# CROSS LAYER NETWORK ARCHITECTURE FOR EFFICIENT PACKET FORWARDING IN WIRELESS NETWORKS

by

**SACHIN GANU**

A Dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Professor Dipankar Raychaudhuri

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2007

**ABSTRACT OF THE DISSERTATION**

# Cross Layer Network Architecture for Efficient Packet Forwarding in Wireless Networks

By SACHIN GANU

Dissertation Director:
Professor Dipankar Raychaudhuri

With the evolution of 802.11-based wireless networks from hotspots to mesh networks, there has been a tremendous increase in the number of wireless users and density of deployments. Consequently, current wireless network face several problems due to interference, uncoordinated medium access, packet processing overheads at each hop and sub-optimal route selection. While radio technologies continue to improve speeds upto hundred megabits per second, the inadequacies of medium access and routing protocols severely impact the overall network capacity and end-user experience. In this thesis, we focus on improving the scalability and packet forwarding efficiency of multihop wireless networks.

We introduce a self-organizing hierarchical ad-hoc network design (SOHAN) based on a three-tier hierarchy with dedicated forwarding nodes to address the scalability of existing multihop networks. We focus on realistic system design considerations and develop a Linux-based system prototype including novel protocols for bootstrapping, discovery and topology control to enable hierarchical self-organization. Experimental and simulation-based evaluations indicate a $\sim$2.5 times

performance improvement over *flat* network models.

We address packet forwarding inefficiencies of existing techniques over multihop networks due to queuing, contention and reprocessing at each hop and propose an interface contained forwarding architecture (ICF) using a combination of cut-through MAC protocol and label-based forwarding to enable "atomic" channel access for downstream transmissions and reduce self-interference. Next, we design a cross layer enabled cut through architecture (CLEAR) that extends the ICF mechanism with novel airtime metric-based route selection to mitigate the interference between flows. We further outline a time-based coordination scheme using soft reservations during route discovery phase to coordinate multihop "burst" transfers amongst flows. This model can be adapted to support differentiated services and provide a "low-latency socket" for real-time traffic over multiple hops. Our work can be the basis for a switched multihop wireless network design that enables conflict-free transfers resulting in efficient utilization of channel capacity and providing a viable alternative to wired network deployments.

A substantial contribution of this thesis also includes the design and development of the ORBIT wireless testbed with focus on cross-layer experimental framework to facilitate rapid prototyping of wireless protocols and experimental evaluations at scale.

# Dedication

To Aai, Baba,

Aaji, Ajoba

and Bapu Kaka

for their blessings

# Acknowledgements

I truly appreciate the support and guidance of many people who have helped me in this intellectual quest. This milestone would not have been reached otherwise.

First, I would like to express my sincerest gratitude to my advisor, Dr. D. Raychaudhuri, for presenting me the opportunity to work as his student at WINLAB during the early stages of my PhD. This journey would not have been so rewarding and memorable without his vision, patience and encouragement at every stage. Based on his vast industrial and academic experience in networking and wireless systems design, he has impressed upon me the importance of a holistic outlook to approaching systems oriented problems. He has always led us by example when it came to raising the bar for research excellence, thus motivating us to strive for the best in this extremely competitive research area. I appreciate his latitude and support in allowing me to gain industrial experience during summers that has helped me in shaping my overall career and skills. I thank him for the opportunity to present my research at various forums as well as demonstrating prototypes of research ideas on the testbed. This experience and training gained through his mentoring will undoubtedly be beneficial in my future career.

I would also like to thank Ivan Seskar for sharing his wealth of knowledge and hands-on experience in all aspects of communications systems design and filling the big void between textbooks and practise. I also appreciate the support and quality feedback of my co-advisors, Dr. Trappe, Dr. Gruteser, Dr. Zhang and Dr. Acharya in shaping this thesis.

Being a part of the ORBIT wireless testbed development team has been the most exciting opportunity of my life and an invaluable learning experience. For this, I am thankful to entire ORBIT team: Dr. Raychaudhuri, Ivan Seskar, Dr. Max Ott, Dr. Rich Howard and my colleagues, Kishore Ramachandran, Pandurang Kamat, Faiyaz Ahmed, Mesut Ergin, Darshan Sonecha, Manpreet Singh and Haris Kremo for teaching me the importance and value of team work that has helped us accomplish this extremely challenging task. I will always cherish the enjoyable moments (after successful demos) and experiences during this project. I also would like to acknowledge NSF NRT research grant ANI0335244 that supported my PhD studies.

My research colleagues deserve a special mention, especially Zhibin Wu, for his helpful suggestions and feedback, as well as Suli Zhao, Sumathi Gopal, Xiangpeng Jing, Wenyuan Xu, Lalit Raju, Bhaskar Anepu and Saurabh Mathur for the intellectually stimulating conversations and exchange of ideas.

I thank Dr. P. Krishnan and Dr. A.S. Krishnakumar for providing me the opportunity to spend a summer at Avaya Research Labs and the invaluable experience and knowledge I gained while building our first indoor wireless localization system prototype. Discussions and collaboration with Dr. Arup Acharya and Dr. Archan Misra from IBM Research Labs have contributed substantially in motivating the theme of my thesis work.

A special note of thanks to the staff of WINLAB (Elaine Connors, Noreen De Carlo and James Sugrim) who have shielded the students from the administrative affairs and procedures and always taken the extra step towards prompt resolution of student matters even at short notices. They have made my stay at WINLAB an extremely pleasant and comfortable experience.

Words are insufficient to express my heartfelt regard and appreciation towards my wife, Seema, who has been my friend, mentor and guide, sharing every moment of this journey. She has led the way (during her own PhD) showing me the value of hard work and sincerity. Her motivation and

unconditional support has been my beacon during moments of disillusionment and I deeply admire her patience and understanding during these years.

The sacrifices and moral support from my parents, my brother Nikhil, the inspiration from my parents-in-law, Dr. K. K. Sharma and Mrs. Sharma and the blessings of the elders and Almighty have carried me through this journey.

Finally, I would like to acknowledge that some of the results reported in this dissertation appear in my prior publications as listed in the vita.

# Table of Contents

# List of Tables

# List of Figures

# List of Abbreviations

**ACK**: Acknowledgement

**AIFS**: Arbitration Interframe Space

**AODV**: Ad-hoc on-demand distance vector routing

**AP**: Access point

**ASR**: Airtime-based source routing

**BEAD**: Beacon Assisted Discovery protocol

**BOOST**: Bootstrapping protocol

**CLEAR**: Cross Layer enabled cut through architecture

**CM**: Chassis Manager

**CSMA**: Carrier sense multiple access

**CTS**: Clear to Send

**CW**: Contention window

**DCF**: Distributed Co-ordination Function

**DCMA**: Data driven cut through multiple access

**DIDC**: Dual interface dual channel

**DIFS**: DCF Interframe space

**DISC**: Dual interface single channel

**DSDV**: Destination sequenced distance vector routing

**DSRC**: Dedicated short range communications

**DSR**: Dynamic source routing

**EEBL**: Extended Electronic brake light

**ESG**: Vector Signal Generator

**ESSID**: Extended Service Set Identifier

**ETX**: Expected Transmission Count

**FDMA**: Frequency division multiple access

**FN**: Forwarding node

**GENI**: Global Environment for Network Innovations

**ICF**: Interface Contained Forwarding

**IRMA**: Integrated Routing and MAC scheduling

**LDP**: Label distribution protocol

**LQSR**: Link Quality Source Routing

**MANET**: Mobile Ad Hoc Networks

**MIMO**: Multiple Input Multiple Output

**MN**: Mobile node

**MPLS**: Multi-Protocol Label Switching

**NIC**: Network Interface Card

**OFDM**: Orthogonal Frequency Division Multiplexing

**OML**: ORBIT Measurement Library

**ORBIT**: Open Access Research Testbed for Next Generation Wireless Networks

**PARMA**: PHY/MAC aware metric

**PLC**: Physical Layer Capture

**QoS**: Quality of Service

**RREQ**: Route Request

**RREP**: Route Reply

**RREP-ACK**: Route Reply Acknowledgement

**RRM**: Radio Resource Manager

**RSSI**: Received Signal Strength Indicator

**RTS**: Request to Send

**SAP**: Service Access Point

**SDMA**: Spatial division multiple access

**SISC**: Single interface single channel

**SM**: Slice Manager

**SNAP**: Subnetwork Access Protocol

**SOHAN**: Self Organizing Hierarchical Ad Hoc Network

**TDMA**: Time division multiple access

**TxOp**: Transmission Opportunity

**VAP**: Virtual access point

**VMAC**: Virtual MAC

**VSA**: Vector Signal analyzer

**VS**: Virtualization support

**WCETT**: Weighted Cumulative Expected Transmission Time

**WLAN**: Wireless LAN

# Chapter 1

# Introduction

Since the allocation of the Industrial, Scientific and Medicine (ISM) frequency band for unlicensed access, wireless usage for information access has become extremely popular due to commoditization of wireless products in these frequencies and their ready availability. With the growing popularity of usage, the typical infrastructure-based *hostpot* model has been extended to multihop mesh networks to serve metropolitan areas as well as in indoor environments such as offices and warehouses. The mesh networks uses a backbone of interconnected wireless links to extended the coverage of the AP-based model.

As the number of wireless users and the density of wireless deployments grows, the overall capacity and the end-user experience greatly depends on how well the underlying channel resources are utilized. Most of the current wireless deployments are based on the IEEE 802.11 standard that uses Carrier Sense Multiple Access technique with Collision Avoidance (CSMA-CA) for distributed medium access control. With advances in physical layer technologies such as Orthogonal Frequency Division Multiplexing (OFDM), Multiple Input Multiple Output (MIMO) and beamforming techniques, the achievable wireless rates are now becoming equivalent to wired Ethernet speeds.

However, the overall performance of these networks is still significantly lower than the underlying channel capacity offered by the physical layer. This wide gap in the performance based on the IEEE 802.11 Distributed Co-ordination Function (DCF) [1] has been the subject of extensive research [2–4] for several years and can be attributed to a variety of reasons including the following.

- **Hidden nodes and exposed nodes:**

As shown in Figure 1.1(a), the *hidden node problem* arises when two transmitters are not within range of each other and communicate simultaneously with a receiver that is in the range of both transmitters. This causes collisions at the receiver. In order to mitigate this problem, the use of Request-to-Send (RTS) and Clear-to-Send (CTS) packet exchanges prior to transmission has been proposed in [5]. This imposes the constraint that no node in the vicinity of a transmitter or receiver can transmit during the subsequent RTS-CTS-Data-Ack exchange. However, this approach is too conservative and prevents parallel transmissions from taking place even if they are compatible and non-interfering as shown in Figure 1.1(b). This results in a poor spatial reuse in the extended neighborhood of an ongoing transmission.

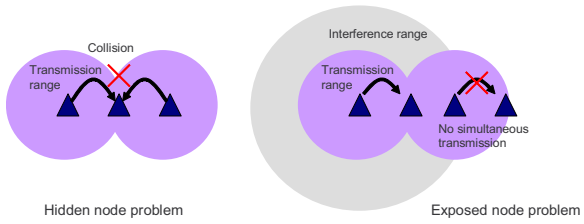



Figure 1.1: a) Hidden node problem b) Exposed node problem

- **Inefficient medium utilization due to collision induced backoffs:**

According to the DCF mechanism, nodes choose a random backoff after detecting the medium idle for the DCF Interframe Space (DIFS) interval prior to sending a frame. The backoff interval is selected uniformly from the interval [0,Contention Window] in multiples of *aSlotTime*. If no acknowledgement is received from the intended recipient of the MAC frame, the contention window is doubled and a random timer is chosen from the new window. This may

result in periods of low channel utilization immediately following a collision due to the conservative medium access control.

- **Performance dominated by low data rate connections:**

  In WLAN networks, clients may be distributed around the Access Point (AP) such that some of the clients are at the edge of the AP's coverage. This results in low SNRs link and consequently lower PHY rates for communication. This is illustrated in Figure 1.2 for different client distributions. The traffic is from the AP to the clients with RTS/CTS enabled. It can be seen in Table 1.1 that the overall performance of the network is significantly impacted by the clients that have poor connectivity to the AP due to their longer transfer durations during which other clients with good connectivity to the AP are forced to remain idle.

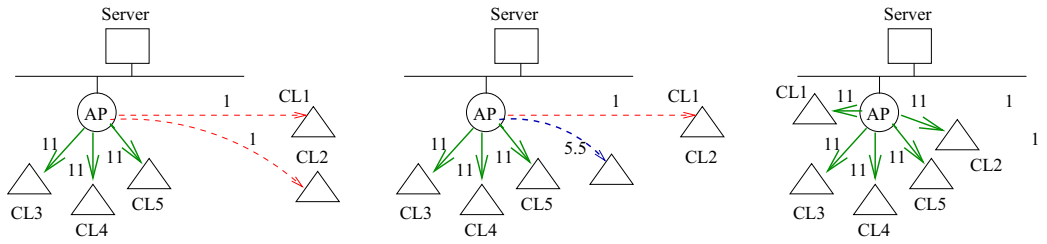



Figure 1.2: Different client distributions result in low SNR and PHY rate links lowering the overal system performance

Table 1.1: System performance with different client position distributions

| Scenario | System throughput (Mbps) | Avg latency (sec) |
|---|---|---|
| 1 (two 1 Mbps clients) | 2.189 | 1.198 |
| 2 (one 1 Mbps and one 5.5 Mbps) | 2.6207 | 1.011 |
| 3 (all 11 Mbps) | 4.856 | 0.128 |

- **Self interference and inter-flow interference:**

  The 802.11 DCF was primarily designed for the WLAN model but has been applied to multihop wireless networks to extend the range where intermediate nodes may participate in

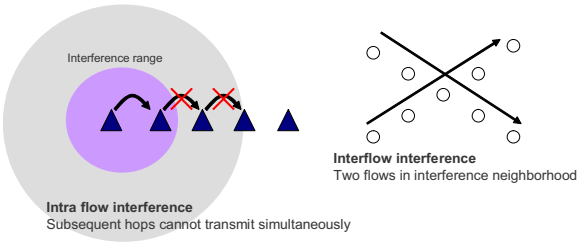



Figure 1.3: a) Self interference b) Inter-flow interference

relaying traffic towards the destination. This results in several problem as outlined in [6]. Based on the CSMA model and the inability of commodity hardware to transmit and receive simultaneously, consecutive hops belonging to the same flow compete for channel access as shown in Figure 1.3(a). This results in a suboptimal use of the underlying resources as shown in [7]. In addition, two or more flows using overlapping paths experience interflow interference resulting in collisions, backoff and poor channel utilization, collisions and multiplicative backoffs (shown in Figure 1.3(b)).

- **Bottlenecks at information sinks, performance degradation over multiple hops:**

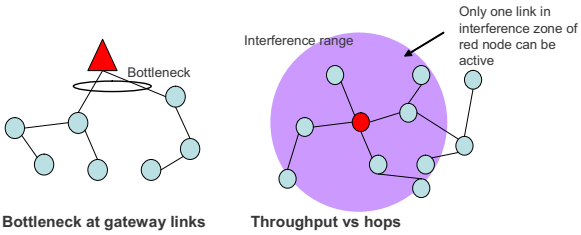



Figure 1.4: a) Bottleneck at gateways b) Only one hop can be active in the entire interference neighborhood

As seen in Figure 1.4, it can be seen that information flowing out of the network through a gateway creates a local bottleneck in the neighborhood of the sink due to increased contention. Examples of such networks include sensor networks with information flowing from

the sensors to the collector as well as infrastructure-based wireless mesh networks that provide access over a wireless backhaul via a portal connected to the Internet. Since the CSMA protocol allows only one hop in a given interference neighborhood to be active, the overall performance deteriorates with increasing number of hops.

- **Protocol and header overheads:**

  Each layer of the protocol stack introduces its own headers into the outgoing frame. In addition, the IEEE 802.11 standard mandates inter-frame spacing for supporting effective medium co-ordination as well as supporting differentiated services. These overheads result in additional reduction in the achievable application layer performance despite the availability of physical layer capacity.

- **In band control messaging and routing protocol inefficiencies:**

  In addition to the above problems, the control traffic for co-ordination (such as RTS/CTS packets and routing messages) competes with the data traffic for channel access resulting in reduced throughput. Also, earlier routing protocols [8–10] have been primarily adapted from the wired domain and use hop-count based route selection. This may result in longer links characterized by low SNR and poor reliability as reported in [11].

In summary, as addressed in [12], *existing wireless networks suffer from poor performance because the available wireless capacity is shared across many links and largely wasted by overheads and inefficiencies*. While the vision of all-wireless deployments becoming increasingly popular as an alternate access technology to wired networks, the performance of wireless multihop networks still continues to suffer because of the various problems outlined earlier.

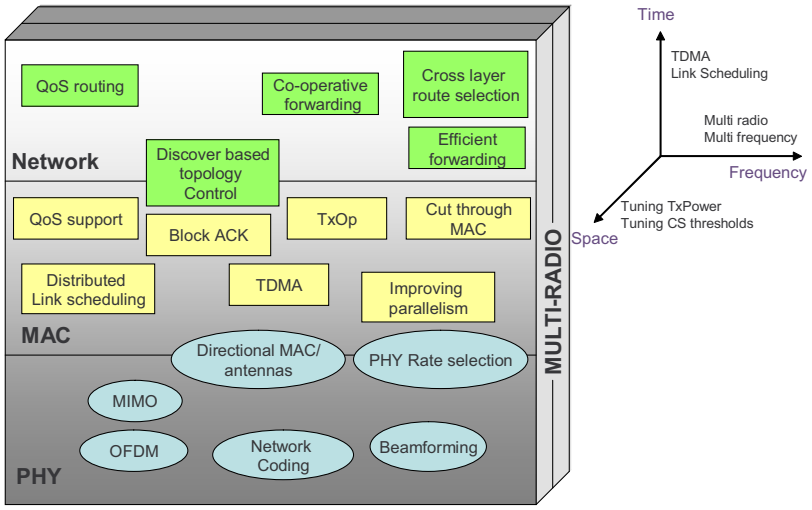



Figure 1.5: a) Summary of various enhancements for multihop networks at the PHY/MAC and network layers b) Dimensions explored for interference reduction

## 1.1 Related Work

Multihop wireless network performance and design has been actively studied for the past several years and many enhancements have been proposed at various layers of the protocol stack. A brief summary of these approaches can be found in [13, 14]. These techniques primarily focus on improving the utilization of the underlying PHY resources based on the using the frequency, time or spatial dimensions and span different layers of the protocol stack as represented in Fig 1.5(a).

- **Physical layer improvements:**

  Since the earlier 802.11 standard [1], there have been improvements in the supported PHY layer rates due to improved modulation and coding techniques. IEEE 802.11a [15] and 802.11g [16] use OFDM technology in the 5 GHz and 2.4 GHz bands respectively and are capable of speeds upto 54 Mbps. The upcoming IEEE 802.11n standard [17] further extends OFDM with multiple antennas (MIMO) and advanced techniques such as beamforming to achieve speeds comparable to wired Ethernet in conducive environments. Also, co-operative

forwarding at the physical layer (network coding) has been investigated [18–20] and shows significant promise in improving the raw capacity.

- **Addressing MAC inefficiencies:**

  Several techniques have been proposed to address the exposed terminal problem in multihop networks. In [21], the authors have proposed enhanced local coordination between neighboring nodes to enable non interfering parallel transmissions and improve the channel utilization. A distributed link scheduling algorithm is outlined in [22] that extends the 802.11 DCF with local multihop coordination resulting in a dynamic TDMA-like schedule in the local neighborhood. This allows several nodes in the interference vicinity to coordinate their transmissions hop-by-hop. In [23], a two stage channel access model using a separate channel (called busy tone channel) has been proposed. The first phase (conducted using busy tone channel) involves contention resolution and in the second stage, the nodes enter the data transmission phase on the main channel. In [24, 25], the authors have also used out-of-band control messages to improve channel utilization. The IEEE 802.11e standard [26] has proposed several enhancements to the basic 802.11 DCF access mechanism in order to handle QoS support for different traffic classes. It also provides a *burst* mode based on transmission opportunities (TxOp) to reduce per-packet overheads and improve the channel utilization. Note that these approaches explore the time dimension (as shown in Figure 1.5(b)) for coordinated medium access.

  Other approaches have explored the spatial dimension by investigating transmit power control [27] and tuning the sensitivity to reception based on adjusting carrier-sense thresholds [28]. These methods reduce the interference caused by transmissions of other flows in the interference range and increases the spatial reuse of the network. In [29], the use of directional

antennas combined with directional network allocation vector (NAV) is evaluated to reduce exposed nodes resulting from omnidirectional transmissions.

Since the ISM band supports multiple non-overlapping channels, several studies [30–33] have looked at the problem of extending the single channel wireless networks to use multiple frequencies in order to improve the capacity of the network and reduce interference within a single channel. In [34], the authors propose a link-layer solution for striping data over multiple interfaces to improve channel utilization.

- **Routing enhancements:**

  In parallel, there have been several cross-layer routing metrics [35–42] proposed to provide high quality routes over multihop wireless links as compared to shortest path. These metrics incorporate various physical and MAC parameters such as PHY link quality, link speeds, error rates, MAC contention and interference effects into route selection. Additionally, [43–45] have looked at explicit admission control and QoS routing based schemes in order to find routes in the network that can meet the flow requirements.

- **Multi-dimensional optimizations:** Recent work has focused on multi-dimensions optimizations for performance improvements. In [46,47], the authors propose and evaluate joint channel assignment and routing algorithms for multi-radio wireless networks. In [48], the authors evaluate the impact of joint transmit power control, carrier sensing adaptations and data rate selection on the spatial reuse of the network. An analytical framework has been proposed in [49] to study joint scheduling, frequency assignment and route selection to find theoretical upper limits on achievable capacity in multihop wireless networks. In [50], the authors have investigated the impact of interference on multihop networks based on the notion of conflict graphs to represent the subset of links that cannot be active simultaneously.

## 1.2 Motivation

Despite several years of research in this area, multihop wireless network deployments continue to be plagued by poor performance resulting from incremental layered solutions, low utilization of the underlying medium and inefficiencies due to control and protocol overheads. While analytical work [49, 51] has provided an estimate on the capacity bounds achievable through optimal scheduling, routing and frequency assignments, there is clearly a lack of work in the area of systematic design of protocols and architecture for wireless multihop networks to achieve improved scalability and performance. While addressing these issues, it is important to consider realistic system aspects such as heterogeneity of devices that participate in the network which imposes a set of constraints for protocol design that are different from homogeneous network assumptions.

Additionally, with the popularity of VoIP and other media applications over wireless networks that require low latencies and high bandwidth, the existing multihop networks severely lack the capability to offer performance equivalent to their wired counterparts despite improvements in the physical layer speeds. It has been pointed out in [52], that most of the multihop network design is still *hop-centric* in which packets are consumed and recreated at each hop along the path following the wired domain design. While this may be due to a variety of reasons ranging from implementation complexity to lack of hardware support, there is clearly a need to re-evaluate this basic store-and-forward paradigm in favor of cut-through forwarding techniques that reduces the overall packet latencies and improves the utilization of the channel. Also, as wired LANs have evolved from hubs to switch based model, it is important for wireless networks to follow the trajectory in order to provide viable alternatives to existing wired deployments with comparable system performance experienced by the end user. Towards realizing these objectives, it is necessary to investigate novel techniques to minimize interference and conflicts amongst flows to match the quality and reliability of wired networks and efficiently using the increasing PHY layer speeds.

## 1.3 Thesis Contributions

In this thesis, we address the issue of scalability in existing multihop wireless networks and propose a three-tier hierarchical network architecture with protocols for self organization, integrated discovery and routing to accommodate device heterogeneity and multi-radio capabilities.

Next, we focus on in-network enhancements to improve the packet forwarding efficiency of the network. Specifically, an interface-contained forwarding architecture is proposed that uses label-based forwarding and MAC enhancements for efficient packet transfers over multiple hops. We also look at the inadequacies of existing cross layer metrics and link quality estimation techniques in terms of accuracy and overhead and propose a low overhead *airtime* metric to supplement the cut-through architecture. Our main focus is to revisit the *hop-based* wireless network model and motivate a novel *path-centric* design enabling efficient data delivery and improved performance in terms of latencies over multihop wireless networks. Towards this end, we have designed a cross layer enabled cut through architecture (CLEAR) that extends the ICF mechanism with novel airtime metric-based route selection to mitigate the interference between flows. We further outline a time-based coordination scheme using soft reservations during route discovery phase to coordinate multihop "burst" transfers amongst flows. This model can be adapted to support differentiated services and provide a "low-latency socket" for real-time traffic over multiple hops. Our work can be the basis for a switched multihop wireless network design that enables conflict-free transfers resulting in efficient utilization of channel capacity and providing a viable alternative to wired network deployments.

A substantial contribution of this thesis also involved the design and development of the ORBIT open access testbed for next-generation wireless networking at Rutgers University to facilitate protocol design and experimental evaluations at scale.

The organization of this thesis is as follows: In Chapter 2, we describe a three-tier self-organizing hierarchical ad-hoc network architecture (SOHAN) to address scalability in wireless networks and focus on specific protocols for bootstrapping, discovery and topology control as well as route selection. Practical design considerations for dual radio forwarding nodes are also investigated.

In Chapter 3, we address the basic inefficiencies associated with multihop packet forwarding over the 802.11 DCF MAC protocol and propose an interface contained forwarding (ICF) architecture which establishes cut-through paths between the source and destination. This approach mitigates the self interference problem in multihop data transfers and reduces channel idle times resulting in improved packet latencies.

In Chapter 4, we evaluate the impact of cross layer parameters such as frequency diversity and physical layer rates on the system throughput and address the inadequacies of existing metrics in terms of the overhead and accuracy of estimation. A novel low-overhead *airtime* is proposed to accurately quantify the effort in transferring a packet across the network.

Based on the ICF architecture and airtime-based route selection, we propose a cross layer enabled cut through (CLEAR) architecture in Chapter 5. This uses interference aware route selection techniques in order to improve the cut-through success ratio of the ICF mechanism using spatially diverse routes. In higher spatial densities, where alternate diverse paths cannot be easily found, we propose further enhancement to the CLEAR architecture for time-based co-ordination amongst asynchronous flows for efficient "multihop burst" mode transfers. This model has also been adapted to provide differentiated services for high priority traffic.

Chapter 6 summarizes our contributions and outlines future directions in terms of an integrated routing and MAC scheduling approach to enable conflict-free data transfers and realize the vision of "switched" wireless networks. The experimental testbed and its overall architecture used for evaluation of our system prototypes is described in the Appendix. This focusses design considerations

for rapid experiment deployment, measurement framework for data collection as well as supporting repeatable experiments.

# Chapter 2

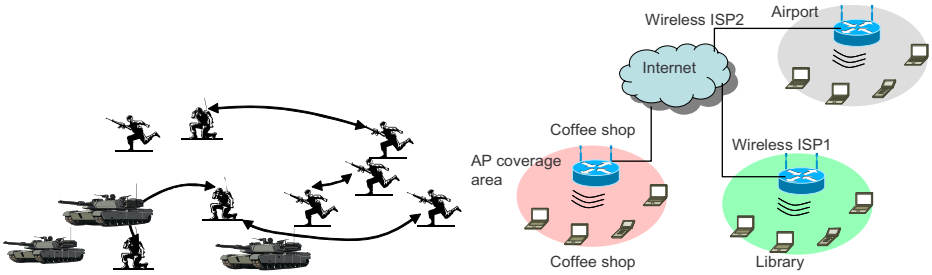
# Addressing Scalability in Multi-hop Wireless Networks

## 2.1 Introduction

In this chapter, we address the scalability issue in existing wireless network models and motivate the need for a hierarchical network architecture design. We propose a Self-Organizing Hierarchical Ad-hoc Network architecture (SOHAN) with a three-tier design including dedicated forwarding nodes. Our focus is on practical design considerations including novel protocols for bootstrapping, discovery and topology control to enable hierarchical self-organization. The benefits of this design are validated using detailed prototype implementation on Linux platform and evaluated using systematic simulation-based studies.

### 2.1.1 Motivation for hierarchical design

Wireless networks based on the popular 802.11 standard [1] can be broadly classified into mobile ad-hoc networks or infrastructure based networks and typically are designed with different objective. Mobile ad-hoc networks (MANETs) are infrastructure-less networks of mobile nodes that can be rapidly deployed and enable connectivity and dynamic communications in emergency situations or military networks and are essentially decentralized. Since the nodes are mobile, the network is dynamic and connectivity changes over time. The traffic flows in such a network may be uniformly distributed between nodes (any node may initiate a connection towards any destination) as shown in Figure 2.1(a). The information is relayed from one node to the other via paths that may span




(a) Ad-hoc networks: No infrastructure

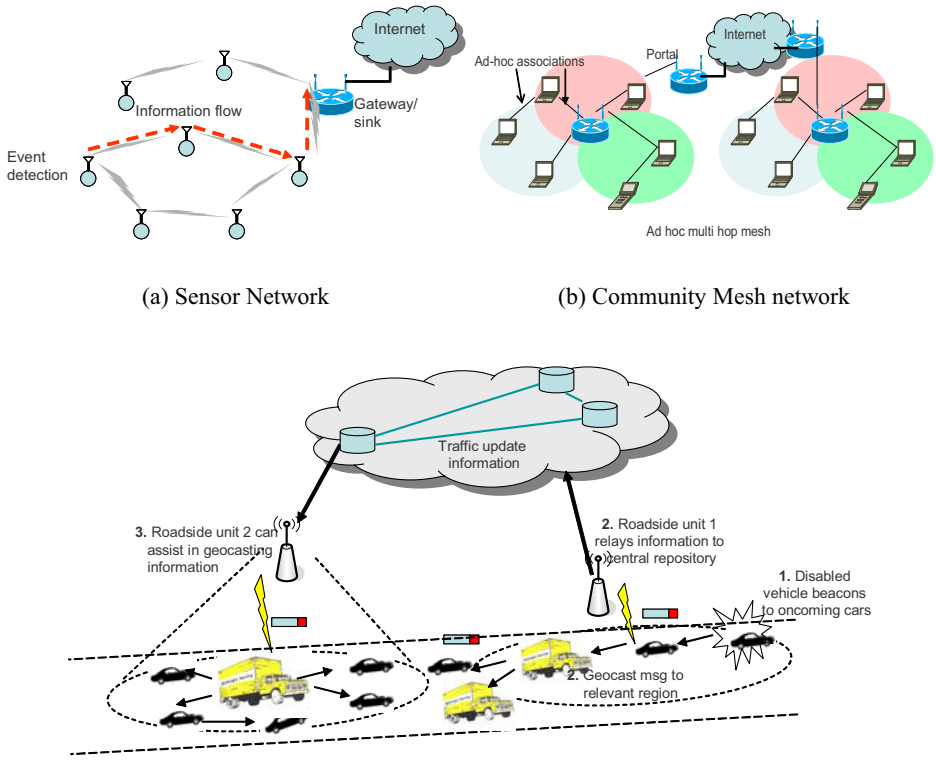
(b) Hotspot deployment: Pure infrastructure based

Figure 2.1: Ad-hoc networks and infrastructure based networks

several hops using either proactive or reactive routing protocols such as DSDV [8], AODV [9] and DSR [10]. Examples of infrastructure-based wireless networks include the typical hotspot AP model commonly deployed at airports, hotel lobbies, cafes etc. to provide Internet access to users within the range of the AP.

More common wireless usage scenarios predominantly involve traffic flows to and from the wired Internet, thus requiring effective integration of wired "access points" with the ad-hoc wireless network nodes. These networks include sensor networks (Figure 2.2a.) with information flows from the sensors to the sinks and community mesh networks (Figure 2.2b.) to extend coverage of existing hotspot based deployments for providing Internet access. More recently, there have been advances in wireless networking for vehicular safety applications such as extended electronic brake light (EEBL) (Figure 2.2c.), since the announcement of the Dedicated Short Range Communications standard for vehicular communications. These networks also use a combination of ad-hoc links between vehicles with few roadside infrastructure gateways for information transfer.

As the size of the network increases, the conventional *flat* network design that uses shared medium access protocols along with multi-hop routing suffers from a degradation in the throughput per node. The seminal paper by Gupta and Kumar [53] showed that in a network comprising $n$




(a) Sensor Network

(b) Community Mesh network

(c) Vehicular networks

Figure 2.2: Hybrid wireless networks with ad-hoc and infrastructure links

identical nodes, each of which is communicating with another node, the throughput capacity per node is $\Theta(\frac{1}{\sqrt{nlogn}})$ assuming random node placement and communication pattern and $\Theta(\frac{1}{\sqrt{n}})$ assuming optimal node placement and communication pattern. The scaling problem in flat network architectures becomes even more aggravated due to traffic bottlenecks around gateway nodes when a significant fraction of packets have to be routed to a correspondent host within the wired Internet.

