponential with mean 13715 Bytes. All the simulations use UDP transport protocol.

4.3.3.1 Changes in Traffic Rate due to Heavy-Tails in Inter-Request Times

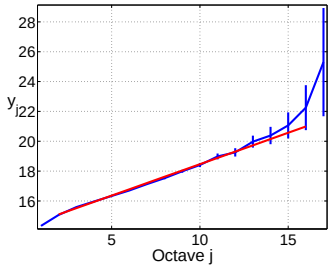
We study the presence of LRD in aggregate traffic rate specified by N_k^τ. We use wavelet analysis to observe scaling and obtain Hurst estimates. Similar results are obtained when looking at the individual connection traffic.

At first, we vary the shape of heavy-tails in the inter-request times. Fig. 4.5 shows that as the heaviness of the inter-request time distribution is increased, the traffic rate shows strong scaling. The x-axis is scale j, and the corresponding time-scale is given by $T = 2^j * \tau$ sec. The y-axis gives the y_j from Eq. 4.11. The exact value of y-axis is not important in our analysis, except in determining the slope of

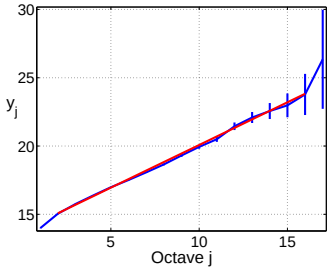


(a) Exponential

(b) Pareto a = 2



(c) Pareto a = 1.5



(d) Pareto a = 1.2

Figure 4.5: Variation in traffic rate due to heavy-tails in inter-request times

the regression fit. For exponential inter-request time, the slope for regression fit to the scaling coefficients is zero. This indicates there is no LRD. But as shape of the Pareto inter-request time decreases, the slope of regression fit increases. This, in-turn, indicates that the Hurst estimate for LRD increases. Such result is in agreement with the *On/Off model* source model that heavy-tails in off-period result in LRD in traffic, and the Hurst estimate is determined by the shape of the heavy-tailed distribution.

4.3.3.2 Changes in Traffic due to Changing Mean and Shape of Inter-Request Times

Now we study the LRD properties of traffic rate with changing mean and shape of inter-request times. Fig. 4.6 shows the Hurst estimates for various distributions. For exponential inter-request times, irrespective of the mean, $H = 0.5$, indicating absence of LRD. As the inter-request times have heavier tails (shape changing from 2 to 1.2), H increases indicating stronger LRD. Also, for a particular shape, as the mean inter-request time decreases (reading from right to left in the figure), the load increases, but the Hurst estimates are pretty much the same. Again, agreeing with the *On/Off model's* theory that the shape of heavy-tails determine the strength of LRD. However, for high load, or low means, the increased packet losses come into play and somewhat affect the Hurst estimate of LRD.

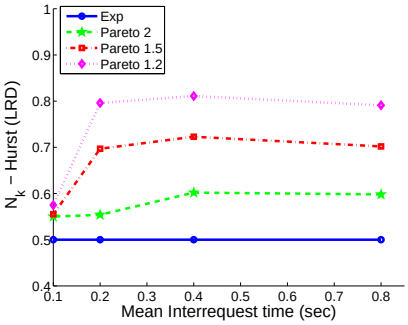


Figure 4.6: Variation in traffic rate due to changing mean and shape of inter-request times

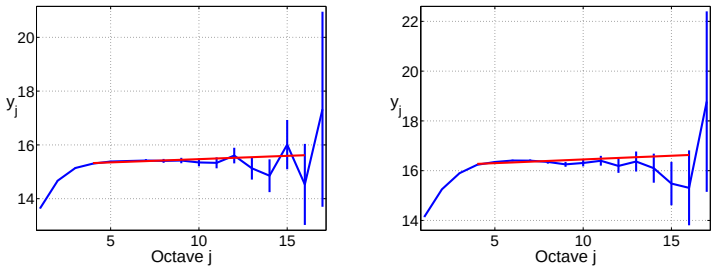
4.3.4 Heavy-Tails in File Sizes

Now we study what happens with heavy-tails in request (or file) sizes. Here the inter-request time are exponential, but we vary mean across simulation runs in order to vary the offered load. The file sizes are Pareto with mean 13715 Bytes, but shapes varying across simulations. The results are from aggregate traffic rate N_k^τ (packet-rate) measured at $\tau = 0.1$ *sec* intervals. Similar results hold for aggregate as well as individual connection's traffic bit-rate R_k^τ. In this set of simulations, we use the 10-node clique scenario with UDP protocol unless stated otherwise.

4.3.4.1 Increasing Traffic Load by Decreasing Inter-Request Times

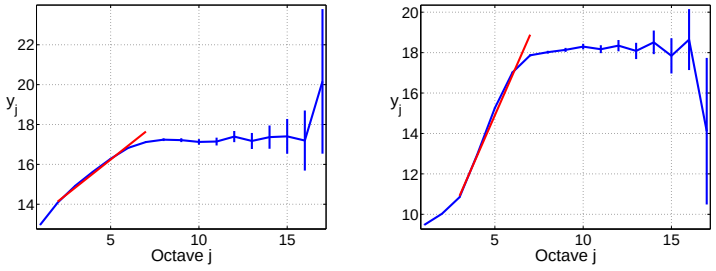
We use Pareto distribution to file sizes with shape = 1.5 and mean file size of 13715 Bytes. We increase the load by decreasing the mean inter-request times. Fig. 4.7 shows the wavelet plots for the traffic rate. As the load increases, there is increased scaling at lower-time scales, indicating a stronger correlation. However,

the scaling seizes to exist beyond scale $j = 8$ which corresponds to $T \sim 25$ *sec*. For low load, there is little queueing as the UDP protocol greedily transmits the packets. Hence there is little memory. With increasing load the queueing ensures some stretching-in-time effect and scaling is observed for small-time scales. However, at larger time-scales the wavelet plot is almost flat implying the stretching-in-time effect is not strong enough to preserve correlation at longer time scales. Hence there is no LRD. We attempted to see if higher load increases the queueing resulting in scaling for longer duration. However, with higher loads, the queues get full, significantly increasing the delays and losses.



(a) Mean inter-request time = 0.8 sec

(b) Mean inter-request time = 0.4 sec



(c) Mean inter-request time = 0.2 sec

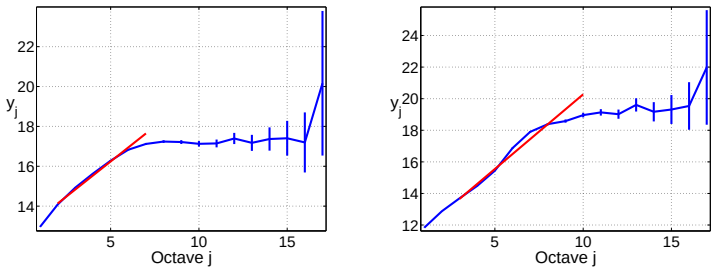
(d) Mean inter-request time = 0.1 sec

Figure 4.7: Variation in traffic rate due to decreasing inter-request times

This result is somewhat unexpected. Based on the *On/Off model*, heavy-tails in on-periods (or file-sizes) should also result in LRD even if the off-periods are exponential. Even applying the $M/G/\infty$ model for Poisson arrivals of heavy-tailed jobs, we should see LRD. However, the results show that the packet-train and the infinite server assumptions do not seem to hold for networks using UDP and 802.11 MAC. The greedy nature of UDP gives packet-trains with very small on-periods. Even huge file sizes cannot ensure longer on-periods due to losses at the MAC because of its finite buffer (even with a high buffer size of 128 MB).

4.3.4.2 Increasing Load by Increasing Number of Sources

Now we try to increase the number of sources from 9 to 19. We use the 20-node 19-connection scenario with Pareto file sizes (mean = 13715 Bytes, shape = 1.5) and exponential inter-request times (mean = 0.4 sec), and compare the traffic rate with corresponding 10-node scenario (mean inter-request time = 0.2 sec). However, again as seen in Fig. 4.8, increasing the sources has little effect. In particular, there is no LRD.

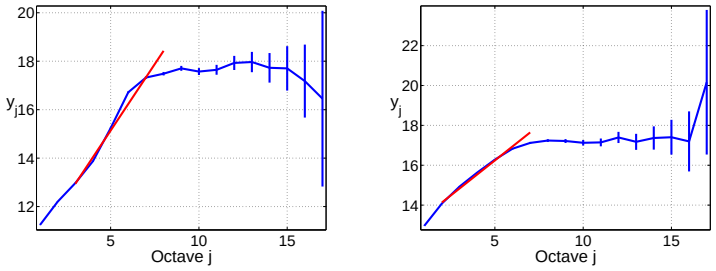


(a) 9 connections (b) 19 connections

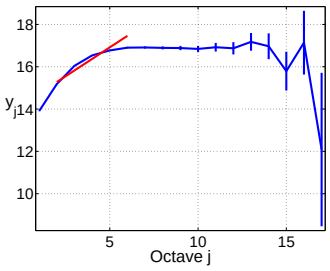
Figure 4.8: Variation in traffic rate due to increasing number of sources

4.3.4.3 Changing Shape of Pareto File Size Distribution

As was observed in [13], UDP connections showed little LRD. But Hurst estimates for heavier file sizes had slightly stronger LRD (higher Hurst estimates). Hence, returning to the 10-node clique scenario, we study the effect of heavier tails in file sizes. Keeping the Pareto file sizes with fixed mean (13715 Bytes), we change the shape and study the traffic rate. Fig. 4.9 shows that even though the heavier tails (lower shapes) make the scaling stronger at lower time scales, the scaling is not preserved at higher time scales and there is no LRD.



(a) Pareto a = 2 (b) Pareto a = 1.5

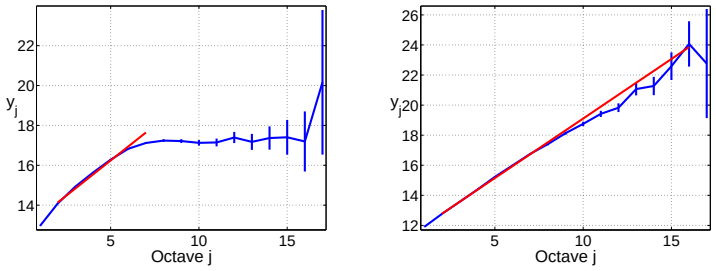


(c) Pareto a = 1.2

Figure 4.9: Variation in traffic rate due to heavy-tails in file sizes

4.3.4.4 Comparison with TCP traffic

Now, we compare the UDP traffic with TCP traffic. In both cases, we use the 10-node clique scenario with exponential inter-request times (mean = 0.2 sec) and Pareto file sizes (mean = 13715 Bytes and shape = 1.5). Fig. 4.10 shows the wavelet plots for the UDP scenario and TCP scenario. Fig. 4.10(b) clearly shows strong scaling at larger time-scales. Hence, there is strong LRD in TCP scenario, whereas no LRD in UDP scenario. The stretching-in-time of TCP ensures the applicability of the packet-train model and the reliable communication ensures that there is no packet drop. These seem to be critical in ensuring LRD. For the UDP scenario, the on-durations are small compared to the TCP scenario. Even with smaller on-periods, the asymptotic results for the *On/Off* or $M/G/\infty$ model should hold with higher number of sources. However, increases in load level or number of sources does not translate into more carried traffic due to the finite capacity of the wireless channel.



(a) UDP (b) TCP

Figure 4.10: Comparison between UDP and TCP traffic

The simulation results show that aggregate traffic in wireless networks need

not be long-range dependent in spite of the presences of heavy-tails in sources. Specifically, the role of UDP and 802.11 MAC are important in shaping the traffic characteristics. The finite capacity of the wireless channel, and the greedy and unreliable nature of UDP leads to crucial packet losses which destroys the long-range dependence. Thus, the *On/Off* or $M/G/\infty$ models cannot be applied in such cases.

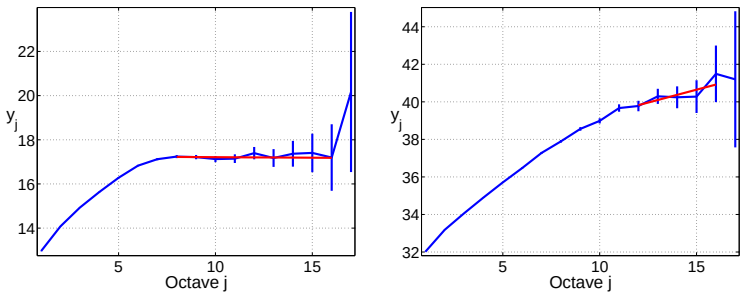
4.3.5 Impact of Finite Buffer

Here, we show that losses due to finite buffer lead to absence of traffic scaling. If we increase the buffer sizes, then traffic LRD can be preserved. Lastly, the buffer sizes required for preserving scaling depend on the PHY data rate. For smaller data rates, even a smaller buffer sizes leads to traffic scaling for longer duration. Similarly, as data rates will increase, traffic scaling will diminish due to finite buffers.

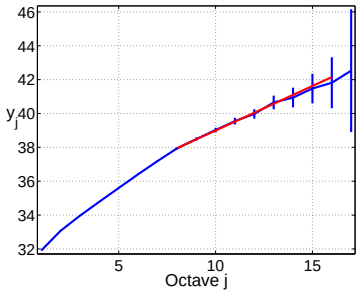
4.3.5.1 Role of Buffer Size

For the 10 node clique scenario, we increase the buffer size and observe the maximum range of scaling. Fig. 4.11 shows that as we increase the buffer size, the observed scaling range increases. For larger buffer, traffic appears to be LRD. This confirms the role of truncation of heavy-tails because of losses at MAC buffer. For greedy UDP protocol, packets are passed on to the MAC as soon as they are generated by the source. Hence, packets are queued up at the MAC buffer. For large files, there will be a lot more queueing. Hence, if the buffer size is small, there

will be a lot of drops. Drops will be larger for heavy-tailed files. These packet drops in the queue truncate the ON-duration of packet transmission. Thus, unless the packet are paced appropriately, finite buffer will destroy traffic LRD.



(a) 10 MB (b) 100 MB



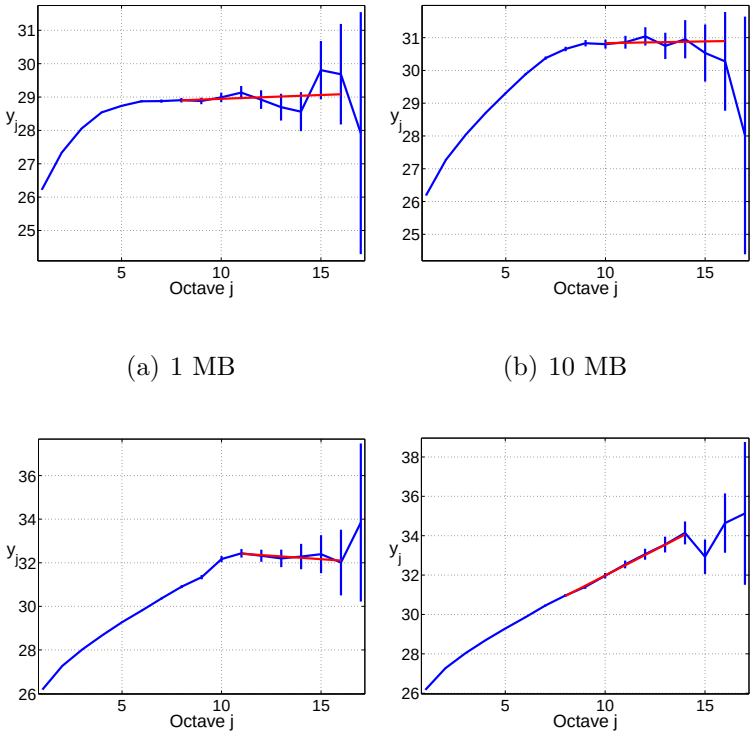
(c) 500 MB

Figure 4.11: Role of Finite MAC Buffer: In 10-Node Topology

4.3.5.2 Role of Number of Users

For a larger topology with multi-hop connections, the coupling in the queues is expected to be more complex. Even if there is losses due to finite buffer at the source, traffic at intermediate hops can be different. However, we find that even in large topologies, the effect of finite buffer in destroying traffic LRD is strong. Fig. 4.12 shows the traffic scaling in a 30-node scenario. The results are similar to the

10-node scenario (Fig. 4.11).



(a) 1 MB (b) 10 MB

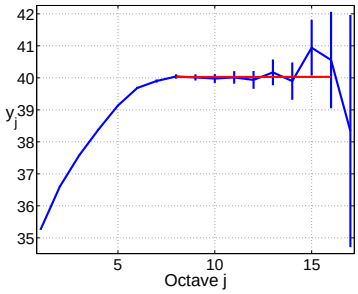
(c) 100 MB (d) 500 MB

Figure 4.12: Finite Buffer: In 30 Node Topology

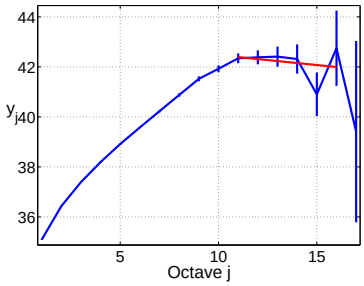
4.3.5.3 Role of PHY Data Rate

As explained above, UDP translates heavy-tails in file sizes into heavy-tails in ON-durations. However, finite MAC buffer will truncate these ON-durations. The level of ON-duration depends not just on the buffer size, but also the rate of emptying the buffer. This means, with smaller data rate, the value of maximum ON-duration will be larger resulting in larger range for traffic scaling. Indeed, Fig. 4.13 shows that for the same 10-node scenario as before, but with a PHY rate of

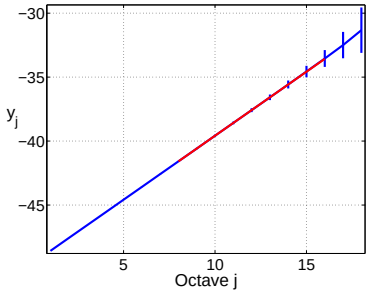
1Mbps, the scaling range is higher even for smaller buffer (Compare with Fig. 4.11). Conversely, as data rate go on increasing, finite buffers will lead to shorter range for traffic scaling, implying lesser chances of traffic LRD.



(a) 1 MB



(b) 10 MB



(c) 500 MB

Figure 4.13: PHY Data Rate: 1 Mbps

4.4 Summary

We looked at the role of MAC retransmissions in introducing heavy-tailed delays. Using extensive network simulations, we investigated service time distribution for 802.11 MAC frames. We found some evidence of power-laws in MAC service time. In particular, the MAC service times can show power-law distribution. However, the simulations show the service times are not pure heavy-tails and have a finite

support which does not stretch beyond a few seconds. Furthermore, even though the service times have power-laws, the delay (including the queueing delay) do not show any power laws. Also, since the power-laws in service times are also bounded, MAC protocol does not impact traffic burstiness above few tens of seconds. Certainly, we found no evidence of traffic LRD due to MAC protocol dynamics alone. However, the power-laws in MAC service time indeed lead to some traffic scaling up to few orders of magnitude (tens of seconds) above the mean MAC service time (few milliseconds).

We also carried out extensive simulations to analyze the role of wireless protocol in preserving high input (source) traffic variability, resulting in traffic LRD. We observed heavy-tails in TCP traffic lead to traffic LRD, consistent with the previous wired network observations. However for UDP traffic, finite buffers and losses at the MAC layer, lead to truncation of heavy-tails and the absence of any traffic LRD. We studied the impact of various network and protocol configurations - like buffer size, PHY data rate, MAC retry limit, etc. on the level and extent of traffic scaling. In particular, when the buffer sizes are huge, traffic scaling is observed at higher timescales. Thus, MAC retransmissions do not shape traffic scaling beyond a few seconds, and finite buffers destroy traffic scaling for beyond a few seconds. The scaling range preserved by the MAC is a function of data rate, number of users sharing the channel and the buffer size. In particular, if the data rate is low, then a smaller buffer size will also preserve traffic scaling as the buffer gets emptied over a longer duration of time. The basic idea is - UDP dumps all the traffic to the MAC layer as soon as it receives it from the application. Hence, the application file size

translates into an ON duration, when there is traffic load at the MAC, and the MAC is serving the packets at a fixed rate. Thus, heavy-tails in file size results in heavy-tails in ON duration, provided the MAC buffer does not lose any packets. However, finite buffer leads to larger losses for heavy-tailed files. Essentially, heavy-tails are truncated because of the finite buffer size. These truncated heavy-tails do not lead to traffic LRD.

Chapter 5: Role of Routing Protocols in Creating Traffic LRD

In the previous chapter, we focused on the role of MAC layer in traffic LRD. We looked at both single-hop and multi-hop topologies, but did not focus on the role of the routing protocol. Routing protocols, especially for ad-hoc wireless networks, play a much bigger role in shaping the traffic at intermediate nodes. In fact, difficulty in finding stable and uncongested routes in most scenarios, are the main reasons for network performance bottlenecks. Even ignoring the channel fading and mobility, significantly higher MAC losses in wireless medium can result in routing changes even in static topologies. Hence, we expect routing protocol dynamics to significantly impact the traffic burstiness at large timescales and possibly traffic LRD.

At first, we carry out a simulation-based study (models given later) to find the relation between traffic burstiness and the impact of protocols in multi-hop wireless networks. In particular, we focus on the joint role of routing protocols and contention-based MAC on traffic burstiness. We study two of the most popular routing protocols - OLSR [38] and AODV [50]. We observe that losses of OLSR routing control packets, at the MAC layer, can lead to route changes even in static networks. Such route changes, in turn, result in bursty traffic at intermediate nodes.

Even though, the typical timescale of routing protocol packets is of the order of few seconds, we see that traffic burstiness is observed even at time-scales of up to several hours. Thus, unlike the wired networks studied before, the varying nature of wireless 'links' and their impact on routing protocols, result in traffic brustiness at much larger timescales.

Recently, there has been some criticism about the LRD analysis of finitely sampled time-series. At the same time, the impact of LRD traffic on a network with finite buffers is also sometimes questioned. However, irrespective of true LRD or not, traffic burstiness of the order of several hours has practical significance for both the actual network performance as well as network resource provisioning. Our own preliminary experiments show that the performance of contention-based MAC, like 802.11, is affected by the traffic burstiness at large-timescales. Also, LRD or not, the observed traffic oscillations directly impact the carried load in the network. For some of the scenarios, we observed a throughput drop of up to 15 % due to avoidable route oscillations. Hence, it is important to characterize the large-timescale behavior of traffic for more accurate performance evaluation and network design. Otherwise, the actual QoS will be much worse than predicted by traditional models.

For the OLSR scenario, we also model the observed traffic burstiness phenomena using models in two steps. At first, using a simple On/Off packet train model, we explain how route oscillations can lead to traffic burstiness, and how a Markovian framework for route oscillations is not sufficient to explain the large-timescales burstiness. Using a simple 4-node scenario with a single connection, we provide evidence for sub-exponential On and Off durations, providing some evidence for

protocol-induced 'heavy-tails'. In the second part, we model the 802.11 MAC and OLSR protocols to show how input traffic load levels can impact protocol dynamics and lead to topology changes even in a static network. We use approximate models to predict the presence of traffic burstiness at large timescales. In particular, we use the 802.11 MAC and OLSR models from Ch. 2 and Ch. 3, and modify one of the components - NDC (Neighbor Discovery Component, Sec. 3.1.1), to find the probability of topology changes; thereby, leading to traffic burstiness in at least some of the intermediate nodes. Lastly, while protocols can lead to traffic burstiness at large-timescales, appropriate tuning of protocol parameters can mitigate the effect. We use our models to correctly predict what choices of protocol parameters will results in traffic burstiness and what choices will not. Thus, we use the models not just to explain the observed phenomenon, but also to aid in network design.

The chapter is organized as follows: In Sec. 5.1, we provide the relevant protocol dynamics at play in our study. Sec. 5.2 includes our preliminary simulation studies which show evidence of traffic burstiness due to OLSR protocol and contention-based MAC. In Sec. 5.3, we give the details about the On/Off model for route oscillations. Sec. 5.5, provides a simplistic model linking load-levels, MAC failure probabilities and the resulting impact on OLSR routing. In Sec. 5.6, we test our models on newer scenarios and demonstrate how we can use them to tune the network parameters appropriately.

5.1 Routing Protocol Dynamics

For multi-hop scenarios, the routing protocol plays a role in shaping traffic. Even in a static topology, the routing protocol may change routes due to losses. Loss of routing protocol control packets at the MAC (due to congestion or physical layer losses) can lead to the inference of topology changes. This in turn, can result in route changes even for static topology. Such changes in the routes may lead to additional dynamics in the traffic at each node. In our simulations we either use the OLSR [38] or AODV [50] routing protocols. We summarize the details relevant to our study here.

OLSR is a proactive routing protocol. It relies on periodic HELLO and TOPOLOGY CONTROL (or TC) packets to discover local and global topology, respectively. Each node periodically broadcasts HELLO packets (with period T). Whenever a neighboring node receives a HELLO packet, the node assumes that the link to the sender is up. If D consecutive HELLO messages are lost (timed-out), the node infers link failure. Based on the local neighborhood information from HELLO packets, TC packets are broadcast throughout the network, and each node stores its limited view of the global topology. Based on this topology, each node chooses the best next-hop for a given destination. Any changes in local topology, inferred due to HELLO packets, triggers network-wide updates in global topology. Thus, losses in HELLO packets (even due to congestion or fading) can result in changes in inferred topology and hence, the routes.

AODV is a reactive routing protocol. Whenever a new connection is arrives,

the source attempts to find the best path to the destination. For a new destination, the source broadcasts Route Request (RREQ) packets which are propagated in the network. The destination node sends out Route Reply (RREP) packets along the best path, using the information stored in the RREQ packets. Once a path has been established, the liveliness of the links on the path is determined using both explicit and implicit methods. Explicit methods rely on periodic HELLO packets on the active links. Implicit methods rely on MAC layer ACKs or control messages. Any of these packets could be dropped at the MAC and PHY layers.

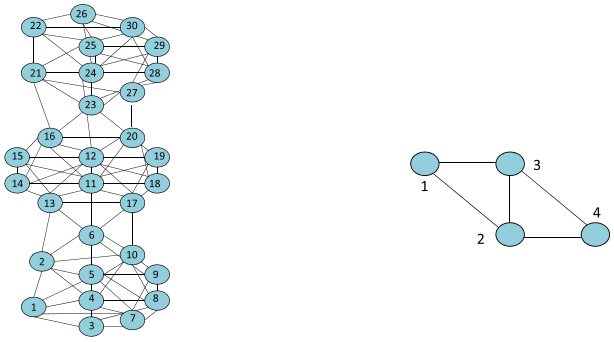
In OLSR, nodes keep an updated view of the entire topology. In the event of a link failure, nodes will quickly route paths on newer links. If that new path becomes congested later, it might trigger more link 'failures'. Thus, it is possible that the routes keep oscillating between two paths, leading to traffic burstiness at the intermediate nodes. In AODV, in contrast, nodes only store active routes. In the event of link failure, there is route recovery. Compared to an OLSR scenario, it is more likely that the route recovery mechanism returns the old path, although route oscillations can occur in AODV as well.

5.2 Simulation Study

At first, we conduct some preliminary experiments to analyze the role of protocol dynamics in shaping traffic at large timescales. This serves as the motivation for a more detailed modeling and analysis that follows. We used OPNET [37], to study traffic scaling in wireless networks. We studied multi-hop scenarios using

IEEE 802.11b MAC (DCF mode). MAC and PHY parameters were set to the standard values. However, to keep our focus on the MAC and higher layers, we set a unit-disc model for the communication range. To achieve this, we used a two-ray ground propagation model, and modified the CCK modulation scheme's BER curve to be either 0 or 1 depending upon the SNR. Thus, a pair of nodes within a given distance threshold can communicate without any physical layer errors. Even though we did not report the results here, we repeated our experiments with realistic PHY and SNR models, to obtain similar results qualitatively.

We used a 30-node topology as shown in Fig. 5.1(a). There are multiple short connections within the clusters, but there are some connections across the clusters as well. The traffic applications use TCP/IP protocols (we obtained similar results using UDP). We ran multiple simulations with varying load levels, and different random seeds for each load level. For each of the scenarios studied below, we ran each simulation for 1 day. At each node, we measured the MAC layer throughput every $\Delta = 0.1$ *sec*. To study the impact of routing protocols, we ran two sets of simulations - one each with OLSR and AODV protocols. We initially chose the default parameters according to the respective RFCs. However, route oscillations were clearly visible in OLSR scenarios. For AODV scenarios, we did not see any evidence of route oscillations. To study the OLSR route oscillations in more detail, we tuned the OLSR routing parameters as described later.



(a) 30-Node Scenario (b) 4-Node Scenario

Figure 5.1: Simulated Topologies

5.2.1 Routing Protocol Leads to Traffic LRD

For each connection, we used exponential file sizes (with mean 13715 Bytes) and exponential inter-request time, with mean 2 sec (for medium load) or 10 sec (for low load). For medium load (mean inter-request time = 2 sec), the total offered load was around 1.5Mbps. While it is difficult to characterize the capacity of the network, we call the load-level "'medium"' (and not "'high"'), because, for properly tuned routing protocols (as described later), all of the offered load was carried by the network. Even for the default OLSR scenario, the throughput was 99.6%.

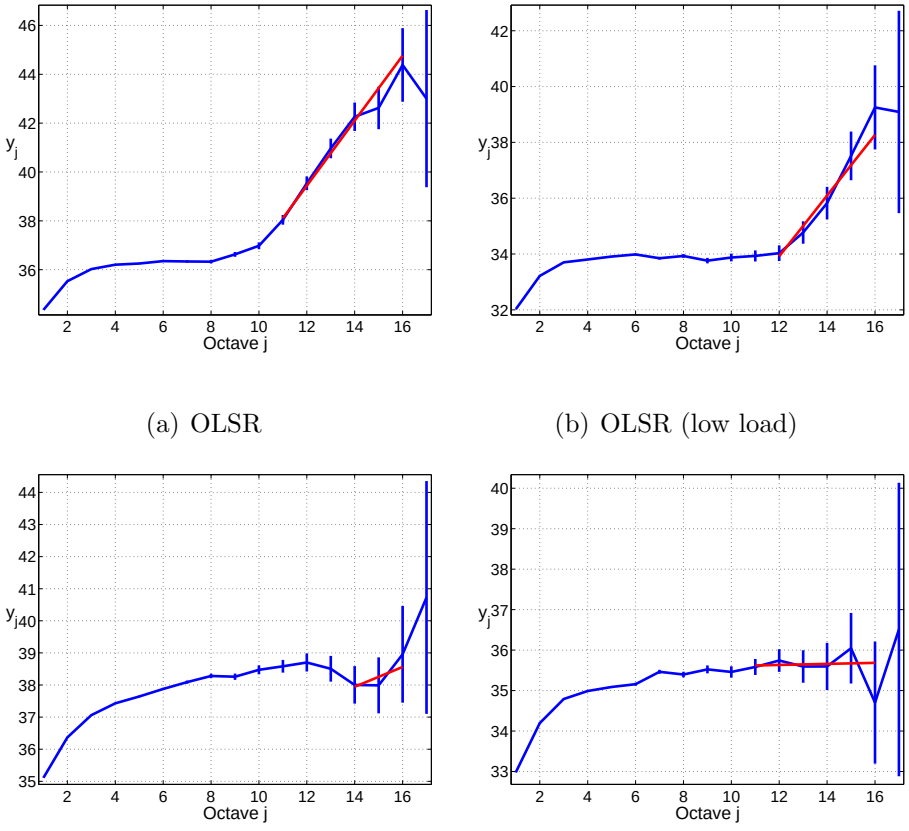
Fig. 5.2 shows the wavelet analysis for the traffic successfully received at the MAC layer of a sample Node 25 (measured in bps). On the x-axis, octave j corresponds to $t = 2^j * \Delta$, where $\Delta = 0.1$ seconds. Thus, *octave* $j = 10$ implies $t = 100$ seconds. Scaling at large timescales is indicative of traffic LRD (Refer Sec. 4.1). Fig. 5.2(a) shows that, the traffic at the intermediate Node 25 displayed LRD, when using the OLSR routing protocol. For exactly the same parameters, but using

AODV, there was no clear evidence of traffic LRD (see Fig. 5.2(c)). We made similar observations for low load scenarios. While we may debate whether the finite series is actually LRD or is 'pseudo-LRD', the traffic is definitely bursty at scales from hundreds of seconds to several hours. This is the same timescales studied in most of the measurement-based papers for traffic LRD [10–12]. Thus, to the best of our knowledge, unlike any other previous study, our simulations show for the first time that wireless network protocols can impact traffic at large timescales for several hours. This impacts the network performances as discussed before. For the sake of brevity, we call this traffic burstiness as traffic 'LRD' in the following discussions, although 'pseudo-LRD' will be more appropriate.

Fig. 5.3 shows the time series plot for traffic at sample node 25 for low-load scenario. Clearly, Fig. 5.3(b) shows that there were periods of very low traffic at node 25 when using OLSR. During the times when node 25 was carrying less traffic, neighboring nodes (especially node 24) carried more traffic (Fig. 5.3(c)). These results are evidence of route oscillations, in OLSR, leading to traffic LRD. At least for the scenarios studied, AODV induced little traffic burstiness.

5.2.2 Tuning OLSR Parameters to Prevent Traffic LRD

To study the impact of the OLSR protocol in inducing traffic LRD, we changed the OLSR protocol parameters. In OLSR, the default values of T is 2 sec, and D is 3 (Sec. 5.1). That is, if $D = 3$ consecutive HELLO messages are lost (timed-out), the node infers link failure, which results in topology and route updates. In our scenario,



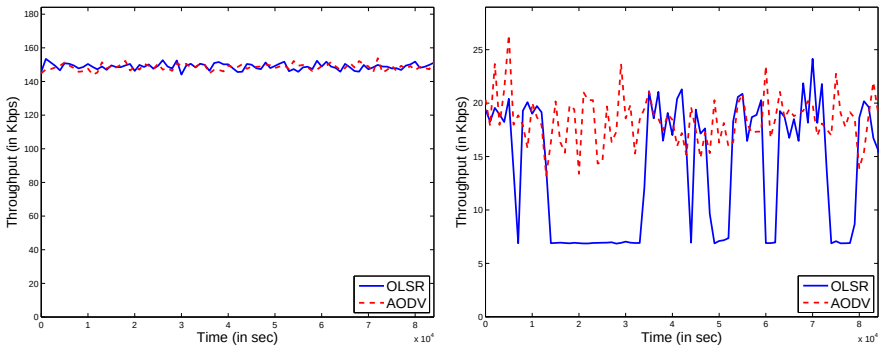
(a) OLSR

(b) OLSR (low load)

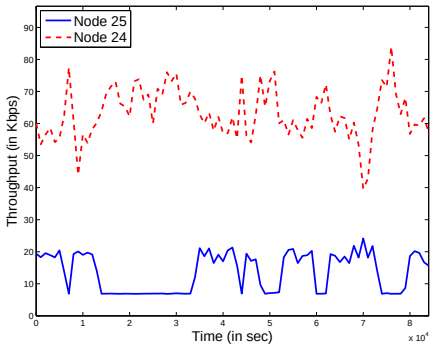
(c) AODV

(d) AODV (low load)

Figure 5.2: LRD due to Routing Protocol



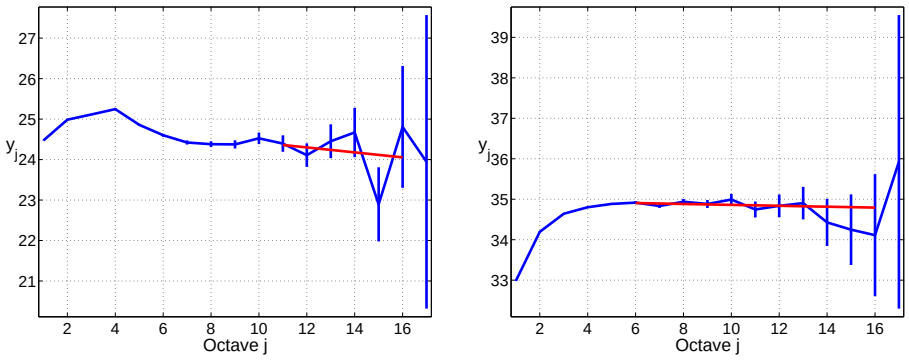
(a) Aggregate Traffic (b) Traffic at Intermediate Node 25



(c) Traffic Oscillation between Node 24 and 25 for OLSR scenario

Figure 5.3: Evidence of Route Oscillations in OLSR scenario

even though the topology is static, periodic HELLO packets may be lost at the MAC layer. To test our hypothesis that route changes are due to drops of consecutive HELLO packets, we set $D = 10$. Hopefully, the probability of 10 consecutive packet drops (spread over $T * D = 20sec$) is low enough to avoid unnecessary route updates. We re-ran our simulation and analysis for the scenarios as above with $D = 10$ for OLSR. We denote these scenarios as *Tuned-OLSR*.Fig. 5.4 shows that there was no LRD in traffic. Also, there were no route or topology updates after the initial setup time. Clearly, by tuning the OLSR parameters, unnecessary route oscillations were avoided. This in turn, leads to absence of routing-protocol-induced LRD.



(a) Tuned-OLSR (b) Tuned-OLSR (low load)

Figure 5.4: Parameter tuning in OLSR to avoid route oscillations and traffic LRD

5.3 Traffic Burstiness due to Route Oscillations

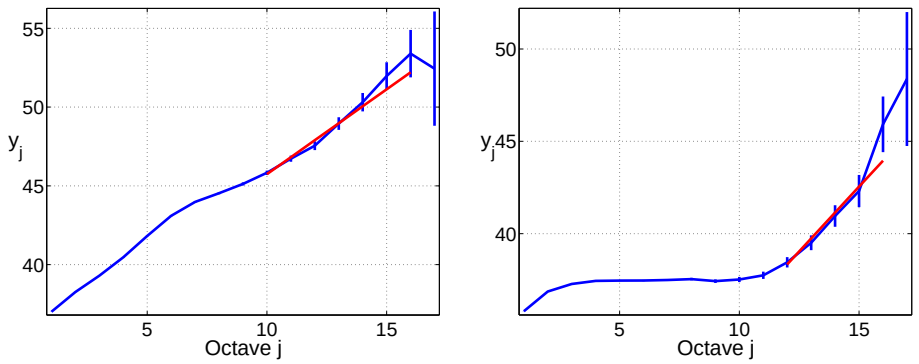
In this section, we investigate the relationship of the observed traffic burstiness at large timescales to the presence of route oscillations. We attempt to model the route oscillations using a simple On/Off packet train model and use it to explain

the observed traffic scaling.

We use a simple topology (Fig. 5.1(b)) to explain the observed phenomenon in more detail. Nodes 1 and 4 are the source and destination. Nodes 2 and 3 forward traffic depending upon routes chosen by the protocols. OLSR will typically select route $1 \rightarrow 2 \rightarrow 4$ or else $1 \rightarrow 3 \rightarrow 4$. However, due to intermittent link losses of OLSR Control Packets, OLSR can infer the topology incorrectly. Hence, the chosen route (and forwarded traffic) oscillates between the two paths. For the intermediate nodes, then, the traffic at it's MAC layer depends on the chosen route at the source. And hence, incoming traffic can be modeled using an On/Off model packet train model. Since we are focusing on the large time-scale behavior, we can ignore the small time-scale dynamics (due to the source traffic arrival pattern and the MAC layer). Hence, we assume a constant traffic rate during the On periods and no traffic during the Off periods.

We measured the traffic at intermediate nodes (2 and 3) and analyzed their scaling properties. Fig. 5.5 shows a few sample plots indicating a strong presence of traffic burstiness at large timescales. The high (/low) load scenario corresponds to mean inter-arrival time was 0.06 seconds (/0.8 seconds). We observed similar behavior for different load levels.