In order to alleviate this problem, earlier work on hierarchical radio networks [54, 55], and more recently on hybrid networks [56] has reported that hierarchical organization of the network yields scalability improvement in these radio networks. In particular, Liu and Towsley [56] have shown that for a hybrid network with a two-tier hierarchy comprising several APs and multi-hop connections to the APs, the capacity scales linearly when the number of APs is greater than square

root of the number of nodes. These results indicate that a network with more than one tier of ad-hoc radio nodes with lower tiers aggregating traffic up to intermediate radio relays can scale well with the right number of infrastructure nodes to support a given number of clients. This motivates the design of our three-tier self-organizing hierarchical network architecture (SOHAN). We describe the SOHAN architecture including protocol design considerations for hiererchical self-organization of devices with heterogeneous capabilities.

## 2.2 Self Organizing Hierarchical Ad-hoc Network Architecture (SOHAN)

In the SOHAN architecture, we propose a three-tier hierarchy of devices: low power "mobile nodes" with limited functionality, higher power "radio forwarding nodes" that route packets between radio links and "access points" that route packets between radio links and the wired infrastructure (Figure 2.3). This model can be directly extended to the current mesh networks where the forwarding nodes are *mesh points* and the access points are the *mesh portals*. Devices with different capabilities such as laptops, PDAs, dual-mode phones, sensors etc. may participate in the network. These have inherent limitations in terms of computational capacity and energy consumption. The SOHAN architecture and protocol design accommodates the heterogeneity of devices. The individual system components of our hierarchical model are described next.

### 2.2.1 System model and assumptions

- *Mobile Nodes (MN)*: The MN is a mobile end-user device (such as sensor, PDA or a laptop) at the lowest tier (tier 1) of the network. The MN attaches itself to one or more nodes at the higher tiers of the network in order to obtain service using 802.11x radios. As an end-user node, the MN is not required to route multi-hop traffic from other nodes. Note that, as a battery-operated end-user device, the MN will typically have energy constraints.

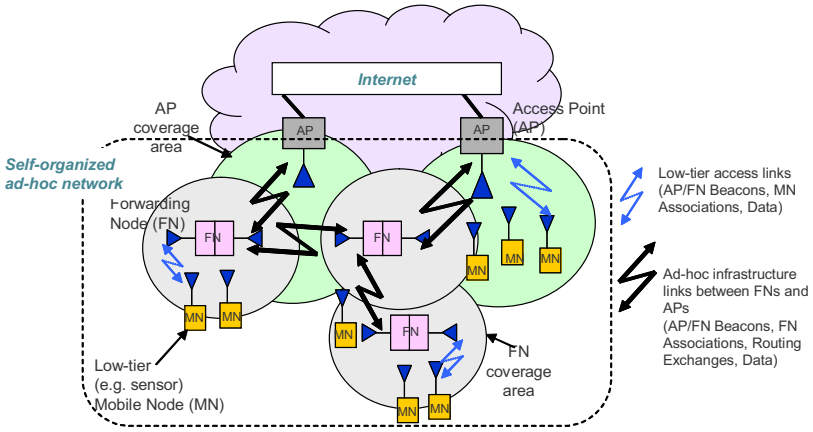



Figure 2.3: Three tier hierarchical ad-hoc network architecture

- *Forwarding Nodes (FN)*: The FN is a fixed or mobile intermediate (tier 2) radio relay node capable of routing multi-hop traffic to and from all three tiers of the network's hierarchy. They typically do not originate traffic of its own and are provisioned solely for multi-hop forwarding of transit packets. An FN with one 802.11 radio interface uses the same radio to connect in ad-hoc mode to MNs, other FNs and the higher-tier APs defined below. Optionally, an FN may have two radio cards, one for traffic between FNs and MNs (known as the *beaconing* interface or the *access* interface) and the other for inter FN and FN-AP traffic flows (known as *scanning* or *infrastructure* interface typically on a different frequency). The FN can be plugged into an electrical outlet, but in certain scenarios, may also be also be a battery-powered mobile device. Thus, the FN is also energy constrained, but the cost is typically an order of magnitude lower than that of the MN defined above. This node is equivalent to the *mesh point* in the IEEE 802.11s [57] mesh network terminology.

- *Access Points (AP)*: The AP is a fixed radio access node at the highest tier (tier 3) of the network, with both an 802.11x radio interface and a wired interface to the Internet. The AP is capable of connecting to any lower tier FN or AP within range but unlike typical 802.11

WLAN deployments, it operates in ad-hoc mode for each such radio link. The AP also participates in discovery and routing protocols used by the lower tier nodes, and is responsible for routing traffic within the ad-hoc network as well as to and from the Internet. Logically, the tier 3 APs are no different from tiers 1 and 2 when routing internal ad-hoc network traffic - the wired links between APs are reflected in (generally) lower path metrics. Since the AP is a wired node, it is usually associated with an electrical outlet and energy cost is thus considered negligible. This node is equivalent to the *mesh portal* in mesh networks.

Our goal is to design the system such that ad-hoc network advantages of dynamic self-organization and low routing overhead/complexity are retained at the lower tiers of the system, while providing the capacity and scaling advantages of a hierarchical network structure. Note that our design does not require a strict hierarchical model: the lowest tier of nodes can directly connect to the APs; similarly FNs may use other FNs to connect to the APs.

### 2.2.2 Design considerations

In this section, we address several design criteria and protocol considerations for the SOHAN architecture. Specifically, the following questions arise and are addressed in the subsequent sections:

- How do forwarding nodes select initial frequencies for operation upon joining a network?
- How do heterogeneous devices efficiently self-organize into hierarchies across multiple frequencies?
- How is traffic routed between the devices and the gateways?
- How does the hierarchical architecture perform as compared to the flat topology?
- What are the prototype design considerations for a multi-radio platform (for forwarding nodes)?

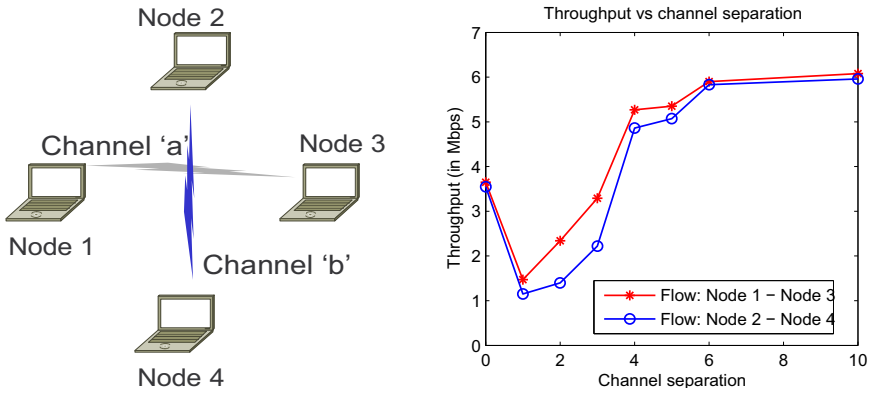
## 2.3 Bootstrapping protocol: Initial frequency assignment at FNs (BOOST)

Bootstrapping the network typically involves the selection of radio frequencies as well as initial transmit power levels to be used when a new node joins the network. While the APs/FNs can be initially provisioned on non-overlapping channels by the network administrator to minimize interference, the goal of the bootstrapping protocol is to eliminate the necessity for manual configurations and automate the process by making the devices capable of sensing the environment and adapting accordingly. In this particular design, we focus on the frequency assignment for the *access interface* at the FN that minimizes interference and maximize system throughput. The AP channels are assumed to be pre-configured and the infrastructure interface of the FN will associate with an AP/FN as controlled by our discovery protocol (described in Section 2.5.2).

### 2.3.1 Frequency Separation: Effect on flow throughput

The motivation behind the BOOST protocol [58] is based on experimental observations from Figure 2.4a. In this setup, two flows use 802.11b radios (Linksys radios based on Prism chipsets) and are located within interference range. The channel separation between the flows is increased from zero to ten during the course of the experiment resulting in a frequency separation of 0 MHz to 50 MHz respectively between the center frequencies.

It can be seen that when the channel separation is one or two, both the flows suffer considerably in terms of throughput. This can be attributed to the fact that when the two flows operate on adjacent overlapping channels, the received power from Flow 1 transmitter at Flow 2 transmitter is close to the carrier-sense threshold of the Flow 2 transmitter and vice versa. This may result in failed carrier sensing at each transmitter prior to attempting frame delivery resulting in collisions at the respective receivers and reduced throughput. Beyond a channel separation of five, the channels are orthogonal and hence both flows experience a throughput of $\sim$6 Mbps.




(a) Experiment setup (b) Throughput vs channel separation

Figure 2.4: Effect of adjacent-channel interference: Throughput degradation due to failed carrier sensing

**Algorithm:**BOOST-A Algorithm

operating_channel = 1;
**for** $(i = 1; i < numChannels; i++)$ **do**
    Set current_channel = i;
    Stay for $t_{dwell}$;
    Record (RSSI, channel, src) for each packet from each src;
    Compute median RSSI for each source and record the minimum value ($RSSImin$) from these;
    Record ($RSSImin_i$, $i$) for each channel $i$;
**end**
operating_channel = channel with least $RSSImin$;

**Algorithm 1**: BOOST-A channel selection

### 2.3.2 BOOST: Interference aware frequency selection

Based on the above observations, we propose two heuristic algorithms for initial frequency selection at the *beaconing* interface of the FN (i.e. the access interface towards MN). In BOOST-A Algorithm, the goal is to avoid channels where the receptions are close to the receiver sensitivity of the node, thereby reducing the likelihood of missed carrier sensing and collisions that follow. The algorithm is illustrated using an example in Figure 2.5 where the circles indicate packet RSSI.

- When a new FN is powered on or joins the network, the scanning interface (i.e. infrastructure interface) sweeps through the channels in order to discover its neighbors and associate with a given AP/FN based on the discovery procedure described in the next section

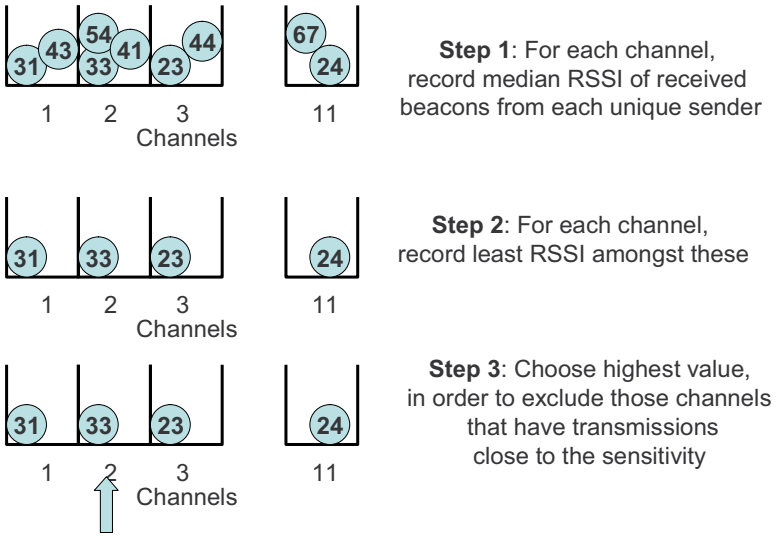



Figure 2.5: BOOST-A Algorithm Illustration

- During this sweep, it receives beacons from APs or other FNs in its neighborhood and records the received power (RSSI) and the number of terminals heard on each channel

- Out of the received power per packet on each channel, the FN then records the minimum value of the received power. By doing this, we are taking into account all the possible neighbors whose packets are received at power levels close to the receiver sensitivity threshold and hence might potentially induce failed carrier sensing and reduce throughput

- Of these minimum values, we choose the highest across all the channels. This gives us the best possible channel to use assuming that there is no channel on which no beacons were detected

- Once the channels are assigned weights based on the above procedure, the beaconing interface of the FN is set to that particular channel and it starts sending beacons on that channel

In an alternative approach (BOOST-B), each FN records the number of beacons (from all senders) received on each channel during scanning and chooses the channel with minimum number

Table 2.1: BOOST Simulation Parameters

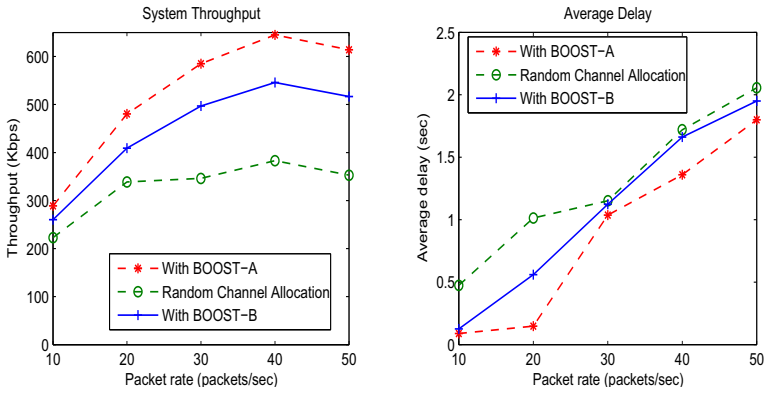
| | |
|---|---|
| Simulation area | 400×400m |
| Number of nodes (AP:FN:MN) | 1:6:10 |
| PHY Rate: Range | 1 Mbps: 250m |
| MAC | CSMA (adapted for hierarchical associations) |
| Traffic | CBR 10-50 packets per sec (steps of 10) |
| Packet size | 512 bytes |
| Number of flows | 10 |

of beacons. BOOST-B assumes that all the received beacons come from stations that are always active and thus accounts for the number of interferers on a given channel. The advantage of BOOST-B is its simpler implementation.

### 2.3.3 Performance evaluation

The above algorithms were implemented in ns-2 [59] simulator and compared with random channel selection at the FNs. Note that the ns-2 simulator was enhanced to include the concept of frequencies and channel overlap factor as outlined in [60] which is based on Prism based chipsets used in our experiments as well. Our simulations were conducted on a topology comprising 1 AP, 6 FNs and 10 MNs with 802.11b channel settings. The flow of traffic was from the MNs to the AP using intermediate FNs. The MNs scan all eleven channels to identify a suitable FN to forward their traffic. The simulation parameters are summarized in Table 2.1.

As seen in Fig 2.6, channel selection has a significant impact on the performance of the network, both in terms of throughput and average delay. Both BOOST-A and BOOST-B outperform the random channel allocation. In BOOST-B, the performance is improved since we reduce the number of interfering nodes while selecting channels. This results in a performance gain of approximately 34%. As the number of beaconing nodes (1 AP and 6 FNs) is more than the number of orthogonal channels (3), BOOST-B may select adjacent overlapping channels. BOOST-A minimizes the likelihood of interference by avoiding channels on which reception is close to the carrier sense threshold.




(a) System throughput (b) Average delay

Figure 2.6: BOOST: Simulation results

This provides approximately 56% higher throughput than the random channel selection. BOOST frequency adaptation may be activated during regular network operation as well in order to continuously monitor interference from existing deployment as well as uncontrolled interference from any co-located networks. While the BOOST protocol is specified for the beaconing interface of the FN, it can be applied to a new AP that is provisioned in the network. In the next section, we look at how the heterogenous devices self-organize into hierarchies suitable for data transfers.

## 2.4 Topology control in heterogenous networks: Discovery Protocol

So far we have looked at the case when a new infrastructure device such as FN or AP joins the network, and uses the interference-aware BOOST mechanism to select the initial frequency of operation. Prior to supporting active data transfers, it is necessary for these devices to discover each other and organize into topologies in a distributed manner. In this section, we motivate the need for introducing an explicit discovery mechanism to assist topology formation. The objective is to mitigate the control overhead and the routing messages as well as accommodate the underlying constraints of different heterogeneous devices.

### 2.4.1 Motivation

In traditional wireless networks, the routing protocol itself is responsible for building the topologies either using on-demand broadcast of route requests or by exchanging neighbor information pro-actively with one hop neighbors. While this may be sufficient for smaller networks, as the number of nodes increases, it results in a high volume of routing message exchanges that compete with existing data traffic for channel access. The problem is more severe in a multi-channel network where the multiple nodes that need to communicate could be on different radio channels. In this case, the routing messages need to be propagated across multiple channels in order to enable data transfer from one node to the other. In addition, topology formation based on routing assumes the devices to be homogeneous with the same capabilities and constraints whereas in real networks, there are many kinds of devices from battery operated PDAs/sensors to powered APs with relatively higher processing capabilities that participate in the network.

In order to address these issues, we propose a "discovery" sub-layer between the MAC and routing layers in the protocol stack as shown in Figure 2.7. Discovery can be described as the process through which a node becomes aware of its surroundings, that includes determining the presence and type of neighbors, assessing quality of links to other nodes, and providing information to the routing protocol to identify the most efficient path to the destination. While the MAC layer detects the physical topology, the discovery protocol processes this information to determine the logical topology that should be visible to the routing protocol. Routing overhead is thus reduced as the routing protocol has to deal with fewer links. In addition, the discovery protocol may also provide a metric that can be used by the routing protocol for choosing paths to forward data based on system objective functions (such as minimizing power consumption, maximizing throughput, minimizing delay etc.).

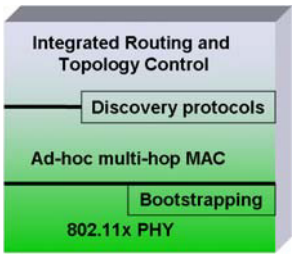



Figure 2.7: SOHAN Protocol Stack

### 2.4.2 Centralized performance bounds

In order to motivate a distributed protocol design, we first consider a centralized approach using linear programming using Matlab [61][1]. This is used to determine bounds on the network performance under different network objective functions such as minimum energy, minimum delay and maximum throughput.

1. The minimum delay optimization finds the topology that minimizes the number of hops from each MN to an AP. This corresponds to the shortest-path metric commonly used in routing protocols.

2. An important criteria at the battery operated MNs is energy consumption. In the hierarchical architecture, the cost of energy at the FNs is at least an order of magnitude less than that at the MNs, while the cost at the AP is negligible

3. For the throughput maximization, we assume that the MNs offer identical loads to the network and are the only sources of data. In the case of the dual-interface FN, uplink and downlink traffic are assumed to operate on orthogonal channels, enabling concurrent transmissions. Maximizing the throughput is done by balancing the MN load over the various APs of the network.

[1]Portions of this work were done in collaboration with L. Raju and appear in [62].

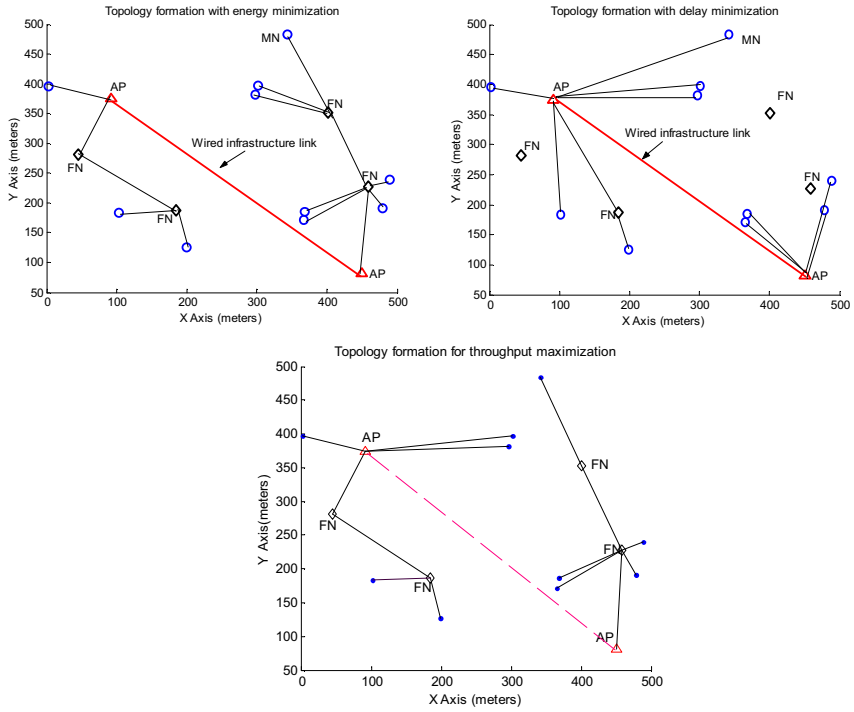



Figure 2.8: Topologies based on only primary objectives: Energy minimization, delay minimization and throughput maximization

A topology of 500m×500m is considered with a random deployment of 2 APs, 4 FNs and 10 MNs. The energy consumption at APs and FNs is normalized w.r.t. to that of the MN using the ratio 0:0.01:1 (AP:FN:MN). The propagation model is assumed to have a path loss exponent of 2.5. We considered several random placement of nodes and show topology formation for one such placement under different *primary* objectives.

As seen in Figure 2.8(a), the minimum energy criteria forces the MNs (and FNs) to associate with nearest FN or APs. This may result in an imbalanced distribution of clients amongst APs resulting in overall throughput degradation. Similarly, in Figure 2.8(b), the delay minimization criterion results in MNs (and FNs) choosing least hop count paths (longer links) towards APs. This may result in an increased energy consumption (and low reliability) in order to connect to distant FNs or APs. For throughput maximization, the MNs may choose to associate with APs over longer links and hence may experience higher energy consumption.

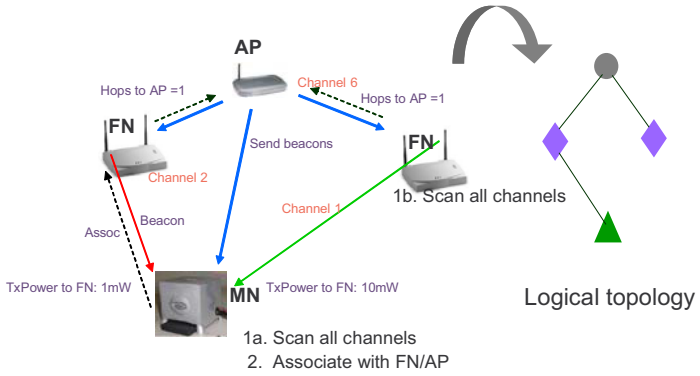
In order to supplement our *primary* conditions, we impose *auxiliary* constraints in our optimization objective functions. Assuming a traffic flow from battery operated MNs to the infrastructure, we particularly focus on energy minimization as the primary criterion with additional constraints: 1) energy minimization with delay constraints and 2) energy minimization with throughput constraints. A large number of random placements of nodes have been considered. We observe that the MN consistently selected the nearest AP/FN irrespective of the optimization criteria. Additionally, the FN's choice of neighbor is heavily influenced by the choice of the network objective. Also, the throughput based constraint did not yield consistent results in terms of the choice of neighbors at the FN. Based on these observations, we choose a decoupled objective function in the design of our distributed protocol: Energy minimization at the MNs and delay minimization at the FNs as our network objective for the distributed protocol design.

### 2.4.3 Distributed heuristics: Beacon assisted discovery protocol

Our design goal is to minimize the energy consumption at the power constrained MN layer while maintaining low latencies based on controlling FN neighbor selection. We propose a beacon-assisted discovery protocol (BEAD) to approximate the performance obtained by the centralized topology selection that uses augmented MAC beacons as shown in Figure 2.9(b).

The basic operation of the BEAD protocol is illustrated in Figure 2.9(a) and consists of:

- **Beaconing phase**: Assuming frequency selection based on the BOOST protocol, the APs and the *access interface* of the FNs transmit periodic beacons announcing their presence. During this time, the MNs and the *infrastructure interface* of the FNs scan the frequencies and collect beacon information for each channel. In addition to the source and destination address, the beacon contains the node type to indicate that the source is an AP or FN. Additionally, it may contain metrics such as hop count which indicates the number of hops to reach the




(a) BEAD protocol: discovery and association

| Source MAC | Broadcast MAC | Node ID | Packet Type (beacon) | Cluster ID | Seq. No | Node Type | Hop Count | TxPower |
|---|---|---|---|---|---|---|---|---|

(b) BEAD: Beacon format

Figure 2.9: BEAD: beacon format and discovery process

infrastructure through the Access Point.

- **Association phase**: The MNs and the infrastructure interface of the FN record the beacons observed during the scanning phase and associate with a parent node based on the discovery metric. In our design, we choose the received beacon SNR as the metric for association at the energy-constrained MNs and hops to AP at the FNs. Upon selecting the parent node for association, the MNs and FNs send an association message to this neighbor. The parent may optionally acknowledge this association message. Following the association phase, the initial topology is created. The routing protocol uses the subset of neighbors selected during the discovery phase to exchange topology updates for discovering paths to the AP.

### 2.4.4 Performance evaluation: centralized vs BEAD

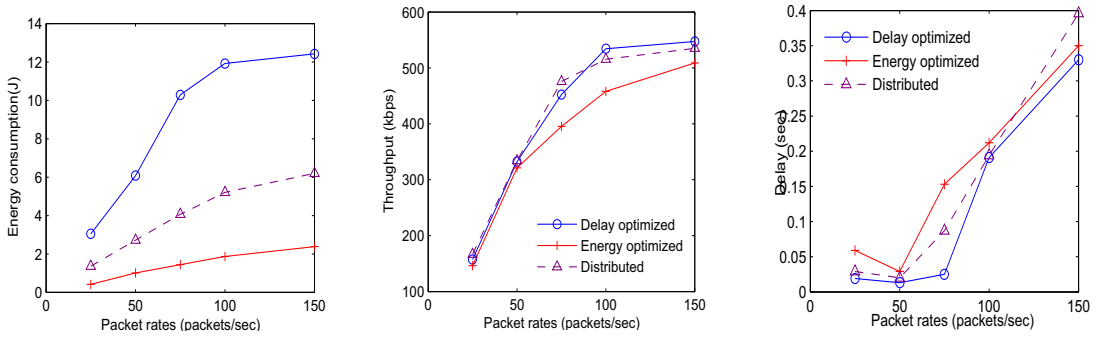
We evaluate the performance of the BEAD protocol implementation against the topologies specified by the centralized computation using ns-2 simulations. For BEAD evaluation, we augment AODV

to propagate only those route discovery broadcasts initiated by the subset of neighbors based on discovery protocol. The routes are selected from this subset presented by the discovery process. Also, the MNs do not participate in the routing process and discard all the received routing packets. For the centralized cases (energy optimized and delay optimized), we import the topology obtained from centralized calculations using MATLAB by configuring the routing tables prior to data exchange. Note that the routing messages are still exchanged for overhead computation but do not impact the topology. A random deployment of 2 APs, 4 FNs, and 10 MNs over an area of 500mx500m is considered. Each MN offers the same load, destined for the Internet through any available AP. The simulation parameters are summarized in Table 2.2.

Table 2.2: BEAD Simulation Parameters

| | |
|---|---|
| Simulation area | 500×500m |
| Number of nodes (AP:FN:MN) | 2:4:10 |
| PHY Rate: Range | 1 Mbps: 250m |
| MAC | CSMA (adapted for hierarchical associations) |
| Traffic | CBR 25-150 packets per sec (steps of 25) |
| Packet size | 64 bytes |
| Number of flows | 10 |
| Beacon Interval | 100ms |
| Missed beacons allowed | 5 |
| Rescan interval | 50×beacon_interval |

In Figure 2.10(a), the energy consumption of BEAD is compared with that of the centralized approaches for energy and delay minimized topologies respectively. Likewise, in Figures 2.10(b and c), the throughput and average delay for BEAD is compared to the centralized minimum delay and the minimum energy topologies respectively. As seen, the performance of the distributed BEAD protocol is comparable to that of the centralized case. This implies that topology chosen by BEAD (using discovery and routing) is close to the set of optimal paths chosen by the routing protocol. We also evaluate the performance in terms of routing overhead and consider two scenarios: an increase in the number of MNs (50 to 200) and increased mobility for 100 MNs (5-20 m/s). As




(a) BEAD energy consumption

(b) BEAD system throughput

(c) BEAD average delay

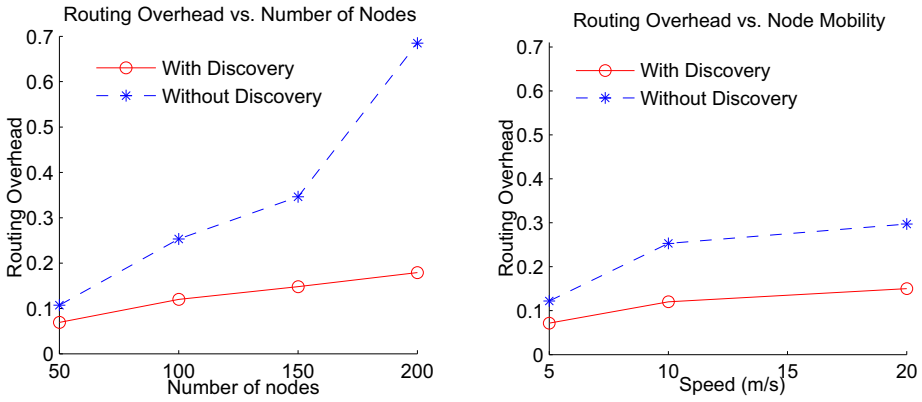
Figure 2.10: BEAD Performance compared to centralized algorithms

seen in Figure 2.11, it can be seen that the discovery protocols provides a significant improvement in routing overhead under increasing mobility as well as increasing number of nodes while meeting the desired network objectives.

In summary, the distributed beacon assisted discovery protocol provides the flexibility to control the topology and performance based on the network objective and minimizing control overheads by presenting a logical topology to the routing layer.

## 2.5 Prototype design and implementation on the ORBIT testbed

Experimental evaluations are important to complement and verify the simulation-based evaluations that are typically based on simplifying assumptions and may not capture real-world physical layer effects and other hardware and software limitations. Thus, it is increasingly important to have a controlled experimental platform for rapid prototyping and system evaluations. Such a platform should facilitate experiment setup and repetition while providing visibility at various layers of the protocol stack and the ability to control experimental parameters such as traffic and topologies. In




(a) Increasing node density

(b) Increasing node mobility

Figure 2.11: BEAD: Reduction in routing overhead

addition, the platform should provide access to a reasonably large number of nodes in order to enable evaluations at scale. We have designed and developed the ORBIT indoor wireless experimental testbed [63, 64] in order to address these requirements and provide a valuable tool for the research community for large-scale experimental evaluations with remote experimentation capabilities. Further details of the testbed along with the control and management framework can be found in the Appendix.

In order to validate SOHAN protocol design, we have implemented a Linux based prototype on the ORBIT experimental testbed. The implementation of the discovery and routing protocols was done using C programming with the Libnet [65] library to create custom MAC beacons, associations and handle their transmissions and Libpcap library [66] for packet reception. The receiver uses the Libmac [67] library, which has been implemented to provide easy access to underlying cross layer information (Received Signal Strength Indicator (RSSI), PHY rate, timestamp) from the device driver on a per packet granularity.

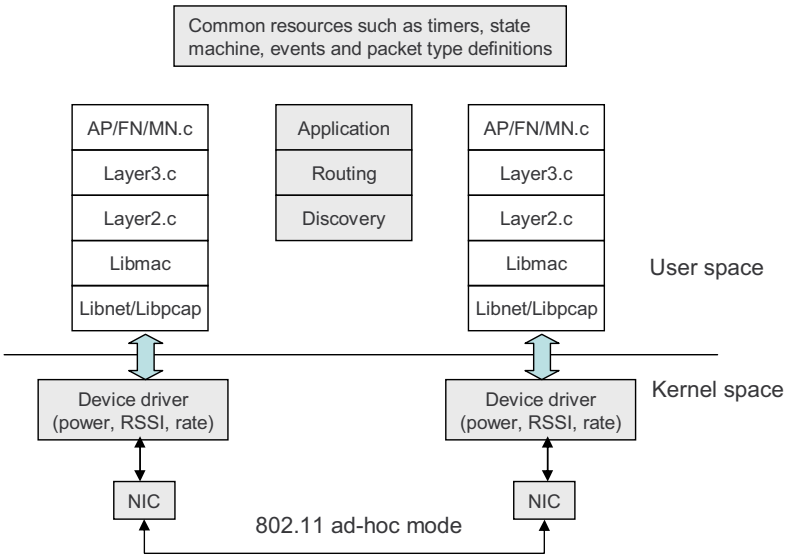



Figure 2.12: SOHAN software stack

### 2.5.1 Software stack

As shown in Figure 2.12, the modular software design is consistent with the SOHAN protocol stack and thus provides an easy way to modify functionality and add features at any layer. Note that this is predominantly a user space implementation with some kernel modifications to provide access to cross layer information such as RSSI, transmit power etc.

**send/recv**: We use the Libnet library APIs [65] for sending unicast association and data packets and broadcast beacons bypassing the TCP/IP stack. All our packets are marked with a unique Ethernet type (0x0800) for appropriate filtering and demultiplexing on the receiver side using the Libpcap APIs [66].

**Layer2.c**: This handles the discovery layer functionality and is responsible for maintaining local "neighbor" tables based on exchanged beacons and association messages. Layer 3 uses information collected from the beacons and associations for actual route selection.