We also measured routing table statistics and deduced the On/Off durations for traffic at the intermediate nodes. For example, whenever node 2 is present on the chosen path, it corresponds to the On duration for node 2. And when the chosen path is $1 \rightarrow 3 \rightarrow 4$, it corresponds to the Off duration for node 2. Then, using the On/Off Model, we attempt to predict the traffic burstiness. Initially, motivated

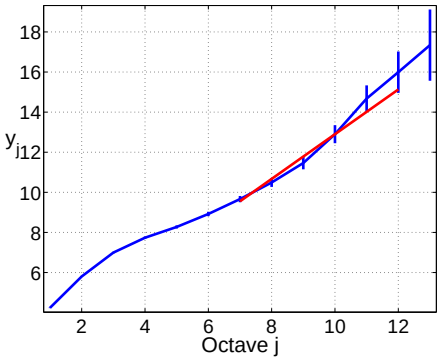


(a) High Load (b) Low Load

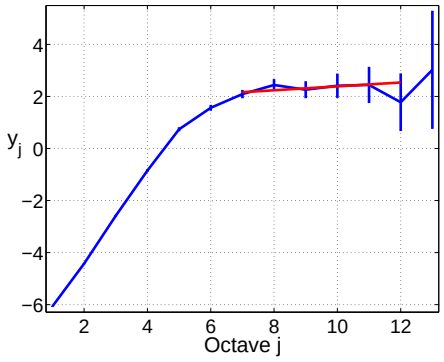
Figure 5.5: Timescale of Traffic Burstiness in 4-Node Scenario

by the usual Markovian framework, we assumed i.i.d. Exponential distributions for the On and Off Durations with appropriate mean (17.48 and 23.67, respectively) corresponding to the observed values. Fig. 5.6(b) shows that the Markovian model for route oscillations cannot explain the observed traffic burstiness (Fig. 5.6(a)) at large timescales of several hours, which is 3-4 orders of magnitude higher than the mean. However, if we use the empirical distributions, the prediction about traffic scaling is accurate (Fig. 5.6(c)). Note that, Fig. 5.6 uses $\Delta = 1$ (unlike $\Delta = 0.1$) seconds as the sampling interval, hence the shift in the x-axes.

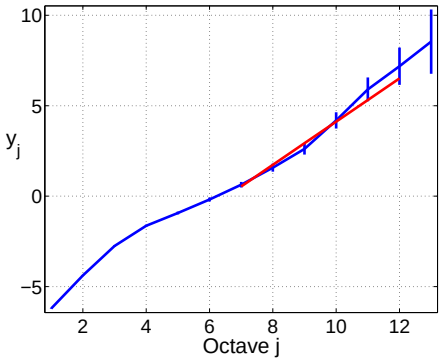
In Fig. 5.7, we plot the On and Off duration distributions for the high load scenario (mean inter-arrival time of 0.06 seconds). It shows the CCDF and quantile-quantile (QQ) plot for the On and the Off (empirical) distributions compared to exponential distributions with respective means. As can be seen, the On durations have much heavier-body, and slightly fatter tail compared to the exponential. Note that the CCDF plot is on a log-log scale. Repeated experiments with different load conditions and different random seeds yielded similar observations. These obser-



(a) Observed Traffic



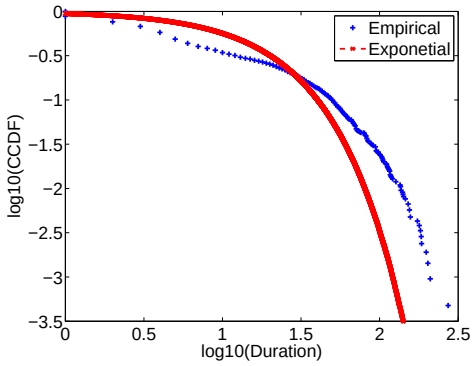
(b) On/Off Model with Exponential Duration Distributions



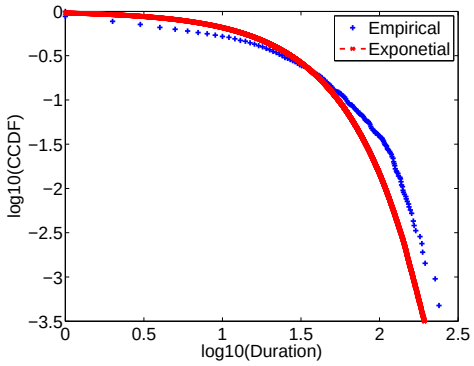
(c) On/Off Model with Empirical Durations

Figure 5.6: On/Off Model and Traffic Burstiness

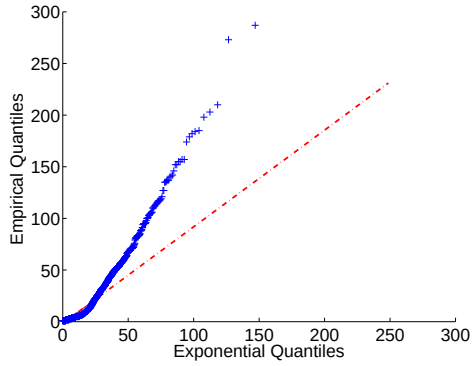
vations are in line with sub-exponential tails leading to burstiness at much higher scales. Whether the empirical distributions are true heavy-tails, needs more detailed evaluation.



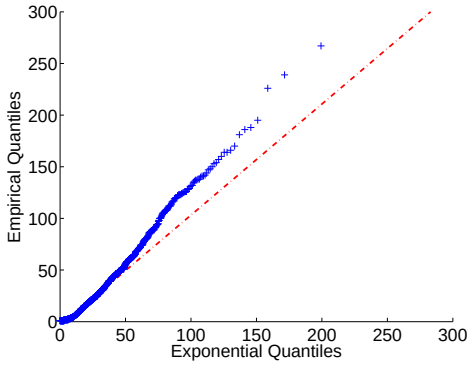
(a) CCDF for On Durations



(b) CCDF for Off Durations



(c) Quantile-Quantile Plot for On Durations



(d) Quantile-Quantile Plot for Off Durations

Figure 5.7: Empirical On and Off Durations Distribution Compared With Exponential

5.4 Modeling the MAC Collisions and OLSR Link Failures

5.4.1 MAC Model

We use the 802.11 MAC model developed in Ch. 2, to find the MAC loss probabilities depending upon the given topology and offered load. In our model, given the input load, the *MAC model* approximates the average MAC frame drop probabilities β (Sec. 2.1.1) over various links. We use these MAC frame drop probabilities as the probabilities of OLSR control packets losses, which form the input to our *OLSR model*.

5.4.2 OLSR Model

Here, we modify the OLSR model from Ch. 3 for our new metric of interest: the probability of topology changes.

For brevity, we assume $f_{ij} = f_{ji} = f$. (The general case of $f_{ij} \neq f_{ji}$ is a simple extension). The probability of change of bidirectional link status is, then:

$$p_{change} = p_{BU \rightarrow BD} \cdot p + p_{BD \rightarrow BU} \cdot (1 - p) \tag{5.1}$$

where

$$p_{BU \rightarrow BD} = \frac{2\pi_{U+D-1} \cdot f \cdot q}{p} \tag{5.2}$$

And,

$$p_{BU \rightarrow BD} = \frac{2\pi_{U-1} \cdot s \cdot q + \pi_{U-1}^2 \cdot s^2 - 2\pi_{U-1}\pi_{U+D-1} \cdot s \cdot f}{1 - p} \tag{5.3}$$

Thus, $p_{change}(f)$ gives the probability of state change for a particular link, as a function of it's MAC layer frame drop probability f. Change in any link status may results in global topology change and may lead to route calculations. Since, route recalculations are performed at every fixed intervals and aggregated for all link changes, we make another approximation. We assume that the probability of topology change ν is dominated by the probability of link status change of the most suspectible link. Hence, we assume $\nu = p_{change}(\beta_{max})$, where β_{max} is the maximum of (vector) β over all links, obtained from the *MAC model* described previously.

5.5 Intermediate Model Validation: Route Oscillations due to MAC Losses

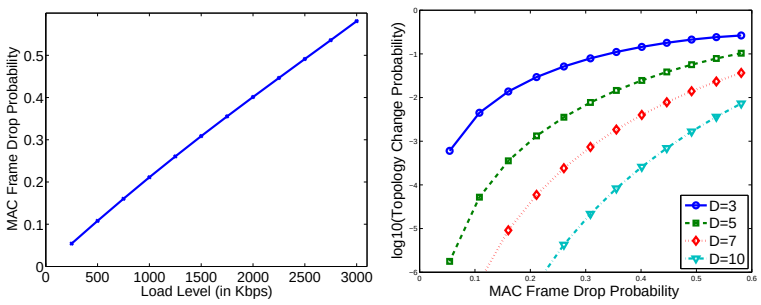
In Sec. 5.3, we used the On/Off model to demonstrate how the route oscillations lead to traffic burstiness, in a simple scenario. In this section, we aim to predict when route oscillations may occur in a large topology, when using OLSR. Depending on the input load levels, and the choice of OLSR design parameter D, we model the 802.11 MAC losses due to congestion and resulting OLSR topology changes. For this purpose, we use the performance models developed in Sec. 5.4.We present the model and simulation results here to demonstrate that the MAC and routing protocol dynamics are the origins for the observed traffic burstiness, and also this traffic burstiness can be mitigated by appropriate parameter choices. The *MAC model* takes the offered load level and topology as inputs to predict MAC frame drop probabilities for various links. The *OLSR model* uses these MAC loss

probabilities as input to give topology change probability and predict route oscillations. Thus, in this section, we use our models to connect input load levels, with MAC frame drops and OLSR topology / routing changes. We use the 30-node topology (Fig. 5.1(a)) to validate our models, and also demonstrate how we can design the protocol parameters to avoid traffic LRD.

5.5.1 Modeling: Link Losses and Topology Changes

5.5.1.1 Packet Collisions due to Increasing Load Levels

At first, using the *MAC model* for 802.11, we find the MAC packet or frame drop probabilities depending upon the input load level. Depending upon the topology and different traffic demands, each wireless 'link' will have varying probability for MAC frame drop due to collisions. OLSR HELLO packets will suffer losses according to these probabilities. Since more MAC loss probability leads to more dynamism in OLSR link changes, the overall rate of OLSR topology changes will be mainly governed by the most susceptible link. Hence, amongst all the different links, we consider the link with highest MAC layer losses (as input to the OLSR model). In Fig. 5.8(a), we show the maximum (over links) average (over time) MAC failure probability as a function of input load. Clearly, as the load increases, there are more collisions at the MAC level.



(a) (b)

Figure 5.8: Load Level, MAC Losses and OLSR Topology Change Probabilities

5.5.1.2 OLSR Link Failures due to Packet Collisions

Now, using the maximum MAC failure probability β_{max} as an input to *OLSR model*, we obtain the steady-state probability of OLSR link changes in the most susceptible link. Note that any local change in topology is likely to result in global topology changes and routes calculations. Also, changes at different links are aggregated and sent periodically. Hence, even if two links change in a slot, only one update will be propagated network wide. Thus, the rate or probability of OLSR topology changes (globally) is mainly determined by the rate of link status change of the most susceptible link. Hence, we assume that the *OLSR model* output gives the probability of global route table calculations. Using the *OLSR model*, Fig. 5.8(b) shows the probabilities of OLSR topology changes as a function of MAC loss probabilities, for various choice of OLSR parameter D. As conjectured in Sec. 5.2.2, higher D mean more resilience to MAC layer losses.

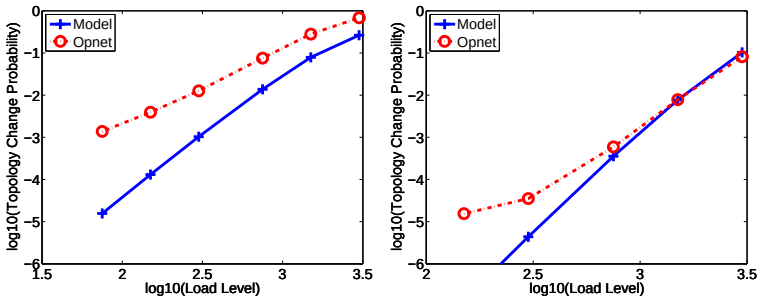
We compare the *OLSR model* output with actual statistics obtained from OPNET simulations. For a given value of OLSR parameter D, Fig. 5.9 shows

the model and OPNET probabilities of topology changes (and resulting route re-calculations), for various load-levels (normalized to 1 for maximum load of 3 Mbps). The topology change probabilities from OPNET are obtained as the fraction of instances when HELLO message exchanges (slotted) lead to topology changes. The y-axes are truncated to hide low data points. Also the missing data points for the OPNET data-set corresponds to $log(0)$, when no topology changes were observed during the entire simulation run. The model output matches very closely with the OPNET results. There is some mismatch at higher levels of MAC loss probabilities, especially for $D = 3$. We believe that is due to the assumption about the most susceptible link dominating the network-wide topology changes. However, even other links can increase the topology change rate, and their impact can be higher at higher MAC loss levels. Nevertheless, the model predicts the order of topology change probability correctly.

5.5.2 Analysis: Traffic LRD

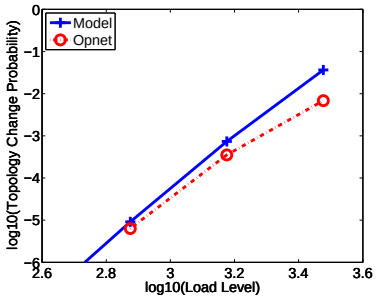
5.5.2.1 Traffic Burstiness due to OLSR Topology Changes

Next, we turn our attention back to traffic burstiness and wavelet analysis. We ran wavelet analysis on traffic at intermediate nodes (as in Sec. 5.2). For each value of D and different load-levels, we ran the simulation for 1 day, and repeated the experiment with 5 different random seeds. As before, we present the sample results for central node 25. However, instead of showing the numerous wavelet plots, we just list the results in Tab. 5.1. The table should be read as follows: the value

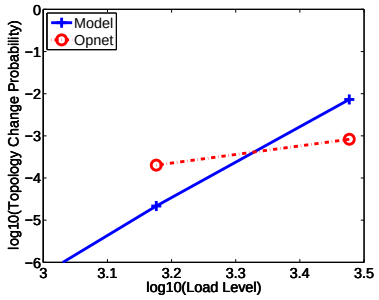


(a) OLSR with D = 3

(b) OLSR with D = 5



(c) OLSR with D = 7



(d) OLSR with D = 10

Figure 5.9: OLSR Topology Changes due to Increasing Load Levels

Load (in Kbps)	75	150	300	750	1500	3000
D=3	0	1	1	1	1	1
D=5	0	0	0	0	0.6	1
D=7	0	0	0	0	0	0.4
D=10	0	0	0	0	0	0
AODV	0	0	0	0	0	0

Table 5.1: Fraction of runs where we observed LRD at node 25 for different load levels in 30-node scenario

give the fraction of scenarios (out of the 5 random seeds), in which we observed traffic LRD according to the wavelet analysis. Clearly, we see that for low values of D, collisions at the MAC couple with OLSR to give rise to traffic LRD. And as we increase D, such traffic burstiness can be avoided. The last row also shows the results for AODV, validating the absence of traffic burstiness in any of the AODV scenarios.

Combining the topology change probabilities from the *OLSR model* and observed LRD in traffic, we look at traffic LRD as a function of topology change probabilities. In Fig. 5.10, shows the fraction (out of 5 random seeds) of scenarios where traffic was LRD at Node 25 (irrespective of the value for D). Thus, there is a strong relation between topology change and traffic LRD.

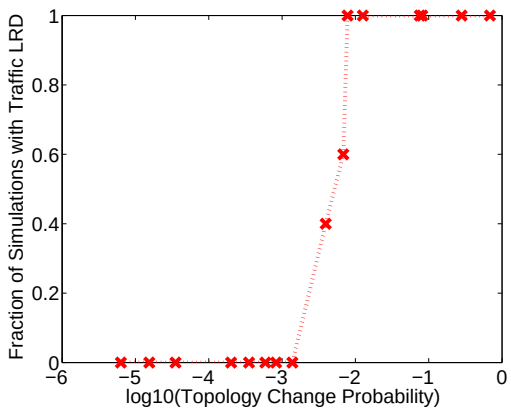
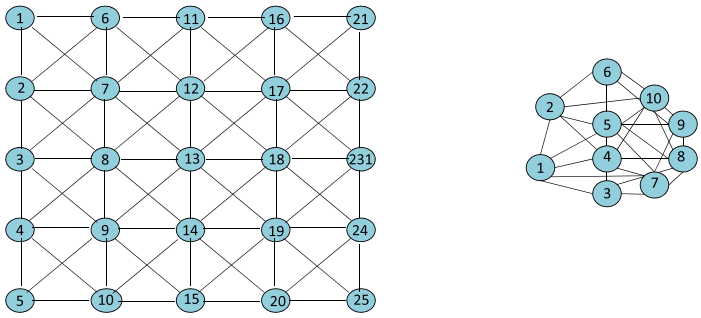


Figure 5.10: Traffic LRD with Topology Change Probability Estimate from the Model: 30-Node Topology

5.6 Model Validation and Network Design

We give additional evidence for traffic LRD. We, also, validate the models and their utility in choosing an appropriate value for OLSR design parameter D to avoid traffic LRD. For these purposes, we used the topologies shown in Fig. 5.11.



(a) 25-Node Grid (b) 10-Node

Figure 5.11: Other Topologies

We used our model to justify a choice of design parameter D. In particular,

given the topology and traffic demands, we relied on our model to find: 1) the MAC failure probability, and 2) Topology change probability, for a given choice of D. Then, we use the empirical relationship (in Fig. 5.10) between topology change probability and fraction of simulation runs where we observed traffic LRD. Thus, we attempt to use the empirical model obtained in one specific 30-node topology to work with any arbitrary multi-hop topology, provided it has enough richness in topology to allow for multiple paths. In particular, if the probability of topology change , obtained from our model, is non-negligible, then we should see traffic LRD at least in some nodes (as in Fig. 5.10). Hence, the particular choice of D is not suitable. Tab. 5.2 and Tab. 5.3 give the results for 25-node and 10-node scenarios respectively. They show the model outputs - the MAC loss probabilities and topology change probabilities, for different input load levels and design choice D. Then, the last row also gives the presence (or absence) of traffic LRD as observed from simulation traces. The table, and Fig. 5.10, show that the model correctly predicts which values of D will give rise to traffic LRD and which won't. In particular, if the topology change probability is too low, traffic will not be LRD. For high probability of topology change, traffic will be LRD (in atleast some of the intermediate nodes). The only exception seems to be at load level of 0.1 with $D = 3$. According to the model, the probability of topology change was negligible and hence we should not have observed traffic LRD. We believe this is due to the approximations of the model which seem to be inaccurate for low load levels and low value of D (similar to the mismatch in Fig. 5.9(a)). Overall, the model predicts traffic LRD accurately, and can be used to chose the right D.

Load (in Kbps)		36	360	720	1200	1800
MAC Failure Probabilities		0.0219	0.2190	0.4152	0.6354	0.6913
Topology Change Probability	D=3	4.1e-5	0.032	0.155	0.278	0.273
	D=10	9.9e-17	7.9e-7	3.5e-4	0.015	0.03
	D=20	2.5e-33	2.0e-13	5.4e-8	1.6e-4	7.6e-4
Traffic LRD	D=3	0.4	1	1	1	1
	D=10	0	0	0	0.2	1
	D=20	0	0	0	0	0

Table 5.2: Analysis for 25-Node Scenarios

To cross-validate the empirical relationship of Fig. 5.10 that used to predict traffic LRD from topology change probability, we plot similar graphs for each of the 25-node and 10-node topologies in Figs. 5.12 5.13. In particular, we plot the topology change probability with the fraction of simulation runs where we observed traffic LRD. We observe that the transition region from 'no LRD' to 'LRD' is similar in all three scenarios. Given the observed limitation of the model at low load levels for D=3, we have omitted the first value in each of the two scenarios when plotting the relationships.

Load (in Kbps)		120	1200	2400
MAC Failure Probabilities		0.0471	0.4303	0.7517
Topology Change Probability	D=3	3.9e-4	0.167	0.242
	D=10	2.0e-13	4.9e-4	0.053
Traffic LRD	D=3	1	1	1
	D=10	0	0	1

Table 5.3: Analysis for 10-Node Scenarios

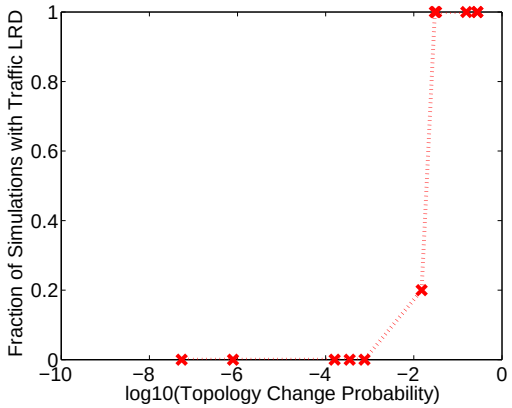


Figure 5.12: Traffic LRD with Topology Change Probability Estimate from the Model: 25-Node Topology

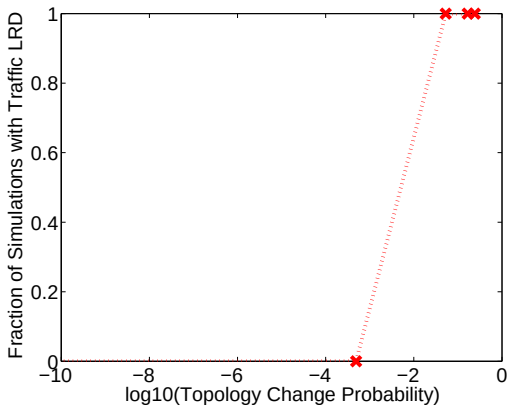


Figure 5.13: Traffic LRD with Topology Change Probability Estimate from the Model: 10-Node Topology

5.7 Summary

We explored the impact of routing protocols in large time-scale traffic burstiness. In particular, we observed that packet losses at the MAC (even due to congestion) can lead to route oscillations in OLSR. Route oscillations, in turn, lead to traffic burstiness at large timescales and pseudo-LRD. We showed that protocols in wireless multi-hop networks can induce traffic burstiness at timescales much larger than previously thought. We also showed that the route oscillations are not accurately modeled by relying on Markovian models for exponential durations. Lastly, we also used performance models for 802.11 MAC and OLSR routing, to correctly predict the MAC losses due to input load, and resulting route oscillations and traffic burstiness. Thus, using these models, we can design the network parameters to avoid such phenomena.

Chapter 6: Impact of Bursty Traffic on MAC Performance

In the previous two chapters, we looked at the role of MAC and routing protocol in creating or preserving traffic burstiness at large timescale or traffic LRD. In particular, we saw that the TCP traffic with heavy-tails in traffic sources was LRD. Secondly, even if there are no inherent heavy-tails in traffic sources, wireless protocol dynamics themselves can create traffic burstiness due to route oscillations. One of the main objective of the previous chapters (Ch. 4 and Ch. 5) was to show that traffic can be bursty in such networks. As we argued before, this is important aspect in performance modeling. In this chapter, we now focus on testing this hypothesis. In particular, we turn our attention to throughput and packet service times in the 802.11 MAC.

We look at how traffic burstiness at large timescales affects MAC's performance. Based on performance models developed for 802.11 MAC in Ch. 2, the MAC service times and throughput for a link depends on the load in its neighborhood. In our previously developed models, we used the average (offered) load conditions as an input to obtain the throughput under varying load conditions. Average load seems a reasonable input parameter, provided that the traffic is smooth (Possion) as timescales much higher than MAC service times. However, last chapter

shows that traffic can be bursty at even at large timescales, lasting up to at least several hours. Hence, we study, using simulations and models, to see if bursty traffic impact the throughput or the carried load. In particular, we set either smooth traffic (Poisson arrival of packets) or else bursty traffic (batched Poisson, where packets arrive in large bursts), keeping the mean load same. For the smooth traffic case, we use Exponential distribution for packet inter-arrival times with mean of the order of milliseconds. We change the mean to reflect the desired load level. For bursty traffic, we choose exponential inter-batch arrival time with mean of the order of a few seconds. Each batch has a fixed number of packets. We do not chose heavy-tailed distribution for bursty traffic in our this part of our analysis, because we were not able to establish if the On/Off durations are true heavy-tails or sub-exponential (Fig. 5.6). However, we observed interesting results even with this simple level of burstiness. Even if the traffic is bursty of the order of few tens of seconds, it it much higher than the MAC scale of operations (Sec. 4.3.2). We show that even with this level of traffic burstiness, the load successfully carried by the MAC layer can be different from the smooth traffic case. Hence, modeling the MAC service times in the presence of bursty load, can help us in developing and improving a MAC model for the problem described in Ch. 2. However, the problem is of significant importance on its own. They may show if and how the network performance can be impacted due to contention-based MAC and high variability in traffic.

To compare the role of contention and bursty traffic, we also compare the results with scheduling based MAC. In particular we look at two popular distributed scheduling-based MAC algorithms for mobile ad hoc networks - USAP [51] and

Greedy Backpressure [52]. USAP - Unifying Slot Assignment Protocol is a popular reservation-based MAC for multi-hop wireless networks. We use the implementation by [53] to simulate the USAP performance. Greedy Backpressure has been proven to be throughput optimal [52]. For our limited focus of interest, it means if the load level is below the capacity, then the traffic arrival pattern, including it's burstiness, will not impact the performance of Greedy Backpressure algorithm and it will schedule and deliver all the traffic. We expect, even in our simulations, that bursty traffic does not change the achieved throughput in Greedy Backpressure scenario.

6.1 Performance of 802.11 MAC for Input Load with High Variability

Performance of contention-based MAC protocols like 802.11 depends significantly on the congestion perceived in the wireless channel. If a sender perceives the channel as busy or experiences a lot of collisions, it may significantly increase the backoff time and take longer time to serve the packet. With this in mind, bursty traffic from the higher layers can lead to significant performance degradation. During the peaks of the bursty traffic, there may be increased contention, collisions and losses. This will increase the service time, and may also have some spill-over effect on the subsequent period of low load. The bursty traffic can also lead to increased losses due to retry attempt limit and finite buffer. While at the same time, during non-peak periods of bursty traffic, there can be low contention and collision, hence higher MAC efficiency. We wish to evaluate how the performance of 802.11 MAC is impacted by bursty traffic compared to smooth / Poisson traffic model usually used

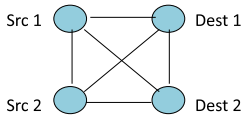
for performance modeling.

6.1.1 Motivating Examples

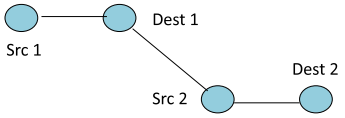
6.1.1.1 4-Node with 2 Connections Scenarios

To analyze the impact of bursty traffic on MAC performance, we initially start with simple 4 node topologies as shown in Fig. 6.1. In particular, we set either smooth traffic (Poisson arrival of packets) or else bursty traffic (batched Poisson, where packets arrive in large bursts), keeping the mean load same. For the smooth traffic case, we use Exponential distribution for packet inter-arrival times with mean of the order of milliseconds. We change the mean to reflect the desired load level. Each packet is of fixed size of 1000 Bytes. For bursty traffic, we choose mean inter-arrival time for each batch of packets to be of the order of a few seconds. Each batch has a fixed number (1000) of packets. And again, the inter-batch arrival times are still exponential (not heavy-tailed). We compare the following metrics - carried load, MAC service times, MAC retransmission attempt distribution and retry limit losses.

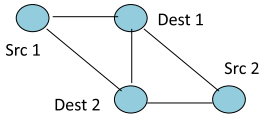
In general, we observe that if the traffic is bursty above a few tens of seconds (much larger than the MAC service times), then the average MAC service time is actually lower in most the scenarios. Simulations show that bursty traffic can lead to higher throughput in some scenarios. For the Asymmetric node scenario (Fig. 6.1(b)), when the offered load on both the links is capacity, the results are shown in Fig. 6.2. This is un-intuitive and unexpected. This is because, for bursty input



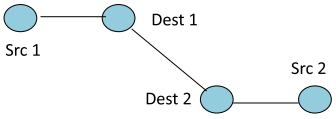
(a) Coordinated Nodes Scenario



(b) Asymmetric Nodes Scenario



(c) Near Hidden Nodes Scenario



(d) Far Hidden Nodes Scenario

Figure 6.1: Topologies

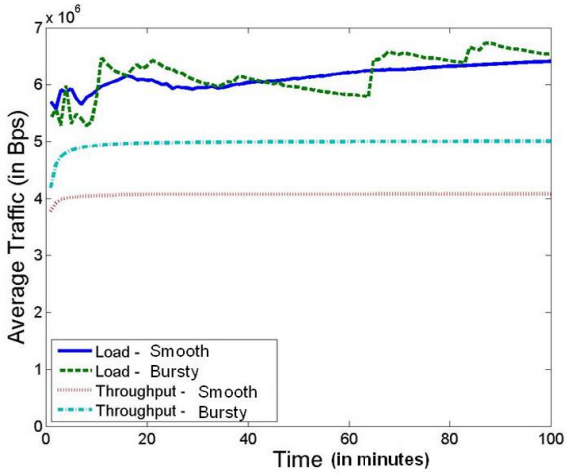
load, the retransmissions and losses are bursty. While these leads to slightly higher frame drops during periods of high burstiness, the overall MAC service times are lower. This leads to higher throughput.

We observed similar results for

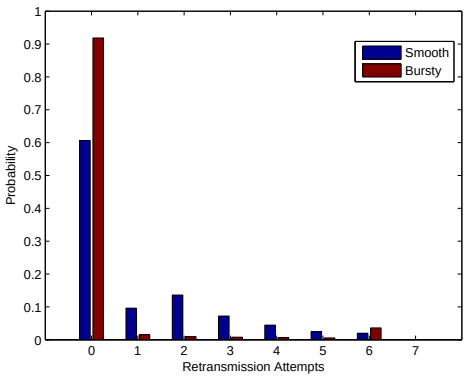
- Simple wireless channel model - unit disc model with BER curve as a step function
- Realistic channel model and BER curve
- Without using RTS-CTS (with hidden node problem)
- Even when using RTS-CTS

We repeated the experiments (with/without unit disc model and with/without

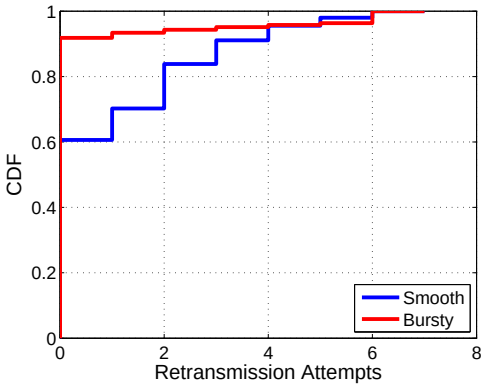
RTS/CTS). The carried load and the MAC service times for the above 4 topologies, when using RTS-CTS with unit disc model are given in Tab. 6.1. For other three topologies (Figs. 6.1), the difference in throughput is not as significant as in Fig. 6.2 for similar input / offered load levels. Even for the other scenarios (without RTS/CTS, etc.), we observed similar results.



(a) Average Input Load



(b) Retransmission Attempts (PMF)



(c) Retransmission Attempts (CDF)

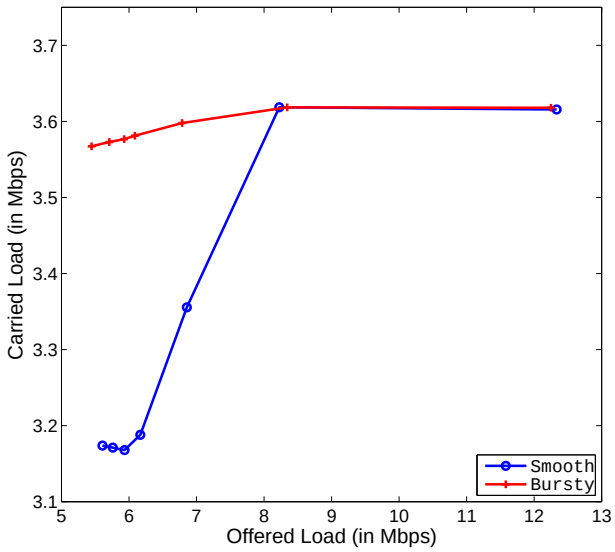
Figure 6.2: Variation in MAC Performance due to High Input Variability.

6.1.1.2 Simulation-based Analysis of 4-Node Asymmetric Scenario

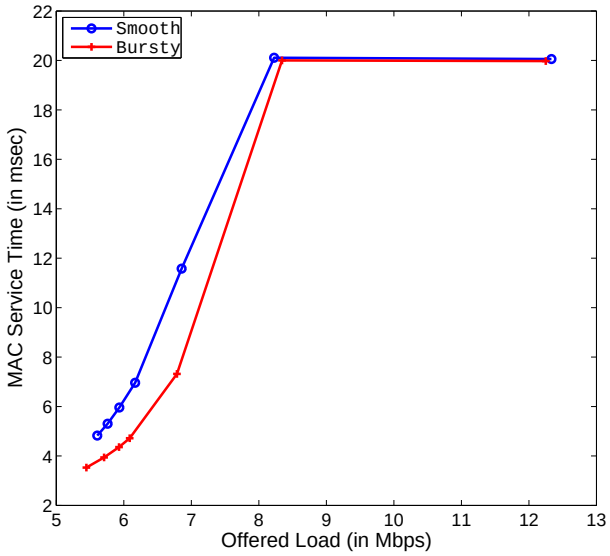
We varied the load levels and compared the carried load for smooth and bursty traffic. The results are shown in Fig. 6.3(a). We observed that for offered load closed to capacity, the carried load for bursty traffic was higher. However, when the offered load is above capacity, then both bursty and smooth traffic achieve similar throughput (carried load equal to the capacity). In particular, the higher observed throughput for bursty traffic, is indeed due to lower MAC service times. Fig. 6.3(b) shows the MAC service times for connection 1. Higher offered load leads to higher service times due to contention. However, for bursty traffic, the average service times are lower, leading to higher throughput, when offered load is slightly below capacity. The MAC service time for connection 2 was close to 2.4 msec irrespective of the load level, since connection 2 does not have to contend for the channel.

6.1.2 Multihop Scenarios

Based on the previous section, we saw a limited impact of bursty traffic on network performance for the 4-node scenarios. To understand the impact on performance in more complicated scenarios, we chose a few multi-hop scenarios and use simulations for analysis. We repeated our experiments with 10-node clique, 25-node grid (5-by-5) and 30-node complex multi-hop scenario. For the 10-node clique scenario, simulation results show no variation in throughput, but MAC service time lower for bursty traffic.



(a) Overall Carried Load



(b) MAC Service Time for Connection 1

Figure 6.3: 4 Node Asymmetric Topology

Topology	**Maximum Carried Load** (in Mbps)		**MAC Service Time** (in milliseconds)	
Traffic	Smooth	Bursty	Smooth	Bursty
Asymmetric	3.17	3.57	5.2	3.9
Coordinated	3.73	3.73	4.1	4.3
Far Hidden	3.48	3.52	4.2	3.8
Near Hidden	3.23	3.26	5.1	5.1

Table 6.1: Impact of Bursty Traffic on Carried Load and MAC Service Time

For the 25-node and 30-node multi-hop scenarios, results show some variation in connection throughput, but overall throughput remains unchanged. Similarly, MAC service times show variation per hop. However, exact nature of impact depends on the underlying network topology and traffic demands.

Observations for General Multi-hop Scenarios:

- Preliminary results show that when the load is close to 'capacity' (our region of interest), the results of lower service times for bursty traffic holds.
- The impact of bursty traffic in enhanced in the presence of realistic PHY (using default BER curves as opposed to unit disc model). This needs to be examined.
- More complicated scenarios like the 30-node scenarios show some difference in carried load. However, it is difficult to pinpoint the exact trends due to

multiple connections.

6.2 Modeling: 802.11 MAC Models For Bursty Input Traffic

Simulations above show that there are some scenarios in which bursty traffic can lead to higher overall carried load / network capacity. While most other scenarios show little impact on the network capacity. We now develop a model to explain the observations from the above simulations. We attempt to understand the impact of bursty traffic on network capacity in more detail. Through our models, we attempt to explain the observed throughput results as well as analyze other metrics like retries, losses and service times. But, more importantly we wish to predict the network performance in more general scenarios. We limit our focus on obtaining the maximum throughput in terms of total carried load in the network. Interesting results in terms of throughput performance is possible mainly near the high load region. For low load, irrespective of the input offered load being smooth or bursty, carried load is expected to be close to the offered load. But, as seen from simulations, carried load, when close to network capacity, can depend upon traffic pattern. Thus, we limit our focus on characterizing this 'network capacity' defined in terms of maximum carried load for a given network topology and its dependence on the burstiness of its input traffic.

We modify our previously developed MAC performance model (Ch. 2) to take into account the presence of bursty traffic. To quickly recap, given the topology, connections and offered load on each of the connections (denoted by vector $\underline{\lambda}$), the

model calculates the performance metrics such as MAC service times ($\underline{T}$), link layer loss probabilities ($\underline{\beta}$), and carried load. In that model, we only deal with averages - average offered load, average service times, etc. However, here our focus is on the impact of bursty traffic. Hence, we modify our model to account for burstiness of offered load as explained below.

Instead of taking the average offered load on each path as the input (Eq. 2.9), we assume an ON/OFF model for the input load for bursty traffic. During the ON period, the offered load (λ^{on}) is assumed to be higher than the capacity ($\rho^{on} = 1$) and the average duty cycle ($D = t_{on}/t_{on} + t_{off}$) is adjusted to keep the average offered load as desired

$$\lambda^{avg} = D * \lambda^{on} + (1 - D) * \lambda^{off} \tag{6.1}$$

The length of ON and OFF periods (t_{on} and t_{off}, respectively) are assumed to be sufficiently higher than MAC service times. Thus, we assume that steady-state for the MAC is reached within each ON or OFF period, before transitioning into the other state. Furthermore, we assume small queues at the MAC, hence data buffered during the ON period is dropped and does not cause traffic load during the OFF period. Hence, for each period, we can use the steady-state equations developed in Ch. 2. We note that the assumption about small queues is for the simplifying assumptions about traffic in On/Off durations. Our model does not explicitly capture queue length. We ensure small buffer size during simulations to match with our model assumptions.

As we saw earlier, the MAC model is described using set of model parameters using implicitly defined equations. To obtain the performance results for given input conditions, we use a fixed point algorithm to numerically arrive at the solution. We can use a similar approach here. However, such approach will not give us the insights we desire. Hence, to gain more insights, we first focus on the simple 4-node scenarios and attempt to simply the equations for each of the scenarios. This way, we can qualitatively understand the role of bursty traffic on contention, losses and network capacity. For the 4-node asymmetric topology, the simplified equations give us an intuitive explanation for the observed higher carried load. Similarly, the equations enable us to explain why we don't see similar impact of bursty traffic in the other 4-node scenarios at similar load levels. Also, interestingly, the model helps to predict under what circumstances or at what other load levels, can we see a noticeable impact of bursty traffic.

The four 4-node scenarios are the four fundamental topologies forming the basis for other complicated / multihop topologies. In particular, when focusing on the MAC throughput, losses and service times on a particular hop, that link's performance can be explained using a composition of these four 4-node scenarios. While we cannot simplify and analyze the set of equations for more general topologies, we hope that we can gain sufficient insights from the 4-node scenarios to predict the impact of bursty traffic on more complicated network scenarios.

6.2.1 Analysis of 4-Node Asymmetric Scenario

We characterize the maximum carried load on one of the two connections for varying offered load on the other connection. We consider both cases when the neighboring connection has smooth traffic or bursty traffic. But for the link under consideration (for maximum carried load), we assume smooth traffic. For brevity, we re-label the nodes $Src1, Dest1, Src2, Dest2$ (Fig. 6.1(b)) as $s1, d1, s2, d2$ (sources and destinations for connections 1 and 2), respectively.

6.2.1.1 Maximum Carried Load on Connection 2 when Bursty Traffic on Connection 1

We first focus on the carried load on connection 2, for varying offered load on connection 1. Here, we assume the offered load on connection 1 is:

$$\lambda_{s1}^{avg} = D * \lambda_{s1}^{on} + (1 - D) * \lambda_{s1}^{off} \tag{6.2}$$

For bursty traffic on connection 1, we assume $\lambda_{s1}^{on} = \lambda^{max}$ and $\lambda_{s1}^{off} = 0$. Here, we use the notation of λ^{max} to indicate offered load equal to the link capacity assuming no other link is active. Thus, λ^{max} is the maximum carried load in a network with a single source and a single destination. For such conditions, $\rho = 1$ on that particular link. We vary D for varying the average offered load on connection 1. For connection 2, we assume a constant offered load of $\lambda_{s2}^{avg} = \lambda_{s2}^{on} = \lambda_{s2}^{off} = \lambda^{max}$ to obtain the maximum carried load.

The carried load on connection 2 is given by

$$\lambda_{d2}^{avg} = D * \lambda_{d2}^{on} + (1 - D) * \lambda_{d2}^{off} \tag{6.3}$$

where we use the steady-state equations for each of the *on* and off periods. In particular, $\lambda_{d2}^{on} = k_{s2}^{on} * (1 - (\beta on_{s2})^m)$, where k is the serving rate at the source of the link and β is the probability of link layer loss. See Ch. 2, Eq. 2.7 for more details.