**Layer3.c**: This layer is responsible for handling the functionality related to maintenance of the

local forwarding tables at each node created by periodic exchanges of neighbor tables amongst one-hop neighbors. Upon the expiration of the route update timer, a periodic forwarding table exchange takes place. Entries are purged upon expiration of the refresh timer.

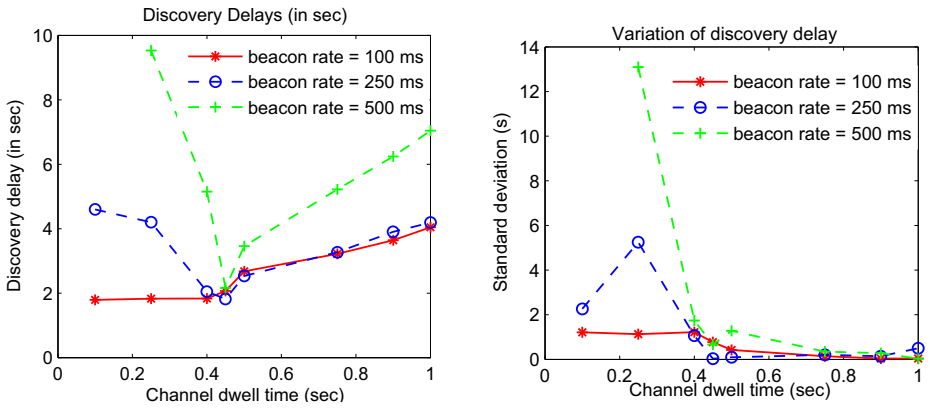
**Application Layer**(AP/FN/MN.c): This layer handles the application specific functionality that depends on the type of the nodes. MNs are responsible for scanning on all channels, listening for beacons, determining the best cost parent to associate with and sending associations to the chosen parent node. They are not involved in routing. FNs have been implemented as MNs on one interface and APs on the other. They are involved in routing and route table exchanges. The APs and access interface of FNs send beacons at regular intervals and listen for associations received from the lower layer nodes. They also send periodic topology updates to the management station.

**Common functions**: The common functionality such as timer management, event management, finite state machine, packet type definitions and common wireless utilities is handled by programs common to all layers and is defined in the corresponding files.

### 2.5.2 Discovery protocol implementation

For the Linux prototype, the beacons used in the prototype are application-level packets and not 802.11 management frames. In our implementation, we modified the device drivers to append transmit power to each outgoing beacon at the APs/FNs and the received signal strength (RSSI) for each incoming beacon at the MNs. Using this information, the node with the maximum $(\frac{RSSI}{TxPower})$ ratio was chosen as the next hop neighbor. Assuming reciprocity of channel, this metric selects the neighbor that can be reached with minimum transmit power while maintaining a high RSSI. In case, there were two or more such nodes, the node whose beacon was received with the higher signal strength was chosen. This metric is chosen to minimize energy consumption at MNs.

**Choice of beacon intervals and dwell times:** We configured the MNs to scan every channel




(a) Average discovery delay

(b) Standard deviation of discovery delay

Figure 2.13: a. Discovery delays with different beacon intervals and dwell times b. Variation of the discovery delays with different beacon intervals and dwell times

and varied the channel dwell times (from 100 ms to 1 sec) for different experimental runs. At the AP, we varied the beacon interval from 100 ms to 500 ms. Also, scanning across channels at the MNs was performed at the application layer using *ioctl* calls to the device driver. We measured the discovery delays for a scenario consisting of a single AP and MN. Discovery delay is the time interval between beginning of the experiment (both nodes starting at the same time) until the first "association" message was received by the AP from the MN. This was repeated for different beacon intervals (100 ms, 250ms and 500 ms) at the AP with varying dwell times (from 100 ms to 1 sec) at the MNs. Figure 2.13(a) show the results for the average discovery delays (in sec) for several sample runs for each setting along with the standard deviation of the delays.

As shown in Figure 2.13(b), we found that for dwell times below 450ms, discovery delay showed a high variation. This was because the MN missed a lot of beacons during its scan and hence the channel sweep iteration during which the first beacon was received was highly variable. When the dwell time per channel was higher than 450 ms, the variation of the discovery delay is significantly lesser than in the previous case. As explained before, the application controls changing of the channels on the card through *ioctl* calls and not done by the firmware. Also, the performance with

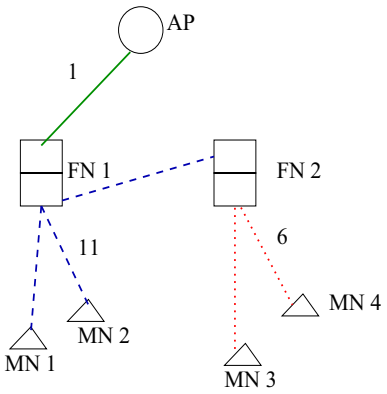



Figure 2.14: Sample topology for illustrating routing table updates

beacon intervals of 100ms and 250 ms was very similar. Hence, we chose 250 ms as the beacon interval at the APs/FNs with a channel dwell time of 450 ms at the MNs as a compromise between discovery delays and injecting more beacons in the network which increased the control overhead.

### 2.5.3 BEAD assisted route selection

The routing protocol is based on the existing discovery mechanism which is used to detect any change in the topology. The goal is to create and maintain local neighbor tables at each of the FNs and APs that are populated based on the beacons and associations received during the discovery phase. These local neighbor tables are then exchanged periodically between neighboring nodes. The neighbors use the tables to learn new information from the sender and update their local tables accordingly. Note that the MNs do not participate in multi-hop routing procedure for power conservation. The MNs send their data to the default best cost parent selected during the discovery procedure. We assume for the FNs that *eth0* is the interface towards the APs and *eth1* is the interface towards the MNs (and other FNs). Also, we use a combination of MAC addresses and node ID of the nodes for the routing protocol. This is to handle the case of the FN that has two different MAC addresses for the two different interfaces but the same node ID.

**Neighbor Table Formation and Updates:** Based on the beacons and associations received

Table 2.3: Local neighbor table at FN1

| MAC[1] Addr | Node[1] ID | Node[3] Type | Refresh[4] Timer | Channel to [5] Next Hop | Cost to [6] Destination | Interface to[7] Next Hop | Next[8] Hop |
|---|---|---|---|---|---|---|---|
| AP1 | AP1 | AP | xx | 1 | 1 | eth0 | AP1 |
| FN2 | FN2 | FN | xy | 11 | 1 | eth1 | FN2 |
| MN1 | MN1 | MN | xz | 11 | 1 | eth1 | MN1 |
| MN1 | MN1 | MN | yz | 11 | 1 | eth1 | MN1 |

Table 2.4: Local neighbor table at FN2

| MAC Addr | Node ID | Node Type | Refresh Timer | Channel to Next Hop | Cost to Destination | Interface to Next Hop | Next Hop |
|---|---|---|---|---|---|---|---|
| FN1 | FN1 | FN | xy | 11 | 1 | eth0 | FN1 |
| MN3 | MN3 | MN | xz | 6 | 1 | eth1 | MN3 |
| MN4 | MN4 | MN | yz | 6 | 1 | eth1 | MN4 |

from the nodes, each node forms a neighbor table with entries for the nodes that have been heard during discovery phase. Each entry is associated with a refresh timer and upon the expiration of the timer, the entry is purged. If a beacon (from a node previously heard) is not received in three (configurable) successive scans, then the entry for that node is purged from the neighbor table. Also, if a beacon is received from the nodes that already exist in the local neighbor table, then the refresh timer for those entries is reset. Otherwise, the refresh timer is decremented by one for every sweep through the channels. Two such local neighbor tables at FN1 and FN2 based on the topology in Figure 2.14 are shown.

**Table Updates/Exchanges**

[1]MAC - This is the MAC address of the node whose beacon/association was received

[2]Node ID: This is the unique id of the node in the network. We need this in addition to MAC address since the FN has two interfaces and hence two different MAC addresses

[3]Node Type: MN/FN/AP

[4]Refresh timer: Initial value is three, decremented for every sweep through the channels, reset upon hearing a beacon/association

[5]Channel to NH: Channel on which this neighbor can be reached

[6]In the current implementation we use hop count

[7]Interface to NH: Interface to reach the next hop

[8]Next Hop: Next hop to reach the destination specified in entry

Table 2.5: Updated table at FN2 after receiving update from FN1

| MAC Addr | Node ID | Node Type | Refresh Timer | Channel to Next Hop | Cost to Destination | Interface to Next Hop | Next Hop |
|---|---|---|---|---|---|---|---|
| AP1 | AP1 | AP | xx | 11 | 2 | eth0 | FN1 |
| FN1 | FN1 | FN | xy | 11 | 1 | eth0 | FN1 |
| MN3 | MN3 | MN | xz | 6 | 1 | eth1 | MN3 |
| MN4 | MN4 | MN | yz | 6 | 1 | eth1 | MN4 |
| MN1 | MN1 | MN | yy | 11 | 2 | eth0 | FN1 |
| MN2 | MN2 | MN | zz | 11 | 2 | eth0 | FN1 |

The neighbor table exchange mechanism only takes place between FNs and APs. The MNs are not involved in the routing updates. Each FN broadcasts its table on the following channels: 1) Channel of its interface 0 (interface towards the APs/other FNs), 2) Channel of its interface 1 (interface towards MNs/other FNs) and 3) Channel of every entry for FN and AP in its table (excluding those from 1 and 2). The actual entries can be maintained in a sorted order (either according to link metrics, hop count, or AP's first etc). For example, as shown in Table 2.5, the cost to AP at FN2 is the cost to destination advertised by FN1 incremented by one hop. If all the data flow is from the MNs to the APs and no inward data flow towards the MNs, then the FNs may exclude all the entries of node type MN while exchanging information with the neighbors. Not shown in the table is a sequence number associated with every neighbor update from a node in order to prevent routing loops.

Once the nodes have been discovered, re-discovery is done on a periodic basis. The interval depends on: a) Network dynamics (node mobility, channel conditions, node failure), b) Routing updates that show a particular path as unreachable (this is a trigger-based mechanism) and c) Power consumption and connectivity requirements (listening to beacons consumes power and may not be necessary if we do not require connectivity; on the other hand, if connectivity is more important than power consumption, we might want to increase discovery frequency)

| Topology | Flat | Mesh |
|---|---|---|
| No. of APs | 1 | 2 |
| No. of FNs | none | 4 |
| No. of MNs | 100 | 100 |
| No. of channels | 1 | 11 |

| Area | 1000m × 1000m |
|---|---|
| No. of MNs | 100 |
| Packet size | 64 bytes |

Table 2.6: SOHAN evaluation: Simulation parameters

## 2.6 SOHAN: Performance evaluation

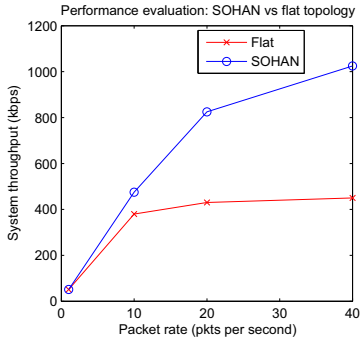



Figure 2.15: Performance evaluation: SOHAN vs flat topology

In order to enable a systematic evaluation of the SOHAN architecture and the potential benefits over flat network models, the bootstrapping and discovery protocols were implemented both as extensions to the ns-2 simulation model as well as a prototype Linux implementation. We next discuss the details of the simulation and prototype model.

### 2.6.1 Simulation Results

Using extensions to ns-2 simulator, the forwarding node with dual interfaces was implemented along with the support for multiple channels [62]. The simulation parameters are described in Table 2.6. In both cases, the traffic originates at the MNs and terminates at the APs. In the hierarchical case, there are four FNs and 2 APs that run BEAD with AODV to route traffic over 11 channels at a data rate of 1 Mbps. For a fair comparison with flat topology using a single channel, a normalized data

rate of 11 Mbps for the flat topology is used. As seen in Figure 2.15, the system throughput saturates at $\sim$ 450 kbps for the flat case, whereas the system supports almost $\sim$ 1Mbps for the hierarchical case.

### 2.6.2 Experimental validation

Using the ORBIT testbed, we compared the performance of "flat" versus hierarchical topologies with pre-configured routes and channels. We created two topologies (as shown in Figure 2.16), to represent the flat multi-hop mesh network and a hierarchical architecture with FNs. Each topology has two clusters with 20 MNs, 4 FNs and 2 APs. Each run has 20 users generating increasing offered loads in steps of 0.75Mbps (from 0.75 - 3 Mbps) towards the sink (AP). These flows represent a few users who are trying to access the Internet using a gateway or an aggregation of traffic flows from several such individual users. We measure the total system throughput, average delays and packet loss for both the scenarios under increasing offered loads.

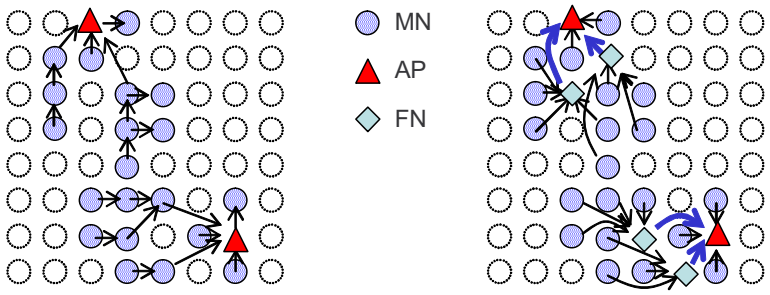
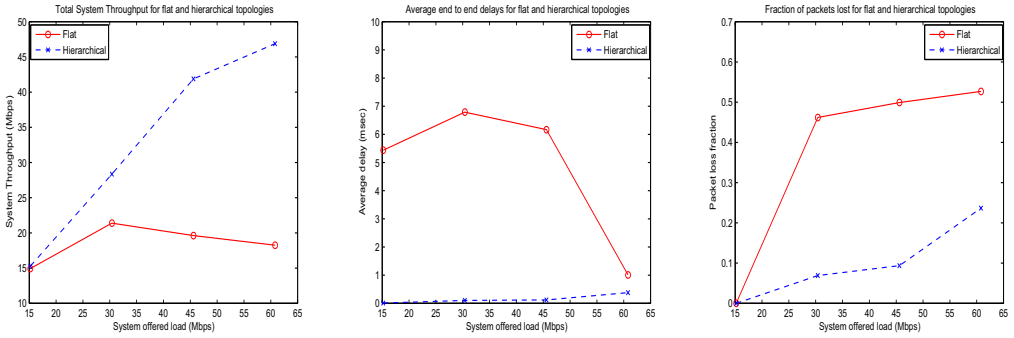



Figure 2.16: a) Flat topology b) corresponding hierarchical topology emulated on ORBIT grid

As seen in Figure 2.17, the hierarchical system is able to support upto 50 Mbps system throughput as compared to flat ad-hoc network model which saturates at 20 Mbps. These results are indicative of the advantages of the proposed hierarchical structure with self-organizing discovery and routing protocols. In [68], we have examined an application of this model for metropolitan area




(a) System throughput (b) Average delay (c) Packet loss

Figure 2.17: SOHAN: Performance improvement of hierarchical network over flat network

mesh networks of particular interest for rapid deployment of broadband Internet access. A system model for an 802.11 metro-area mesh is developed and used to estimate capacity, performance and cost for sample scenarios. It can be seen that with a right mix of APs and FNs, the SOHAN architecture scales well for a given coverage area with a 2.5 times performance improvement over flat network topologies, and can provide a viable alternative technique for broadband wireless Internet access.

## 2.7 Experiences in the design of forwarding nodes for SOHAN

We next investigate some of the important criteria in the physical design of multi-radio forwarding nodes. Based on our experimental observations, compact designs for FNs (with antennas in close proximity to each other) may adversely impact the performance of flows and may negate some of the performance benefits of using multiple orthogonal channels assumed in simulation and analytical studies. We believe this to be due to near field coupling (also reported in [47, 69]) and study the effect of varying a number of parameters such as channel separation, transmit power levels as well as physical separation between antennas.

### 2.7.1 Experimental methodology

The interesting observation regarding the coupling effect was made during experiments to measure the impact of using orthogonal channels at the FN. The original experiments were done to measure the throughput and packet loss by using multi-hop forwarding using an FN with two wireless interfaces operating on different channels and compare the results to those obtained by using a single channel and a single interface at the FN. This is shown in Figure 2.18

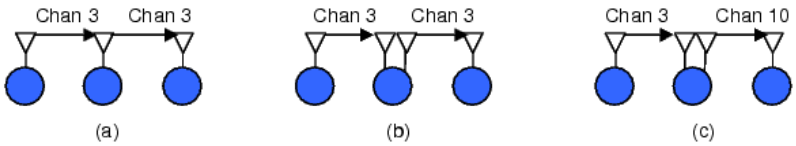



Figure 2.18: a) Baseline: SISC b) DISC c) DIDC

**Single interface, single channel (SISC):** In the baseline scenario, all the wireless interfaces were configured on the same channel (channel 3) and the FN used a single interface to forward packets from the source to the destination.

**Dual interface, single channel (DISC):** This scenario was similar to SISC, except that the FN used two interfaces; one to receiving incoming packets from the sender and the other to forward these packets downstream to the destination.

**Dual interface, orthogonal channel (DIDC):** In the third case, the forwarding node used two interfaces operating on mutually orthogonal channels (channels 3 and 10 respectively). The goal of this experiment was to study the performance improvement achieved by using multi-frequency forwarding using orthogonal channels that would enable simultaneous transmission and reception at the forwarding node. In each run, the offered load was increased from 1 Mbps to 10 Mbps using 1024 byte UDP datagrams and the resulting throughput and packet loss were measured. All the experiments were run for duration of 60 seconds using transmit power of 50 mW.

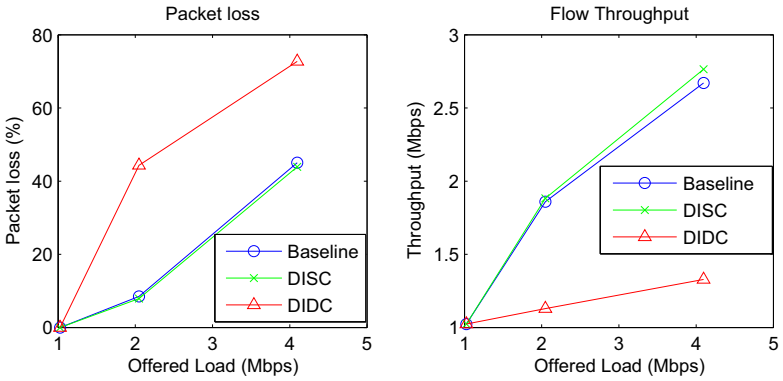



Figure 2.19: a) Packet loss b) Throughput for the three different scenarios

As seen in Figure 2.19(b), the observed throughput for the SISC case is around 3 Mbps. This is because only one of the two links can transmit at a time since it competes with the other for access to the channel. For the DISC case, we note that having two separate interfaces does not improve the throughput since the interfaces still operate on the same channel and hence they still have to share the medium.

Not surprisingly, the throughput and the packet loss curves are similar to the earlier case. The results appear slightly worse than the first case which can be attributed to the fact that in this case, the packet arriving on the incoming interface now has to be copied into the buffer of the other interface after a route lookup which accounts for the additional overhead.

For the third scenario, the observations were quite surprising. We expected to see much higher throughput, since there were two interfaces operating on orthogonal channels, thereby allowing simultaneous transmission and reception at the forwarding node. However, despite a channel separation of 7 channels (∼35MHz), the throughput is considerably lower than the earlier scenarios. Subsequent experiments with transmit power, frequency and physical separation reveal the reasons for this behavior.

### 2.7.2 Effect of transmit power control

We conjecture that the throughput degradation was attributed to the coupling effect due to antenna proximity. We further investigated the above effect by lowering the transmit power of the interfaces at the forwarding node from 50 mW to 20 mW and then to 1 mW. As expected, we observed (Figure 2.20) that the throughput degradation caused by the *front-end coupling* effect reduces when the transmit power is reduces from 50mW to 1mW.

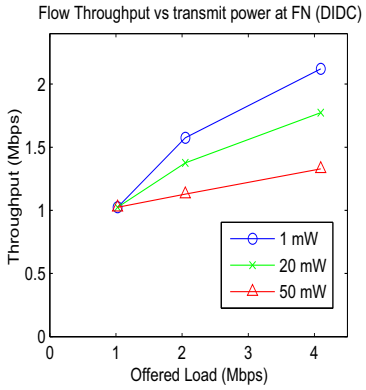



Figure 2.20: Effect of transmit power on the coupling in terms of throughput

### 2.7.3 Effect of transmit power control, frequency and spatial separation

To further investigate the effect of frequency separation, we conducted the following detailed experiments varying the transmitter power levels as well as the channel separation between the two interfaces of the forwarding node. In the first case (Figure 2.21(a)), the forwarding node had two interfaces, both operating on channel 1. The second interface was moved away (in frequency) one channel at a time until it operated on channel 11. This was repeated for three different transmitter power settings of 1 mW, 20 mW and 50 mW. For each setting, we measured the packet loss for an offered load of 4 Mbps using 1024 byte UDP datagrams. The separation between the two interfaces of the forwarding node was about one inch. The second case (Figure 2.21(b)) was similar to case 1

except that the antennas of the two interfaces on the forwarding node were physically separated by 12 inches using an external connector on the wireless card attached to external antenna.

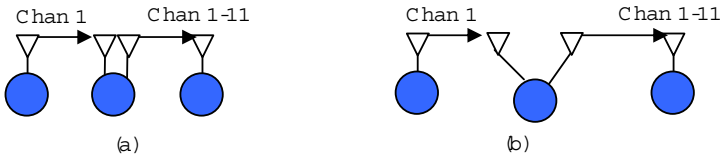
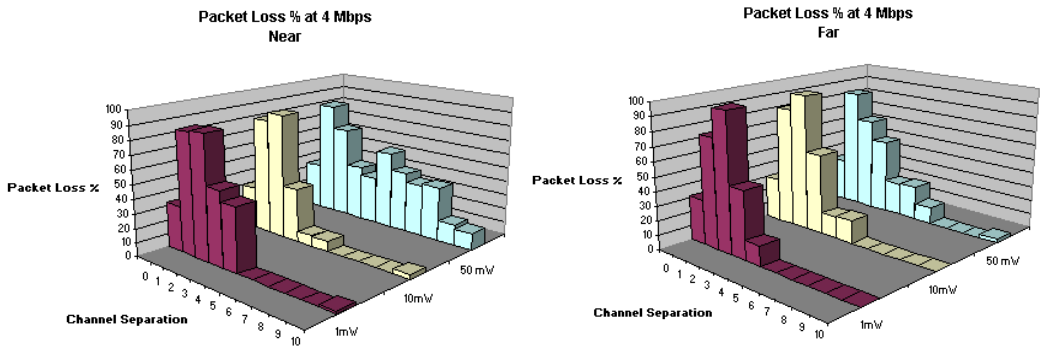



Figure 2.21: Experimental setup



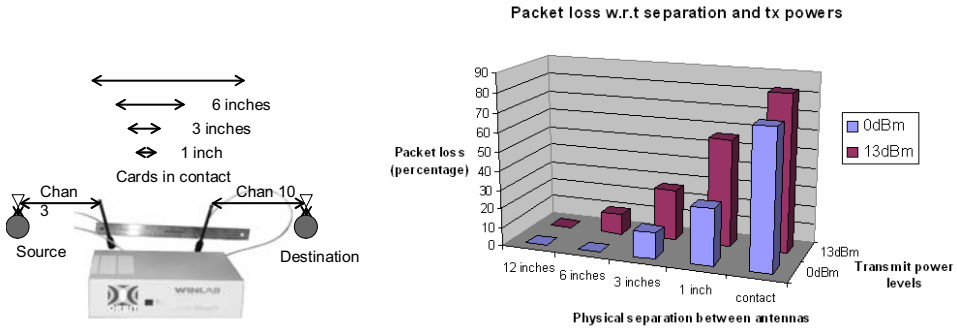

(a) Antennas near (b) Antennas separated

Figure 2.22: Packet loss at 4 Mbps offered load with antennas close and spaced apart

The results show the packet loss at an offered load of 4 Mbps. The packet loss is plotted as a function of channel separation between the two interfaces and the transmit power levels of the sending interfaces. Figure 2.22(a) corresponds to the first case with the antennas close together. We see that as the channel separation between the two interfaces on the forwarding node increases, the packet loss initially increases (due to the carrier sensing effects described in Figure 2.4) and later drops when the frequency separation is greater than four channels (20 MHz). An interesting trend to note is the effect of reducing the transmit power levels from 50 mW to 1 mW. At 50 mW transmit power levels; the packet loss is still significantly high even at a channel separation of greater than 6

or 7 channels (which are completely non-overlapping). However, this packet loss drops as we reduce the transmit power and hence the induced coupling. Figure 2.22(b) corresponds to the second case with the antennas separated by a foot. As seen in the figure, after physically separating the antennas, the coupling reduces significantly (even at transmitter power levels of 50 mW) and packet loss drops significantly.

### 2.7.4 Effect of transmit power control and physical separation




(a) Experimental Setup (b) Packet loss

Figure 2.23: a. Experimental setup b. Packet loss with physical separation of antennas

We next investigated the packet loss behavior as a function of the physical separation between the two antennas on the forwarding node. It is known that electromagnetic waves propagate spherically from the source and at a distance D from the source, the received power $P_D$ is given by $P_D = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 D^2 L}$ where $P_t$ = transmit power, $P_D$ = received power at a distance D, $G_t$ = transmitter antenna gain, $G_r$ = receiver antenna gain, $D$ is the separation in meters, $L$ is system loss factor and $\lambda$ is the wavelength. This equation holds when the distance D is large. However, at shorter distances, the planar approximation for radiated waves does not hold true and this region is called the near-field of the antenna. Typically, this separation between the near and the far fields is given

by the formula $R = \frac{2D^2}{\lambda}$ [70]. The near field distance is about 10 inches for 802.11b frequencies for an antenna length of $\sim$5 inches.

In our setup, as shown in Figure 2.23(a), the antenna separation was gradually reduced from 12 inches all the way until the antennas were in physical contact with each other. The total end-to-end packet loss was recorded at an offered load of 4 Mbps using 1024 byte UDP datagrams. This was done for two different transmitter power levels, 1 mW and 20 mW.

Fig 2.23(b) summarizes the packet loss w.r.t. reduced separation of the cards on the forwarder. Note that we are operating at distances that are close to the boundary distance (10 inches) between the near and far fields. The resulting packet loss increases as the separation between the cards is reduced until it reaches nearly 70% when both the antennas are in physical contact with each other. These results confirm the coupling effects in the near field when two antennas are in close proximity. Our observations motivate careful antenna placement considerations in the design of forwarding nodes with multiple radios.

## 2.8 Summary

In this work, we have proposed a novel three-tier hierarchical architecture that yields improvements in scalability and system performance as compared to conventional flat ad-hoc networks. This includes novel protocols for bootstrapping and discovery assisted topology formation to enable hierarchical self-organization. A proof-of-concept prototype was developed for evaluation of protocol design options and validation of system performance. Experimental results show a 2.5x improvement in system throughput over flat ad-hoc networks and are fairly consistent with predictions from simulation. These are indicative of the advantages of the proposed hierarchical structure with self-organizing discovery and routing protocols. Design considerations for dual radio FN have also been identified. In SOHAN, we have used default 802.11 MAC and hop-based routing techniques in

the hierarchical self-organization. We next focus on improve the performance of a multihop wireless network in terms of overall end-to-end packet delay and throughput by looking at fundamental inefficiencies associated with existing MAC and routing techniques.

# Chapter 3

# Interface contained forwarding architecture for multihop wireless networks

## 3.1 Introduction

In this chapter, we investigate the basic inefficiencies associated with multihop packet forwarding over the 802.11 DCF MAC protocol and propose an Interface-contained forwarding architecture (ICF) which helps to establish cut-through paths between the source and destination. This approach mitigates the self interference problem in multihop data transfers by reducing channel idle times as well as addresses the route lookup latencies at each hop. We evaluate the performance of our proposed ICF architecture as compared to 802.11 DCF mechanism using simulation-based evaluation.

## 3.2 Motivation: Existing packet transfer in multihop wireless networks

Our work is motivated by the observation that the action of packet forwarding undertaken by an intermediate node in a multi-hop wireless network is significantly different from the corresponding operations performed in a wired network. In a wired network, a router typically has at least two physical network interfaces, with the forwarding functionality consisting of receiving a packet over one physical interface and subsequently sending it out over a second interface[1]. In contrast, in a wireless network, a node with a single wireless interface may act as an intermediary for two nodes

[1] In high-end routers/switches, a packet is transferred from one interface to another via dedicated switching fabric, while in software-based routers, an incoming packet is processed by the CPU before subsequent transmission on the outgoing interface.

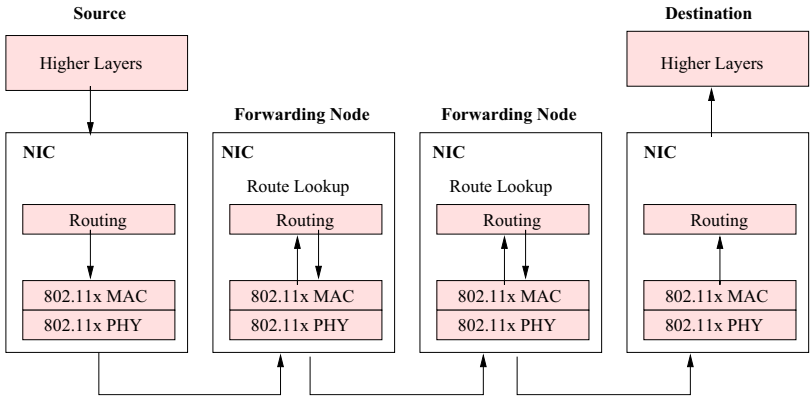



Figure 3.1: Typical packet forwarding in a multihop wireless network

that are each within the communication range but not directly within the range of each other. Thus, we see that packet forwarding in the wireless environment does not typically imply the transfer of a packet between distinct interfaces on a single host. However, all implementations of 802.11-based packet forwarding operate at the network layer, treating the process of receiving a packet from the upstream node and of sending it to the downstream nodes as two independent channel access attempts. Figure 3.1 shows a conventional implementation of software-based packet forwarding. This approach involves the reception of a packet on the wireless interface, transfer of the packet up the host's protocol stack to the IP layer where a routing lookup is used to determine the IP (and MAC) address of the next hop, and subsequent transmission of the packet using the same wireless interface to the MAC address of the next hop. This mechanism introduces two forms of latency in the multi-hop wireless forwarding process that are independent of all the other 802.11-related drawbacks enumerated earlier: 1) latency related to independent channel accesses required for each hop along the path and 2) latencies associated with interrupt-handling, packet copying and route lookup at each forwarding node. These latencies can impact the performance of delay-sensitive applications over multihop paths. To further understand the source of these latencies, we look at multihop packet forwarding using the 802.11 DCF mechanism.

### 3.2.1 Basic 802.11 DCF packet transfer

In this section, we briefly explain the 802.11 Distributed Coordination Function (DCF) contention resolution mechanisms commonly employed in multi-hop ad-hoc networks. Each node essentially acts a peer to all nodes within its transmission range. To avoid the hidden node problem, unicast communication in the DCF mode involves a 4-way handshake mechanism (shown in Fig 3.2 between sender A and recipient B using RTS/CTS exchange prior to data transmission.

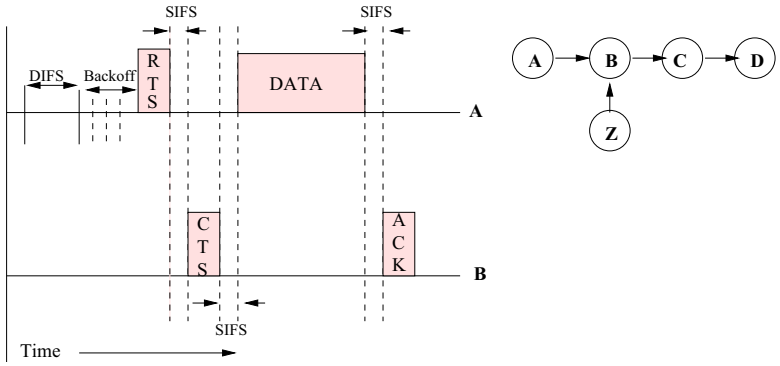



Figure 3.2: 802.11 DCF with RTS/CTS

The interaction consists of an RTS-CTS exchange that silences the neighbors in the vicinity of the sender (A) and receiver (B) respectively, followed by the data transfer and an acknowledgement. For contention resolution, 802.11 uses a timer-based exponential back-off scheme where the node selects a random back-off time in the range [0, Contention Window] (specified in terms of slots) if the channel is busy. Each time the medium becomes idle, the station waits for a DIFS and then decrements the backoff timer in units of aSlotTime. The node makes a fresh attempt at sending an RTS packet upon the expiration of the timer. Upon failure of the RTS packet, the contention window is doubled and a random timer is chosen from the new window. Each 802.11 node also maintains a Network Allocation Vector (NAV) that monitors the state of the channel. Whenever the node overhears a control packet (RTS or CTS) transmitted by a neighboring node (to some other node),

it updates its NAV appropriately to reflect the duration of the corresponding 4-way data exchange. Thus, using a combination of PHY carrier sensing and NAV-based virtual carrier sensing, the nodes co-ordinate access to the medium in a distributed manner. Even in the case of the 802.11a and 802.11g, the basic carrier sensing and channel access mechanism remains the same.

### 3.2.2 Forwarding Operation in 802.11 based multihop wireless networks

We now look at the overheads associated with a forwarding operation when using the 802.11 MAC in a multi-hop wireless environment. The upstream node (node A) sends a data packet to the forwarding node (node B); which then forwards the packet to the downstream node (node C), as shown in Fig 3.3.

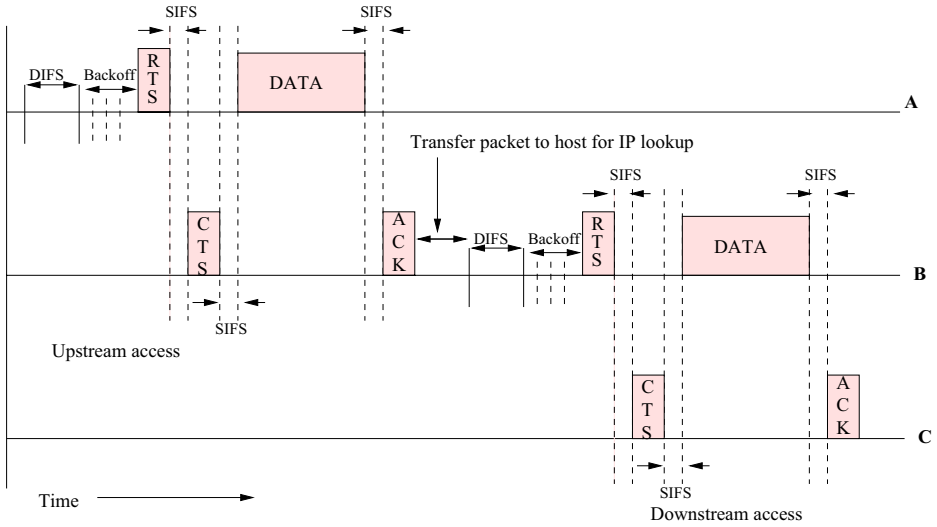



Figure 3.3: Packet forwarding operation in existing 802.11 based multihop networks

After the IP lookup function, host A determines that B is the next hop of the DATA packet, and the packet is transferred to A's NIC. The MAC implementation on A's NIC then performs a 4-way handshake (including any backoff timer-based countdown that may be needed to gain access to the channel) to forward the packet to B's NIC. At B, the packet is transferred from the device to the

main memory either using DMA or PIO (Programmed I/O) techniques, and the host CPU is notified (e.g. via interrupts or soft IRQs) [71]) for further processing of the packet. The host software (IP protocol stack) typically queues up the packet in a transmission queue and selects packets for transmission based on a scheduling algorithm (typically, FIFO). When this packet reaches the head of the queue, the same steps as those executed at A, are taken, e.g. route lookups to determine the IP address and then the MAC address of the next hop (C), inserting the MAC-layer header (corresponding to next hop C) and transferring the packet to the NIC[2]. This packet is now treated as an independent data transfer between the nodes B and C; accordingly, B performs the usual backoff timer countdown before initiating an RTS-CTS-DATA-ACK exchange with C. Once this handshake is successfully completed, the packet is received by C's NIC, at which point the whole forwarding process is repeated. As with the initial data transfer (from A to B), the NAV of node A is blocked (by the RTS sent by B) for the entire duration of the 4-way exchange between B and C. Thus, as the path length increases, the packet latencies over multiple hops may be quite high.

## 3.3 Interface contained forwarding (ICF) architecture

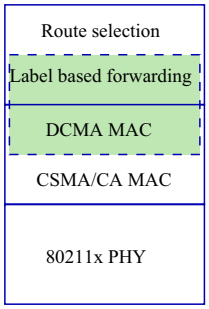



Figure 3.4: Overview of proposed enhancements to the protocol stack

Our primary goal is to define an architecture and protocol for cut-through interface contained

[2] [72] have benchmarked the sources of latency in typical packet handling operations on an MPI architecture using 20 byte packets to about 100$\mu$s for NIC-host-NIC (excluding route-lookups and related overheads). Even though these values may be OS and driver specific, we use similar values to the ones described.

forwarding and to evaluate its effectiveness compared to basic DCF over multihop networks. To realise this, we first propose an architecture for a forwarding node in a multi-hop wireless network that shifts the packet forwarding functionality away from the host processor to the wireless network interface card (NIC). This is done by combining medium access control (MAC) for packet reception and subsequent transmission with address lookup in the interface card itself, using fixed-length addressing labels in the MAC control packets. Thus, our focus is on MAC and forwarding enhancements to the existing protocol stack as shown in Figure 3.4.

### 3.3.1 Label based forwarding

In this section, we describe the label-based forwarding approach that reduces the latency associated with the NIC-host-NIC interaction at the forwarding node. More importantly, eliminating the NIC-host-NIC interaction enables us to subsequently perform atomic packet-forwarding at the MAC layer, eliminating the more significant latency component associated with independent channel accesses in 802.11. To support label-based forwarding, the network interface card (NIC) is enhanced to store a label-switching table, consisting of an incoming MAC address, an incoming label, an outgoing MAC address and an outgoing label. Figure 3.5 shows a schematic diagram of the interaction between the host software and the enhanced NIC that contains the label-switching table.

The information needed to perform NIC-resident lookups can be established offline using a separate label-distribution algorithm. The choice of the actual mechanism for label distribution does not affect the performance of the ICF architecture. A label itself can be similar to an MPLS label [73]. For label distribution, existing routing protocols such as AODV [9] may be adapted or distribution mechanisms such as LDP [74] may be used. This allows packet forwarding to be confined entirely to the NIC, which matches the label of an incoming packet with an entry in a data structure to determine the MAC address of the next hop node and the label to be used for that hop.

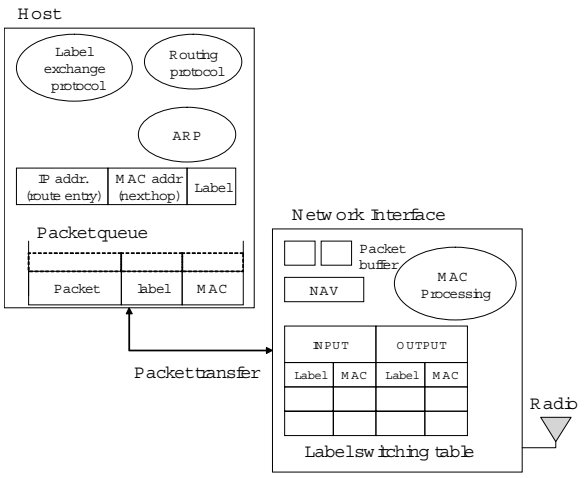



Figure 3.5: Host and NIC components for packet forwarding using labels

As in basic MPLS, labels are associated with routes or destinations, i.e. at any node, all entries in the label switching table that refer to the same route (the same path to the same destination) will share the same outgoing MAC address (of the next hop) and outgoing label. For example, let an entry in the switching table of B be (A, $L_{AB}$, C, $L_{BC}$). This means that any packet received at B from A with a label $L_{AB}$ will use C as the next downstream hop with a label $L_{BC}$[3].

The combination of the outgoing label $L_{BC}$ and the MAC address of the next hop node C, essentially defines a specific route to a destination, say D. If B has another neighbor, say Z, which uses B to reach D as well, then there will a corresponding entry in the label-switching table (Z, $L_{ZB}$, C, $L_{BC}$). The number of distinct outgoing labels is equal to the number of destinations in the network. It should be noted that each label is unique only to a single hop, and the same label may be re-used by different nodes of the network. We shall see that the label information is not needed in the DATA packets, but is carried only in control packets such as the RTS and CTS frames. This is possible because the MAC protocol reserves a time duration (via control packets) during which a forwarding node can expect to receive a DATA packet.

[3]The MAC address itself cannot be used as a label, since packets that are received at B need to be further distinguished based on their individual destination. Thus, two identifiers are needed, one for the next hop node and the other for the eventual destination

### 3.3.2 Data-driven cut-through MAC: DCMA

To reap the full benefits of the optimized forwarding process, it is also necessary to define an efficient medium access protocol for packet forwarding. We propose an atomic channel access scheme that pipelines the reception of a packet from an upstream node and the subsequent transmission to the downstream node, to avoid the overhead of separate channel accesses on the upstream and downstream links. Without these enhancements, a packet would need to be buffered at the NIC between the two separate channel accesses thereby nullifying the performance benefits achieved by the elimination of the routing lookups.

We present a simple modification of the 802.11 contention resolution scheme, called Data-driven Cut-through Multiple Access (DCMA) that provides preferential access to expedite forwarding, using a modified ACK/RTS control packet. Both of these enhancements work in tandem: to exploit the "cut-through" capability of DCMA, the NIC must be capable of determining the identity of the next-hop node from the signaling information in the contention resolution phase (without transferring the packet from the NIC to the host CPU and invoking a routing table lookup).

Our proposed MAC scheme is based on enhancements to the IEEE 802.11 Distributed Coordination Function (DCF) mode of channel access and follows the associated 4-way handshake involving RTS-CTS prior to data transmission. We term this scheme as Data-driven Cut-through Multiple Access (DCMA). DCMA attempts to replace the two distinct channel accesses, upstream and downstream, with a combined access. The reservation for the downstream hop (B to C) is attempted only after successfully receiving the DATA packet from the upstream node (A). The advantage is that a downstream reservation is made only after the upstream channel access has been granted and the packet reception from the upstream node is successful. Accordingly, as shown in Figure 3.6, DCMA combines the ACK (to the upstream node) with the RTS (to the downstream node) in a single ACK/RTS packet that is sent to the MAC broadcast address.

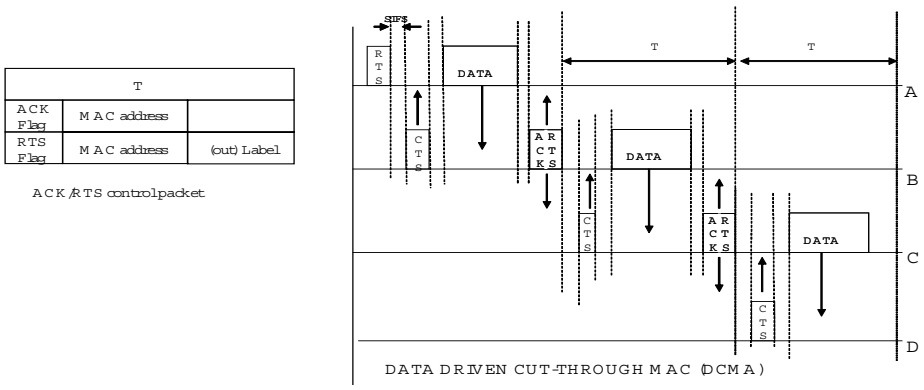



Figure 3.6: DCMA Timing Diagram

The payload of the ACK/RTS packet now contains the MAC address of the upstream node, A, and the MAC address of the downstream node, C. It also includes a label intended for use by the downstream node to figure its next hop. Since the downstream node (and all other neighboring nodes of the forwarding node) is assured to be silent until the completion of the ACK, piggybacking the RTS packet provides the forwarding node with preferential channel access for the downstream transmission. Before sending the ACK/RTS, the forwarding node (B) performs channel sensing to check whether the medium in its vicinity is idle. This reduces the likelihood of backoffs that might be generated at node B when its cut-through request (RTS/ACK sent to C) fails due to hidden node effects (e.g., when a currently transmitting downstream node D prevents node C from responding with a CTS) associated with the discrepancy between channel sensing and transmission ranges. If cut-through does fail; the forwarding node simply queues the packet in the NIC queue and resumes normal 802.11 channel access. DCMA requires no modification of the 802.11 NAV mechanism- a node simply stays quiet as long as it is aware of activity involving one or more of its neighbors. Any node that overhears an ACK/RTS not addressed to it merely increments the NAV by the specified time interval; this NAV increment is also performed by the target of the ACK (the upstream node).

In DCMA, the label is carried in the RTS/ACK (or RTS). In principle, the DATA field could also have carried this label, since the label lookup (to find the downstream node) is not strictly necessary

until after the DATA is received. However, by carrying the label information in the RTS, we provide the forwarding node additional time to complete the lookup (in parallel with the DATA transfer from the upstream node). Thus, DCMA provides an end-to-end pipelined transfer for the flow, assuming that all cut-through attempts are successful. IEEE 802.11e standard [26] specifies a TxOp (Transmit Opportunity) mode in which, after gaining access to the channel, the sender does not wait for the required DIFS interval between two frames. Instead, it waits only for SIFS and then transmits the second data frame. While the TxOP mechanism is used between pairs of nodes only, DCMA can be thought of as an "end-to-end" extension to this bursting mode.

### 3.3.3 Related Work

The use of MPLS (or labels) for providing fast and efficient packet forwarding on the wireless channel has not been extensively reported in literature. In [75], the authors have suggested using MPLS to support packet routing and handoff in wireless cellular networks along with the use of label merging to accommodate multiple links between a mobile node and the cellular infrastructure. To the best of our knowledge, there appears to be no prior public work in the area of devising MAC algorithms for providing label-based forwarding in multi-hop wireless networks. DCMA's pipelined mode of channel access is similar to the contention free "burst" mode of 802.11e [26]. While the contention free burst is on a per-hop basis, DCMA extends this burst by providing an end-to-end cut-through access across multiple hops along the path. Several approaches for pipelining data transmissions have been proposed in [76, 77]. More recently, in [25], the authors propose a Control Channel Based MAC ($C^2M$) which uses a combination of advanced channel reservation and packet aggregation on the low data rate control channel to improve the efficiency of the data channel. However, these approaches are based on using out-of-band signaling for co-ordinating

data transmissions on the main channel, whereas our approach uses does not need a separate control channel. Also, based on our earlier work [78], alternative mechanisms for distributing labels amongst nodes have been considered such as [79, 80]. Additionally, in [81], the authors propose a queue driven cut-through model that performs cut-through access at an intermediate node only for packets buffered in its queue irrespective of the flow to which they belong. Thus, cut-through is enabled only when the buffer at the intermediate relays builds up.

## 3.4 Analyzing the impact on the Latency using cut-through transfers

As stated earlier, the conventional packet forwarding process results in two types of latencies, one associated with the multiple NIC-to-host packet transfers, and the other with the separate independent channel access attempts. While the overhead associated with the NIC-to-host packet transfers and route lookups will be hardware and operating system dependent, we use an upper bound of 1 millisecond at each hop to quantify the total latency in all the required operations. Consider a single data path consisting of N links, defined over the hosts $H_1$ to $H_{N+1}$. Let us consider the 802.11b standard and assume that each of the N links can sustain a raw "data" transfer rate of X Mbps (where X is one of 2 Mbps, 11 Mbps and 54 Mbps). RTS/CTS packets are sent out at the base rate of 2 Mbps for 802.11b and 6 Mbps for 802.11g while the ACKs are sent out at the data rate. By considering the additional overhead imposed by the PHY layer, we can see that for a MAC layer DATA payload of L bytes, each individual packet transfer consumes a total delay (we ignore propagation delays in our analysis) as shown in the second column of Table 3.1. The total latency to send L bytes of DATA payload at X Mbps over N hops, is given in Equation 3.1.

For DCMA, RTS packets have an additional label field (4 bytes) intended for the downstream neighbor. The ACK/RTS packet is the same as 802.11 ACK with the following additional fields: upstream node MAC address (6 bytes), label for the downstream neighbor (4 bytes), and a flag to

Table 3.1: Summary of 802.11b/g and DCMA parameters

| Event | Time(in $\mu$s) (with b/ **with g** |
|---|---|
| SIFS | 10 (**10**) |
| aSlotTime | 20 (**9**) |
| $T_{rtlookup}$ (Route lookup delay at each forwarder) | 1000 (**1000**) |
| DIFS | 50 (**28**) |
| $T_{PHY}$ = (PLCP header + long preamble) | 192 (**20**) |
| $T_{RTS}$ = ($T_{PHY}$ + 20 bytes at 2Mbps) (802.11b) or 6 Mbps (802.11g) | 272 (**46.6**) |
| $T_{CTS}$ = ($T_{PHY}$ + 14 bytes at 2Mbps) (802.11b) or 6 Mbps (802.11g) | 248 (**38.6**) |
| $T_{ACK}$ = ($T_{PHY}$ + 14 bytes at X Mbps) | 192 + 192/X (**20 + 192/X**) |
| $T_{DATA}$ = $T_{PHY}$ + 8*(L+36 byte MAC hdr)/X | 192+8*(L+36)/X (**20+8*(L+36)/X**) |
| $T_{backoff}$ = 0.5*$CW_{min}$*aSlotTime | 310 (**67.5**) |
| $T_{RTS\_DCMA}$ = ($T_{PHY}$ + 20 bytes RTS + 4 byte label) at 2Mbps) | 288 (**52**) |
| $T_{ACK\_RTS}$ = ($T_{PHY}$ + 14 byte ACK + 1 byte flag + 4 byte label + 6 byte MAC addr at 2Mbps) | 292 (**53.33**) |

indicate ACK/RTS (1 byte). Now, consider the case of pipelined transfers from $H_1$ to $H_{N+1}$ using the DCMA protocol. In this case, the channel access delay is incurred only in the first host (original sender). Moreover, since ACK and RTS share the same frame (on all intermediate hops), the total latency to send L bytes of DATA payload at X Mbps over N hops is given in Equation 3.1.

$$\begin{aligned} T_{80211}(N,L,X) &= N \times (T_{backoff} + DIFS + T_{RTS} + T_{CTS} \\ &\quad + T_{DATA} + T_{ACK} + 3 \times SIFS) + (N-1) \times T_{rtlookup} \\ &= N \times \left[1294 + \frac{480 + 8 \times L}{X}\right] + (N-1) \times T_{rtlookup} \end{aligned} \tag{3.1}$$

$$\begin{aligned} T_{DCMA}(N,L,X) &= T_{backoff} + DIFS + T_{RTS\_DCMA} \\ &\quad + N \times (3 \times SIFS + T_{CTS} + T_{DATA} + T_{ACK\_RTS}) \\ &= 658 + N \times \left[762 + \frac{8 \times L + 288}{X}\right] \end{aligned} \tag{3.2}$$

We plot the latency for a 7-hop chain for DCMA and 802.11 with different data rates (2 and 11 Mbps) and different packet sizes (80 bytes and 1536 bytes) in Fig 3.7(a). Additionally, latencies

using 54 Mbps data rate provided by 802.11g/a cards and the appropriate interframe spaces are calculated using similar analysis. We also plot the % improvement in latency of DCMA vs 802.11 in Figure 3.7(b). The percentage improvement is calculated as $100 * (\frac{Latency_{80211} - Latency_{DCMA}}{Latency_{80211}})$. At higher data rates and packet sizes, the contribution of packet transmission time to the total latency is much smaller compared to the channel access latency. Hence, by reducing the channel access latencies, DCMA will provide a significant improvement in end-to-end latency, especially as wireless technology evolves to support higher and higher data rates. As seen in the Fig 3.7(b), the improvement in latency for 1536 byte payload transmitted at 54 Mbps is $\approx 73\%$ as compared to $\approx 17\%$ at 2 Mbps.

## 3.5 Performance evaluation

The existing 802.11 based network adapters have partial or no access to MAC layer functionality. Most of the time-critical functions such as sending acknowledgements within SIFS interval are mainly done by the hardware. A few control knobs such as the RTS threshold and fragmentation threshold are usually exposed through *ioctl* based interface. The current ORBIT experimental testbed does not support fully programmable MAC although this is a long-term design objective. As a result, we implemented the DCMA protocol as part of the ns-2 simulator [59] with the CMU wireless extensions [82]. We focus on three metrics: a) the throughput improvement achieved by the cut-through protocols b) the potential reduction in end-to-end latency due to the expedited MAC forwarding and c) percentage of cut-through out of the total packets received. To study the throughput and latency behavior of flows, we consider CBR UDP flows with varying packet arrival rates. The buffer size at each node is set to 50 packets. The routing tables are pre-configured with the shortest path routes to their respective destinations. Each node keeps track of the number of packets sent out and the number of cut-through acknowledgements received. The cut-through percentage is

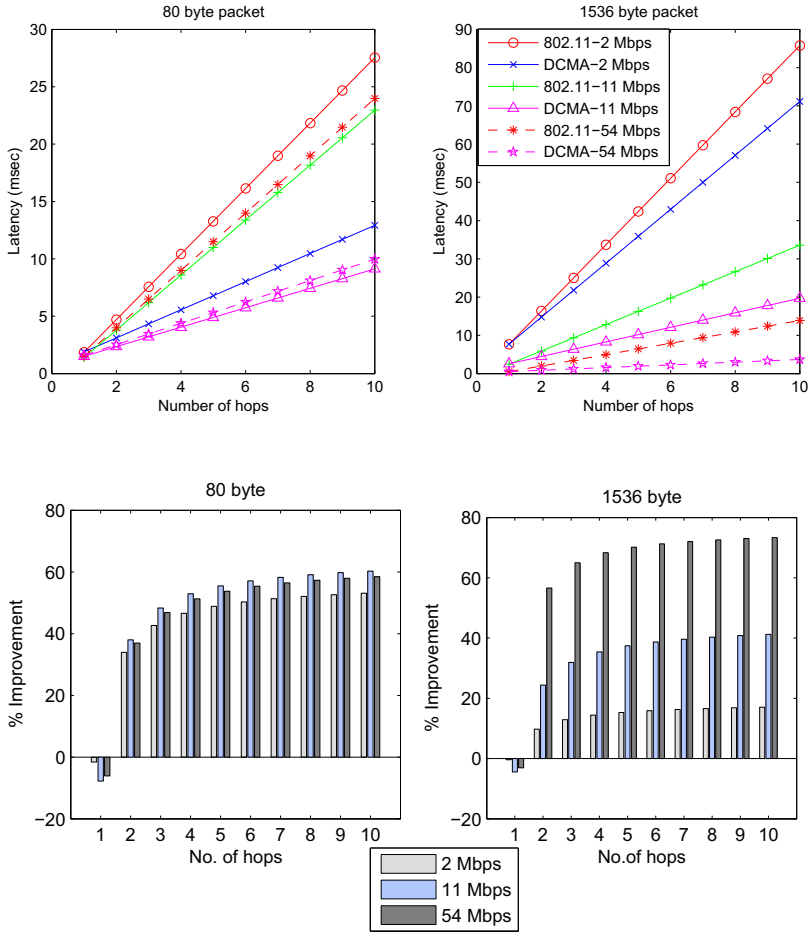



Figure 3.7: a. Latency for different packet sizes and rates b. % improvement in latency

calculated as the ratio of sum total of the cut through ACKs (ACKs for RTS/ACK-driven transmissions) received to the total number of packet transmissions (each packet transmission on a link is considered separately). The parameters of the ns-2 simulator are summarized in the Table 3.2.

### 3.5.1 Baseline scenario: single flow

In this scenario, the traffic consists of a single UDP flow between the two end nodes of the chain. Even though this scenario has only a single flow as shown in Fig 3.8(a), it helps to understand the benefits of DCMA over 802.11 under different offered loads in terms of reduced self-interference.

Table 3.2: Common Parameters for all simulation results

| | |
|---|---|
| Topology | 7 node chain and 100 node grid |
| Distance between adjacent nodes | 248 m |
| Traffic model | CBR (UDP) |
| Data Rate | 11 Mbps |
| PHY model | SINR based reception |
| Transmission/Interference Range | 250m/550m |
| RTS Threshold | 0 (always on) |
| Host processing delays at each relay for (802.11 DCF) | 1 millisecond |

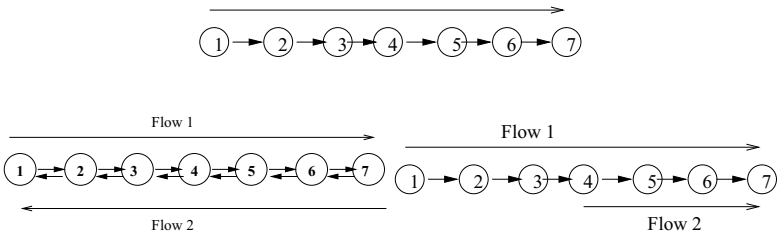



Figure 3.8: a. Single flow b. Two flows - reverse c. Two flows - same direction

The offered load was increased from 125 kbps to 1.75 Mbps in steps of 125 kbps using two different packet sizes: 256 bytes and 1536 bytes. Figure 3.9 shows the throughput and delay results for DCMA and 802.11 for two different packet sizes. We see that for both packet size, the DCMA throughput is higher than 802.11. As expected, DCMA offers almost a 50% reduction in end-to-end delay, especially at higher offered loads.

Moreover, the lower throughput for 256 byte packets is due to the proportionally larger overhead of the MAC/PHY headers. One of the most important advantages with DCMA is that the pipelined access mechanism essentially reduces the channel contention effects among consecutive packets belonging to the same flow - by allowing most packets to cut-through faster to downstream nodes, it lowers the incidence of contention-induced backoffs at upstream (hidden) nodes for subsequent packets. This intra-flow contention becomes especially more pronounced for larger packet sizes with 802.11, where the throughput actually declines from $\sim$ 1 Mbps to $\sim$750 Kbps-since each packet transmission now takes longer, each transmitting node now “holds” the channel for a longer

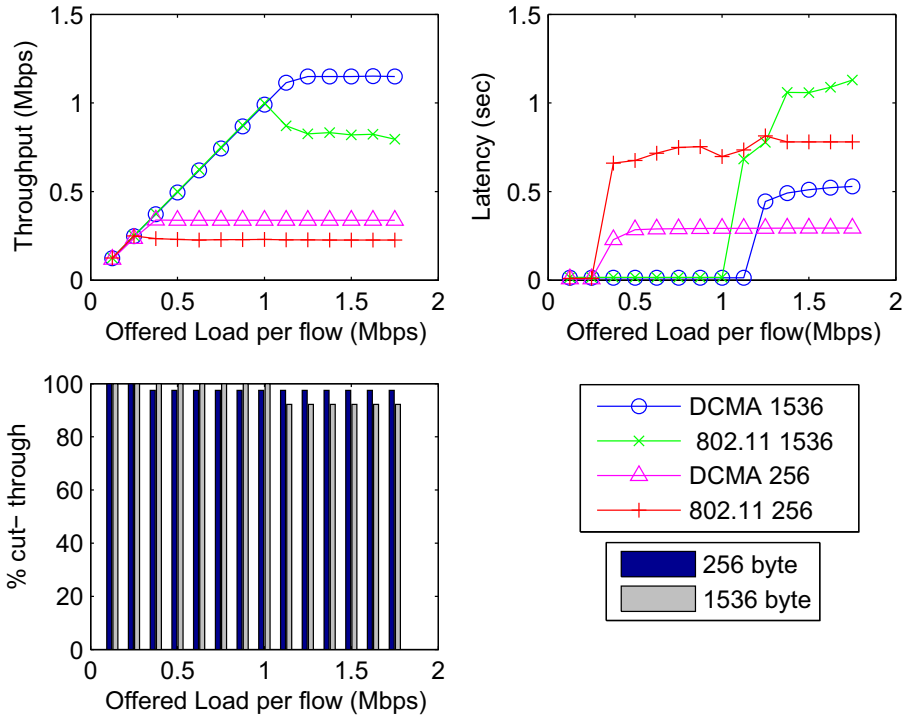



Figure 3.9: a. Throughput b. Latency and c. % cut-throughs for a simple chain topology (CBR traffic)

duration and may thus cause repeated multiplicative backoff for upstream hidden nodes (an observation also reported in [7]). At relatively low traffic rates, the packets arrive sufficiently spaced apart to avoid this problem of intra-flow contention. We expect the latency difference between 802.11b and DCMA to be much higher for higher data rates (54 Mbps) as shown in Fig 3.7(b). Fig 3.9(c) shows the percentage of cut-through packets to the total number of packets delivered from the source to the destination. For lower offered loads, there are 100% successful cut-throughs at all intermediate nodes. However, as the offered load increases beyond 1 Mbps, the packet injection rate is much higher than the cut-through delivery time, resulting in queue build up at the intermediate nodes and reduced cut-throughs.

Fig 3.10 provides a zoomed view of the average delays for each packet size. Note that many representative interactive or multimedia applications require the end-to-end latency not to exceed

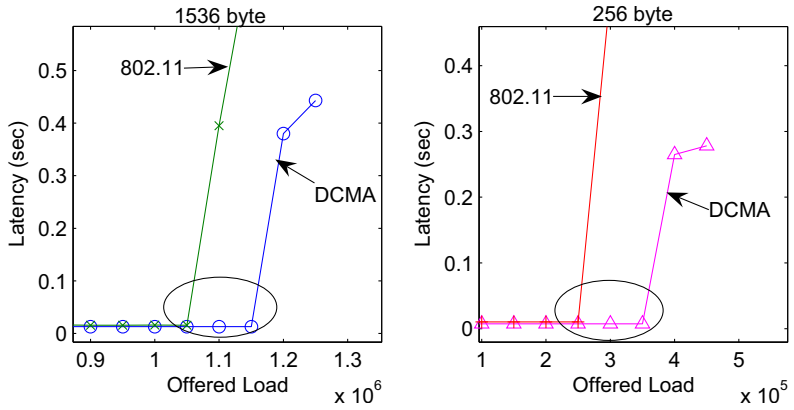


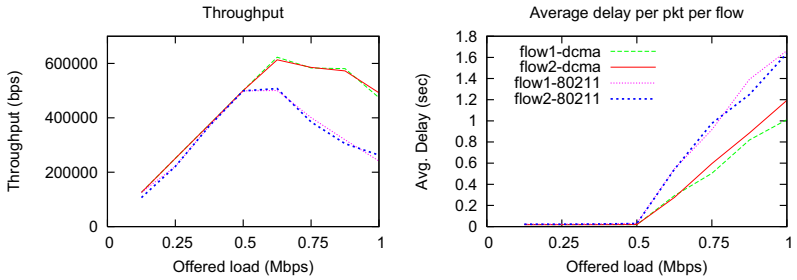

Figure 3.10: Zoomed view of flow latencies for a. 256 byte and b. 1536 byte payload

100 to 200 ms. Accordingly, we define the *operating range* of a protocol to be that where the end-to-end latency does not exceed 200 msec. As seen, DCMA extends the operating range from 1.05 Mbps to 1.15 Mbps for 1536 byte packets and from 250 Kbps to 350 Kbps for 256 byte packets while maintaining reasonably low latency.

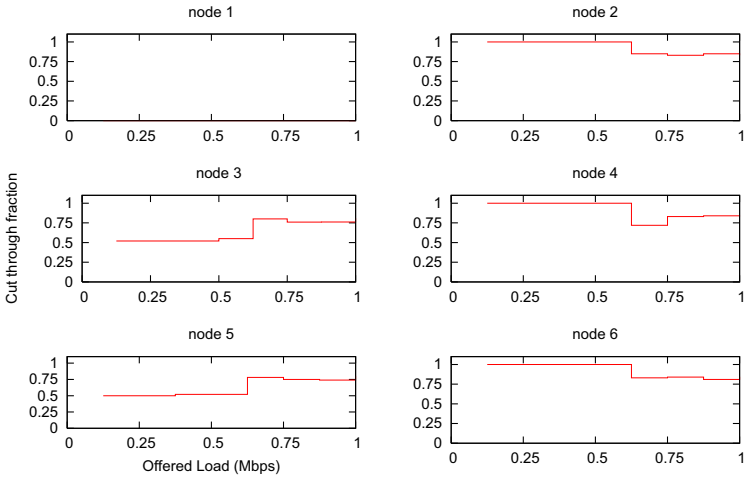
### 3.5.2 Simple chain with two flows in reverse direction

We consider two CBR flows in reverse direction (from node 1 to 7 and from node 7 to 1 respectively, as shown in Fig 3.8(b). This scenario also provides insights into the behavior of TCP traffic, which has the data flowing from the source to the destination and ACKs flowing in the reverse direction. In this case, the offered load (CBR-UDP with 1500 byte payload) is increased from 125 kbps to 1 Mbps per flow. Per flow throughputs, average packet transfer delay are recorded for both DCMA and 802.11 along with the successful cut-through transfers at each node using DCMA. This is calculated as $(\frac{No.of cut through packet sent}{No.of total packet sent})$.