Focusing first on the off duration, we simplify the model equations. Since connection 1 is not active and only connection 2 is active, link loss probability for the active link is $\beta_{s2}^{off} = 0$. Thus,

$\lambda_{d2}^{off} = k_{s2}^{off} * (1 - \beta_{s2}^{off}) = \lambda_{s2}^{off} = \lambda^{max}$.

For the *on* duration, $\beta_{s2}^{on} = 1 - (1 - \theta_{d2,s2}^{on})$ and $\beta_{s1}^{on} = 1 - (1 - \theta_{d1,s1}^{on})$ where θ is the probability of hidden node transmission. Now, for the given topology, there is no hidden node for $d2$, hence $\theta_{d2,s2} = 0$. And $\beta_{s2}^{on} = 0$ too. Therefore, as for the off duration,

$\lambda_{d2}^{on} = k_{s2}^{on} * (1 - \beta_{s2}^{off}) = \lambda_{s2}^{on} = \lambda^{max}$.

Thus, $\lambda_{d2}^{avg} = \lambda_{d2}^{on} = \lambda_{d2}^{off} = \lambda^{max}\ \forall D \in [0, 1]$. Hence the maximum carried load on connection 2 is independent of the offered load on connection 1. In particular, bursty or smooth traffic on connection 1 does not impact carried load on connection 2.

For the *on* duration, we also measure the carried load on connection 1. When $\lambda_{s2}^{on} = \lambda^{max}$, $\rho_{s2}^{on} = 1$ and the probability of transmission from hidden node (neighbor for $d1$ hidden from $s1$), $\theta_{d1,s1}^{on} \sim 1$. Thus, $\beta_{s1}^{on} = 1 - (1 - \theta_{d1,s1}^{on}) * (1 - \alpha_{s2,d2}^{on})^V$. Since

$\theta^{on}_{d1,s1} \sim 1$, we get $\beta^{on}_{s1} \sim 1$. Similarly, it can be shown that the MAC service time T is very high. Thus, very high link layer loss probability and high service time for Connection 1, results in negligible carried load on connection 1, $\lambda^{on}_{d1} = k^{on}_{s1} * (1 - \beta^{on}_{s1}) \sim 0$.

We summarize and explain the results in more intuitive way: Connection 2 has an asymmetric advantage over connection 1. $d2$ does not have any hidden node collisions and $s2$ has to only contend with the CTS from $d1$ and not the data transmission from $s1$. Hence, at steady-state, the carried load on connection 2 equals the offered load (provided it is below the capacity). The traffic level or burstiness on connection 1 has little impact on connection 2, except a negligible increase in service time. However, the carried load on connection 1 is adversely impacted by the offered load on connection 2. We turn to this in the following section.

6.2.1.2 Maximum Carried Load on Connection 1 when Bursty Traffic on Connection 2

The more interesting and non-intuitive case is of traffic burstiness of connection 2 impacting the average carried load on connection 1. As shown in Fig. 6.3, bursty traffic on connection 2 leads to higher carried load on connection 1.

Here, the offered load on connection 2 is $\lambda^{avg}_{s2} = D * \lambda^{on}_{s2} + (1 - D) * \lambda^{off}_{s2}$, where $\lambda^{on}_{s2} = \lambda^{max}$ and $\lambda^{off}_{s2} = 0$. For connection 1, the offered load is $\lambda^{avg}_{s1} = \lambda^{on}_{s1} = \lambda^{off}_{s1} = \lambda^{max}$. Now, we solve for the carried load on connection 1: $\lambda^{avg}_{d1} = D * \lambda^{on}_{d1} + (1 - D) * \lambda^{off}_{d1}$.

For the off duration, since connection 2 is not active, link loss probability for connection 1 is $\beta_{s1}^{off} = 0$. Thus,

$\lambda_{d1}^{off} = k_{s1}^{off} * (1 - \beta_{s1}^{off}) = \lambda_{s1}^{off} = \lambda^{max}$.

For the on duration, the input offered loads are similar to the on duration handled above. That is, both $\lambda_{s1}^{on} = \lambda_{s2}^{on} = \lambda^{max}$. Also, $\beta_{s2}^{on} = 0$ and $\beta_{s1}^{on} \sim 1$. Hence,

$\lambda_{d2}^{on} = k_{s2}^{on} * (1 - \beta_{s2}^{off}) = \lambda_{s2}^{on} = \lambda^{max}$.

and,

$\lambda_{d1}^{on} = k_{s1}^{on} * (1 - \beta_{s1}^{on}) \sim 0$

Thus, $\lambda_{d2}^{avg} = D * \lambda_{d2}^{on} + (1 - D)\lambda_{d2}^{off} = D * \lambda^{max}$.

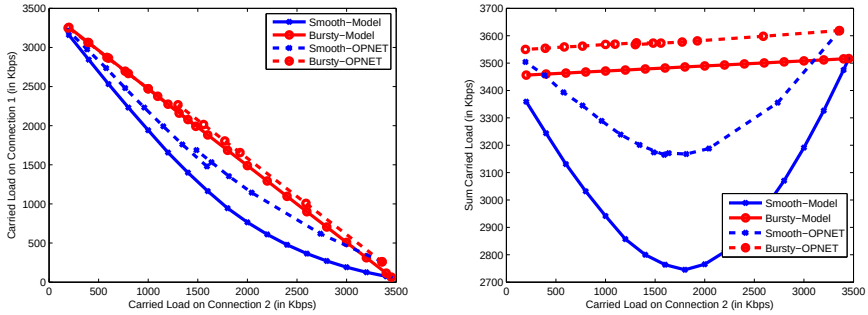
and,

$\lambda_{d1}^{avg} = D * \lambda_{d1}^{on} + (1 - D)\lambda_{d1}^{off} = (1 - D) * \lambda^{max}$.

Hence the maximum carried load on connection 1 depends on the offered load on connection 1, controlled by the parameter D. However, the total carried load (for both the connections) is λ^{max}. This occurs only if the traffic is bursty at large timescales. Since, the $on - off$ durations are long enough (to reach steady state), it allows both the connections (especially connection 1) to utilize the other connection's inactive duration effectively. Especially, since connection 2 is inactive for sufficiently long duration, connection 1 can utilize the channel without contention or collision. On the other hand, when the traffic model is assumed smooth, then there are negligible extended periods of low load on connection 2. Hence, throughput (or average carried load) on connection 1 is lower because of the time wasted in collisions and backoffs. This phenomenon explains the observed earlier in simulation results of Fig. 6.3(a). Thus, high traffic burstiness on connection 1, improves the

average total carried load as well as the carried load on connection 1.

We now plot the maximum carried load on connection 1 for varying carried load on connection 2 in 6.4. The connection 2 offered load (not shown) is either smooth or bursty as indicated. The figure also plots the total carried load on the two links. As is seen clearly, bursty traffic leads to higher carried load in at certain load levels, compared to smooth traffic scenario.



(a) Carried Load on Each Connection (b) Total Carried Load

Figure 6.4: Carried Load for 4-Node Asymmetric Topology

6.2.1.3 Model Validation Using OPNET Simulations

We also validate our model through OPNET simulations. As before, we either use smooth (exponential inter-arrivals with mean in few milliseconds) or bursty (batch arrival with mean inter-arrivals of a few seconds). We plot and compare the OPNET and our model results measured in terms of carried load on both the connections. We note that, our model assumes steady state is reached in each of the On and Off durations. Whereas the OPNET simulations results also take into account the transient behavior. Hence, we expect the OPNET result to show

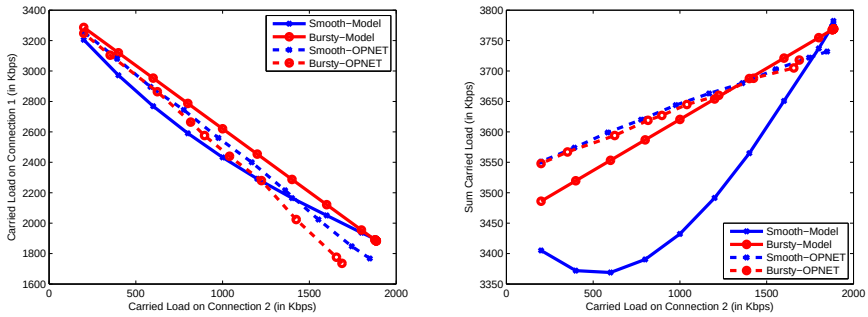
reduced impact of bursty traffic, as compared to our model. However, Fig. 6.4 (and later figures) show that, the difference in OPNET and our model results is very little. More importantly, the model captures the simulation results qualitatively very well. In particular, the range of load when the impact is highest is same in simulations and our model.

6.2.2 Analysis of 4-Node Coordinated Scenario

Unlike the 4-node asymmetric scenario, we only need to consider one case, because of symmetry of the two connections. We characterize the maximum carried load on connection 1, assuming connection 2 to have bursty (or smooth) traffic with varying offered load.

In this scenario, there are no hidden nodes. Hence, $\theta = 0$ (hidden node collision probability) for all the links. However, link-layer loss probabilities β and service time T for both the connections depend on the load on other connections. Hence, unlike the 4-node asymmetric scenario above, we cannot completely decouple the set of equations for two connections. In particular, while characterizing the maximum carried load on connection 1, connection 2 metrics cannot be assumed to be independent of offered load on connection 1. Hence, we have to fallback to numerically solving the coupled set of equations. We give the numerical results in Fig. 6.5. Compared to maximum difference of 3500 Mbps (Bursty) vs 2750 Mbps (Smooth) in total carried load in Asymmetric topology (Fig. 6.4(b)), the maximum difference between smooth and bursty traffic scenario is negligible (3530

Mbps vs 3360 Mbps). We only show the zoomed-in portion of the results, hence the axes in Figs. 6.4 and 6.5 are not similar. Thus, for coordinated node scenario, there is little difference in throughput, losses or other MAC metrics of connection 1 when considering either smooth or bursty traffic on connection 2. In other words, maximum carried load in this scenario is not affected by presence of bursty traffic.



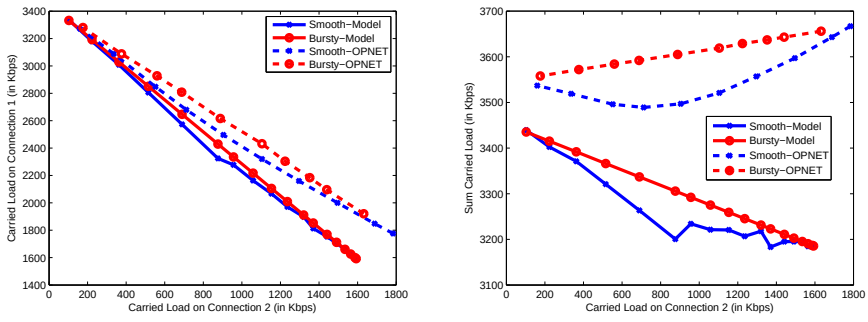
(a) Carried Load on Each Connection (b) Total Carried Load

Figure 6.5: Carried Load for 4-Node Coordinated Topology

Analyzing the results in more detail, we see that there is little difference in the MAC service time for high offered load scenarios. Since there are no hidden nodes, there are little collisions and retransmissions. The higher service times are simply because of the waiting for neighboring transmissions to finish. This small variation in MAC service times and near perfect knowledge of the channel conditions, result in negligible impact of neighbor's traffic pattern - smooth or bursty. Furthermore, the symmetry of the topology implies, whenever there is little traffic one connection, the other connection effectively utilizes the available channel resources. Hence, when characterizing the (sum) capacity region of the topology, traffic arrival pattern does not play any role.

6.2.3 Analysis of Other 4-Node Scenarios

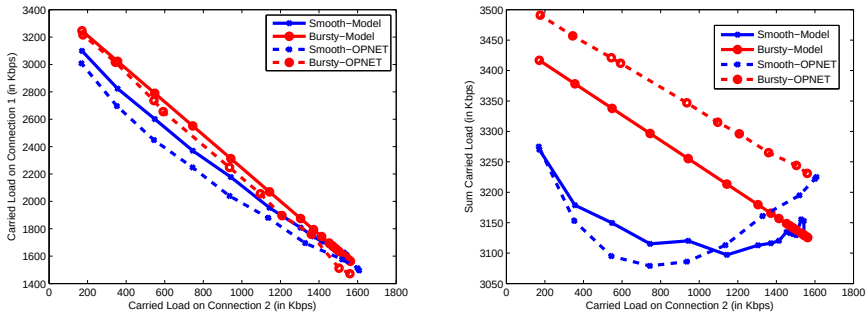
We now analyze the remaining two topologies (Fig. 6.1) - near hidden and far hidden scenarios. Like the 4-node coordinated scenario, the equations for each of the connections cannot be completely decoupled from the other connection. Also, like before, the two connections are symmetric. Hence, any reduction in one connection's input load is quickly captured by other connection; and, any increase of load on one link results in reduction on the other. However, unlike the 4-node coordinated scenario, there are hidden nodes which make the traffic compensation not exact. The MAC service times are much higher due to collisions and retransmissions due to hidden nodes. Because of the imperfect knowledge of channel conditions (due to hidden nodes), there is some loss of MAC efficiency. Overall, from the model, bursty traffic has some impact on maximum carried load, but the difference is not as high as in the 4-node asymmetric scenario.



(a) Carried Load on Each Connection (b) Total Carried Load

Figure 6.6: Carried Load for 4-Node Far-Hidden Topology

Thus, the models above agree with the simulation based observations in Tab.



(a) Carried Load on Each Connection (b) Total Carried Load

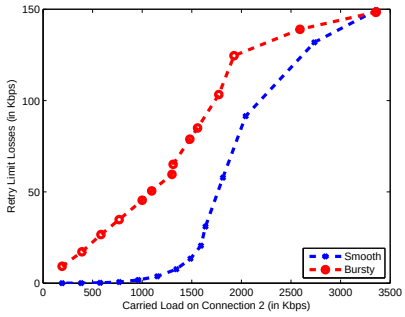
Figure 6.7: Carried Load for 4-Node Near-Hidden Topology

6.1. The role of bursty traffic is most evident in 4-node asymmetric scenario. Because of the imperfect knowledge, the disadvantaged connection 1 is not able to effectively utilize the channel capacity when connection 2 is not transmitting (when the traffic on connection 2 is smoothly paced out). Symmetry of the network and the near perfect feedback, in the other 3 scenarios, ensure that network capacity is effectively utilized and traffic burstiness has little impact on network performance.

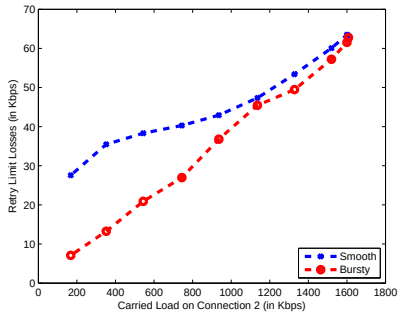
Based on these observations, we believe that for more complicated network topologies, traffic burstiness will cause some impact on network performance. However, higher carried load on one link (as in the case of 4-node asymmetric scenario) can result in lower carried load in it's vicinity. Thus, while some connections can enjoy higher throughput in case of bursty traffic, overall carried load for the network may not be much different in most scenarios.

As observed in Fig. 6.2, even though the carried load can be higher for bursty traffic, it can come at the cost of increased losses due to retry / retransmission limit. Hence, we also study the losses due to retry limit in each of the scenarios studies

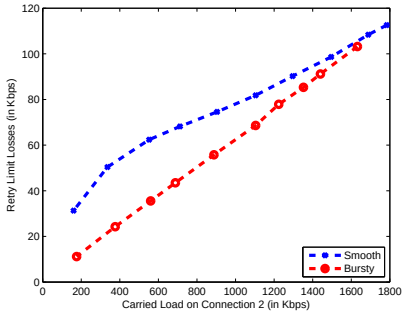
above. Fig. 6.8 shows the losses due to retry limit in 3 of the 4 topologies considered. For the 4th topology - coordinated node topology, the retry limit losses are negligible (less than 0.1 Kbps) for all the load levels for both smooth and bursty traffic. For the asymmetric topology, we indeed see higher losses for bursty traffic case. However, for near-hidden and far-hidden topologies the retry limit losses are lower for bursty traffic scenario. This may seem counter-intuitive, at first. However, for bursty traffic scenario, the average retry limit losses for over the alternate On/Off durations can be lower than the average retry limit losses for the smooth traffic scenario.



(a) Asymmetric Topology



(b) Near-Hidden Topology



(c) Far-Hidden Topology

Figure 6.8: MAC Losses due to Retry Limit

6.3 Scheduling-based MAC and Bursty Traffic's Impact

6.3.1 USAP - Scheduling-based MAC

6.3.1.1 4-node scenarios

Simulation results show no impact of bursty traffic in 4-node scenarios (Fig. 6.1), when trying to find the capacity region.

The interference pattern is different in USAP than 802.11 MAC. Unlike 802.11 MAC, USAP model uses 2-hop interference model for scheduling. Hence, there are no hidden nodes impacting the scheduling algorithm in the 4-node topologies. Hence, the scheduling for 4-node is perfect and is not impacted by traffic burstiness or any other aspects of traffic arrival pattern (as long as it is below capacity).

6.3.1.2 6-node scenario

In this scenario, 3 links are stacked adjacent to one another. While the middle link interferes with both the bordering links, the two bordering links can be scheduled independently. Simulations show that, depending on the scheduling algorithm, bursty traffic does affect the maximum carried load. In particular, the scheduling duration relative to the timescale of traffic burstiness (or arrival pattern) can result in imperfect scheduling. In some arrival patterns, traffic burstiness can lead to higher carried load as well. For example, for the default scheduling algorithm used in USAP, the carried load for each of the 3 connections for varying offered load is shown in Tab. 6.2.

Offered Load (in Mbps)	**Carried Load (in Mbps)**				
	Smooth				Bursty
	Conn 1	Conn 2	Conn 3	Overall	Overall
1.6	1.6	1.6	1.6	4.8	4.8
1.7	1.7	1.65	1.7	5.05	5.1
1.8	1.8	1.45	1.8	5.05	5.4
1.9	1.9	1.25	1.9	5.05	5.7
2	2	1.05	2	5.05	6

Table 6.2: Carried Load for USAP MAC 6-Node Scenario for Different Offered Load

6.3.2 Greedy Backpressure-based MAC

There is no impact of bursty traffic on maximum carried load. This is expected as the algorithm is throughput optimal. In particular, the arrival pattern does not impact the scheduling performance / throughput if the load lies within the capacity region. In other words, if the input load is below the capacity (maximum carried load), the algorithm will stabilize the queue (and hence serve the throughput) irrespective of how bursty the traffic arrival pattern is.

6.4 Summary

By and large, bursty traffic does impact the MAC performance in some scenarios. We gave a On/Off model and validated the results for simple 4-node topologies. In particular, we identified traffic load regions where traffic burstiness can impact the MAC performance using both our model as well as simulations. We tried to characterize the 'capacity' region based on traffic being smooth or bursty. However, in most multi-hop topologies, the impact seems to be balanced out. Bursty traffic in one link may enhance carried load in one neighborhood (or clique); but the increased load in that neighborhood can decrease the carried load in the adjacent neighborhood (or clique). Hence, when characterizing the maximum carried load, or when building approximate performance model that are general, we should take care to look the traffic burstiness. But overall carried load may not be much different in most multi-hop topologies.

Chapter 7: Modeling Heavy-tails in Traffic Sources for Network Performance Evaluation

In this chapter, we shift our focus from the protocols to the traffic sources themselves. While it is important to understand the impact of protocols to characterize the network traffic, it is also important to capture all the features of the input traffic from applications for accurate analysis. Hence, we focus on statistical modeling of the traffic generators, keeping our focus on heavy-tailed distributions.

The negative impact of heavy tails in work loads on the performance of systems is well-known in the queuing literature. Indeed, many new scheduling strategies came to be invented primarily to avoid these bad effects of very large work loads (even if they be infrequent and from a small set of customers) for systems with schedules such as the First-in-First-Out discipline. Concern about heavy tails nevertheless holds even in the context of modern day systems like high speed and wireless networks. Indeed, the increasing presence of bandwidth intensive video and streaming audio has heightened the concern particularly in wireless networks as evidenced for example by the AT&T experience soon after the introduction of the iPhone.