It can be seen in Figure 3.11 that upto 500 Kbps offered load per flow, the network is able to sustain the offered load with DCMA supporting higher throughput than 802.11 because of efficient packet forwarding and reduced self-interference thereby being able to transfer more packets during the simulation duration. Note that the flows do interfere with each other, but the channel capacity




(a)

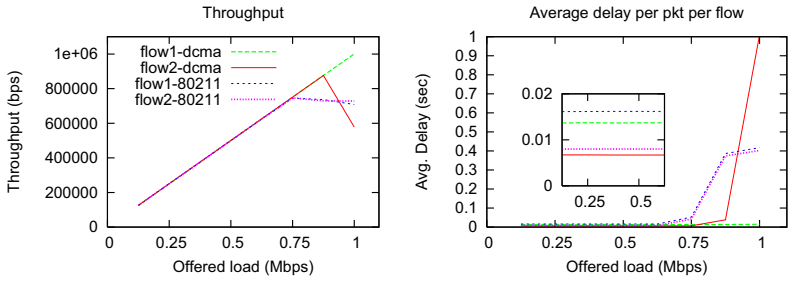



(b)

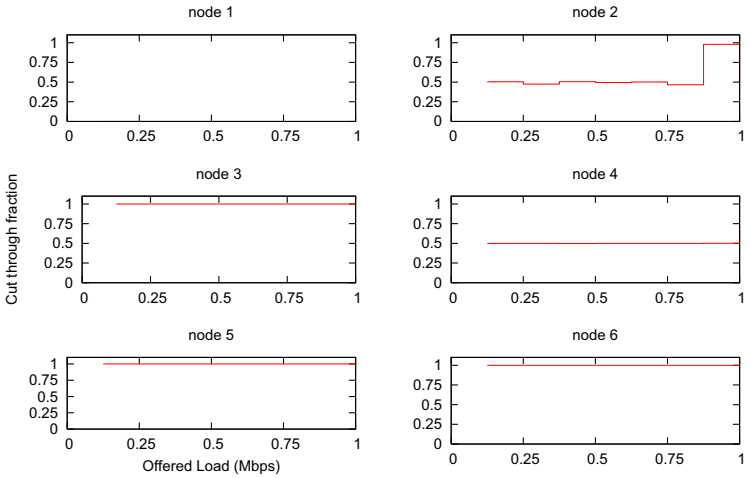
Figure 3.11: Reverse chain topology: a. Throughput and latency performance b. Successful cut through transfers at each node out of the total packets sent

is enough to sustain both flows. As the traffic load increases beyond 500 Kbps per flow there are two effects that start to dominate - self-interference between hops of the same flow due to faster packet injection rate, and mutual interference for the other flow. This manifests as reduced number of cut through transfers at the node in the middle (Node 4) (shown in Figure 3.11b.) which has to handle simultaneous cut through requests from each flow. While the 802.11 throughput and latency performs suffers considerably due interference induced collisions and backoff, DCMA is still able to support higher throughput and lower latencies as shown in Figure 3.11a.

### 3.5.3 Simple chain with two flows in the same direction




(a)




(b)

Figure 3.12: Same direction chain: a. Throughput and latency performance b. Successful cut through transfers at each node out of the total packets sent

Next, we look at a case where the two flows are in the same direction (1 to 7 and 4 to 7 as shown in Fig 3.8(c)). This scenario deals with the case when the intermediate relay nodes may also have traffic of their own to send.

It can be seen in Figure 3.12(b) that node 4 cuts through 50% of the total packets. This is because it is the source of the second flow and cannot cut through the packet it originates. An interesting observation is that even for higher offered loads, node 4 still maintains the cut-through ratio of 50%, whereas node 2 reports an increase in the number of cut through transfers. This seems

to indicate that node 4 continues to serve the flow from node 1 to node 6, thus starving its own local traffic. This is seen correspondingly in Figure 3.12a. as the throughput and latencies for the flows diverge significantly. We acknowledge this fairness issue and have proposed a heuristic approach by modifying the interface behavior to monitor if the node has any local packets pending in its MAC buffer. The interface can then probabilistically decline the cut-through attempt from other flows by rejecting an RTS/ACK request. In [81], the authors propose a queue driven cut-through model that performs cut-through access at an intermediate node only for packets buffered in its queue irrespective of the flow to which they belong. Thus, cut-through is enabled only when the buffer at the intermediate relays builds up. In our work, however, we focus on the performance benefits from efficient forwarding.

### 3.5.4 Understanding self-interference and interflow interference

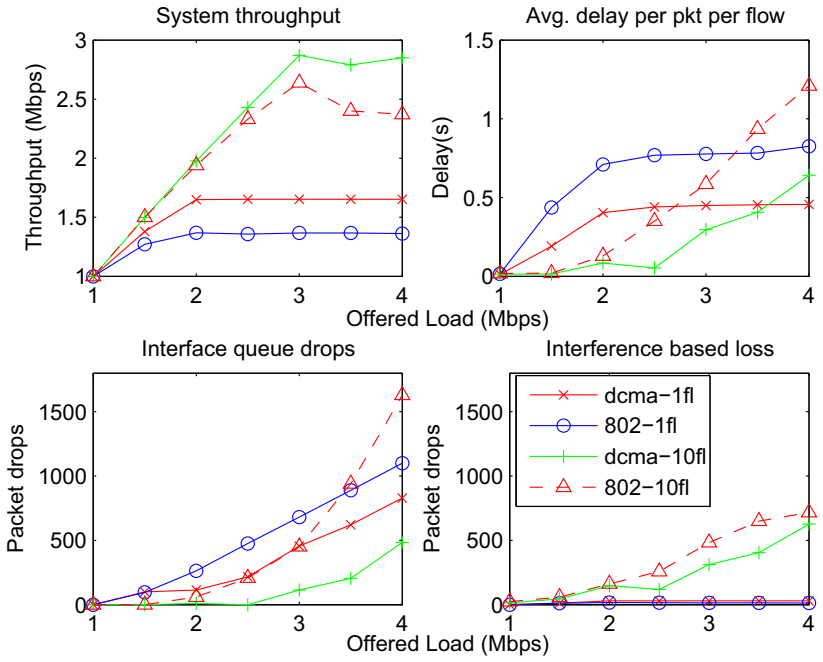



Figure 3.13: Understanding the impact of self interference and inter-flow interference on cut through transfers

After studying the chain topologies described earlier, we tested DCMA on a general $10\times10$ grid topology with randomly selected sources and destinations. The grid topology represents a combination of the scenarios described earlier (multiple flows - in the same direction, reverse directions as well as interfering flows). In order to understand the impact of self interference and interflow interference, we look at two scenarios: with one flow and ten flows with the same total offered load. The node separation is 248m so such that each node only has one node in each direction within its communication range. Total offered load for both scenarios is varied from 1 Mbps to 4 Mbps in steps of 500 Kbps. The single flow case is used to understand the impact of self interference on the behavior of DCMA and 802.11 since there is no interflow interference. The ten flow case represents a mix of both self interference and interflow interference. We also keep track of the packet losses due to buffer overflows at the source and due to collisions and interference. All the simulations are run multiple times and the results are averaged over different topologies.

**Offered loads and self-interference:** For the scenario with a single flow, the time gap between two packets should be such that the previous packet has had an opportunity to exit the interference range ($\sim$3hops) of the current packet. For DCMA, the 3-hop latency for 1500 byte packet at 11 Mbps (assuming 300$\mu$s backoff interval) is $\sim 6.2msec$ (Eqn. 3.1). This corresponds to 158 packets per second of 1500 bytes or 1.9 Mbps offered load. Similarly, for the case of 802.11, this value is $\sim 10.28msec$, corresponding to 98 packets per second of 1500 bytes (or 1.1 Mbps). Beyond these operating points, we expect to see effects of self interference where the new packet entering the system encounters interference from the previous packet still within its interference range.

As seen in Figure 3.13(a and b), for the one flow case, the 802.11 throughput saturates at an offered load of around 1.2 Mbps where as DCMA saturates around 1.6 Mbps offered load due to self interference. Similar observations are made for the packet latencies. Also, as expected, beyond the operating point of $\sim$1.5 to 2Mbps, the buffer overflows at the source increase and the throughput

saturates. Note that, there is no interflow interference in this case and hence losses due to collisions are negligible. (seen in Figure 3.13d.)

**Interflow interference:** In order to understand the impact of interflow interference, we maintain the same total offered loads for the network as before. In this case, we choose ten randomly chosen (unique) source-destination pairs with a maximum hop length of ten hops. For the case of ten flows with the same total offered load as the single flow case, it can be seen that interference and collision based losses are dominant as seen in Figure 3.13d. Beyond $\sim$3 Mbps total offered load, the interflow interference causes queues to build up at intermediate nodes resulting in increasing buffer overflows and packet drops ( 3.13c.). The overall throughput is higher than the single flow case because of better spatial reuse. Also, the packet latencies are much lower than the corresponding single flow case with the same offered load. This is because the offered load per flow in the ten flow scenario is scaled, so the effects of self interference are minimized. Also, in this operating region (upto 3 Mbps offered load), the network has enough capacity to sustain interference between flows as well. Based on these observations, we choose an system operating load of 3 Mbps which indicates a balance between an under-loaded and saturated system for further evaluations.

### 3.5.5 General grid topology: performance evaluations

Based on the preceding discussion, we evaluated the performance of DCMA and 802.11 in a general grid topology with different number of flows. We maintain the same grid topology (with 248 m. spacing between nodes) and use 1500 byte UDP payload for CBR based traffic model. The following two scenarios are evaluated. Static routes were used in these scenarios in order to focus on quantifying the performance benefits of the ICF mechanism.

**Constant total offered load, increasing number of flows:** In this case, the total system offered load is maintained at 3 Mbps. The individual flow offered loads are scaled according to

number of flows in the system and we look at 5 to 20 flows (with maximum path length = 10 hops). This results in per-flow offered loads of 600 Kbps down to 150 Kbps with increasing number of flows. As seen in Figure 3.14, DCMA latencies are significantly lower than the corresponding

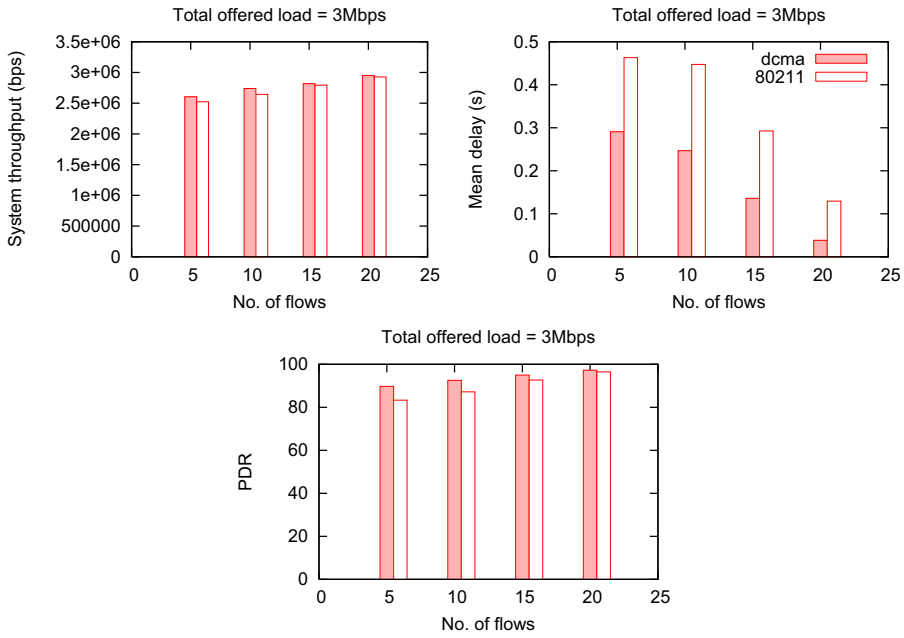



Figure 3.14: System throughput and average delay per flow in a 10×10 grid with 3 Mbps total offered load

802.11 latencies under different number of flows. With 5 and 10 flows, the scenario is dominated by self-interference due to the faster packet arrival rates. It can be seen that the efficient interface contained forwarding reduces the self-interference per flow by allowing packets to traverse beyond the interference range faster, resulting in lower latencies, higher throughputs and packet delivery rates. As the number of flows increases to 15 flows and 20 flows, note that the overall latencies are lower as compared to the smaller number of flows. While this may seem non-intuitive because increasing the number of flows implies increased inter-flow interference, it is important to note that the per flow offered loads are now scaled resulting in lesser traffic being injected into the network per flow. The interflow interference caused by flows is directly proportional to the flow activity. In

both the scenarios, DCMA results in a ∼2 times improvement in terms of packet transfer latencies over 802.11 based multihop transfer.

**Constant offered load per flow, increasing number of flows:** Next, we study the effect of increasing interflow interference, by maintaining the offered load per flow at 200 Kbps and increasing the number of flows from five to twenty. As the number of flows increases, the higher volume of traffic results in higher interflow interference. As seen in Figure 3.15, DCMA outperforms 802.11 in terms of end-to-end packet latencies under all the system loads with a 2x improvement with 20 flows.

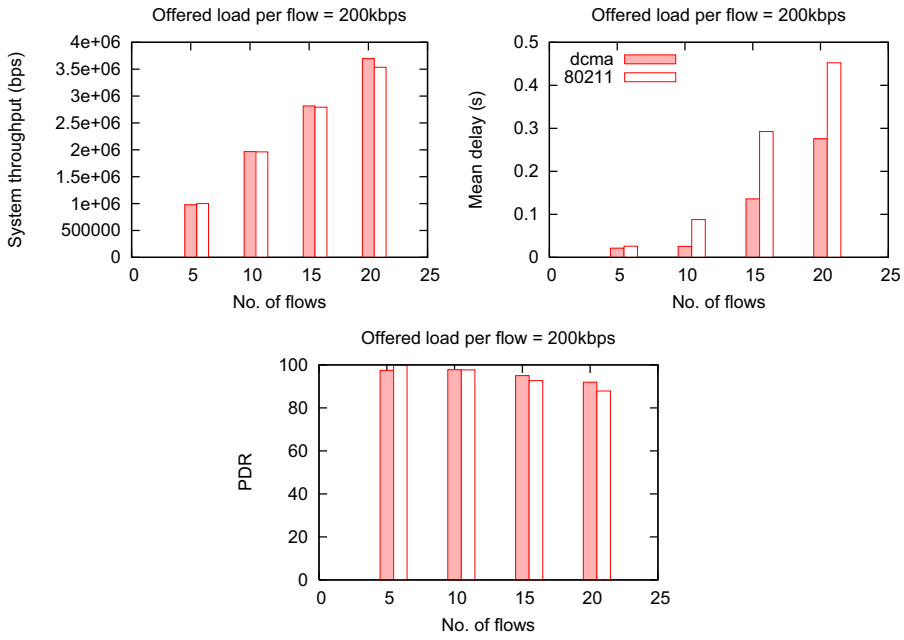



Figure 3.15: System throughput and average delay per flow in a 10×10 grid with 200 kbps per flow

Thus, our simulation studies show that DCMA outperforms 802.11 DCF in all scenarios in terms of end-to-end latencies. Specifically, the efficient interface contained forwarding reduces the self-interference amongst flows by allowing packets to traverse beyond the interference range faster, thereby increasing the flow throughput and the packet delivery rates.

## 3.6 Summary

In this work, we present an efficient interface contained forwarding (ICF) architecture for a "wireless router", i.e. a forwarding node with a single wireless NIC in a multi-hop wireless network that allows a packet to be forwarded entirely within the network interface card of the forwarding node without requiring per-packet intervention by the node's CPU. To effectively forward packets in a pipelined fashion without incurring the 802.11-related overheads of multiple independent channel accesses, we specify a slightly modified version of the 802.11 MAC, called Data Driven Cut-through Multiple Access (DCMA) that uses MPLS-like labels in the control packets, in conjunction with a combined ACK/RTS packet, to reduce 802.11 channel access latencies. Using extensive simulation studies, we evaluate the performance of DCMA and show a $\sim$2x improvement in latencies as compared to 802.11 DCF based channel accesss. In general, packet cut-through is especially useful at relatively low to moderate network loads (which is typically the case in well-provisioned wireless networks). A combination of DCMA and call admission control (so that the network load stays within specified bounds) could prove to be especially useful for relatively low-bandwidth, delay sensitive applications such as VoIP-over-wireless. Note that this scheme may also be used selectively for supporting higher priority traffic.

Also, we have assumed static route selection prior to cut through forwarding. If the paths are chosen independent of the underlying link load, then cut-through transfers may fail at intermediate nodes that are busy due to ongoing transmissions in its vicinity. Hence, an interference aware route selection in conjunction with cut through forwarding might be beneficial to reduce the interflow interference. In the next chapter, we investigate cross layer route selection enhancements that incorporate the awareness of underlying link conditions while choosing between alternate candidate paths and can be used to supplement DCMA cut-through transfers over multiple hops.

# Chapter 4

# Cross layer route selection in wireless networks

## 4.1 Background and motivation

Wireless networks are inherently different from wired counterparts due to the inherent broadcast nature over a less reliable wireless medium. They specifically pose many interesting challenges such as intra-flow and interflow interference in multihop networks, resulting from transmissions within the interference neighborhood of a receiver, variable bandwidth links due to link quality based rate adaptation as well as losses due to low SNR and collisions. In [11], the hop count based metric usage in wireless networks has been shown to be inadequate due to the longer links resulting in losses and lower data rates.

There have been many studies that have proposed and evaluated a variety of link characteristics as metrics for route selection. In [36], the authors have proposed using received signal stability as an indicator of high quality links. In [42], the authors propose the expected transmission count metric (ETX) that incorporate the effects of link loss ratios into route selection and experimentally evaluate the performance on outdoor mesh network. Awerbuch et al. [35] explored the multi-rate functionality of the MAC protocol in route selection and suggested that in a fully interfering network, MTM (Medium Time Metric) is an optimal route-selection metric to maximize end-to-end flow throughput. The work in [41] introduces WCETT (Weighted Cumulative Expected Transmission Time), which combines both ETX and MTM metrics and prefers paths with higher frequency

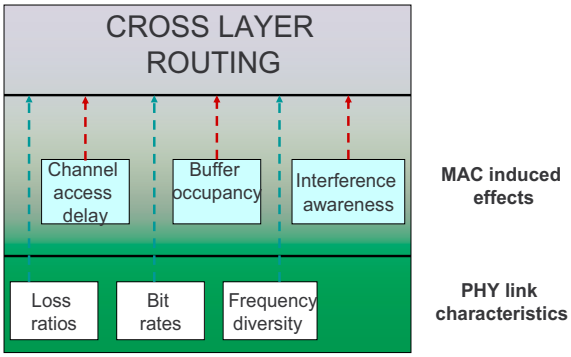



Figure 4.1: PHY and MAC parameters for cross layer route selection

diversity to achieve higher throughputs. More recently in [37], a comparative study on some of current metrics for routing algorithms was reported. The MIC metric [40] scales the WCETT metric by taking into account the interfering links on the channel and channel switching penalties, while the IAware metric [38], leverages the MIC metric by making active channel usage estimates based on ongoing traffic. In [39], the authors have suggested using a combination of PHY and MAC layer information in terms of bit-rate and media access delay for route selections. The various parameters used for cross layer optimization are summarized in Figure 4.1.

In this chapter, we first consider the impact of two critical PHY layer design components, frequency diversity and bit rate selection on the performance of an 802.11-based multihop network. Our initial results indicate that underlying PHY bit rate selection plays an important role in addition to frequency diversity for high quality path selection. We then draw attention to some of the shortcomings of existing link quality estimation techniques used by majority of the existing cross layer routing approaches described above, in terms of overhead and accuracy of estimation and propose a novel *airtime* metric for high throughput route selection. Our proposed metric is based on passive monitoring of ongoing data traffic and quantifies the *effort* (in terms of time), it takes to transmit information to a neighbor. This low overhead estimation technique avoids the active probing and inaccurate estimations of existing approaches.

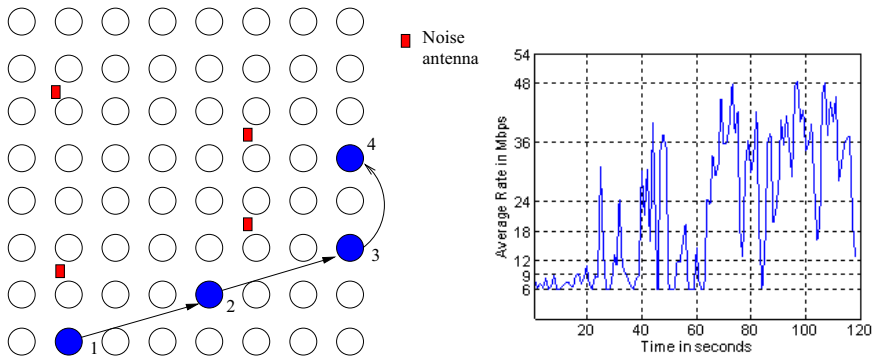
## 4.2 Impact of frequency diversity and bit-rate on path quality

In this section, our goal is to quantify how the choice of PHY rates and frequencies for a given link impact the overall throughput achieved on a path traversing these links. In particular, throughput results are provided for multi-hop ad-hoc networks with and without PHY auto-rate control and for single and multiple frequencies. The study is motivated by the fact that default 802.11-based ad-hoc networks using commercially preset auto-rate PHY and a single frequency channel suffer from performance degradations caused by link quality fluctuations and MAC layer self-interference respectively.

Using the ORBIT radio grid testbed, we evaluate the throughput of a multi-hop flow in terms of end-to-end throughput operating under noisy environment using a single channel and default PHY rate selection algorithm implemented on the commercial 802.11 hardware. This is used as a baseline case to compare the performance with the above-mentioned PHY design choices (i.e. alternative rate control methods and use of multiple channels)

### 4.2.1 Existing bit rate selection algorithms: effect of noise

We conducted initial experiments to observe the effect of noise on the average PHY rate of the links under test. All of our experiments were done on the ORBIT testbed. The testbed also includes a raw waveform generator that is connected to specifiable antennas on the grid. This generator can be remotely controlled to inject AWGN noise at a desired power level and frequency band, thereby enabling the creation of arbitrary link quality levels and related ad-hoc network topologies. Each node has two 802.11 a/b/g cards that can be used to set up arbitrary topologies. By operating the second interface on a different channel, multi-channel forwarding can be implemented where so desired. Supporting software libraries allow us to extract useful information such as RSSI, PHY rate from the device driver on a per packet granularity. Further details of the testbed functionality




(a) (b)

Figure 4.2: a. Experimental setup: Positions of nodes, links and noise antenna on the grid b. Link rate adaptation caused by noise

are described in the Appendix.

Figure 4.2a. illustrates the choice of nodes, links and the positions of the noise antennas. All the wireless cards for this experiment use 802.11a PHY. The noise generator can be controlled to inject AWGN noise at a desired power level and frequency (from baseband to 6 GHz). In this experiment, we measured the average PHY rate of each link over a 120 second interval using 512 bytes UDP datagrams at 50 packets per second from one sender to one receiver under the influence of noise at -12 dBm. The transmit power of the sender was set to 20dBm. The PHY rate automatically adjusts to channel conditions using the default auto-rate adjustment algorithm implemented in the driver/firmware of the card (Atheros AR 5212 chipset). Figure 4.2b. shows the fluctuation of the average PHY rate per second (averaged over 50 consecutive packets since we can report the selected PHY rate per packet) on one such link (1 to 2). This automatic PHY bit-rate adaptation feature is considered to be useful in WLAN systems because it permits end-users to take advantage of good-quality short-range links when available. However, one of the consequences of the automatic rate adaptation is the rapid rate fluctuation. Most auto-rate mechanisms adapt to the link quality based on packet loss measurements. The fluctuations caused by auto rate PHY algorithm may lead to

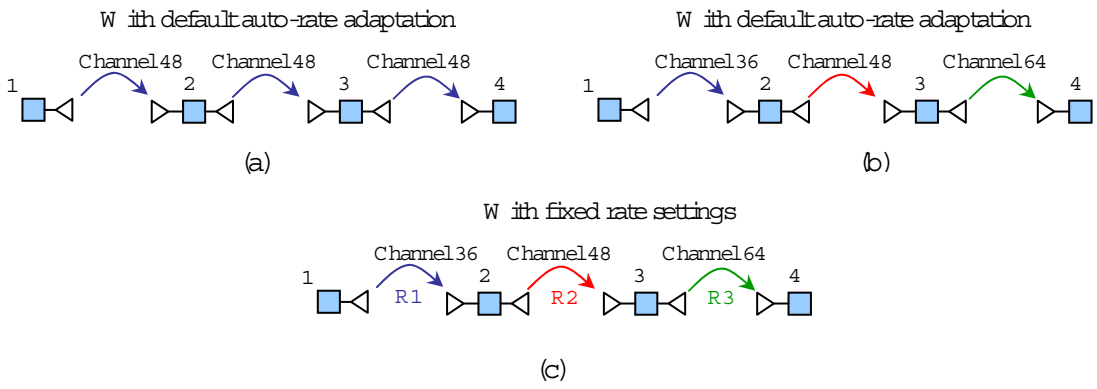



Figure 4.3: Experimental scenarios

an in-efficient utilization of the link capacity resulting in a lower end-to-end throughput. We next evaluate a multihop path under the influence of noise under different combinations of frequencies and rates assigned to each link.

### 4.2.2 Baseline Case: Single Channel with default Auto-Rate PHY

The baseline scenario represents the case where the routing protocol chooses the shortest path with no consideration to cross layer information. As shown in Figure 4.3(a), all the wireless interfaces operate on the same 802.11a channel (channel 48, with central frequency at 5.24 GHz). The transmit power of each radio is set at 20 dBm. We consider a single three-hop flow to isolate any interference from other flows and study the performance of frequency diversity and rate adaptation under the influence of noise. AWGN noise was injected using a raw waveform generator [83] at the center frequency of 5.24 GHz and different levels ranging from -18 dBm to -5 dBm. An offered load enough to saturate the channel was introduced.

Figure 4.4(a) represents the throughput achieved under increasing noise levels and the maximum is 4.2 Mbps. This is because even though there are two radio interfaces at the forwarding node, since they are on the same channel, only one of them can operate at a given time. Also, the achieved throughput decreases with noisier link conditions due to higher packet losses.

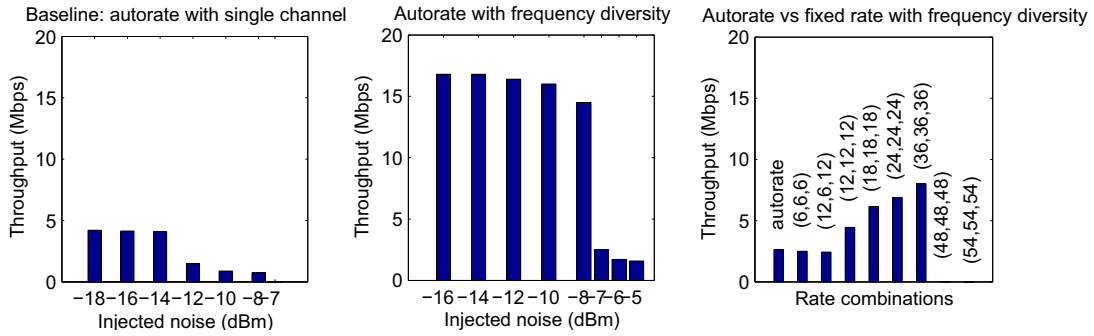



Figure 4.4: a) Autorate with single channel b) Autorate with frequency diversity c) Autorate vs fixed rate with frequency diversity at -6dBm noise on channel 48

### 4.2.3 Autorate with frequency diversity

This case represents the scenario when the routing protocol uses knowledge of the underlying frequency assignments of the paths. The selected path has each hop on orthogonal frequency so that all three hops can operate in parallel. The achievable throughput is tested under the similar noise conditions as the previous case.

In this case, the throughput increases to about 16.8 Mbps (as seen in Figure 4.4(b)), a gain of $\sim$4x for the specific flow under consideration as compared to the baseline case. This is because the noise affects only one of the links (from 2 to 3) in this case. In the previous case, there is zero throughput at -5 dBm, where as in this case, a low throughput ($\sim$2 Mbps) can still be sustained. We believe the reason for this is that the link between 1 and 2 is completely cut off in the first case, even though the other two links are still operational.

### 4.2.4 Selectable PHY rates with frequency diversity

Both previous cases use the default bit rate adaptation algorithms implemented in the driver [84, 85] for the wireless card. In most wireless cards, these algorithms are proprietary and cannot be configured. The goal of rate control algorithms is to adapt to the changing channel conditions by adjusting the modulation scheme for transmission. The main motivation behind the experiment is

to see if the rate selection strategies operate well under the influence of injected noise.

In this experiment, we fix a particular noise level (-6dBm) on the middle link and evaluate different combinations of PHY rate settings (R1, R2, R3) on the three links manually. Note that the cards support the capability to use fixed rates as specified by the user (using iwconfig [86] utility).

Figure 4.4(c) shows the results for an offered load of 8.4 Mbps. It can been seen that the default autorate adaptation performance significantly worse than some fixed rate settings tried on each link. In particular, it can be seen that the combination (36 Mbps, 36 Mbps, 36Mbps) has the highest throughput under the given noise level. This indicates that the default rate adaptation algorithms are conservative in selection of bit rates under noisy link conditions and fall back to much lower rates than sustainable ones.

### 4.2.5 Summary of observations

To summarize, our initial experiments suggest that there is a large scope for improving the system performance using a cross layer protocol framework and proper resource allocations. The information from the underlying layers regarding link bandwidth, link loss ratios, channel diversity etc. can then be used by the routing protocols to select high quality paths from the available ones. Motivated by these preliminary results, we next propose and evaluate a low overhead *airtime* metric for route selection in multihop wireless networks.

## 4.3 Low overhead metric for path selection: Airtime

In this section, we point out the inadequacies of the existing link quality estimation techniques proposed thus far. Specifically, most of the proposed cross layer metrics, with the exception of [39] use broadcast packets to estimate the pairwise link error rates between neighboring nodes. These are then exchanged across the network by suitable routing protocols such as AODV [9], DSDV [8]

or DSR [10] and used for metric based route selection. For example, in [42], nodes periodically broadcast packets of 134 bytes every second and the link loss rates are calculated as the ratio of lost probes to the total number of expected probes in a given observation interval in both directions. A composite metric known as Expected Transmission Count (ETX) is introduced to capture the link delivery information. The path with minimum ETX metric is selected as the candidate path. There are two drawbacks of these approaches.

- **Inaccuracies due to channel estimation based on broadcast packets**: Lundgren et al. [87] have shown the presence of *grayzones* which are defined as links that are capable of delivering broadcast based routing messages, however, they are unable to support data traffic. This can be due to the presence of asymmetric links as well as the fact that broadcast packets are typically sent at one of the basic rates (using more error resilient modulation schemes) as compared to data packets. This may lead to an inaccurate estimation of link delivery rates resulting in route selection that may be inadequate in terms of delivery rates or throughputs for actual data traffic.

- **Control overheads due to active probing**: Moreover, periodic probe messages impose additional overheads that compete with control messages (such as routing updates) or data traffic. Even though these overheads may be small in terms of bytes transmitted (134 bytes per second per node), the secondary effects of additional control traffic manifest as increase in delay for data packets due to higher channel load and/or possible collisions caused between control and DATA traffic by hidden nodes.

To summarize, the broadcast probe based link loss estimation for route selection may be inaccurate and incurs overhead on actual data traffic. Contrary to the above approach, we propose a passive *data driven estimation* using the novel *airtime* metric, which quantifies the “effort” (in

terms of time) to send one byte of data over the link. In a sense, this is inversely proportional to the "throughput" supported by the link. Thus, by selecting paths with minimal air-time, a higher end-to-end throughput can be achieved. The airtime metric similar in spirit to the ETT metric proposed by [41]. However, we estimate the airtime based on actual data traffic rather than broadcast beacons (based on ETX) that typically have a wider range of reception than actual data traffic. Also, no additional probes are needed to estimate the link conditions. This local metric can be either disseminated throughout the network and a suitable routing protocol can use this metric for path evaluations. The key goal of our design is the simplicity and low overhead required for the link estimations and more accurate measurement techniques to cope with the problems of *communication gray zones*.