An early work drawing attention to the presence of heavy tails in Internet

file sizes is that of Crovella (1996) [12]. We use Crovella's data set and model the distribution as a LogPH distribution and also in terms of classical models like Pareto, Weibull, LogNormal and Log-t. The LogPH distribution was proposed by Ramaswami who identified it to have a power law tail and dense (in the weak convergence metric) in the class of all distributions on $[1, \infty)$. A LogPH random variable Y is a a random variable that can be written as $Y = e^X$ where X is a phase type random variable as defined by Neuts [31, 54]; see also [55] for a discussion on phase type (PH) distributions. The first formal reference to LogPH was made in the paper by Ghosh, et al. [32] on modeling traffic to a public Wi-Fi network. A detailed mathematical treatment of the LogPH class of distributions has been given in Ahn et al. [56]. That work of Ahn et al. [56], demonstrates the power of the LogPH class to model heavy tailed distributions in their entire range in the context of several financial examples. In this chapter, we demonstrates its power in the context of network performance modeling.

There are many attempts in the literature (see [30, 56, 57]) in queuing, performance analysis, risk theory, and finance to model heavy tailed distributions. A key idea is to use some distribution (like Pareto or Weibull) with a known heavy tail to model the tail and then a mixture to obtain a fit across the entire range. Unfortunately, these attempts have not resulted in a single class of models that can be used dependably in a large number of contexts, and, furthermore, many aspects of the fitting methodology appear to be ad-hoc. Also, it would appear prudent to adopt more stringent standards in assessing the statistical quality of the fits in terms of a test data set that is separate from the training data set used for fitting a model.

Also, it is desirable not only to consider the fitted random variable, but assess the quality of the fit in terms of its ability to predict performance of systems in which the models are used. Judged in this context, this paper may be found interesting and useful by many.

This chapter is organized as follows. In Section 7.1, we provide a quick discussion of various classical models used in the context of heavy tails and also of the LogPH distribution. A brief discussion of a method based on the EM algorithm to fit LogPH distributions is given there. In Section 7.2, we fit LogPH distributions to two data sets, the Crovella Internet file size data set [12] and a very recent (2012) data set on file sizes downloaded by mobile phone users in a cellular mobility network. We also provide a comparison of the LogPH fit to various other models like the Pareto, Weibull, LogNormal and Log-t. In addition to visual comparisons of the empirical and fitted distributions we also provide some quantitative measures that aid such comparison. We wish to note that in the comparisons related to wireless mobility, since several data sets were available we have taken a more stringent approach of having a "training set" based on which the model fits were made and a separate "test set" for comparing the models. Unfortunately, we had only one data set in the case of the Crovella data; we did comparisons with bootstrapped samples generated from this data set and found the fit to hold good for them as well. This step was undertaken primarily to make sure that we did not run the risk of overfitting. In Section 7.3, we take the Crovella data set and make a trace driven simulation of a bottleneck Internet link and compare the performance results (queue lengths, throughput) against simulations run with fitted models using LogPH, Pareto, Weibull and Log-t.

Our results show that the LogPH gives more accurate results and thereby increase our confidence in the LogPH model class. More importantly, the simulations show the need to model the entire distribution for better performance analysis.

7.1 Background

7.1.1 Phase-Type (PH) Distribution

A PH distribution is defined as the distribution of the time until absorption of a Markov chain with an absorbing state. This general class was introduced by M.F. Neuts [31, 54]. To be specific, consider a Markov chain with states $0, 1 \ldots, n$, initial probability vector $(0, \boldsymbol{\tau})$ and infinitesimal generator Q. The row vector $\boldsymbol{\tau}$ is of size n and satisfies $\boldsymbol{\tau}\mathbf{1} = 1$, where $\mathbf{1}$ is a column vector of 1's. Assuming state 0 is an absorbing state, Q can be denoted as

$$Q = \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{t} & T \end{bmatrix}$$

where $\mathbf{t}$ is a column vector of size n and T is a $n \times n$ nonsingular matrix satisfying, $T(i,i) < 0$, $T(i,j) \geq 0$ for $i \neq j$, and $T\mathbf{1} + \mathbf{t} = \mathbf{0}$. Thus, the Markov chain is completely characterized by parameters $\boldsymbol{\tau}$ and T. The random variable X describing the time until absorption of the Markov chain into state 0 is called a PH random variable, denoted $PH(\boldsymbol{\tau}, T)$. The number of non-absorbing states (n) is called the order of the PH random variable. The distribution and density functions of the PH

random variable defined above are given by

$$F(x) = 1 - \boldsymbol{\tau} exp(Tx)\mathbf{1},\ for\ x \geq 0, \tag{7.1}$$

$$f(x) = \boldsymbol{\tau} exp(Tx)\mathbf{t},\ for\ x > 0. \tag{7.2}$$

PH-distributions are known to be dense in the class of all distributions on $[0, \infty)$. That is, they can approximate any distribution arbitrarily closely. Furthermore, they have many interesting closure properties and are highly tractable due to the connection to a Markov chain which makes conditioning arguments easy. For these reasons, they have attracted much attention in applied probability. A property of the PH distribution is that its tail is asymptotically exponential; more specifically, for the distribution above, $P(X > x) \approx Ke^{-\eta x}$ for large x, where $-\eta < 0$ is the eigenvalue of T closest to zero.

7.1.2 LogPH Distribution

The LogPH distribution, denoted by $LogPH(\boldsymbol{\tau}, T)$, is defined as the distribution of the random variable Y that can be written as $Y = exp(X)$ where X has a PH distribution with $(\boldsymbol{\tau}, T)$. The LogPH random variable Y has its distribution function and density function as

$$F_Y(y) = 1 - \tau e^{Tlogy}\mathbf{1},\ y \geq 1$$

and

$$f_Y(y) = \frac{1}{y}\,\tau e^{Tlogy}t,\ y \geq 1,\ t = -T\mathbf{1}.$$

From the exponential decay of the tail of the PH-distribution, it easily follows that the LogPH random variable has a power law tail. Specifically, for large y, $P(Y > y) \approx K/y^{\eta}$, where η is as defined earlier. Also, from the fact that PH type distributions are dense on $[0, \infty)$, it follows by standard continuity theorems governing weak convergence (see Whitt [58]) that LogPH distributions are dense in the set of all distributions defined on $[1, \infty)$. These properties make LogPH an attractive candidate class for modeling heavy tailed random variables. In this context, we wish to note that the restriction of its range to $[1, \infty)$ is not particularly limiting for two reasons: (a) in many cases, one could rescale the data and fit LogPH to the scaled data set; or (b) one can use a similar construction based on the Bilateral PH random variables which generalize the PH distribution to the entire real line; see Ahn & Ramaswami [59].

7.1.3 Some Classical Heavy-Tailed Distributions

Traditionally, modeling of heavy-tailed random variables has been focused mainly on the tail of the distribution. Some of the commonly used distributions are Pareto, Weibull and lognormal. Pareto, has the following tail distribution -

$$Pr[Z > z] = \left(\frac{b}{z}\right)^{a}, \qquad z \geq b \tag{7.3}$$

where a and b are respectively the shape and the scale parameters. Various enhancements of the Pareto distribution have also been used to match the mean, to select the cut-off at which the asymptotic power law takes over. A Pareto random variable Y can be realized as $exp(X)$ where X is an exponential random variable;

in this sense one may consider the LogPH class as a natural generalization of the Pareto distribution since the exponential distribution is the most trivial example of a PH-distribution.

The Weibull distribution does not have a power law in the tail distribution, but still the tail decays more slowly than the exponential. Denoting by a and b as the shape and scale of the distribution,

$$Pr[Z > z] = exp\{-(z/b)^a\}, \quad z \geq 0,\ a < 1. \tag{7.4}$$

Lognormal distribution is modeled as a distribution which is Normal in the log scale. Specifically, Z has a lognormal distribution if we can write

$$Z = e^X,\ where\ X \sim \mathcal{N}(\mu, \sigma^2) \tag{7.5}$$

The normal distribution has a fast decaying tail $e^{-x^2/2}$, and this has led some researchers to use the t-distribution in place of the Normal and define a Log-t distribution as a model for heavy tailed distributions. Like the normal, the t distribution also is symmetric about the origin and that could limit some of its applicability.

While the literature abounds in many applied examples where the above distributions and mixtures involving them have been used successfully for specific situations, there are some basic challenges in their use. These distributions do not form a dense class that provides a guarantee that one may effect a fit from any one of the members to a desired accuracy. Also, often one is forced to make a trade off between matching the tail and matching the head of the distribution and to come up with ad-hoc procedures for fitting a mixture that attempts to give a good model in the entire range of the data. We refer to [56], for a discussion of the issues in

light of a famous data set – the Danish fire insurance data – that has been used as an important test data set in the statistical literature.

7.1.4 Fitting a LogPH distribution

A LogPH distribution is fitted to the data by fitting a PH-distribution to the logarithms of the data values (to the base e). If only a very small fraction of data values exist that are less than 1, we may discard them; or alternately, we may rescale the data by dividing all the elements by the minimum value so that all logarithms are positive and then fit a LogPH to these log values.

The standard approach to fitting a phase type distribution is to use the EM algorithm whose details are provided in a paper by Asmussen et al [60]. The EM algorithm is based on the following observations. Suppose the true distribution is a phase type distribution of order n. If one knew the number of visits to each of the n transient states in the Markov chain and the amount of times spent in each of them before the Markov chain gets absorbed, these values together would constitute a set of sufficient statistics for the unknown parameters of the Markov chain. Now the EM algorithm starts with a trial phase type distribution of order n and iterates on the following two steps. (i) the E-step: consider the number and duration of visits to the transient states as missing values and replace them with their conditional expectation evaluated with respect to the current estimate of the parameters; (ii) the M-step: now considering as though we have a complete sample on the sufficient statistics, maximize the likelihood function to obtain an improved estimate of the

Markov chain parameters. The general theory of EM guarantees convergence to (a local) maximum of the likelihood function.

7.2 Data Sets and Fitted Models

This paper deals with two data sets. The first is the well known World Wide Web file size traces collected by Crovella in 1995 commonly used by researchers to model heavy tailed data. We will demonstrate that LogPH provides a much better fit for the entire data range as compared with previously used models, particularly in its ability to predict network performance more accurately as evaluated in the context of a bottleneck Internet link. Secondly, we will also use a very recent data set of file sizes from the *mobile Web*. We show once again that LogPH provides a good fit to the entire range of the mobile data set. Our interest in the second data set is due to our current focus on wireless and mobile networks and the fact that the Crovella data is now quite dated and pre-dates some major new bandwidth intensive applications like video and streaming audio that have become much more popular. A priori, it was not even clear if there would be any need for heavy tailed models in the wireless context, but as our results show heavy tails seem to pervade the wireless and mobile networks as well. The mobile Web data set is obtained from a national cellular service provider. The data set is collected over a period of 24 hours during 2012 inside a geographic area (spanning hundreds of cell sectors within a large metropolitan area).

In the case of the mobility data, the presence of many different data sets

collected over different time intervals gave us the unique opportunity to fit the model with one data set but test it with another one not used in testing. This is obviously a more stringent comparison in tune with practices in the area of machine learning and modern data analysis using separate training and test data sets. For the Crovella data set itself, we used a simulation where one of the inputs was driven by the empirical distribution of the Crovella data; that gave us the opportunity to treat the simulated values as a bootstrapped sample and to assess the quality of the fit against that. Unfortunately, Crovella does not provide a multitude of data sets. We did run our algorithms for fitting on a random subset of the Crovella data using the rest for testing, and the results were entirely satisfactory and as good as in the mobility data sets.

7.2.1 Crovella's WWW data set

Crovella's dataset [12] contains the file sizes for a large number of HTTP requests. There are around 130,000 data points for file sizes. Fig. 7.1 shows the empirical density of the file sizes on a natural log scale. Fig. 7.2 shows the tail distribution on the log-log scale clearly indicating the presence of an asymptotic linear decay which for the files sizes entails a power law decay of the form $O(1/x^{\eta})$. A regression fit for the tail portion of the complementary distribution on the log log scale gave an estimate $\eta = 1.15$; we have shown the regression line for the tail along with the plot of the complementary distribution.

Now, we describe some key steps taken by us in fitting a LogPH to the file size

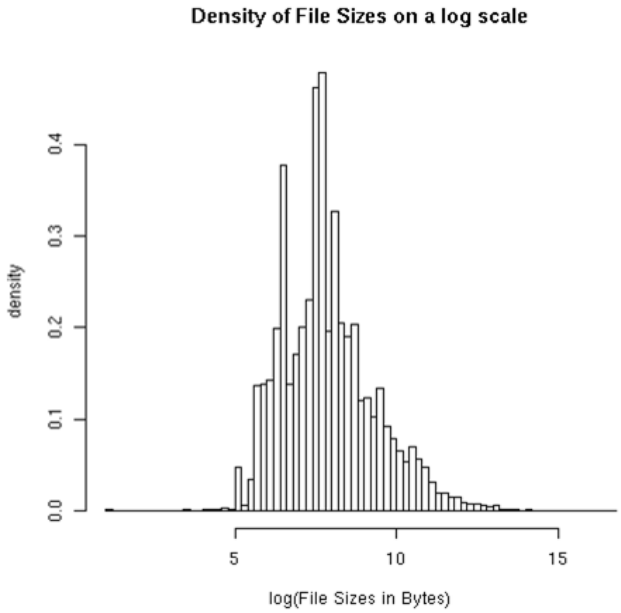


Figure 7.1: Crovella data - density

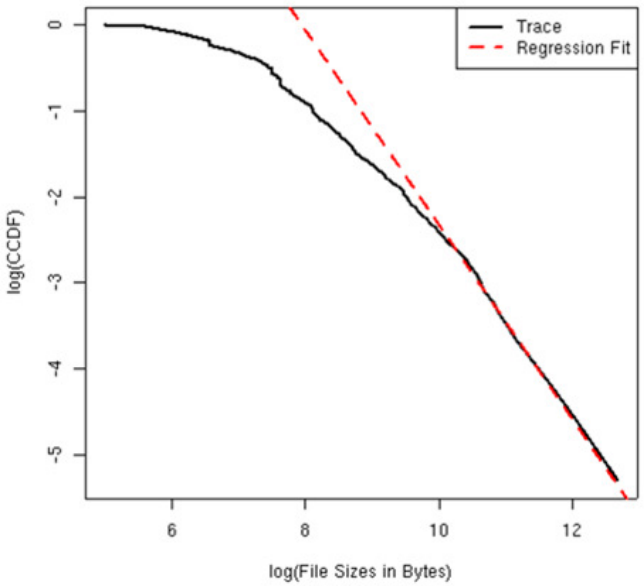


Figure 7.2: Crovella data - CCDF and regression fit

data. Recall that the log of the file size is to be modeled as a PH random variable. A preliminary analysis of the data set showed that an overwhelming majority (98.7%) of the file sizes were over 200 bytes. We took the set of values above 200, divided them by 200, took the logarithm and fitted a phase type distribution to the resulting set of values. This rescaling is equivalent in the log scale to shifting the density (Fig. 7.1) to the left so as to start from zero (see the density in Fig. 7.3(a)). The original distribution of the data was then approximated by the mixture of a point mass at 200 with probability 0.013 and the distribution of a random variable that is 200 times the fitted Log-PH r.v., the latter with weight 0.987. The motivation for not expending much effort on the lower tail below 200 (which is possible through the use of a log-BPH distribution) was that these observations constituted a small percentage and from the perspective of the performance comparisons in a queuing model would not make much of a difference. Indeed, we followed a similar procedure for fitting all other distributions we tried. To be mathematically precise, denoting by X the file size, we model $Y = max(0, X/200)$ as a LogPH random variable, and sweep the mass to the left of 200 bytes to a point mass at 200.

The EM algorithm for fitting a PH distribution was tried with different number of phases in the range 3 to 7. We stopped at a value of $n = 5$ at which point the improvement in the maximum likelihood value appeared negligible. A PH fit of order 5 was found to provide a good match as illustrated in Table 1 by means of

log-likelihoods. The PH parameter estimates are $\boldsymbol{\tau} = (0, 0, 0, 0, 1)$ and

$$T = \begin{bmatrix} -1.31 & 0 & 0 & 1.31 & 0 \\ 1.31 & -1.31 & 0 & 0 & 0 \\ 0 & 1.79 & -2.11 & 0.32 & 0 \\ 0 & 0 & 0 & -2.11 & 0 \\ 0 & 0 & 2.0 & 0.11 & -2.11 \end{bmatrix}$$

The fitted PH-density function and the cumulative distribution function are shown in Figures 7.1 and 7.2 respectively along with their corresponding empirical figures based on data.

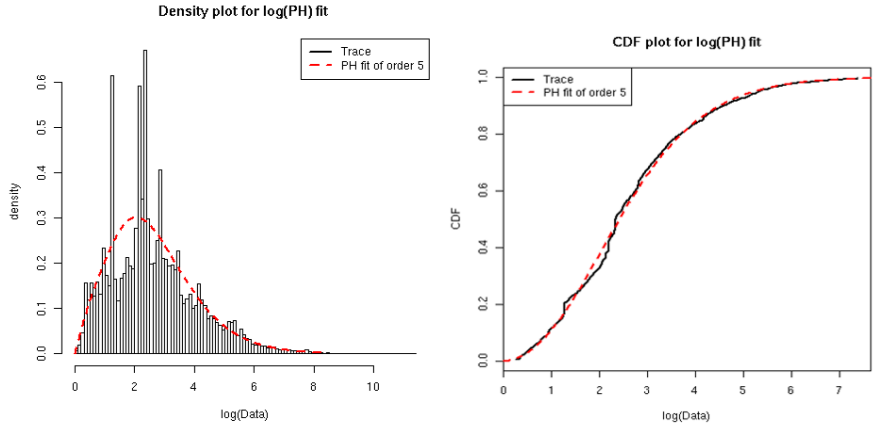
We wish to make a few comments with respect to this fit:

(a) At this time, we are not aware of any methods that will automatically choose the size of a PH-distribution for the fit. Hence we have had to try out a few values and stop in an ad-hoc manner. This is an area where further research is needed.

(b) While the fitted distribution of order 5 appears to do a fairly good job overall (and particularly from the system performance perspectives to be discussed later; see Figures 7.11 and 7.12), it does miss out on a few pronounced peaks in the empirical density. To capture these peaks, one would need a very high order of a phase type distribution and that may run the risk of overfitting. In many applications, such peaks occur due to peculiarities in the application being modeled. For instance, in the wireless data, we saw a pronounced peak around 1500 bytes coinciding with the size of an Ethernet packet size. Perhaps some pre-processing of the data to remove these peaks and model them as point distributions may be warranted in some instances. This is also an area for further examination. In the present case,

with the probability mass near the peak being small, it did not make much difference in our network performance evaluation,

(c) Note that the tail decay parameter for the fitted PH distribution is 1.31 (by computing the eigenvalue of T closest to 0) and is reasonably close to the estimate 1.15 based on a crude regression model for the tail. The tail parameter estimate based on regression is highly sensitive to what is considered the tail, and therefore the value 1.15 should be taken only as a very crude estimate. We need to develop some sound procedures for obtaining reliable tail estimates and to compare them, as well as methods to modify the EM algorithm so that the fitted LogPH distribution will have a given tail decay. Once again, these are topics for further research and beyond the scope of this work.



(a) Density plot and its LogPH fit (b) CDF and its LogPH fit

Figure 7.3: Fitting PH to log(file size in Bytes - 200), (Crovella data set)

7.2.2 Mobile Web data set

The mobile Web data set is obtained from a national cellular service provider. The training and test data sets comprised of the sizes of approximately 11.3 million files that were downloaded by all smart phones over a certain period of 24 hours during 2012 inside a metropolitan area with hundreds of cell sectors. Figure 7.4 shows the empirical histogram and various LogPH model order fits to it. We stopped with a ninth order fit since we found that the gain in the likelihood function in increasing the order did not justify the risk of overfitting. While a crude regression model provided a tail decay parameter estimate of 1.14, the resulting value from our ninth order LogPH fit turned out to be 0.91.

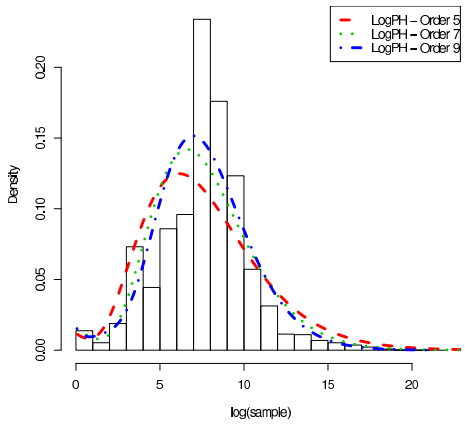
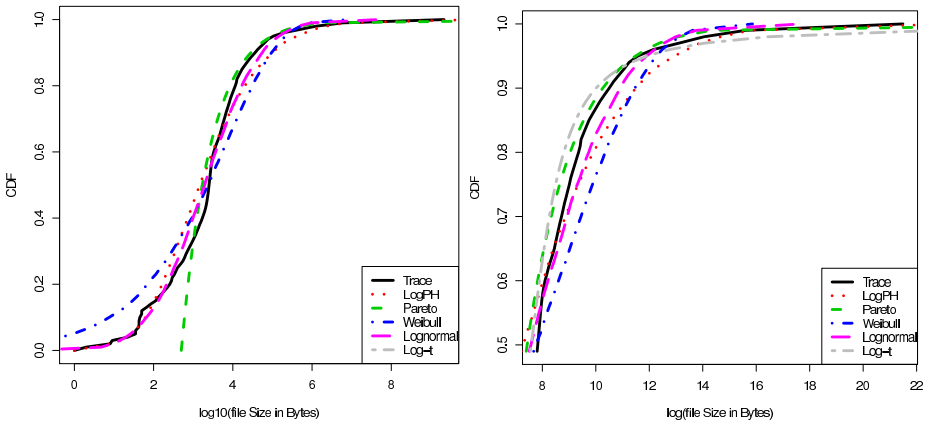
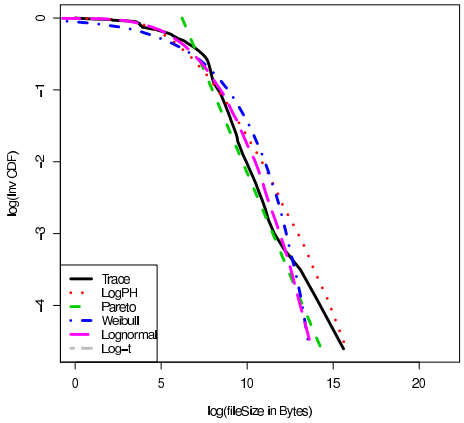


Figure 7.4: Density plot and LogPH fit for different model orders of Mobile Web data set

Figures 7.5(a) 7.5(b) 7.5(c) shows the various fits to the Mobile Web trace. Note that LogPH provides a good fit across the entire distribution. Pareto and Weibull distributions, well known for modeling heavy tailed distributions perform



(a) CDF (b) CDF Tail (magnified)



(c) Complementary CDF

Figure 7.5: LogPH fit (9th order) compared to other classical heavy tail distribution fits for Mobile Web data set

poorly at the tails as well as the head. Using the EMpht code, we obtained a 9th order fit.

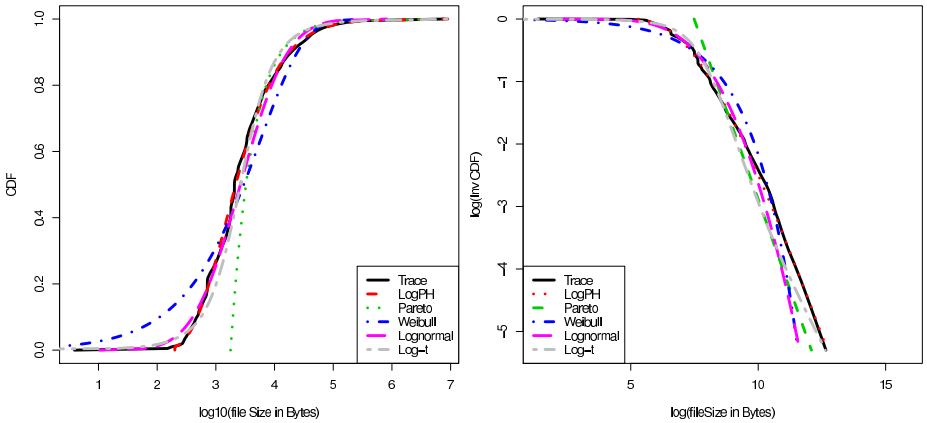
7.2.3 Comparison of Models

For both data sets under consideration, we also fitted the Pareto, Weibull, lognormal and log-t models to the data. A similar preprocessing of the data in terms of scaling etc., was done for the Crovella data as done for the LogPH fit to make the comparisons meaningful. In this section, we make some comparisons of the fits based on some visual plots and the likelihood ratio values.

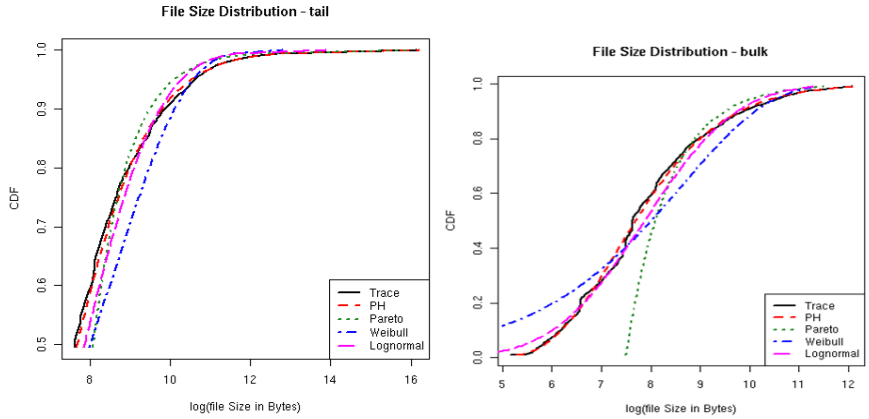
7.2.3.1 The Crovella Data

The parameters of the fitted classical distributions are as follows. The Weibull distribution (Eq. 7.4) has shape $a = 0.57$ and scale $b = 5680$. The lognormal distribution (Eq. 7.5), has $\mu = 7.87$ and $\sigma = 1.46$. The Log-t distribution has $\mu = 7.87$ and $\nu = 3.66$. Using EM algorithm directly for Pareto distribution gives a poor fit. Hence, for the Pareto distribution (Eq. 7.3), we use the shape $a = 1.15$ from the power-law in the tail (Fig. 7.2) and the scale parameter $b = 1788$ so that the mean for the Pareto distribution and empirical file size mean are equal (13710 bytes).

Fig. 7.6 shows the distribution plots for various fits along with the empirical file size distribution from the Crovella's trace. Fig. 7.6(a) shows CDF over the entire support (with x-axis on the log scale). For the CDF, Log-t and LogPH appear to



(a) CDF (b) CCDF



(c) CDF - focusing on the tail (d) CDF - focusing on the bulk

Figure 7.6: Comparing different fits (Crovella WWW trace)

match the entire distribution well. However, a closer examination of the tail through a log-log plot given in Fig. 7.6(b) shows that Log-t does not capture the tail as well as the Log-PH. The log-log plot for the complementary CDF also shows that Weibull and lognormal do poorly in the tail. Finally, although Pareto appears to give a good match for the decay rate of the tail, nevertheless, does not match the bulk of the distribution. Overall, we can conclude that LogPH does indeed provide the best fit.

We also compute the log-likelihood values for the various fits. Table 7.1 gives

Model	Uncond. Crovella	Cond. Crovella	Uncond. Mobile Web	Cond. Mobile Web
Weibull	-9.874	-10.176	-10.148	-10.816
Pareto	$-\infty$	-10.034	$-\infty$	-10.809
Lognormal	-9.670	-10.033	-10.801	-10.924
Log-t	-9.683	-10.017	-10.059	-10.665
LogPH	-9.554	-9.980	-10.057	-10.682

Table 7.1: Log-Likelihood Values(Crovella and Mobile Web)

the log-likelihood values. The unconditional log-likelihood values correspond to the case when all the observations (from the file size trace) are used. Since Pareto distribution had zero probability of taking value less than its scale parameter ($b = 1788$), the log-likelihood turns out to be $-\infty$. Hence, we also compare the conditional log-likelihood values conditioned on observations above the Pareto scale parameter. The log-likelihood values (both unconditioned and conditioned) appear to attest to an overall superior performance by the LogPH fit.

We also plot the percentage error between the modeled and true file sizes, (Estimate - True Value)/(True value) for various percentiles of the file sizes in Figure 7.7. We note that LogPH provides the lowest percentage error compared to other fits for the Crovella data set. The median (50th percentile) error is 5.5% for

LogPH compared to 57.9%, 44.5%, 26.8%, 26.7% for Pareto, Weibull, Lognormal and Log-t respectively.

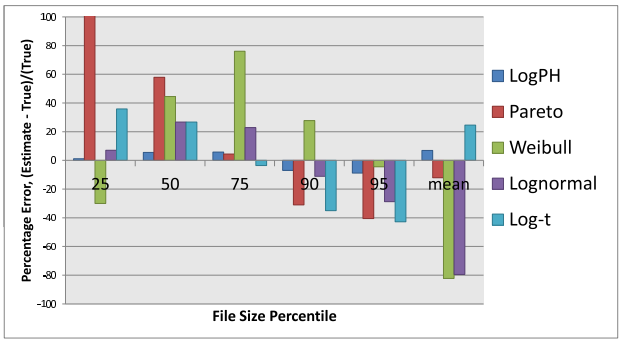
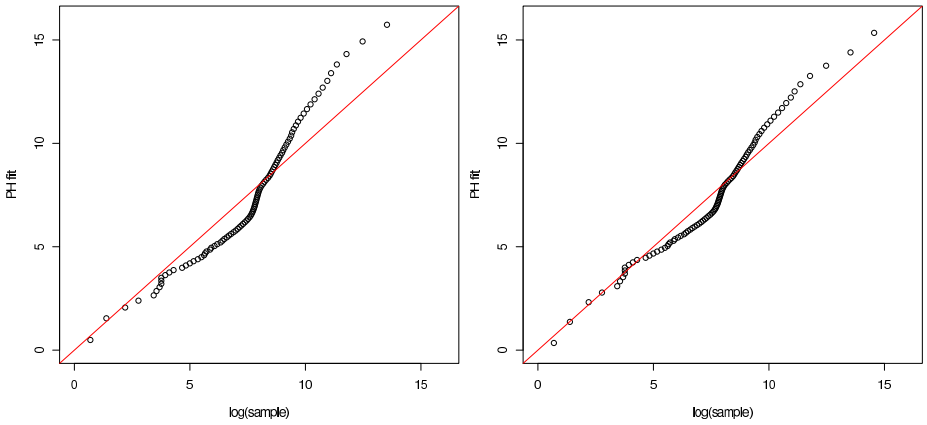


Figure 7.7: Percentage error between modeled and true file sizes, varying percentiles

7.2.3.2 Mobile Web Data

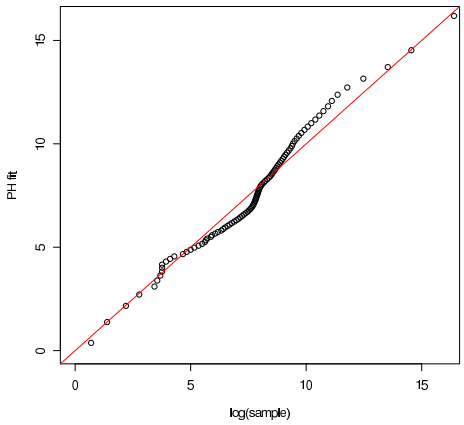
Figures 7.8(a), 7.8(b), 7.8(c) show a comparison of the fitted model quantiles with 5, 7 and 9 phases against the corresponding quantiles of the training data set. The improvement as the order of the phase type distribution is increased as well as the reduction in the marginal improvement effected by increasing the number of phases are both clear from the sequence of graphs shown there. As noted earlier, we stopped with an order 9 phase type fit to the log data.

In the case of the mobile data, since we had data sets from many different cell towers, we could compare the model predictions with a test data set that is different than the training set used to fit the model. The training data set and the validation data set were thus from two different sets of cell towers (i.e. spatially separated, and different users requesting mobile content). Figure 7.9 which compares the fitted model quantiles with the quantiles of the test data set shows an excellent match.



(a) LogPH model order 5

(b) LogPH model order 7



(c) LogPH model order 9

Figure 7.8: Mobile web data

7.3 Impact on Network Performance

Our interest in LogPH as a model class is from its ability to model heavy tailed data in its entire range. But there is a fundamental question as to if it even matters whether the entire distribution is matched or not. In particular, the focus of this work is on performance modeling and analysis; and here, we wish to find whether accurate modeling for heavy-tailed traffic sources leads to better performance pre-

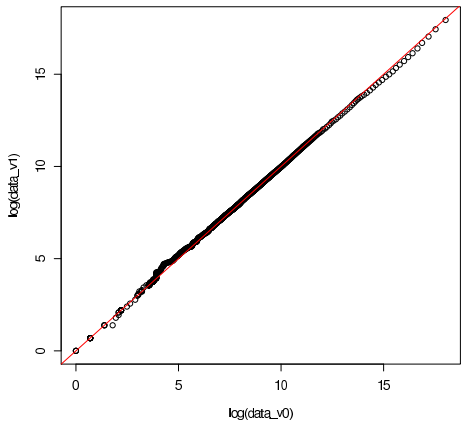


Figure 7.9: Test data quantiles (y-axis) vs modeled data quantiles (x-axis), Mobile Web data set

diction. The goal of this section is to examine this issue in the context of the data and the various models fitted to it. We do this with some NS-2 simulations of a bottleneck link implementing the TCP-Reno protocol. The simulation is first run with a stream generated by sampling the Crovella data trace and then its results (queue lengths and throughput of the bottleneck link) are compared to simulations run with streams generated by various models fitted to the Crovella data.

7.3.1 The Simulation Experiment

We used the NS-2 network simulator [61] to perform an extensive set of simulations. We set up a simple topology of a homogeneous set of clients and a server as shown in Fig. 7.10 connected by one intermediate router. In this configuration, there are N clients each of which is connected to the router using a 10 Mbps link with a 1 msec link latency. The link from the server to the router is 1 Mbps link with a 30 msec latency and depicts a potential bottleneck condition. Even though we use

wireline networks, we expect similar observations for wireless multi-hop networks.

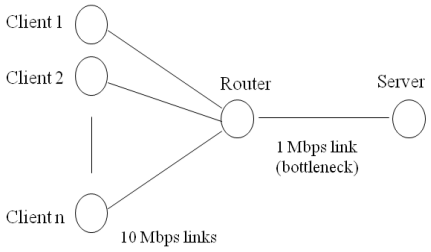


Figure 7.10: Network Scenario

We use two sets of client-server interaction. In the first set, a client can have at most one pending request at any given time. Thus in this set, a client generates a request and upon receiving a download from the server, it will generate the next request after a random amount of idle time. In the second set, a client can have multiple requests pending at the server. Here, each client generates a sequence of requests with a random inter-arrival time distribution without regard to what happened to previous requests. We call the first set as the *finite sources* scenario and the second set as the *infinite sources* scenario.

The connections use the default TCP-Reno protocol. The request size is fixed to a value of 40 Bytes, and the inter-request time is set to be an exponential random variable with a mean of 10 sec. For each request, the server replies with a file size drawn from an assigned file size distribution. The server think time for processing the request is set to (1 ± 0.1) msec. The response file is divided into packets of size 1040 Bytes and passed to the transport layer, which then passes the packets

onto the link from the server to the router, depending upon the congestion on the bottleneck link (from server to router). We vary the number of clients to simulate different load conditions. The buffer sizes for all links are set to a very high value so as to mimic an infinite buffer scenario.

For each configuration – whether finite source or infinite source – we first determine the number of clients corresponding to an assigned load level. Then for each scenario we run 6 different simulations each using a different file size distribution at the server. The 6 different distributions used are: Trace (Empirical distribution based on the Crovella data), LogPH, Pareto, Weibull, lognormal or Log-t, where the latter are the fitted models to the Crovella data. We run each simulation for a clock time of 3600 sec. We measure the queue length for the bottleneck link (from server to router) at the end of the simulation. We also measure the connection throughput (response file size/request completion duration) and take its average over all the connections in the simulation run. Depending on the load, there are about 10000–25000 connections per simulation run. We repeat this experiment 1000 times using different random seeds. Thus, we obtain a distribution for queue length (at T=3600 sec) and an average connection throughput. We then compare the results across different choices of the file size distribution.

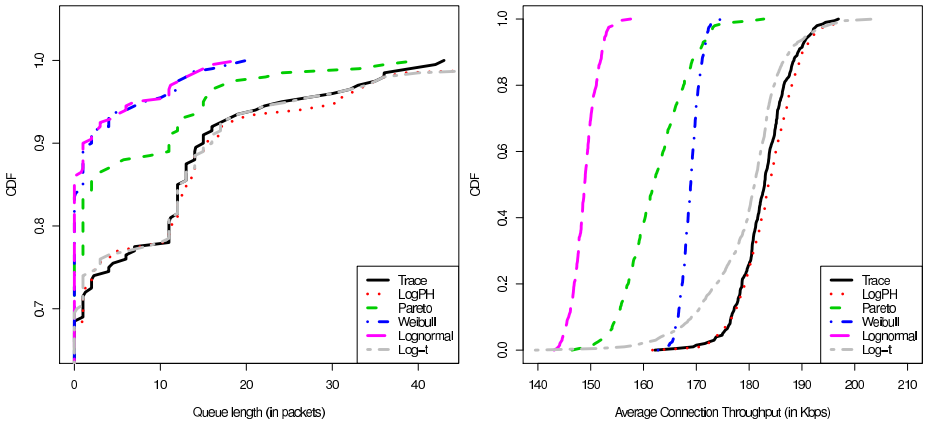
7.3.1.1 Results and Analysis

We run 4 sets of experiments. The first two sets use 30 and 75 clients in the *finite sources* configuration so as to have a link utilization of 40% and 80% on the

bottleneck link, respectively. The third and the fourth sets use 30 and 60 clients in the *infinite sources* configuration to again have a link utilization of 40% and 80% respectively. As mentioned in the previous section, we measure the queue length at epoch 3600 second, and average connection throughput over 1000 runs for file size distributions (at the server) from Crovella's Trace or one of the five modeled fits.

Fig. 7.11 shows the results for finite sources with 40% percent link utilization. The lognormal and Weibull fit show a very low queue length. This can be attributed to the lighter tail for lognormal and Weibull which results in lower load. Similarly, log-t overestimates the tail and results in higher queue occupancy. Pareto fit seems to predict the queue length slightly better, but still is not as close as the predictions from logPH. Looking at the average connection throughput, again LogPH predicts the throughput very well. Surprisingly, even though the queue lengths for Lognormal, Weibull and Pareto runs are low, the corresponding throughputs are also low. This may seem counter-intuitive. However, we believe that this may be because of the way TCP behaves. For smaller file sizes, the setup time and the slow-start phase reduce the throughput. For larger file sizes, the TCP connection has enough time to reach the congestion avoidance phase, thereby increasing the throughput.

In Fig. 7.12, we repeat the experiment with 80% link utilization. Compared to the previous set, the queue length has increased and the throughput has decreased. This is expected because of the increased load. Even in this experiment, Pareto, lognormal and Weibull seem to give poor estimates. Log-t gives good estimates in this scenario, however, they are not as accurate as LogPH. Also, unlike previous experiment, the low queue length for lognormal, Weibull and Pareto leads to higher

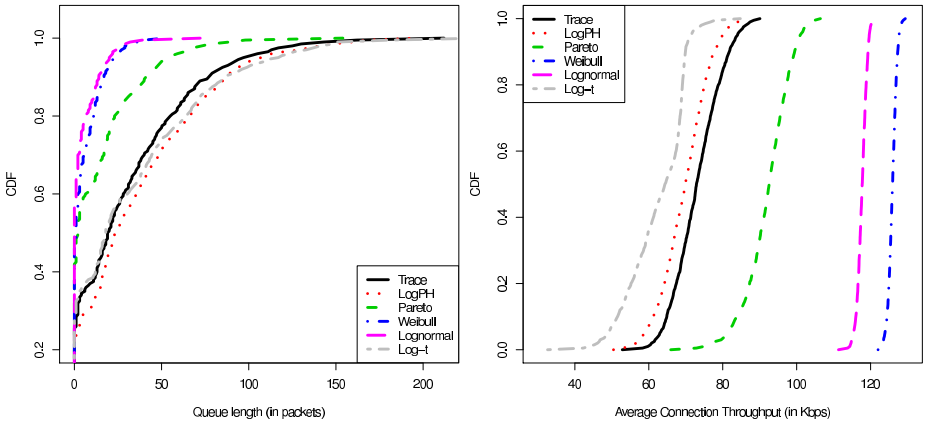


(a) Queue Length (Crovella data set) (b) Average Connection Throughput (Crovella data set)

Figure 7.11: Finite Sources with 40% link utilization

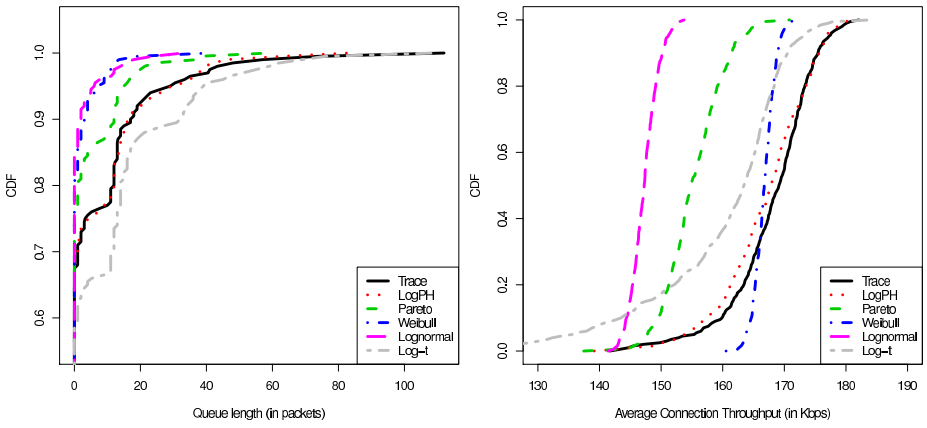
average connection throughput, when compared to the Trace. This can be attributed to the connection throughput being dominated by the queueing delays rather than the TCP dynamics.