### 4.3.1 Calculating the airtime metric

The airtime metric accounts for the medium access delays, channel data rates and the packet delivery ratios based on passive estimation techniques. The measurements are based on ongoing DATA packets combined with a passive sensing based estimation in the absence of active DATA packets through the node. This works as follows: every node maintains the following information tuple $\langle NeighID, Txtime, Txbytes \rangle$ using only DATA packet sent to its neighbor. The node marks the time ($t1$) when a packet is received from higher layers at the MAC and when a corresponding ACK is received for this frame from the neighbor ($t2$). $TxTime$ is the difference $t2 - t1$ and $TxBytes$ is the number of bytes of application data transmitted (minus protocol overheads). Thus, every sender node maintains a record for every DATA packet sent out and airtime is calculated as shown in Equation 4.1

Thus, airtime incorporates channel busy degree ($T_{channel\_access}$), link speeds ($T_{DATA}$) and link delivery ratios ($N$), which quantifies the *effort* in terms of time it takes to transmit one byte of DATA to the next hop. Note that broadcast management messages (such as beacons) or MAC

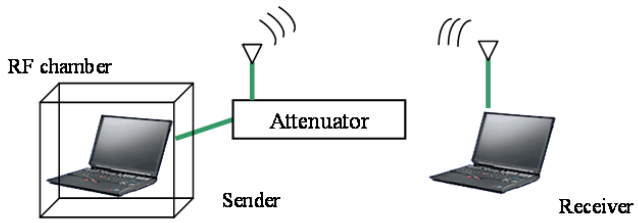



Figure 4.5: Airtime validation: Experimental setup

control messages (RTS/CTS/ACK) are not used in this estimation. A sliding window may be used to capture the recent information to estimate airtime.

$$\begin{aligned} TxTime(neigh) \; += \; & T_{channelaccess} + (T_{RTS} + T_{SIFS} \\ + \; & T_{CTS} + T_{SIFS} + T_{DATA} + T_{SIFS} \\ + \; & T_{ACK} + T_{DIFS}) \times N \end{aligned} \tag{4.1}$$

$$TxBytes(neigh) \; += \; Datapktsize(bytes) \tag{4.2}$$

$$Airtime(neigh) \; = \; \frac{TxTime(neigh)}{TxBytes(neigh)} \tag{4.3}$$

where N = no. of transmission attempts

### 4.3.2 Evaluating the effectiveness of airtime metric

In order to verify the accuracy of our passive estimation technique, we evaluated the effectiveness of the airtime metric in estimating link throughput and per hop delays using both experimental and simulation methodologies.

**Experimental verification**: Our experimental setup, as shown in Figure 4.5, was based on the Intel Calexico wireless cards with modifications to the driver to record and calculate the airtime measurements. We were unable to obtain backoff encountered per packet from the driver and hence the TxTime calculation (Eqn 4.1) does not include $T_{backoff}$.

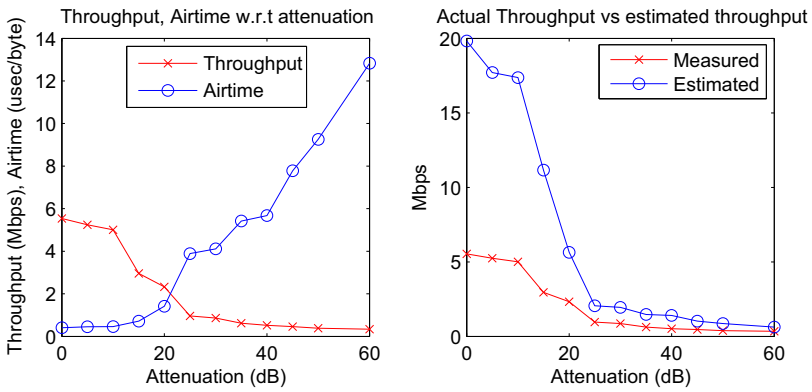



Figure 4.6: Experimental evaluation: a.) Airtime vs throughput b) Measured throughput vs estimated throughput

In order to emulate link variations, we used a Weinschel 8311 Smart Step Programmable Signal Attenuator [88] that is able to provide between 0 and 121 dB of attenuation. The sender was caged in an RF isolation chamber to prevent any RF leakage from the connectors. The output from the sender was sent over the air after desired attenuation to the receiver. The over-the-air path between the sender and receiver was to enable realistic conditions where there would be other stations competing for access to the channel as well. We measured both airtime as well throughput under different channel conditions. The sender and receiver ran Windows 2000 with Intel CX2 wireless adapters and modified device drivers to report airtime as well as average RSSI and PHY Rate over the experiment duration. In order to measure the throughput and packet loss, we used Iperf [89] with 1024 byte UDP flow running between the sender and receiver for duration of one minute.

As shown in Figure 4.6(a), with increasing attenuation, the link throughput decreases and air time increases correspondingly. Figure 4.6(b) compares the throughput estimated using air time and actual throughput reported by Iperf. The estimated throughput ($T_{est}$) using the airtime (which is measured as sec/byte) is calculated as the $T_{est}(Mbps) = \frac{8}{airtime(\mu sec/byte)}$. We note that for lower attenuation levels, the throughput estimate is optimistic since the retransmissions are minimal and

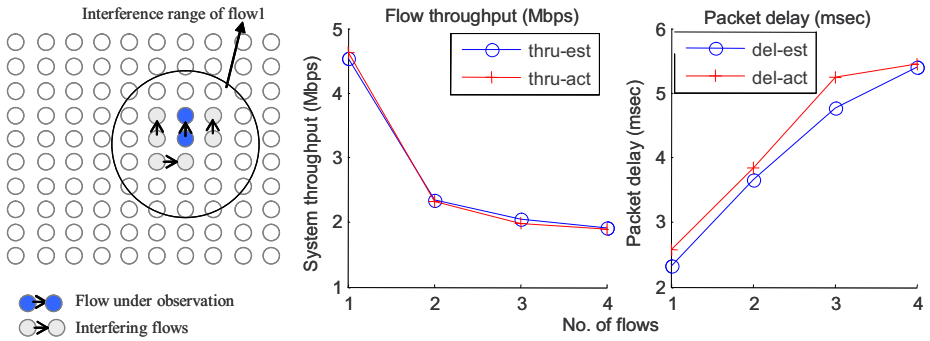



Figure 4.7: Simulation based study: Data driven airtime estimation

the airtime measurement from the driver does not consider the backoff time. As the channel condition worsens with increasing attenuation, the effect of link layer packet losses is more pronounced and is reflected by more accurate throughput estimation based on air time.

Thus, from our initial experiments, it can be seen that airtime can be used to infer outgoing link throughputs using the existing traffic flows without additional overhead especially in high packet loss scenarios. We next conduct detailed simulations for more accurate validation of the metric that is based on Equation 4.1.

**Simulation based evaluations** The airtime metric computation was implemented as a separate module in the ns-2 simulation framework that receives its inputs from the MAC layer in terms of TxTime and TxBytes per successfully acknowledged DATA packet. Airtime calculation using Eqn. 4.1 yields the average $\mu$seconds required to transmit one byte of application data. As shown in Figure 4.7, we measure the airtime of the darker shaded flow in the presence of interfering flows. For throughput estimation, an offered load of 6 Mbps per flow using 1500 byte packets is used such that there is always a packet to send at the sender. For delay estimation, we used a total offered load of 4 Mbps to maintain high packet delivery ratio for measuring delays at the receiver. In both case, RTS/CTS exchange was enabled. We estimate the throughput and average delay per packet at the sender using Equation 4.4 and compare them with the actual measured throughput and delay at the receiver.

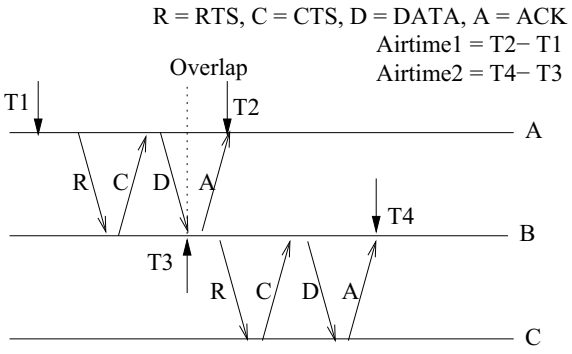



Figure 4.8: Overlap in airtime calculation

$$\begin{aligned} Throughput &= \frac{1}{Airtime} MBps \\ Delay &= Airtime \times packetsize(bytes) \end{aligned} \tag{4.4}$$

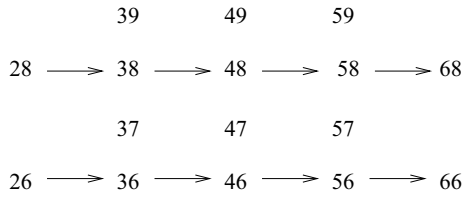
It can be seen that for the one hop case, airtime accurately estimates the available bandwidth as well as the expected packet latency. In order to extend this for the multihop case, it should be noted that simply adding up the per link airtimes to measure the path airtime may result in slightly higher delay estimates. This is a result of the slight discrepancy shown in Figure 4.8 where the overlap period will be accounted for in the link airtime estimates for hop 1 and 2. This "double counting" may result in overestimation of the delay estimate. However, we argue that it still provides a qualitative basis to evaluate candidate paths that can be used by routing protocols for route selection.

### 4.3.3 Estimation in the absence of active flows: Busytime

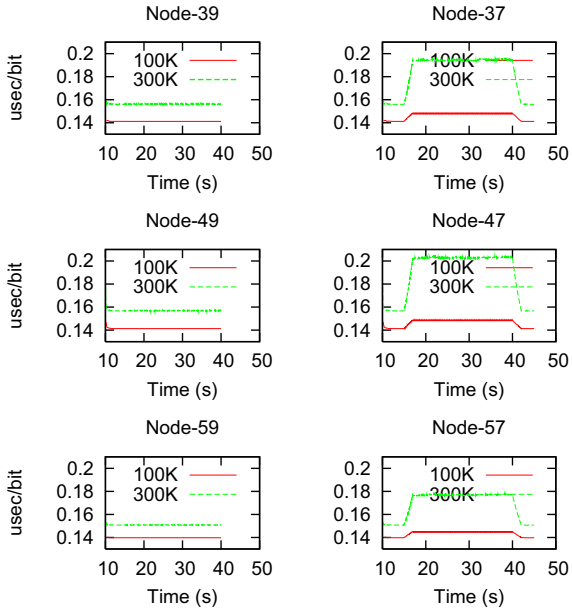
It is evident that in the absence of active DATA traffic flowing through the node, there are no airtime measurements. There are two approaches that can be used to overcome this lack of information. The IEEE 802.11s mesh networking standard [57] suggests the following values:

$$Airtime = (O_{ca} + O_p + B_t/R)(1/1 - e_{pt}) \tag{4.5}$$

where $O_{ca}$(channel access overhead) = 335 $\mu$seconds, $O_p$(protocol overhead)= 364 $\mu$seconds, $B_t$(test payload size) = 8224 bits. Assuming a bit rate, R = 11 Mbps and a loss rate, $e_{pt}$ = 0, an initial value of 0.1759 $\mu$seconds per bit can be used. Note that this may be inaccurate due to the discrepancies in the actual packet loss rate and the link rate as dictated by SNR and the underlying firmware/driver implementation of rate adaptation in the network card. Also, $O_{ca}$ depends on the MAC channel access latencies in the medium surrounding the node and needs to be estimated more accurately.




(a)




(b)

Figure 4.9: Airtime estimation in absence of active data transfer through nodes

We choose an alternate approach by leveraging the channel busy degree estimation techniques

proposed in [39] for bootstrapping the airtime estimation. In this technique, each node passively senses the medium and records the amount of time the medium was sensed busy over a given observation interval, based on the NAV duration. This is used to compute the effective load of the medium in the vicinity of the node based on ongoing transmissions which is a better indictor of $Q_{ca}$ than the constant prescribed in [57]. Passive sensing based estimates are used during initial phase or when there has been no active DATA transfer at a node for $T_{maint}$ interval. We describe a simple experiment (as shown in Figure 4.9) to validate the sensitivity of passive estimation techniques. Two multihop flows (Flow 1: from node 28 to node 68 and Flow 2: from node 26 to node 66) are started at t=10s and t=15s respectively and run for 30 seconds each. The distance between adjacent nodes is 175m and the interference range is 500m. We use two different offered loads (100 Kbps and 300 Kbps per flow). Nodes in the vicinity passively monitor the medium and record airtime estimates based on Eqn 4.5. It can be seen that nodes 39, 49 and 59 sense the activity corresponding to Flow1 and not Flow2 since they are beyond its interference range. Node 37, 47 and 57 detect the presence of Flow 1 and Flow 2 and node-47 senses the medium to be busiest since it is in the interference range of eight other links. The busy estimation also correlates well with the increase in the offered load from 100kbps to 300kbps. In summary, the airtime metric uses a combination of DATA traffic based measurements augmented with passive sensing based estimates in the absence of active DATA flows through the node in order to quantify the per byte per hop transfer time. This is similar to the ETT metric without the inaccuracies and overheads of estimation.

## 4.4 Performance evaluation

In this section, we evaluate the performance of the airtime metric and compare it with the popular ETX metric described in [42] under two different scenarios: Wireless LANs with a spatial distribution of clients and AP (star topology) and multihop wireless networks.

### 4.4.1 Infrastructure WLAN networks

We motivate the utility of the airtime metric in existing WLAN deployments that consists of an access point and spatially distributed clients. The IEEE 802.11e [26] provides an option for setting up a direct path between two clients in a WLAN network. This is known as *Direct Link Setup* (DLS) mode and in envisioned to provide support for devices to connect to each other directly rather than through the infrastructure AP. In home networks, it is commonly the case that the communicating devices are in range of each other and the DLS mode may provide a better path than the two hop penalty going through the AP. However, the 802.11e standard has left open the triggers for enabling DLS mode upto the client implementations. We envision the airtime metric to be useful for evaluating the direct path and the indirect path through the AP and intelligently adapting to the link conditions.

In order to evaluate the performance using the DLS mode based on the airtime metric, we use the following topology representing a WLAN scenario. As shown in Figure 4.10, CL1 intends to communicate with CL2 which is in its range. Several other clients (CL3 to CL6) communicate with a wired server using the AP. Usually, the PHY rate selection is vendor specific and proprietary, however, we assume that since CL1 and CL2 are at the fringe of the APs coverage, they operate at 2 Mbps whereas the other clients are connected at 11 Mbps to the AP.

CL1 can periodically enable the DLS mode in order to attempt direct connection with the peer station CL2. In Table 4.1, we see the airtime measurements of the path through the AP versus DLS using saturated offered load (5 Mbps). It should be noted that airtime metric is able to capture the

Table 4.1: Airtime measurements for legacy vs DLS modes

| Mode | Airtime |
|---|---|
| Legacy | 1.94 $\mu$seconds per bit |
| DLS | 0.75 $\mu$seconds per bit |

effects of both the slow links between CL1-AP and AP-CL2 (based on $T_{data}$ in equation 4.1) as

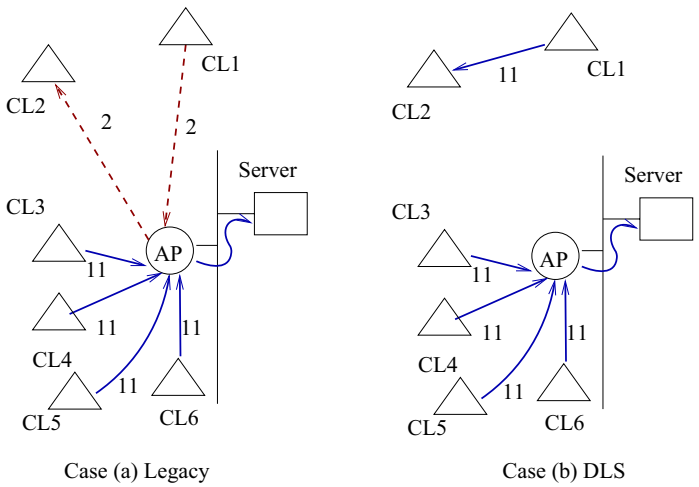



Figure 4.10: WLAN topology to study DLS usage based on airtime metric

well as the load at the AP due to other ongoing client transmissions (based on $T_{channel\_access}$ in equation 4.1). From Figure 4.11, DLS mode triggered by the airtime based estimation, can provide significant improvements in the overall throughput and packet latencies.

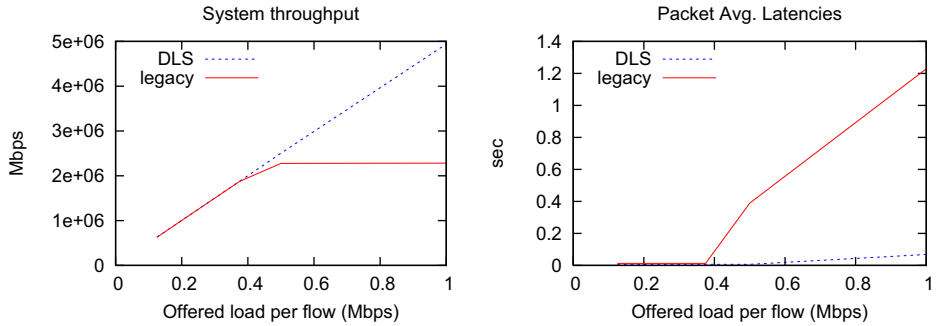
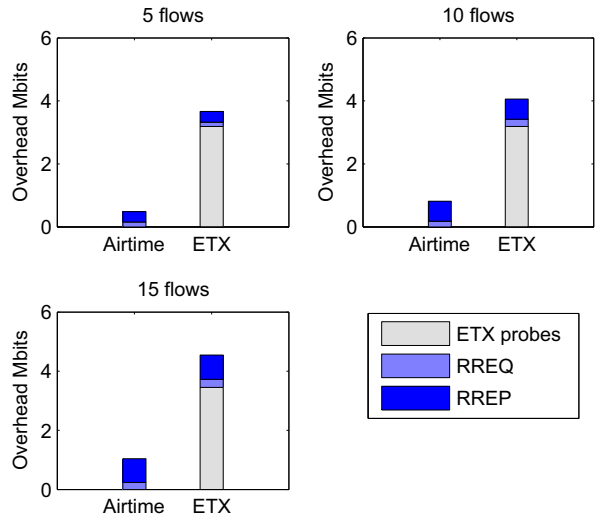



Figure 4.11: Throughput and packet latencies in legacy vs DLS modes based on airtime metric

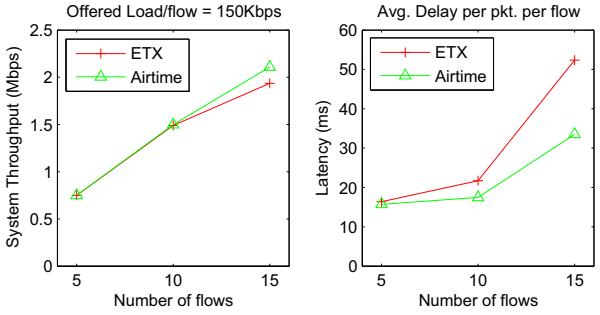
### 4.4.2 Multihop networks

In this case, we used a 10×10 grid with a node separation of 175m and an interference range of 500m. The AODV protocol was augmented to carry information about the calculated ETX (based on periodic broadcast probes) or the airtime (inferred from methods described in Section 4.3.1 and 4.3.3). The offered load per flow is 150 Kbps and we record the system throughput, average per packet latencies and the control overhead for both these methods with the increasing number of

flows from five to fifteen. Each simulation presents an average over several runs.




(a)




(b)

Figure 4.12: Performance evaluation in multihop networks: Airtime vs ETX

As seen in Figure 4.12, it can be seen that the periodic broadcast probes from every node increase the control overhead significantly (about three times as compared to our passive airtime based estimation). We start to observe the impact of this control overhead at higher system loads (15 flows corresponding to 2.25 Mbps total offered load) where the airtime-based route selection results in a higher system throughput and lower latencies ($\sim$1.5x) as compared to the ETX metric based route selection. Thus, our observations indicate that the probe-based estimation may result in inaccurate

evaluation due to the presence of asymmetric links and incur additional control overheads which impact the overall performance.

## 4.5 Summary

In this chapter, we experimentally investigate two important PHY design options that arise in multi-hop wireless ad-hoc networks built with 802.11 radios: PHY rates and frequency of operation and their impact on the performance of the network. Motivated by our observations and the shortcomings of existing cross layer metric based in accurately capturing the underlying link characteristics, we propose a novel *airtime* metric that incorporates PHY rates, link losses as well as medium access delays and enables high throughput path selection in wireless networks with a low control overhead. Using a WLAN topology with multirate clients connected to the AP, we have shown the benefits of airtime-based Direct Link Setup to enable low latency/high throughput communication between nearby stations without using the AP. Evaluations over multi-hop networks using the airtime-based route selection yields a $\sim$1.5x improvement in the average latencies per packet.

In the next chapter, we motivate the design of a cross layer framework that leverages the airtime metric based route selection to augment the ICF architecture proposed in chapter 3 for supporting low latency cut through data transfers in multihop wireless networks.

# Chapter 5

# A path-centric network design for efficient packet forwarding

## 5.1 Introduction

The overall performance of multiple asynchronous flows over multihop wireless network using the distributed channel access mechanisms is inherently limited by the efficiency of the flow co-ordination schemes. As the number of flows increases, the inter-flow interference and contention for access to the channel increases significantly. This results in performance degradation and inefficient channel utilization due to increased overheads associated with collisions and backoffs. In order to mitigate the impact of interference for a given spatial deployment of nodes, there are different dimensions that can be considered to improve the overall network utilization by minimize conflict and collisions. These are outlined as follows:

- *Spatial dimension:* The spatial dimension can be utilized to minimize the conflict between interfering flows based on spatial separation

- *Time based coordination:* Multihop flow coordination can be improved by scheduling the flows at different times to reduce the interference amongst flows. This can be achieved by explicit "reservations" of slots along the path. Alternatively, implicit "soft-reservations" can also be utilized for better flow coordination without the need to make hard reservation of slots. These approaches require time synchronization amongst nodes.

- *Multiple frequencies:* To utilize the available spectrum in the ISM band and reduce interference between flows, spatially overlapping flows may be operated on different orthogonal frequencies. This approach requires dynamic protocols for frequency selection and coordination and the associated channel switching penalties.

- *Multi-dimensional approaches:* Recent work has focused on multi-dimensions optimizations for performance improvements. In [46,47], the authors propose and evaluate joint channel assignment and routing algorithms for multi-radio wireless networks. An analytical framework has been proposed in [49] to study joint scheduling, frequency assignment and route selection to find theoretical upper limits on achievable capacity in multihop wireless networks.

In this chapter, we first investigate the spatial dimension based on interference aware route selection techniques. To achieve this goal, we leverage the airtime metric discussed in chapter 4 and introduce a Cross Layer enabled cut-through Architecture (CLEAR) for low latency data transfer in multihop wireless networks to maximize the number of cut through transfers. Motivated by a "path-centric" design [52] for efficient packet delivery, we propose improvements to the CLEAR architecture for improve flow-coordination based on implicit scheduling using soft-reservations during route setup along with multihop "burst" mode data transfers. This framework can be extended to provide differentiated services for real-time flows with low latency requirements.

## 5.2 Motivation

In chapter 3, we have shown that a combination of label based switching and cut through medium access results in more efficient packet forwarding with reduced overall latencies. In our earlier evaluation, we have assumed fixed routes prior to data transfers based on shortest path metric and focus more on the benefits of fast switching and cut through MAC enhancements. However, route

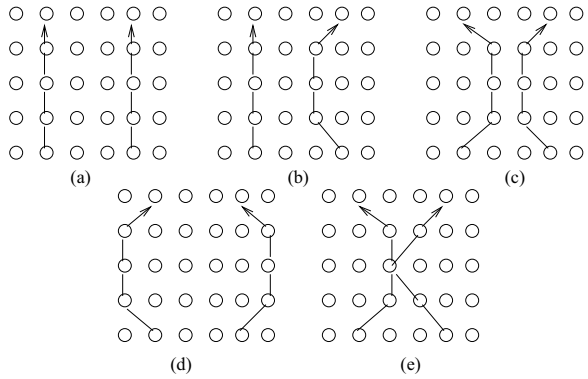



Figure 5.1: Route Selection choices

selection plays an important in maximizing the number of successful cut through transfers over multiple hops. We motivate the need for improved interference-aware route selection techniques in realizing this objective using the following illustration (Figure 5.1)

The topology consists of two flows and the different paths correspond to candidate routes chosen by the routing protocol. In this setup, we have statically configured the routes to explore different alternate routes between the source and destination. These are studied using both default 802.11 and DCMA. The average delay (in milliseconds) per packet per flow is tabulated in Table 5.1.

Table 5.1: Average latency per packet for different topologies

| Topology | 80211 (delay in ms) | DCMA (delay in ms) |
|---|---|---|
| a | 10.332 | 8.617 |
| b | 15.302 | 12.661 |
| c | 16.8 | 13.756 |
| d | 10.38 | 8.615 |
| e | 16.828 | 13.926 |

Thus, it can be seen that in the worst case scenario (Topology e), DCMA alone improves the end to end latency from 16.8 msec to 13.9 msec. Additionally, the choice of low interference routes (e.g. Topology a, d) with DCMA brings the latency down to 8.6 msec which is a $\sim$2x improvement over Topology e with 802.11. In the above scenario, we sampled different routes using static routing.

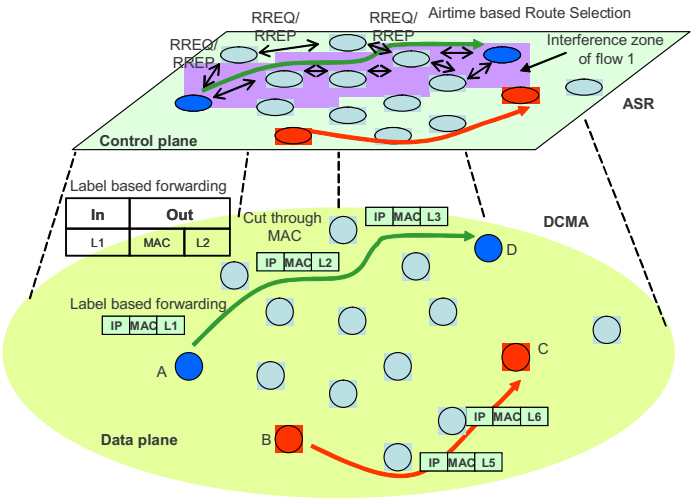



Figure 5.2: Cross layer network architecture for low latency packet transfers in wireless networks

Our goal is to propose a distributed cross layer metric based route selection strategy in order to find low interference paths to maximize the cut through transfers over multiple hops.

## 5.3 Cross Layer enabled Cut through architecture (CLEAR)

Motivated by our observations, we propose the Cross Layer Enabled Cut-Through Architecture (CLEAR) that combines the benefits of interface contained forwarding with airtime based route selection to spatially distribute flows for minimizing interflow interference. The architecture as shown in Figure 5.2 consists of two main components.

- **Data plane enhancements-Interface contained forwarding (ICF)**: We leverage the ICF architecture as outlined in [90] and Chapter 3 which is based on fast label-based next-hop lookups at forwarding nodes that eliminates the route lookup latency (CPU interrupt and packet copying latencies). In order to reap the benefits of fast label switching, we also propose novel cut through MAC extensions (DCMA) to the 802.11 DCF which attempts to pre-reserve the channel for the subsequent data transfer while acknowledging the current packet delivery. The ICF architecture significantly reduces the end-to-end packet transfer latencies due to

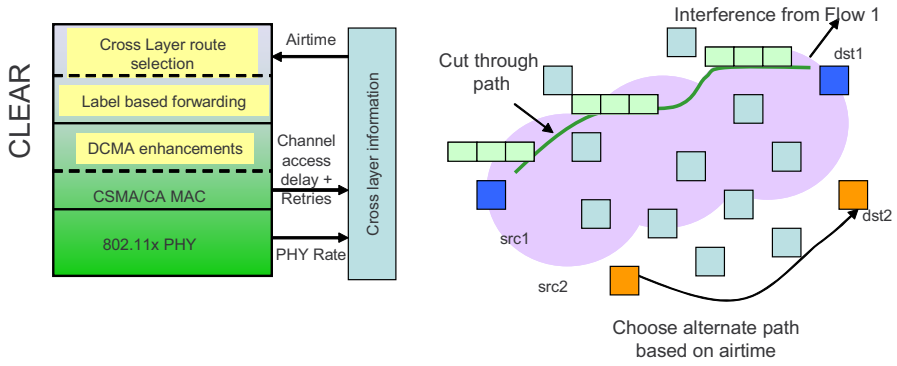



Figure 5.3: Airtime based source routing: protocol stack and concept

reduced channel access overheads at each hop along the path.

- **Control plane enhancements-Airtime based route selection (ASR):** We also propose a cross layer routing protocol (ASR) based on the novel *airtime* metric in order to maximize the successful cut through transfers and reduce interference amongst flows.

The data plane enhancements based on ICF and the airtime metric for interference aware route selection have been described in Chapters 3 and 4 respectively. In the next section, we focus on the control plane enhancements based on our novel airtime based source routing (ASR) protocol.

### 5.3.1 Airtime based source routing: ASR

We leverage the data driven and passive airtime measurements described in Chapter 4 to find high quality routes between source and destination pairs. We propose the ASR routing protocol (shown in Figure 5.3), which is an on-demand source routing technique similar to LQSR [91]. ASR explores several candidate paths based on the accumulated airtime metric, and the source selects the best path amongst these choices. This heuristic approach is loop free since it is based on source routing and allows us to explore paths with low interference so as to maximize the successful cut-through packet transfers. The routing protocol operates as follows.

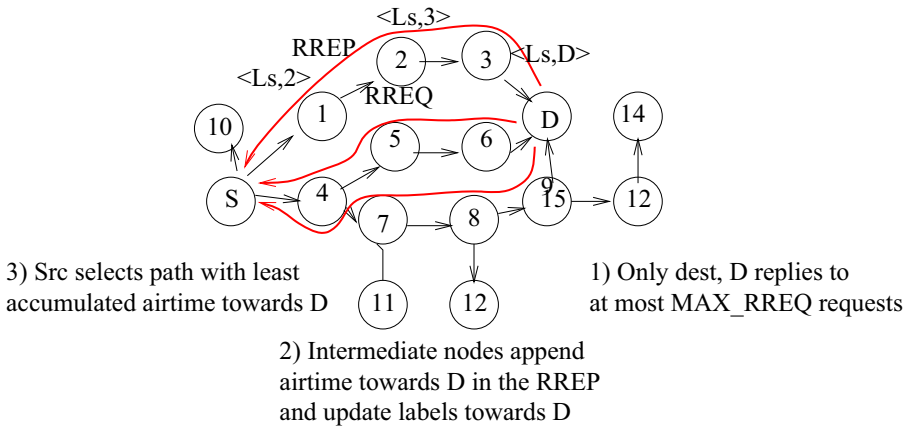



Figure 5.4: Route discovery and label updates

- **Route discovery phase:** In the route discovery phase, the node with data to send (S) towards a destination (D) initiates route discovery by broadcasting an RREQ with TTL equal to $MAX_{TTL}$, the *destination-only* flag set, a unique broadcast ID and a destination label ($L_d$). We choose $MAX_{TTL}$ for the initial discovery since we do not know the average number of hops between source and destination prior to sending the first RREQ message. Also, each node maintains a local label table that contains the following information $\langle Dest.Label, Nexthop \rangle$

  Each intermediate node that receives the RREQ rebroadcasts it **once** per unique ID and creates a reverse route to the source, S. Intermediate nodes do not reply to the RREQ even if they have cached information. The destination, D, upon receiving the RREQ, replies to at most $MAX_{RREP}$ per unique broadcast ID adds a local entry $\langle L_d, D \rangle$. This approach is to chosen so as to balance the control overhead of multiple RREPs traversing back to the source and the diversity of candidate paths that can be chosen by the source. Note that the RREQ does not need to carry the metric information. Since the airtime metric is a source based estimation, the intermediate nodes monitor and append the airtime towards the destination in the RREP and their own ID before unicasting it back towards S. Also, each intermediate node creates

local label entry $\langle L_d, PrevhopMAC \rangle$.

For example, nodes 3, 2 and 1 will have the entries $\langle L_d, D \rangle$,$\langle L_d, 3 \rangle$ and $\langle L_d, 2 \rangle$ respectively (refer Figure 5.4). Upon receiving multiple RREPs across diverse paths, the source is responsible for path selection based on cumulative airtime towards the destination and chooses the path with the least airtime. The source also records the hop count of the selected path to be used as TTL in periodic route maintenance as described in the next section.

- **Route maintenance phase:** Arrival of new flows in the system or wireless link dynamics may affect the quality of the initial route chosen in the route discovery phase. It should be noted that in earlier approaches such as [42], the routing protocol is run for some duration (90 seconds) prior to sending data traffic. During the interval of the experiment, the routing protocol continues to exchange control packets, however, these are used only to provide competing control traffic in the network and not to update or maintain routes. Thus, the route selection is heavily biased by estimates at the beginning of the experiment and fails to adapt to any dynamics during the experiment.