Next we look at similar medium and high load scenarios with infinite sources. Fig. 7.13 shows the results for infinite sources with 40% link utilization. Once again, Lognormal and Weibull give inaccurate estimates. Even though Log-t seems to give very accurate performance estimates in some cases, it does not do so in all the scenarios. This shows that the mismatch due to constrained symmetry (of t-distribution) leads to inaccurate results in some cases. However, LogPH estimates the queue-length and throughput very well in all the cases.



(a) Queue Length (Crovella data set) (b) Average Connection Throughput (Crovella data set)

Figure 7.12: Finite Sources with 80% link utilization



(a) Queue Length (Crovella data set) (b) Average Connection Throughput (Crovella data set)

Figure 7.13: Infinite Sources with 40% link utilization

7.3.1.2 A Modeling Issue

In the context of the discussion of the simulation results, it behooves us to raise an important issue we discovered that raises some interesting challenges for mathematical research. All models fitted to the data sets have an infinite support while the data sets themselves span only a finite interval. Thus, in our methodology while using the fitted model there is the inherent problem of extrapolating far beyond available data. This type of a problem has always existed when model fits are made, but it has some serious consequences when the model fitted is a heavy tailed one as we learned from our experiments. The use of an infinite support model often leads to extremely pessimistic predictions as compared with what would obtain with the sample data trace.

To make the point clearer, consider a FIFO queue and suppose we have a heavy tailed service time with a power law tail index $\eta = 0.5$; i.e. the tail is of the order $O(1/x^{0.5})$ say. For very large x, now the conditional probability $P(X > 2x|X > x) \approx 1/\sqrt{2} \approx 0.71$. What this result points to is an interesting fact, namely, that when we use an infinite support heavy tailed distribution, the resulting queuing model is potentially being subjected to much much larger excursions of the underlying queue length and waiting time processes when such excursions do occur. This affects the tails of various performance measures leading to an overall over-pessimistic assessment. Such effects are negligible when the underlying distributions have fast decaying tails.

Indeed, in the network simulations we described earlier, our use of the heavy

tailed models all resulted in extremely pessimistic results with none of them providing a good match to the trace driven simulations. Having guessed the source of the problem, we normalized the fitted distribution in the range $(0, 1.2M)$ where M is the maximum value seen in the data trace and used the resulting distributions with finite support in our simulations. The results reported above and the attending figures correspond to simulations with such finite support modifications and the results speak for themselves.

Obviously, our choice of the truncation index is quite arbitrary and we had no recourse to any theory to choose that value appropriately. This is a (difficult but much needed) area of research: what kind of steps should one take to ameliorate the problems arising from extrapolating far from the data through the use of infinite support heavy tailed models in queuing systems when in reality the modeled random variables are indeed bounded albeit at a high value?

7.4 Summary

We have demonstrated the power of the LogPH class of models to model heavy tailed network data and provided a set of comparisons with other approaches that provide a compelling argument for its use. More importantly, we have identified an important set of research questions for examination whose resolution will enhance our ability to model heavy tailed distributions and thereby assess more accurately the performance of systems that suffer the phenomena of heavy tails.

Chapter 8: Conclusions and Future Work

In this thesis, we improve a loss-network model for multi-hop wireless networks based on IEEE 802.11 MAC. Our model give good approximations for the network performance metrics like throughput, losses and delays, for arbitrary multi-hop topologies. We also modify the MAC model and integrate it with a model for OLSR protocol. Unlike any work before, the combined model captures cross-layer interaction. In particular, depending upon the routes chosen by OLSR model, the traffic on those paths and resulting congestion in the MAC layer is higher, possibly leading to lower throughput. Similarly, given two choices of path, OLSR model chooses the path with less congestion with higher probability, since the congested links may not always be detected due to MAC losses.

We look at presence of HT and LRD in wireless traffic, which can impact the performance. We find that if the input traffic has Heavy-tails, then the TCP traffic is indeed LRD as expected. The UDP traffic is not LRD in most cases as the Heavy-tails in input traffic are curtailed by the finite MAC buffer. Contrary to popularly held view that network protocols cannot introduce traffic LRD, we investigate the role of wireless network protocol dynamics in shaping traffic at much larger scale. We investigate the role of more complex MAC protocols like CSMA/CA or 802.11

in shaping the delays and traffic burstiness. We find that the MAC service times can indeed have power-law body (from a few milliseconds upto a few seconds), though they are truncated at a few seconds. Furthermore, even though the service times have power-laws, the queuing delay does not. Thus, MAC protocols seem to have little role in introducing traffic LRD by themselves.

We show, for the first time, that some wireless routing protocols can introduce traffic LRD. This is contrary to the popular belief that network protocol dynamics cannot impact traffic LRD or traffic burstiness at large timescale. Cross-layer interaction of MAC and routing protocol lead to route oscillations due to incorrectly perceived link failures. These route oscillations behave like On/Off traffic models leading to traffic LRD. Apart from proving the possibility of traffic LRD purely due to wireless protocols, we also give a model to predict under what circumstances / network scenarios, we can expect the presence or absence of traffic LRD. In particular, given the network topology and offered load levels (/input traffic levels), our model correctly predicts the existence of traffic LRD. Furthermore, from our previously developed models for OLSR enhancements, we also suggest appropriate configuration to prevent route oscillations and traffic LRD

We show that the traffic can be bursty at large timescales, either because of inherently bursty traffic demands or because of route oscillations. The performance of MAC protocols depend on load in the neighborhood. For contention-based protocols like 802.11 MAC, the carried load (or achieved throughput) can potentially depend on traffic arrival pattern. We focus on one particular aspect - we investigate the impact of traffic burstiness on MAC layer throughput. We show that traffic, that

is bursty even at the time scale of a few seconds, can lead to noticeable difference in MAC capacity. With bursty traffic, modeled as On/Off input traffic model on neighboring nodes / links, we find that the connection level carried load and total carried load can be slightly higher. We also give On/Off models to explain such behavior.

Lastly, we discuss the importance of modeling the input / source traffic correctly. We show that using exponential or heavy-tailed distributions may not sufficiently capture their impact on network performance. We use a new family of distribution - logPH which are capable of modeling the entire distribution accurately. Furthermore, we show that these improved models can more closely predict the network performance which are missed by focusing on the tail or head of the distributions.

8.1 Future Work

More investigation needs to be made to understand the role of MAC and routing protocols in shaping traffic characteristics. We should develop analytical models for 802.11 MAC retransmissions, similar to the work for ALOHA in [28]. Secondly, given the finite power laws in MAC service times, we should analyze the role of protocol parameters on the distribution. Moreover, we should study more extensively the simulation results or real measurements making use of important additional features of 802.11n/ac etc. New advancement try to reduce the MAC service time and support much higher capacity. At the same time, a lot of additional

complexities might introduce additional dynamics.

For the routing protocol dynamics, we need to work on establishing the true distribution of route oscillations. In particular, we should strive to verify if they are true heavy-tails and the traffic burstiness is true LRD. Even if not, we should find an appropriate model to give the range of traffic scaling. We should improve the On/Off model for route oscillations and given the on/off distributions, give numerical models (if not analytical) to explain the observed range of traffic scaling. We also can study more extensively the role of more complex topologies. In particular, for multi-hop topologies, our model simply predicts the presence or absence of 'some' route oscillations. We slightly better models, it should be certainly possible to identify specific paths which suffer route oscillations. With slightly more efforts, we should strive to model the on/off durations for the same.

Secondly, we demonstrated that appropriate OLSR parameter tuning can avoid route oscillations. However, we did not fully evaluate the side effects of such parameter choices. For example, while we observed high D reduces route oscillations, this might result in stale routes not getting removed quickly in case of node mobility. In our previous work, we had looked at appropriate tuning for parameters D and U of choosing stable link, but also being able to quickly detect and remove lost links. We should combine the efforts to look at more comprehensive tradeoff analysis in terms of route oscillations and node mobility.

We demonstrated the impact of bursty traffic on MAC performance. We need to analyze more complex topologies to see if the impact is diminished or negligible as observed from preliminary simulations. If we find more evidence of performance

discrepancies, and if those discrepancies can occur fairly regularly, then we should improve our MAC model based on average load levels to also account for bursty traffic or route oscillations. Even in the absence of significant differences in bursty or smooth traffic, we can get good insights into the working of the MAC layer and the dependence between the MAC performance of neighboring nodes. Hopefully this helps us in improving the accuracy of the MAC models even further, even for smooth traffic scenarios.

Bibliography

[1] G. Bianchi. Performance Analysis of the IEEE802.11 Distributed Coodination Function. *IEEE Journal on Selected Areas in Communications*, 18, March 2000.

[2] A. Kumar, E. Altman, D. Miorandi, and M. Goyal. New Insights from a Fixed Point Analysis os Single Cell IEEE 802.11 WLANs. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2005.

[3] A. Kumar, E. Altman, D. Miorandi, and M. Goyal. New insights from a fixed-point analysis of single cell ieee 802.11 wlans. *IEEE/ACM Trans. Netw.*, 15(3):588–601, 2007.

[4] M. Hira, F. Tobagi, and K. Medepalli. Throughput Analysis of a Path in an IEEE 802.11 Multihop Wireless Netwoks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Hong Kong, China, March 2007.

[5] Michele Garetto, Theodoros Salonidis, and Edward W. Knightly. Modeling per-flow throughput and capturing starvation in CSMA multi-hop wireless networks. *IEEE/ACM Trans. Netw.*, 16(4):864–877, 2008.

[6] J. S. Baras, V. Tabatabaee, G. Papageorgiou, and N. Rentz. Performance Metric Sensitivity Computation for Optimization and Trade-off Analysis in Wireless Networks. In *Proceedings of the IEEE Global Communications Conference*, 2008.

[7] Apoorva Jindal and Konstantinos Psounis. The achievable rate region of 802.11-scheduled multihop networks. *IEEE/ACM Trans. Netw.*, 17, 2009.

[8] J. S. Baras, V. Tabatabaee, P. Purkayastha, and K. Somasundaram. Component based performance modeling of the wireless routing protocols. In *IEEE ICC Ad Hoc and Sensor Networking Symposium.*, 2009.

[9] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 1994.

[10] Walter Willinger, Murad S. Taqqu, Robert Sherman, and Daniel V. Wilson. Self-similarity through high-variability: statistical analysis of ethernet lan traffic at the source level. *IEEE/ACM Transactions on Networking*, 1997.

[11] V. Paxson and S. Floyd. Wide area traffic: the failure of poisson modelling. *IEEE/ACM Transactions on Networking*, 1995.

[12] M. E. Crovella and A. Bestavros. Self-similarity in world wide web traffic: Evidence and possible causes. *Performance Evaluation Review*, 1996.

[13] Kihong Park, Gitae Kim, and Mark Crovella. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *Proceedings of the International Conference on Network Protocols (ICNP)*, 1996.

[14] Kihong Park. On the effect and control of self-similar network traffic: a simulation perspective. In *Conference on Winter Simulation*, 1997.

[15] Kihong Park and Walter Willinger, editors. *Self-Similar Network Traffic and Performance Evaluation*. John Wiley & Sons, 2000.

[16] O. I. Sheluhin, S. M. Smolskiy, and A. V. Osin, editors. *Self-Similar Processes in Telecommunications*. Wiley & Sons, 2007.

[17] A. Feldmann, A. C. Gilbert, and W. Willinger. Data networks as cascades: investigating the multifractal nature of internet wan traffic. In *Proceedings of the ACM SIGCOMM*, 1998.

[18] Yoshiaki Sumida, Hiroyuki Ohsaki, Masayuki Murata, and Hideo Miyahara. Effects of upper-layer protocols on the self-similarity of network traffic. *Electronics and Communications in Japan (Part II: Electronics)*, 2001.

[19] O. Tickoo and B. Sikdar. On the impact of ieee 802.11 mac on traffic characteristics. *Selected Areas in Communications, IEEE Journal on*, 2003.

[20] J. Yu and A. P. Petropulu. Is high speed wireless network traffic self-similar? *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2004.

[21] Shahram Teymori and Weihua Zhuang. Queue analysis and multiplexing of heavy-tailed traffic in wireless packet data networks. *Mobile Networks and Applications*, 2007.

[22] Mikael Gidlund and Nicolas Debernardi. Scheduling performance of heavy-tailed data traffic in wireless high-speed shared channels. In *WCNC'09: Wireless Communications & Networking Conference*, 2009.

[23] Qin Yu, Yuming Mao, Taijun Wang, and Fan Wu. Hurst parameter estimation and characteristics analysis of aggregate wireless lan traffic. In *International Conference on Communications, Circuits and Systems*, 2005.

[24] I.W.C. Lee and A.O. Fapojuwo. Estimating heavy-tails in long-range dependent wireless traffic. In *Vehicular Technology Conference, VTC*, 2005.

[25] D.P. Pezaros, M. Sifalakis, and D. Hutchison. On the long-range dependent behaviour of unidirectional packet delay of wireless traffic. In *Global Telecommunications Conference, GLOBECOM*, 2007.

[26] Adam Domański, Joanna Domańska, and Tadeusz Czachórski. The impact of self-similarity on traffic shaping in wireless lan. In *Proceedings of the 8th international conference, NEW2AN and 1st Russian Conference on Smart Spaces, ruSMART on Next Generation Teletraffic and Wired/Wireless Advanced Networking*, NEW2AN '08 / ruSMART '08, 2008.

[27] John S. Baras, Vahid Tabatabaee, and Kaustubh Jain. Component based modeling for cross-layer analysis of 802.11 mac and olsr routing protocols in ad-hoc networks. In *Proceedings of the 28th IEEE conference on Military communications*, MILCOM'09, pages 1240–1246, 2009.

[28] Predrag R. Jelenkovic and Jian Tan. Is ALOHA causing power law delays? In *International Teletraffic Conference*, 2007.

[29] Patrice Abry, Pierre Borgnat, Fabio Ricciato, Antoine Scherrer, and Darryl Veitch. Revisiting an old friend: on the observability of the relation between long range dependence and heavy tail. *Telecommunication Systems*, 2010.

[30] E. Chlebus and G. Divgi. A novel probability distribution for modeling internet traffic and its parameter estimation. In *IEEE Global Telecommunications Conference - GLOBECOM*, 2007.

[31] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*. Johns Hopkins University Press, 1981.

[32] Amitabha Ghosh, Rittwik Jana, V. Ramaswami, Jim Rowland, and N. K. Shankaranarayanan. Modeling and characterization of large-scale wi-fi traffic in public hot-spots. In *INFOCOM*, 2011.

[33] H. Martin Bücker, George F. Corliss, Paul D. Hovland, Uwe Naumann, and Boyana Norris, editors. *Automatic Differentiation: Applications, Theory, and Implementations*, volume 50 of *Lecture Notes in Computational Science and Engineering*. Springer, New York, NY, 2005.

[34] J. S. Baras, V. Tabatabaee, G. Papageorgiou, and N. Rentz. Modelling and Optimization for Multi-hop Wireless Networks Using Fixed Point and Automatic Differentiation. In *Int. Symp. on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, 2008.

[35] S. E. Dreyfus. An appraisal of some shortest path algorithms. *Operations Research*, 17(3):395–412, May 1969.

[36] Automatic differentiation by overloading in c++. URL.

[37] OPNET Modeler. http://www.opnet.com/solutions/network_rd/modeler.html.

[38] T. Clausen and P. Jacquet. Optimized Link State Routing (OLSR). In *IETF RFC 3626*, 2003.

[39] H. Huang and J. S. Baras. Component Based Routing: A New Methodology for Designing Routing Protocols for MANET. In *Proceedings of the 25th Army Science Conference*, Nov. 27-30, 2003.

[40] Sidney I. Resnick. *Heavy-Tail Phenomena. Probabilistic and Statistical Modeling*. Springer, 2007.

[41] Patrice Abry and Darryl Veitch. Wavelet analysis of long range dependent traffic. *IEEE Transactions on Information Theory*, 1998.

[42] Darryl Veitch and Patrice Abry. A wavelet based joint estimator of the parameters of long-range dependence. *IEEE Transactions on Information Theory*, 1999.

[43] D. R. Cox. *Long Range Dependence: A review*, volume IA. Iowa State University Press, 1984.

[44] Thomas G. Kurtz. Limit theorems for workload input models. *Stochastic Networks: Theory and Applications*, pages 119–140, 1996.

[45] T. Issariyakul, D. Niyato, E. Hossain, and A.S. Alfa. Exact distribution of access delay in ieee 802.11 dcf mac. In *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, volume 5, pages 5 pp. –2538, dec. 2005.

[46] P. Raptis, V. Vitsas, A. Banchs, and K. Paparrizos. Delay distribution analysis of ieee 802.11 with variable packet length. In *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, pages 830 –834, april 2007.

[47] Omesh Tickoo and Biplab Sikdar. Modeling queueing and channel access delay in unsaturated ieee 802.11 random access mac based wireless networks. *IEEE/ACM Trans. Netw.*, 16:878–891, August 2008.

[48] P. R. Jelenkovic and J. Tan. Can retransmissions of superexponential documents cause subexponential delays? 2007.

[49] T. Sakurai and H.L. Vu. Mac access delay of ieee 802.11 dcf. *Wireless Communications, IEEE Transactions on*, 6(5):1702–1710, May 2007.

[50] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. In *IETF RFC 3561*, 2003.

[51] C.D. Young. Usap: a unifying dynamic distributed multichannel tdma slot assignment protocol. In *MILCOMM, Military Communications Conference*, volume 1, pages 235–239 vol.1, Oct 1996.

[52] L. Tassiulas and Anthony Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *Automatic Control, IEEE Transactions on*, 37(12):1936–1948, Dec 1992.

[53] S. Perumal, V. Tabatabaee, J.S. Baras, C.J. Graff, and D.G. Yee. Modeling tdma-based usap in jtrs mdl for multicast and unicast traffic. In *MILCOMM, Military Communications Conference*, pages 1924–1929, Oct 2010.

[54] M. F. Neuts. Probability distributions of phase type. *Liber Amicorum Prof. Emeritus H. Flor in*, pages 173–206, 1975.

[55] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. ASA-SIAM Series on Statistics and Applied Probability, 1999.

[56] Soohan Ahn, Joseph H.T. Kim, and Vaidyanathan Ramaswami. A new class of models for heavy tailed distributions in finance and in surance risk. In *Journal of Risk and Insurance*, volume 51, pages 43–52, 2012.

[57] Mitzenmacher M. and Tworetzky B. New models and methods for file size distributions. In *Allerton*, 2003.

[58] Ward Whitt. *Stochastic-Process Limits*. Springer, 2002.

[59] Soohan Ahn and Vaidyanathan Ramaswami. Bilateral phase type distributions. *Stochastic Models*, pages 239–259, 2005.

[60] S. Asmussen, O. Nerman, and M. Olsson. Fitting phase-type distributions via the em algorithm. *Scandinavian Journal of Statistics*, 1996.

[61] NS-2. http://isi.edu/nsnam/ns/.