  In ASR, we have a provision to periodically rediscover routes to destination. The source, S, sends periodic maintenance RREQ every $T_{maint}$ interval with a TTL equal to the hop count of the currently used path. This restricts the scope of the flooding and thereby the associated control overhead. The maintenance RREQ is identical to the original RREQ and is processed by the destination and intermediate nodes in a similar fashion. The maintenance interval is equal to the window used for airtime calculation in order to capture the changes in link status since the last maintenance cycle. In our current implementation, we choose $T_{maint}$ equal to 10 seconds, however, this interval can be configured based on traffic and link dynamics to limit the control overhead.

To summarize, ASR uses the airtime metric based source routing with periodic route maintenance in order to explore low latency paths between communicating nodes.

### 5.3.2 System evaluation

In order to evaluate the delay performance of high priority flows using CLEAR, we have implemented the CLEAR architecture in the ns-2 [59, 82] environment with an improved SINR-based physical layer model. The simulation setup is described in Table 5.2. The topology is a $10\times10$ grid with random source destination pairs. We consider several topologies with increasing number of flows from five to fifteen with random source-destination pairs. Each run lasts for 30 seconds and the results are averaged over several runs (topologies). Note that the initial route discovery mechanism using RREQ's is common to both AODV and CLEAR implementations. We have observed that the candidate routes that are available at the source for metric based route selection highly depend on the broadcast propagation of the corresponding RREQ messages. In some cases, basic AODV route discovery results in the initial paths that are similar to the ones chosen by ASR. Using periodic re-evaluations, ASR is able to select a better path in the next update cycle. The source-destination pairs are selected such that the maximum separation between them is at most $N_{maxhops}$ hops. Using these simulation settings, we compare the end-to-end packet delay performance for the following two protocols.

- Baseline case: AODV + 802.11 DCF
- CLEAR: ASR with DCMA

**Calculating the end-to-end successful cut through transfers:**

We measured the average latencies per flow per packet ($T_{avg\_lat}$) as well as the fraction of successful end-to-end cut through transfer ratio ($P_{cut}$). In order to calculate $P_{cut}$, we used one bit of

Table 5.2: Simulation parameters

| | |
|---|---|
| Number of flows | 5, 10 and 15 |
| Max path length, $N_{maxhops}$ | 3, 7 and 10 hops |
| Offered load per flow | 200 Kbps |
| Traffic type | CBR (1500 bytes) |
| PHY Reception model | SINR based |
| Interference range | 500m |
| Distance between nodes | 175m |
| Transmission range | 250m |
| Airtime refresh interval | 10 seconds |
| Maintenance cycle period | 10 seconds |
| Max RREPs per unique RREQ | 3 |
| Max Maintenance cycles | 3 |

information in the MAC header to indicate the cut-through state of the packet. The source node sets this bit to 1 and if any intermediate node is unable to cut-through to its neighbor, this bit is reset. The destination node keeps track of the number of packets ($N_{cut}$) with this bit set to one. Since we use unique source-destination pairs, the successful end-to-end cut-through transfer ratio, $P_{cut}$ is calculated as $(\frac{N_{cut}}{N_{sent}})$, where $N_{sent}$ is the number of packets sent by the corresponding source.

**Results:**

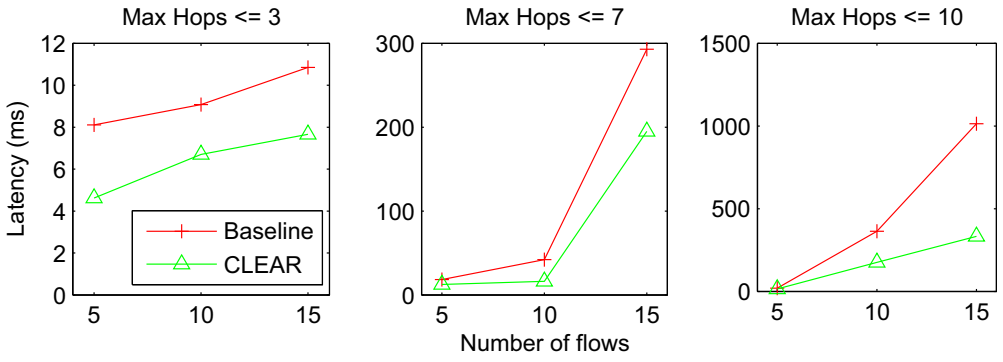



Figure 5.5: Packet latencies versus number of flows (for different path lengths)

We evaluated the CLEAR architecture under different spatial densities. In particular, we control the maximum path length, $N_{maxhops}$, when selecting the unique source-destination pairs as communicating endpoints. A lower value of $N_{maxhops}$ and lower number of flows results in a lower spatial

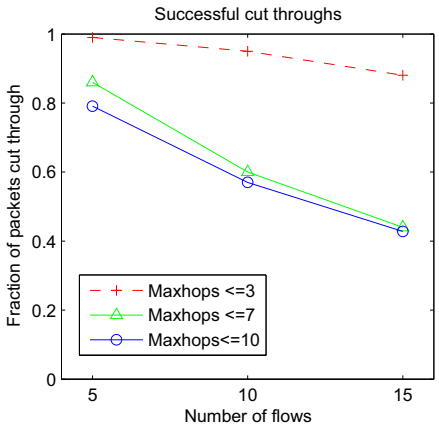



Figure 5.6: Successful end-to-end cut through transfers versus number of flows (for different path lengths)

density. This results in better opportunities for the ASR protocol to find routes with minimum interflow interference. For higher $N_{maxhops}$, there is a higher spatial density of flows due to longer paths. This reduces the likelihood of finding non-conflicting paths. However, CLEAR architecture still provides improvements from fast label based forwarding and cut through MAC transfers. As seen from equation 3.1 the longer the path, the higher the benefits of contention reduction at each hop due to DCMA MAC. We have used medium offered loads of 200 Kbps per flow so as not to saturate the network and maintain a high packet delivery ratio. The following observations can be made.

- **Shorter path length:** The results for latencies for the shorter path lengths (corresponding to $N_{maxhop} <= 3$) are shown Figure 5.5a. It can be seen that the latencies for both baseline as well as CLEAR are relatively low (<20 ms). CLEAR still performs better than the baseline model. The moderate gains are due to shorter path lengths and DCMA benefits are directly proportional to the path length (from Equation 3.1).

- **Longer path length:** For longer paths (7 hops and 10 hops), it can be seen that CLEAR provides significant improvement in latencies as compared to the baseline mode due to the

combination of reduced interflow interference due to ASR and self-interference from DCMA and label-based forwarding. For the case of 15 flows, CLEAR yields ∼1.5x to ∼3x improvement in the overall latencies.

In Figure 5.6, we show the impact of increased path lengths and number of flows on the successful cut through transfers using CLEAR. It can be seen that as the number of flows and the path length increases, the cut through ratio reduces due to increasing interference. CLEAR is still able to support upto 40% successful cut throughs at these loads.

In summary, using a combination of airtime-based route selection, label-based forwarding and cut-through MAC enhancements, the CLEAR architecture yields significant improvements in the end-to-end packet latencies over conventional forwarding in multihop networks.

## 5.4 CLEAR+: Implicit scheduling for improved flow co-ordination

The CLEAR architecture, described in the preceding sections, uses the spatial separation amongst flows during route selection to minimize interference amongst flows that can then use DCMA MAC to cut-through packets over multiple hops. While this approach yields overall improvement in packet latencies and successful end-to-end cut through transfers, the number of cut through transfers is limited by the interference between flows especially when the spatial density of the flows is high. In these cases, it may be beneficial to look at time-based co-ordination amongst flows to reduce interference. Additionally, the following observations can be made.

- DCMA protocol use RTS/CTS handshake prior to every packet exchange to attempt pre-reservations. While this may provide benefits for larger payload sizes, it incurs an overhead in terms of additional control traffic introduced per packet per hop. For a more session-oriented data transfer, the overheads can be avoided by attempting a multihop reservation for

the duration (or fraction) of the session.

- The DCMA MAC is currently responsible for opportunistically establishing "soft" reservations hop-by-hop as the packet progresses towards the destination. Since the routing protocol needs to establish a path to the destination prior to the actual data transfer, we argue in favor of shifting the responsibility of attempting reservations to the route discovery phase.

Some of the related work in this area include explicit reservation-based packet transfers using QoS routing techniques [43, 44] that rely on bandwidth (or delay) estimation techniques to establish "hard" QoS reservations. Given that they operate over links that use independent PHY rate adaptation algorithms [85, 92, 93], it is extremely challenging to predict whether the guarantees can be met. In [94], the authors introduce a model for service differentiation where the real-time flows use Request to Reserve (RTR) and Clear to Reserve (CTR) messages (at the MAC layer) to reserve the medium for the duration, $\Delta$, and periodicity, $\tau$. However, an independent route selection using AODV is done prior to the reservation in order to find a path. A distributed link scheduling algorithm is outlined in [22] that extends the 802.11 DCF for local multihop coordination without explicit global synchronization resulting in a dynamic TDMA like schedule in the local neighborhood. However, this co-ordination is within an interference region for controlling channel access for each packet.

In our approach, we use soft-reservations during the route discovery phase to coordinate efficient multihop "burst" transfers at the MAC. This concept can be considered as a multihop extension to the TxOp (transmit opportunity) mode proposed in the IEEE 802.11e standard [26]. Our work is inspired by [52], which motivates a *path-centric* network design using a relay-oriented physical layer, multi-hop in-network coordination as well as cooperative transport using diversity combining.

While the relay oriented physical layer requires changes to existing hardware, our focus on the co-ordination amongst flows to further improve the end-to-end packet latencies.

### 5.4.1 Design choices

We first look at a few design choices to enable time-based coordination amongst flows and improved packet forwarding at the MAC layer.

**Route discovery - Enabling soft reservations:** In our system design, we assume asynchronous arrivals of traffic flows and that the route discovery procedure is responsible for creating soft reservations. One approach is to explicitly reserve the medium where the intermediate nodes do not accept any packets (data or control), and the nodes in the vicinity of this flow also set their NAV until the end of the reservation. This approach has been proposed in [94]. However, based on our investigations on a two-dimensional grid, we observe that depending on the reservation duration, offered loads, buffer sizes and the arrival of flows, there is a high likelihood of route discovery failure for flows that arrive during ongoing traffic sessions. Route failure leads to periodic rebroadcasting of RREQ messages resulting in increased overheads. It may also cause blocking of flows in case the ongoing flow lasts longer than the maximum route discovery timeout.

In our design, each flow, upon arrival, attempts to make an initial reservation for $T_{softres}$, which is a configurable parameter. It is expected that the application can provide this information to the network layer. By adjusting the $T_{softres}$, channel access for the flows can be controlled. In our initial design, we assume all the flows belong to the same priority level and thus conservative values for $T_{softres}$ are used to allow multihop burst opportunities for all the flows. A key design choice is that we **do not** reserve the Network Allocation Vector (NAV) at the intermediate nodes and **do** allow control traffic from other flows to flow through. We use the control traffic to piggyback information about ongoing active sessions as described later in section 5.4.2. Note that our model

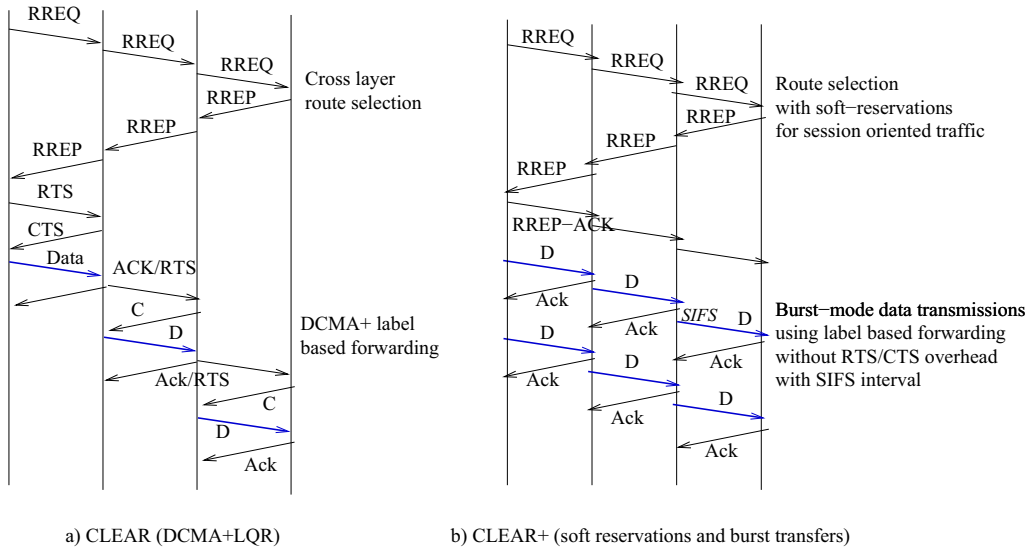
can be extended to support differentiated services, by selecting $T_{softres}$ to be equal to the flow duration for higher priority flows. This is explained later in Section 5.4.4.

**MAC enhancement - Burst mode transfer:** DCMA MAC uses RTS/CTS exchanges per packet to attempt pre-reservation for the next hop along the path. In the case of successful cut-through transfer, this approach eliminates the medium access latency at each hop thereby improving the overall packet transfer latency. However, as described earlier, the route discovery phase is now responsible for soft reservations. Hence, we consider an improved multihop burst mode transfer that is equivalent to the TxOP mechanism proposed in the 802.11e [26] standard for single hop WLANs. In this mechanism, RTS/CTS exchanges are eliminated and a burst of DATA packets (upto $T_{TxOP}$) can be sent separated by a shorter SIFS interval after negotiating the TxOp interval between the AP and the client. In our design, we extend this concept to multiple hops. The following options can be considered for the reducing the control overheads associated with acknowledgements for the burst mode data transfer.

- **No Acknowledgements:** This can be considered for real time traffic that is more latency-sensitive. Alternatively, improved FEC techniques may be used for fast local recovery at the receiver without triggering the transport layer retransmissions.

- **Implicit Acknowledgements:** This approach has used in [94] where the downstream DATA transfer is perceived as a successful reception by the upstream node. Note that the STA will have to listen promiscuously to decipher the transmission from the next hop node. In case this is not overheard, the upstream node re-attempts the frame delivery.

- **Block Acknowledgement:** This enhancement has been proposed in the 802.11e [26] for single-hop WLAN networks to enable multiple frame transmissions in a burst prior to acknowledgement to reduce the per packet control overhead. The size of the burst transfer is




a) CLEAR (DCMA+LQR) b) CLEAR+ (soft reservations and burst transfers)

Figure 5.7: CLEAR Route Setup and proposed enhancements

negotiated based on the buffer sizes at the sender and receiver. The receiver then indicates the missing frames within a burst using a bitmap in the Block Acknowledgement message. The sender performs a selective retransmissions of missing frames. While this can be extended to the multi-hop case, it can results in a lot of state maintenance regarding missing frames at each intermediate node.

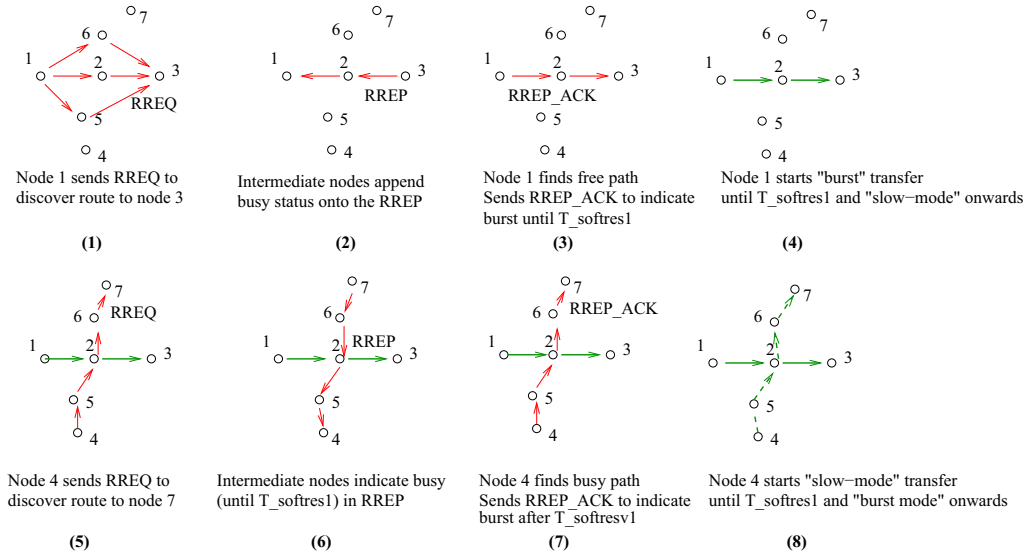
- **Regular ACK:** This approach is similar to the regular DCF mechanism with no RTS/CTS enabled. Each frame is acknowledged per hop and recovery is done at each hop.

In our design, we use the Regular ACK approach with the following enhancement: Intermediate nodes acknowledge the received frame and do not re-contend for medium access for the subsequent data transmission. The DATA frame is sent $T_{SIFS}$ interval after the upstream ACK. This is the "burst" mode of operation. In the regular mode or "slow-mode" , default 802.11 DCF without RTS/CTS is used.

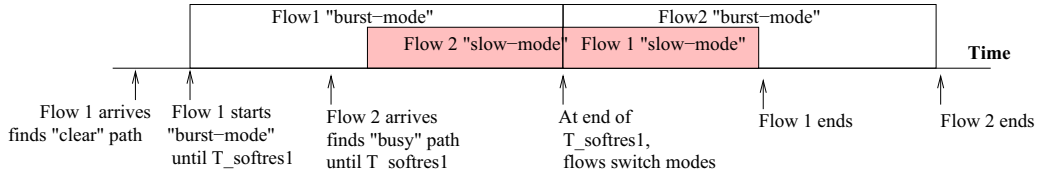
In summary, the key protocol differences between our original CLEAR architecture and the proposed enhancements can be seen in Figure 5.7.

### 5.4.2 CLEAR+: Protocol description

We describe the protocol design using a simple example with two flows as illustrated in Figure 5.8. Note that each node maintains a local *busy_duration*, which represents the node's perception of the ongoing activity in the neighborhood. It is assumed that the nodes are synchronized and *busy_duration* is an absolute timestamp which is set to zero at the beginning. Flow1 and Flow2 arrive asynchronously and use the route discovery procedure based on RREQ and RREPs to find routes to the destination. For simplicity, we assume that flow 1 arrives before flow 2. The CLEAR+ protocol dynamics are explained below.




(a) CLEAR+: Example




(b) CLEAR+: Time sequence

Figure 5.8: CLEAR+: Mode of operation

- Node 1 sends a RREQ to initiate route discovery to the destination, Node 3. As in ASR,

the RREQ flag is set to "destination-only" to disable intermediate nodes from responding with cached information. In this case, we use a $MAX\_RREP = 1$ which implies that the destination responds to only the first RREQ by sending an RREP back to the source along that path. Intermediate nodes use the RREP flowing from the destination to the source to piggyback information about any ongoing activity along (or in the vicinity of) the path. Upon receiving the RREP from the node 3, each intermediate node creates label entries for the flow and checks if they are currently serving other flows based on local *busy_duration*. The RREP is then forwarded towards the source. The intermediate nodes set their state from *idle* to *busy-tentative* and set a timeout to reset the state back to *idle* if no RREP_ACK is received. This is illustrated in Figure 5.8 (1 and 2)

- Node 1 receives the RREP and checks the information in the RREP. Upon finding a "clear" path (with no ongoing transfer reported by intermediate nodes), node 1 sends a RREP_ACK towards the destination, indicating a soft-reservation for the flow for $T_{softres}$ duration. Note that this interval is configurable. Upon receiving the RREP_ACK, each intermediate node and the destination sets their *busy_duration* to $T_{softres}$. Following this, node 1 initiates a "burst" mode transfer until $T_{softres}$. This is illustrated in Figure 5.8 (3 and 4). As described earlier, in the burst mode, DATA packets arriving at intermediate nodes are acknowledged individually and reforwarded to the next hop after $T_{SIFS}$. Node 1 continues to operate in this mode until $T_{softres}$ after which it switches to the slow mode (basic 802.11 with no RTS/CTS).

- During the burst transfer of flow 1, flow 2 arrives and node 4 initiates route discovery to the destination node 7. Note that, in our design, we allow control traffic to interrupt ongoing burst mode transfers to prevent starvation of other flows. The intermediate nodes (node 6, node 2 and node 5) indicate the presence of ongoing burst activity until $T_{softres}$ in the RREP that

travels towards the source node 4. Node 4 records this activity and sends an RREP_ACK to indicate the future soft reservation from t=$T_{softres}$ to $2 * T_{softres}$. Intermediate nodes and the destination record this future reservation in *busy_duration*. This is shown in Figure 5.8 (5,6 and 7)

- Flow 2 then initiates a "slow" mode transfer (regular 802.11 without RTS/CTS) until the expiration of the current burst ($T_{softres}$). At $T_{softres}$, the flows reverse their modes. Flow 2 commences its burst mode until $2 * T_{softres}$ while Flow 1 drops back to slow mode. Thus, the soft-reservations during route discovery enable flows to co-ordinate their burst mode transfers while avoiding starvation of any particular flow.

### 5.4.3 Performance evaluation

We have implemented the CLEAR+ protocol enhancements in ns-2 simulator. The simulation setup is similar to the one described in Table 5.2. The topology is a $10 \times 10$ grid with random source destination pairs. Using these simulation settings, we compare the end-to-end packet delay performance for the following three protocols.

- Baseline case: AODV + 802.11 DCF
- CLEAR: ASR with DCMA
- CLEAR+: Improved route discovery with multihop "TxOp" based MAC transfer

To motivate the potential benefits of flow coordination on the system performance, we first look at several two-flow topologies and results are shown in Table 5.3. The flow duration was 2 seconds with a separation of 0.5 seconds between arrival of flows and $T_{softres} = 1sec$.

It can be seen that for the simple case with two flows, the time coordinated "burst" mode transfers results in $\sim$3x improvement in latencies over the baseline scenario and outperforms CLEAR

Table 5.3: Average latency per packet per flow

| Offered Load per flow (Kbps) | Baseline (delay in ms) | CLEAR (delay in ms) | CLEAR+ (delay in ms) |
|---|---|---|---|
| 300 | 14.89 | 11.55 | 10.8 |
| 500 | 36.57 | 15.47 | 11.94 |
| 750 | 102.67 | 80.69 | 32.03 |

protocol that is unable to find spatially diverse routes to reduce interference.

We next extend the evaluation to more general topologies with increasing number of flows from five to fifteen with random source-destination pairs and the results are averaged over several runs (topologies). The source-destination pairs are selected such that the maximum separation between them is at most $N_{maxhops} = 7$ hops. For our evaluation, we consider an offered load of 200 Kbps per flow and measure the end-to-end packet latencies per flow. The flows last for 4 seconds and start 250 ms apart. We use different soft reservation intervals of (100ms, 500 ms, 1000ms and 2000ms) and present the averaged results for these settings. As seen in Figure 5.9, as the

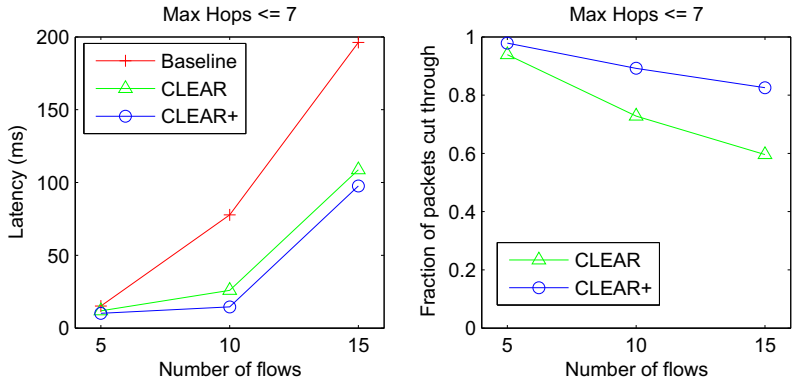



Figure 5.9: Average delay performance comparison: Baseline, CLEAR and CLEAR+

number of flows increases from five to fifteen, the spatial density of flows increases, and the benefits of time-based coordination result in an improvement on overall latencies per packet per flow as compared to the baseline model as well as CLEAR. Due to the high flow density, CLEAR is unable to find spatially diverse paths to reduce interference. Based on time-coordination amongst flows, all

the flows experience an opportunity to use the multihop "burst" mode for efficient packet delivery resulting in lower average packet latencies for all flows.

### 5.4.4 Towards differentiated services for real-time traffic

The CLEAR+ architecture can be adapted to provide differentiated services for real-time traffic. Using a combination of soft-reservations and multi-hop bursts, CLEAR+ can provide a "low-latency" socket for efficient low-latency real-time packet delivery over multiple hops. In order to achieve this, the high priority flows can adjust the $T_{softres}$ equal to the entire flow duration. When the lower priority flows encounter the presence of an existing high priority flow along the path, they strictly defer until the end of $T_{softres}$ instead of continuing to operate in the "slow mode" as described in Section 5.4.2. This is illustrated using the example in Figure 5.10.

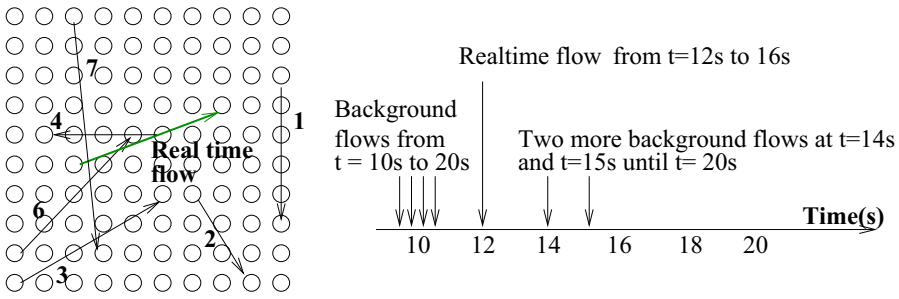



Figure 5.10: Illustration to study the application of CLEAR+ for differentiated services

There are four background flows from $t = 10s$ to $t = 20s$, each offering 50 Kbps using 256 byte packets using 802.11 DCF MAC and AODV. The real-time flow starts at $t = 12s$ and lasts until $t = 16s$. It uses 1500 byte packets to deliver real time traffic. We consider different offered loads for the real time flow (500 Kbps, 750 Kbps and 1 Mbps). During the route discovery stage of the real-time flow, the RREP_ACK is used to inform the nodes on the path of the soft-reservation until $t = 16s$. Two other background flows arrive at $t = 14s$ and $t = 15s$ and initiate RREQs to find a path to the destination. Upon receiving a corresponding RREP from the destination, the

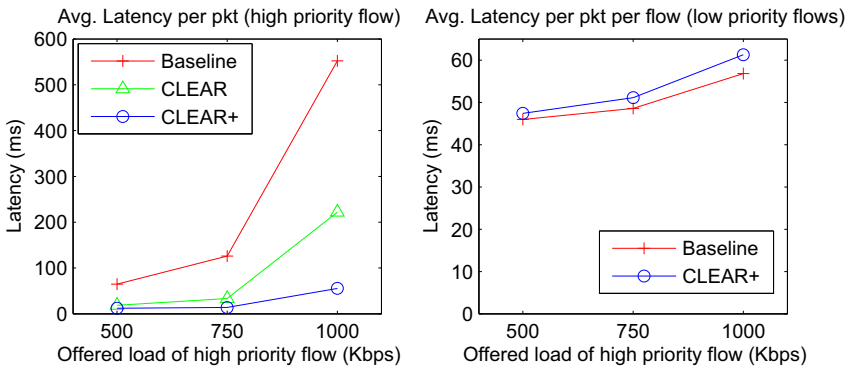



Figure 5.11: a.) Average delay per packet of high priority flow: Baseline, CLEAR and CLEAR+ b) Average latency per packet per flow for the low priority flows: Baseline vs CLEAR+

intermediate nodes piggyback the information regarding ongoing high priority flow until $t = 16s$. Instead of operating in the "slow mode" as before, the newly arrived low priority flows defer until the end of the real time flow to access the medium. We measure the average latency per packet for the real time flow and plot the results for the three schemes: Baseline, CLEAR and CLEAR+.

As seen in Figure 5.11, time-based coordination and burst mode transfers of CLEAR+ protocol yields significant (∼5x to ∼10x) improvements over baseline and (∼2x to ∼4x) over CLEAR protocol with small impact on the low priority traffic. While the above scenario is an illustrative representation, it indicates the flexibility of the CLEAR+ architecture to support differentiated services by tuning the soft-reservations based on the network conditions and traffic requirements. We are currently investigating this approach.

## 5.5 Summary and Future enhancements

In this chapter, we have looked at a cross-layer enabled cut through architecture (CLEAR) based on cut-through MAC enhancements, improved label-based forwarding as well as interference aware route selection techniques. This architecture utilizes the spatial dimension to find routes with minimal conflict based on a novel low overhead *airtime*-based route selection and reduces route lookup and contention overheads at the intermediate nodes. Using simulation based evaluations, we report

$\sim$2x improvement in end to end packet latencies as compared to traditional approaches. We have further proposed a time-based coordination scheme (CLEAR+) using soft-reservations during route setup and subsequent multihop "burst" mode transfers for highly efficient packet delivery. This scheme can be used in scenarios with high spatial density of flows where alternate low-interference paths may not be available. Results indicate a $\sim$2x improvement in average packet latencies per flow over uncoordinated 802.11 based medium access with AODV routing. CLEAR+ has also been adapted to support differentiated services over multiple hops, thus providing a "low-latency socket" for real-time traffic over multiple hops.

There are several improvements that can be made to the existing design.

- In the current protocol model, each flow relinquishes the "burst" mode transfer at the end of the current soft-reservation interval. While this is a conservative approach to accommodate other flows, the "burst" transfer can resume in the absence of any interfering traffic.

- Presently, when the source receives the RREP that reports some ongoing burst mode transmission along the path, it schedules its own burst mode upon the end of the reported interval and continues to operate in the slow-mode. Instead, this throttling can be moved further down the path to the node(s) that are in the vicinity of the other ongoing flow. If more than one flow finds a busy path, they can resort to conventional backoff schemes for "path access control", to avoid aligning their burst transfers.

- Multiple RREPs can be used to explore paths with different busy durations and the source can decide to use the route with least wait time. There is an additional overhead in terms of control traffic associated with multiple RREPs. This may result in combining the benefits of both spatial and time based coordination for further improvements in packet transfer latencies.

- Block Acknowledgements can be used instead of per packet ACKs in order to further improve the efficiency of the multihop burst. Alternatively, the recovery procedure could benefit from improved FEC schemes or rely on transport protocols thereby eliminating MAC layer acknowledgements.

- Admission control techniques can be employed in conjunction with the cut-through transfers to maintain the network load in the optimal operating region. This could prove to be especially useful for relatively low-bandwidth, delay sensitive applications such as VoIP-over-wireless

In summary, as wired LANs have evolved from hubs to switch based models, it is imperative for wireless networks to follow this trajectory to provide competitive access technologies using the increasing PHY capacity more efficiently. The CLEAR architecture and path centric cut-through design presented in this chapter is a step in this direction.

# Chapter 6

# Conclusions and Future work

In this thesis, we have focussed on architectural and in-network enhancements to improve the overall scalability and performance of multihop wireless networks. More specifically, the following contributions have been done.

**Self Organizing Hierarchical Ad-hoc Network Architecture (SOHAN):** We have proposed a novel three-tier hierarchical architecture with dedicated forwarding nodes that yields improvements in scalability and system performance as compared to conventional flat ad-hoc networks. This includes novel protocols for bootstrapping and discovery assisted topology formation, that take realistic system considerations such as device heterogeneity into account, to enable hierarchical self-organization. A proof-of-concept prototype was developed for evaluation of protocol design options and validation of system performance. Experimental results show a 2.5x improvement in system throughput over flat ad-hoc networks and are fairly consistent with predictions from simulation. These are indicative of the advantages of the proposed hierarchical structure with self-organizing discovery and routing protocols. Design considerations for dual radio FN with respect to the physical proximity of antennas have been evaluated using experiments.

**Interface contained forwarding architecture for multihop wireless networks:** We have further investigated the basic inefficiencies associated with multihop packet forwarding over the 802.11 DCF MAC protocol and have proposed an Interface-contained forwarding architecture (ICF) to improve the packet forwarding efficiency in multihop networks. The ICF architecture enables a packet

to be forwarded entirely within the network interface card of the forwarding node without requiring per-packet intervention by the node's CPU. To effectively forward packets in a pipelined fashion without incurring the 802.11-related overheads of multiple independent channel accesses, a slightly modified version of the 802.11 MAC, called Data Driven Cut-through Multiple Access (DCMA) is introduced that uses labels in the control packets for faster next hop lookups, in conjunction with a combined ACK/RTS packet, to reduce 802.11 channel access latencies. Systematic evaluations demonstrate that DCMA outperforms 802.11 DCF and is able to extend the useful operating range of the network (with tolerable latencies)

**Cross layer metric for route selection:** Motivated by our observations and the shortcomings of existing cross layer metrics that have been studied for wireless networks, we have proposed a novel *airitime* metric that incorporates PHY rates, link losses as well as medium access delays and enables high throughput path selection in wireless networks with a low control overhead. The metric uses a combination of DATA traffic based measurements augmented with passive sensing based estimates in the absence of active DATA flows through the node to reduce the overheads associated with link quality estimation.

**Path-centric network design for efficient packet delivery in wireless networks:** The ICF approach mainly focusses on MAC and forwarding efficiencies assuming independent route selection. In order to minimize the interference from other flows, there are various dimensions that can be explored - multiple frequencies, spatial distribution to enable reuse and time-based flow coordination. We first focus on the spatial dimension first by proposing cross layer enabled cut through mechanism (CLEAR) that uses interference- aware route selection techniques based on "airtime" metric to distribute flows spatially for minimizing interflow interference. This increases the number of successful cut-through transfers, thereby improving end-to-end packet latencies. In order to accommodate high spatial density of flows, we further outline a time-based coordination scheme that

uses soft-reservations during route discovery phase to coordinate multihop "burst" mode transfers amongst flows. This model has also been extended to support differentiated services, thus providing a "low-latency socket" for real-time traffic over multiple hops.

A significant contribution of this thesis also involved the design and development of the 400-node ORBIT wireless testbed at Rutgers University as a part of the NSF Networking Research Testbeds Initiative to facilitate rapid prototype and experimental evaluations at scale. In particular, we have proposed a RF calibration protocol to characterize the commercial 802.11 hardware parameters and quantify the differences to enable repeatable experimentation. We have also reported throughput fairness issues caused by physical layer capture at the radio level. Various cross layer techniques have been proposed to address this issue. We are currently working on expanding the ORBIT control and experiment management infrastructure to enable conducting experiments that span heterogeneous networks: wired and wireless.

## 6.1 Future directions

Analytical studies have established upper bounds on the achievable performance using existing 802.11 technologies in multihop environments by joint scheduling, route selection and frequency assignments. This is essentially a three dimensional optimization involving coordination of time, space and frequencies. There is still a wide gap in terms of existing system designs and deployments based on 802.11 MAC and the promises of the advertised performance based on analytical evaluations. In the CLEAR architecture, we have explored the time and spatial dimensions to coordinate asynchronous traffic flows and have focussed on a path-centric design as compared to a "hop" centric model in existing networks.

We are currently exploring integrated routing and MAC scheduling architecture (IRMA) using

a global co-ordination plane [95] for reducing interference among flows and improving the utilization of the network. (Figure 6.1). This approach uses a TDMA based MAC with integrated route selection. We also propose a dedicated global control plane (separate from the data path) along with the associated protocols in order to effectively exchange the slot requests and assignments. While hardware changes may be necessary to accommodate this design, it is expected to yield significant improvements in the overall utilization of the network by enabling conflict-free switched paths for efficient data transfers.

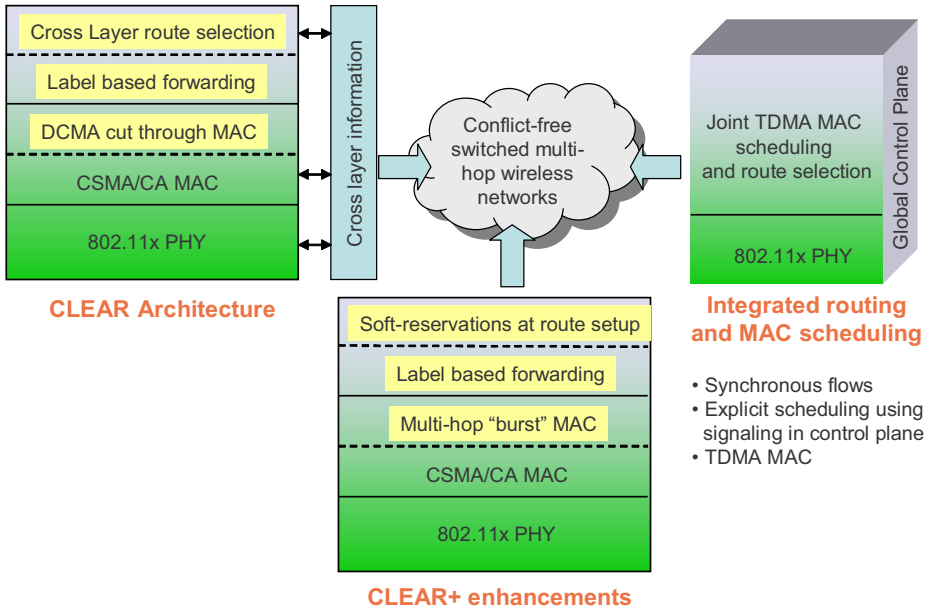



Figure 6.1: Towards conflict-free switched multihop wireless transfers

Along with multi-frequency enhancements to fully utilize the available spectrum, our work can be the basis for realizing the vision of truly switched wireless mesh network backhauls, that can support conflict-free data transfers over multiple frequencies, time slots as well as path segments. Such a network can efficiently utilize the increasing physical layer speeds and can be a competitive alternative for wired deployments in terms of cost and performance. Our contributions will hopefully bring us closer to this goal.

# Chapter 7

# Appendix

## 7.1 Introduction

Research on wireless network performance and protocol enhancements has been predominantly based on analytical and simulations models that are typically characterized by simplified radio interfaces and are unable to capture realistic physical layer effects commonly observed in actual live networks. This drawback can be attributed to the lack of easily available tools for modeling, emulation or rapid prototyping of a complete wireless network systems and protocols as concluded in [96]. Thus, there is an increasing need in the research community to be able to perform controlled experimental investigations of protocols and evaluations of system design using real-world wireless devices.

These considerations motivated the NSF sponsored Networking Research Testbed initiative which aims to provide a flexible, open-access multi-user experimental facility to support research on next-generation wireless networks. A significant part of this thesis involved the design and development of the ORBIT radio infrastructure [63] with 400 wireless nodes (the largest testbed of its kind) and support services in order to conduct experiments repeatedly with ease. In this chapter, we discuss the overall design and architecture of the ORBIT testbed which has been used to conduct the experimental evaluations in Section 2.5 and 4.2.

The organization of this chapter is as follows: First, an overview of our experimental testbed is

presented with key software and hardware components that enable rapid and repeated experimentation. We next focus on the factors that affect the repeatability of the experiment and propose a mechanism to calibrate the physical radio devices to minimize the experimental error caused by hardware differences. Using this testbed, we gain insights into important considerations for high density wireless network deployments. In particular, we focus on the physical layer *capture* phenomenon commonly observed in current radio hardware and investigate its impact on the throughput fairness of the network.

Our ongoing work proposes extensions to the ORBIT control and management framework to wired networking testbeds (in particular, PlanetLab [97]) to provide a global infrastructure for conducting networking experiments across diverse substrates with a single programming interface and experimental methodology.

## 7.2 The ORBIT Wireless Testbed

The radio grid emulator consists of 400 wireless nodes with 802.11a/b/g wireless cards arranged in a 20×20 grid with ∼1m spacing between nodes. Each node is connected to the backend via multiple high-speed Ethernet links to provide dedicated channels for control and management traffic. Users have full access to the radio nodes and can download and install customized operating system images as well as software packages. Support services enable remote rebooting of the nodes as well as access to the console for diagnostic purposes. The software packages include test traffic generators, libraries to provide access to radio level statistics from the hardware as well experimental data collection tools to facilitate measurement reporting and storage. Additionally, both hardware and software support is provided to enable users to create arbitrary topologies using noise injection mechanism described later in this section.

The ORBIT testbed primarily consists of the following key components

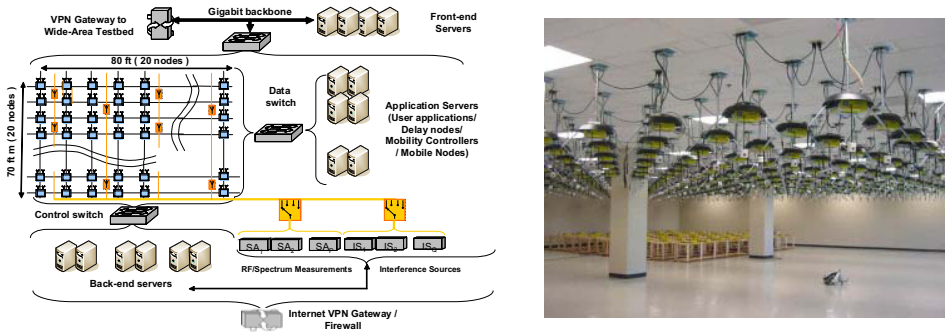



Figure 7.1: ORBIT: System Architecture and Actual Testbed

1. ***Radio Mapping*** that helps in mapping arbitrary topologies with specific dimensions into the ORBIT grid by trying to match the link characteristics of the original topology using controllable noise injection

2. ***ORBIT Hardware*** which is the main component of the grid and comprises ORBIT radio interference generators, spectrum monitors, 802.11 monitors as well as support servers

3. ***ORBIT Software and Services*** that allows quick deployment of custom OS onto nodes, traffic generation, measurement and statistic collection as well as overall experiment orchestration

### 7.2.1 Radio Mapping

The primary goal of the radio mapping algorithm is help create controlled topologies on the radio grid that emulate typical wireless scenarios such as office environments, hotspots etc. as shown in Figure 7.2

Based on the characteristics of the indoor testbed, the mapping algorithm investigates the best location and transmission power for the interferer node in order to inject the desired amount of interference to create the required SNR characteristics corresponding to the scenario to be mapped. Note that this can be done using empirical measurements or profiling of the radio grid using radio

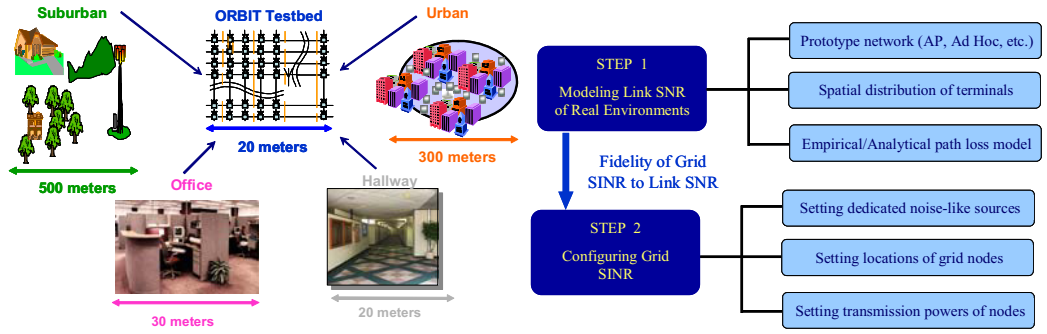



Figure 7.2: ORBIT: Topology mapping into physical dimensions of the grid

sources at fixed locations and power levels at each position in the grid and measuring the corresponding SNR of the link between this source and arbitrary positions on the grid. Further details can be found in [98].

Currently, ORBIT hardware supports injection of AWGN noise at desired power levels and center frequencies using Agilent Vector Signal Generator using distributed noise antennas positioned at appropriate positions on the grid.

### 7.2.2 Hardware components

**ORBIT radio nodes:** The radio nodes serve as the primary platform for user experiments and consists of the following hardware components (as shown in Figure 7.3)

1. 1-GHz VIA C3 processor with 512 MB of RAM and a 20 GB local hard disk

2. two Atheros wireless mini-PCI 802.11a/b/g interfaces

3. two Gigabit Ethernet interfaces for experimental data and control respectively

4. Integrated chassis manager (CM) to remotely monitor the status of each radio node's hardware and enable remote reset, powering on/off of the nodes

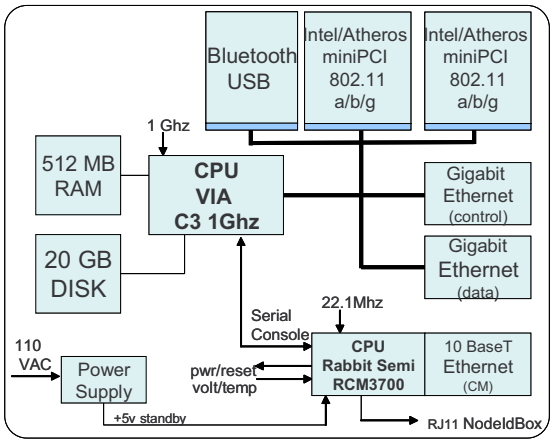



Figure 7.3: ORBIT Radio Node

**Instrumentation Subsystem:** The instrumentation system (Figure 7.4a.) provides capabilities for creating and injecting various types of artificial RF interference (white noise, colored noise, microwave oven like noise etc.) at different locations on the grid. The interference generator is based on RF Vector Signal Generator while the spectrum measurements are done using Vector Signal Analyzers. The goal is to provide the ability to create arbitrary topologies by injecting controlled noise at different positions on the grid.

**Independent 802.11 Monitoring Subsystem:** This system consists of distributed 802.11 frame sniffers that provide a MAC/network layer view of the radio grid's components. The observed traffic is recorded in a database and can be used to independently validate experimental observations in case of any discrepancies.

**Support Servers and sandboxes:** This setup consists of machines to host the various services as well as sets of sandboxes (comprising two ORBIT nodes connected through RF cables as shown in Figure 7.4) for initial testing and development of experiment software prior to experimentation on the main grid. Sandboxes use the same hardware as the grid, hence users can easily migrate their applications to the actual grid after testing on the sandboxes.

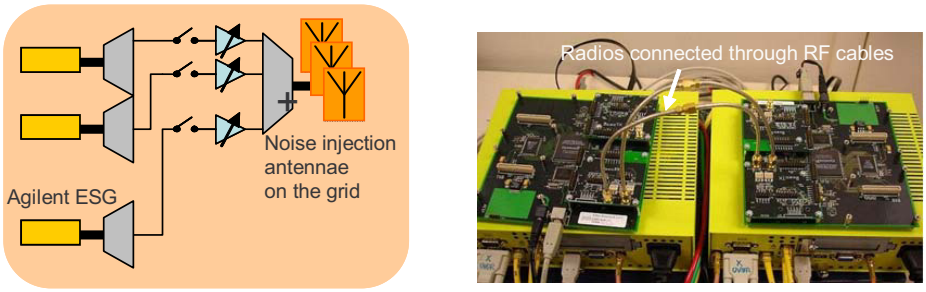



Figure 7.4: a. ORBIT Instrumentation system b. ORBIT Sandbox system

### 7.2.3 Software components

Software packages and libraries have been developed to support both application/protocol evaluations. These include common libraries for traffic generation, measurement collection etc. and also provide easy hooks to enable "expert" users to develop their own applications and protocol stacks on the testbed. A few key components as shown in Figure 7.5 are described below and further details can be found in [99].

**Node Handler:** The purpose of the Node Handler is to disseminate experiment scripts using multicast to the Node Agents residing on the individual nodes, in order to orchestrate the experiment. The Node Handler is Ruby-based and processes the experiment script, keeps track of the experiment steps and events, and sends them to the involved Node Agents at the appropriate time. The Node Agent reports back the state of experiment command execution to the Node Handler.

**Node Agent:** This is the client-side component of NodeHandler that resides on the ORBIT nodes and listens to commands from the ORBIT Node Handler. It can run and stop the applications, dynamically pass the parameters to the applications, and report the experiment state to the controller.

**Orbit Measurement Library (OML)** The purpose of the OML framework [100] is to collect the reported measurements during the experiment at run-time using different granularity (e.g time-based or sample-based). The nodes collect the measurements and send them to the server using

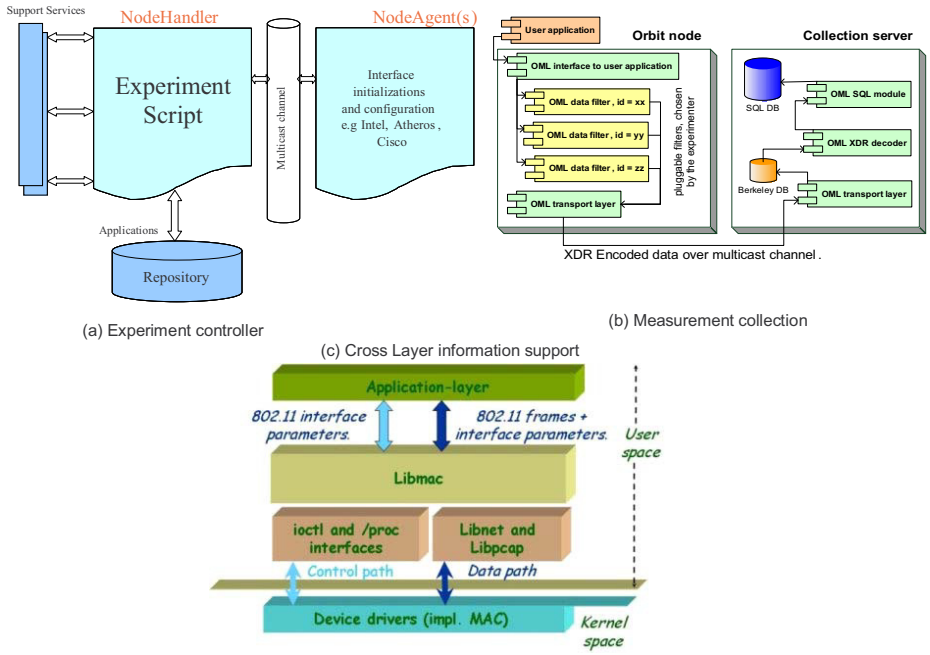



Figure 7.5: ORBIT Software components

XDR format [101] which provides a type-safe mechanism for handling and transporting different data types. This reduces the burden of statistic collection and prevents duplication of effort on the part of experimenters.

**Disk-Imaging:** The purpose of the disk-imaging service is to enable quick installation of custom operating systems on the nodes as per the choice of the experimenters. This service works over a reliable multicast session using Frisbee [102] and allows for different groups of nodes to load different OS images simultaneously between experiments.

**Libmac:** Libmac is a custom user-space C library that allows the applications to inject and capture MAC layer frames. It also allows manipulation of wireless parameters such as transmit power, channel settings and recording RSSI, noise and PHY rate on an aggregate and a per-packet basis. The primary purpose of this library is to provide a bridge between device drivers and the applications such that application developers can easily use a standard interface to communicate

with wireless device drivers. Currently, the Madwifi driver for Atheros chipsets and the Intel driver for Intel wireless cards have been cuztomized for use with Libmac.

### 7.2.4 Experiment Lifecycle

A typical ORBIT experiment involves experiment definition, node assignment, node configuration, loading of software packages, configuration of dynamic parameters and data collection. As shown in Figure 7.6, the following steps are typically involved in an experiment

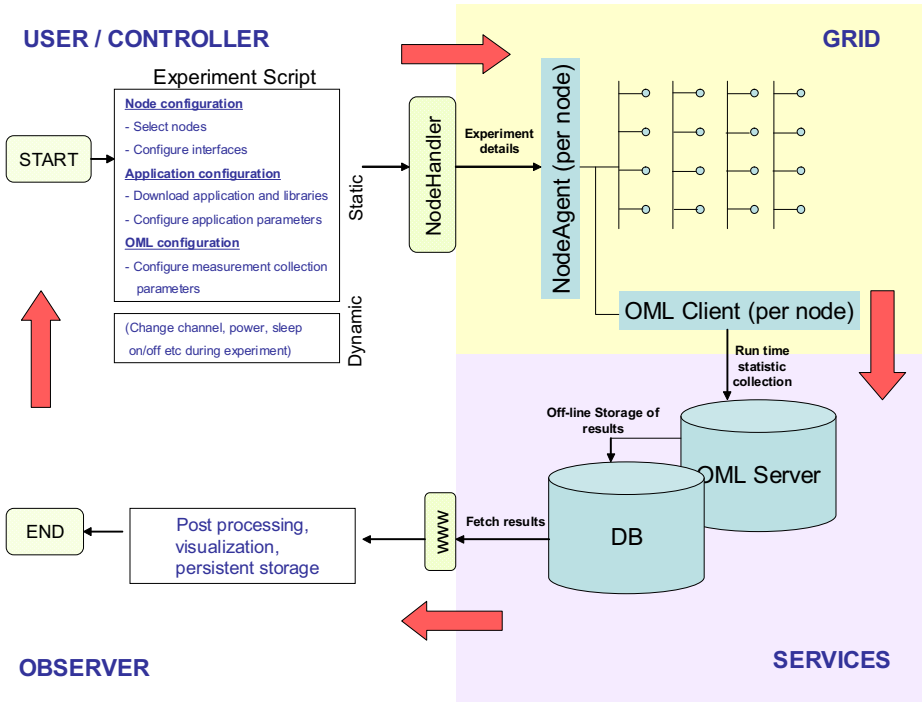



Figure 7.6: ORBIT Experiment Lifecycle

1. The experiment details are translated into a script that identifies the nodes to be assigned for the experiment, configures the wired and wireless interfaces according to the requirements of the experiment, fetches the appropriate application, libraries required to run the experiment and specifies (optional) statistic collection points and intervals

2. This information is disseminated by the NodeHandler software to the corresponding NodeAgent residing on each node

3. The NodeAgent executes the script, performs the experiment which may involve statistics collection done by the OML library

4. A separate run-time and post-experiment database allows users to quickly view results during experiment run-time as well archive them for future retrievals and off-line analysis

A sample experiment script is shown in Figure 7.7. In this experiment, node1-2 sends UDP datagrams of 1024 bytes at the rate of 300 Kbps to the receiver1-4. The wireless settings use 802.11b with the receiver acting as an AP (this is done using the ”Master” mode on the card) and the sender is the client (using the setting ”Managed” on the card

```
Experiment.name = "tutorial-1"
Experiment.project = "orbit:tutorial"

# Define settings used in the experiment

defProperty('rate', 300, 'Sender rate in KBps')
defProperty('packetSize', 1024, 'Payload length')

# Define nodes used in experiment

defNodes('sender', [1,2]) {|node|
# assume the right image to be on disk
  node.image = nil
  node.prototype("test:proto:sender", {
    'destinationHost' => '192.168.1.4',
    'packetSize' => Experiment.property
                    ("packetSize"),
    'rate' => Experiment.property("rate"),
    'protocol' => 'udp'
  })
  node.net.w0.mode = "managed"
}

defNodes('receiver', [1,4]) {|node|
  # assume the right image to be on disk
  node.image = nil
  node.prototype("test:proto:receiver" , {
    'hostname' => '192.168.1.4',
    'protocol' => 'udp'
  })
  node.net.w0.mode = "master"
}
```

```
# Wireless interface configurations
allNodes.net.w0 {|w|
  w.type = 'b'
  w.essid = "helloworld"
  w.ip = "%192.168.%x.%y"
}
# Now, start the application
whenAllInstalled() {|node|
  Experiment.props.packetSize = 1024
  Experiment.props.rate = 300
  allNodes.startApplications
  wait 60
  allNodes.stopApplications
  wait 10
  Experiment.done
}
```

Figure 7.7: ORBIT Sample Experiment Script

So far, we have looked at the overall ORBIT testbed infrastructure which provides hardware access to 400 wireless nodes and software services for deployment of custom software, experiment

orchestration as well as measurement collection. In addition to the general design to support open access experimentation, specific contributions of the thesis include the following.

The script-based execution described before allows the experimenter to capture the different settings used in the experiment for repetition at a later stage. However, facilitating repeatable experiments especially in wireless environments and with commercially available radio hardware poses interesting challenges. We have studied the behavior of different hardware devices and outline a methodology to address the problem of repeatability in wireless experiment.

For cross-layer wireless protocol design and experimental evaluation, it is essential to have the ability to control settings and observe effects at different layers of the protocol stack. The settings include packet sizes and packet rates at the application layer, windows sizes at the transport layer, routing metrics at the network layer as well as control of MAC and PHY layer parameters such as transmit power, frequency of operation, MAC contention window adjustments. We have specifically looked at the capture effect observed in the radio hardware on the ORBIT testbed that causes a stronger packet to be decoded in the event of collision and results in throughput unfairness in the system. Various cross layer parameters have been studied to mitigate the unfairness.

The ORBIT experimental testbed has been specifically used for building a prototype of the SOHAN hierarchical network architecture as described in Section 2.5 and gaining valuable insights into the design of forwarding nodes and protocol design.

## 7.3 Addressing repeatability in wireless experimentation

The key aspect in the experiments on the ORBIT testbed is the ability to control and measure important network properties, such as transmit power, throughput, or error rate, accurately, reproducibly, and quickly enough to characterize complex systems. This is an important factor to consider for any testbed that uses commercially available hardware and does not have anechoic environments to

guarantee RF isolation, which are expensive to provide for 400 nodes.

### 7.3.1 Parameters affecting repeatability

There are several factors in a wireless testbed that may affect the repeatability of experiments and reproducibility of results.

1. First, there are significant differences in commodity radio hardware devices which can be attributed to low-cost design constraints in these products intended for the Industrial, Scientific and Medical (ISM) band.

2. Additional issues include broad tolerances and ageing for low cost components. Also, during the lifetime of the testbed, there may be a need to replace wireless cards that malfunction or are superseded by new models. This may lead to differences in experimental results over time.

3. There is also a possibility of subtle software or firmware bugs that may manifest as inconsistent experimental results.

4. Finally, for a wireless testbed, the environment poses the biggest challenge to repeatability due to uncontrolled interference over time and space. This could be due to interference from co-located infrastructure access points, movement of people, opening and closing of doors

We have observed and reported several inconsistencies with the radio cards in terms of their transmit power settings, RSSI measurements in [103]. We propose a comprehensive calibration procedure to identify and document these inconsistencies that can be fed back into the experiment script appropriately. In addition, by using the script based experimental execution, the experiment settings (at applications level, wireless level etc) are automatically saved. Users can thus repeat experiments easily with the same settings or compare the differences with the previous experiments

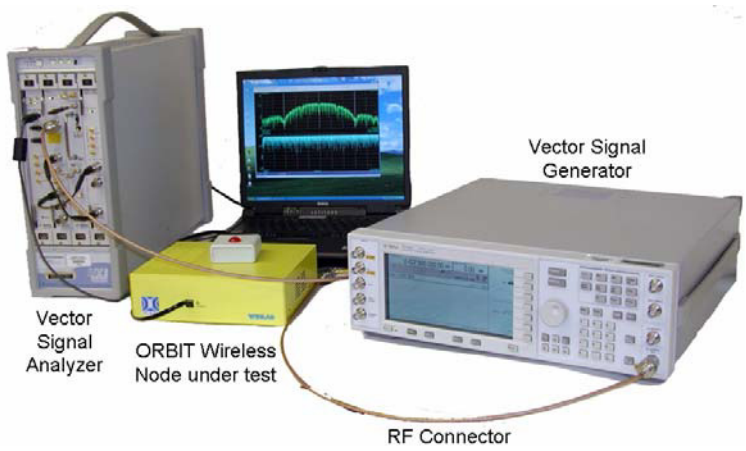


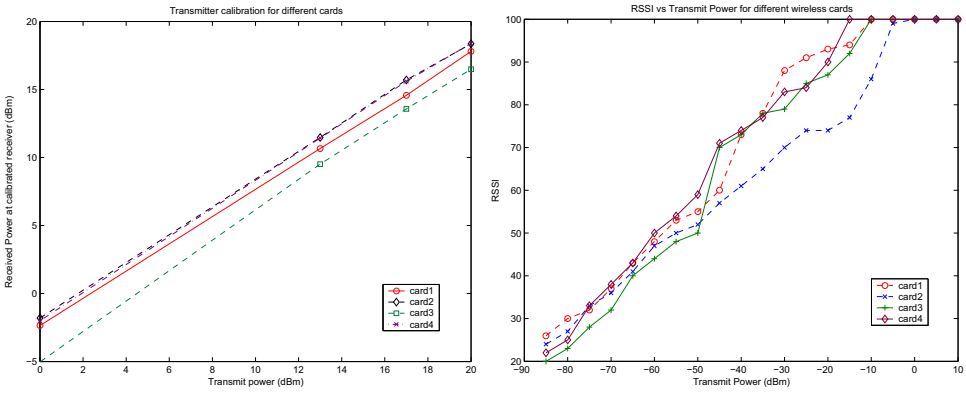

Figure 7.8: ORBIT Node Calibration Setup

by changing the settings in experimental runs thereby reducing any unintentional errors on the part of the experimenter.

### 7.3.2 Calibration procedure

**Transmit side calibration:** In order to calibrate the transmitting side of each card, we use Agilent 89600S Vector Signal Analyzer (VSA) [104] as the calibrated receiver (Figure 7.8). The output of the cards is connected through an RF-cable and a pair of connectors (with known attenuation) into the front end of the VSA. The transmitting card is fixed on a particular channel and generates a continuous stream packets cycling through different transmit power levels. The VSA measures the corresponding received band power for the bandwidth corresponding to the channel for each of the transmitter power settings. These values are recorded in a database indexed by the MAC address of the card.

**Receive side calibration** The receiver side is calibrated by using Agilent E4438C Vector Signal Generator (ESG) [83] as the calibrated transmitter. The ESG supports the capability to injected modulated (1Mbps or 6Mbps rates) 802.11 packets (with custom payload) at a desired frequency




(a) Transmit calibration (b) Receive side calibration

Figure 7.9: Transmit and receive side calibration for cards under test

and power level. We used this feature to generate and transmit test beacons at precise frequencies and powers to exercise the entire range of RSSI measurements reported by the card. As before, the card under test is connected to the ESG using an RF-cable with a known fixed attenuation.

Our preliminary experiments expose the differences in the card behavior and also confirm the accuracy and the exact mapping between RSSI and dBm values for the device under test. This can be immensely useful to the research community that heavily relies on the relative statistics reported by the drivers and design cross layer adaptations based on these readings.

As seen in Figure 7.9(a), the received power from cards 1, 2 and 4 matches their corresponding transmit power settings (after taking into account the 2 dB RF-cable attenuation loss). However, there is a slight deviation from this trend for card 3. The received power for this card is about 3 dB lower than the other three cards. Figure 7.9(b)shows the reported RSSI values by each card for each of the transmit power settings. All the cards are unable to receive packets below a VSG transmit power of -88 dBm (which is equal to -90 dBm at the front end of the card taking into account the 2 dB RF cable and connectors' loss). This roughly corresponds to the receiver sensitivity of each

card. Note that while cards 1, 3 and 4 report similar RSSI values for different transmit powers at the ESG, the RSSI readings reported by card 2 are as much as 10dB lower than the rest for some ranges of the transmit power.

It is precisely this information that we intend to capture for each card and store in the form of a correction factor to be applied during the experiments. The reason for this deviation could be attributed to the ageing of the components as well as differences in their tolerance levels. However, it needs to be accounted for in order to support repeatability in experimental results. We have conducted experiments across different time intervals, days as well as using different set of nodes while maintaining rotational or translational symmetry. Our initial findings [103] motivate the need to feedback the calibration information into the experiment lifecycle in order to accurately account for the hardware specific inconsistencies.

### 7.3.3 Characterizing the repeatability of the existing testbed setup

We conducted to measure repeatability of results in our initial testbed setup in an environment that is not optimized for RF stability. This includes identical experiments conducted over the span of a month (in order to capture time variations) and also on different sets of nodes, while maintaining the same topology (in order to capture the spatial effects and other hardware issues).

**Temporal repeatability:** To investigate repeatability of results, we conducted the same experiment at random times over an extended period of about a month. In this section, we report the results for five sample runs chosen out of this duration ensuring that they span across a time period of a month. To reduce the scope of experimental error, we used the same set of nodes, same wireless cards and the same settings for each of these experiments for the entire duration. Over that period, there were some changes in the physical environment and positioning of the nodes that contribute to any changes noted. The experimental setup, as shown in Figure 7.10, consisted of 7

nodes, with a sender sending UDP packets of 1024 bytes to a receiver that formed the Link Under Test (LUT). Five other interfering nodes broadcasted UDP packets (1024 bytes) on the same channel as the sender-receiver pair. Both the sender and all interferers transmit at 1 mW. All the nodes are configured to be on Channel 1 initially. In order to combat interference, the channel used by the LUT is incremented one channel at a time until it operates on a completely orthogonal channel (Channel 6). We observe the effect on the throughput of the LUT as it is moved to an orthogonal channel away from the interferers. The LUT dwells on each channel for 30 seconds. Hence, the entire experiment duration is 180 seconds.

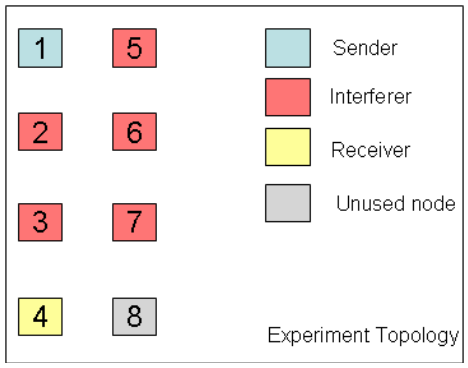
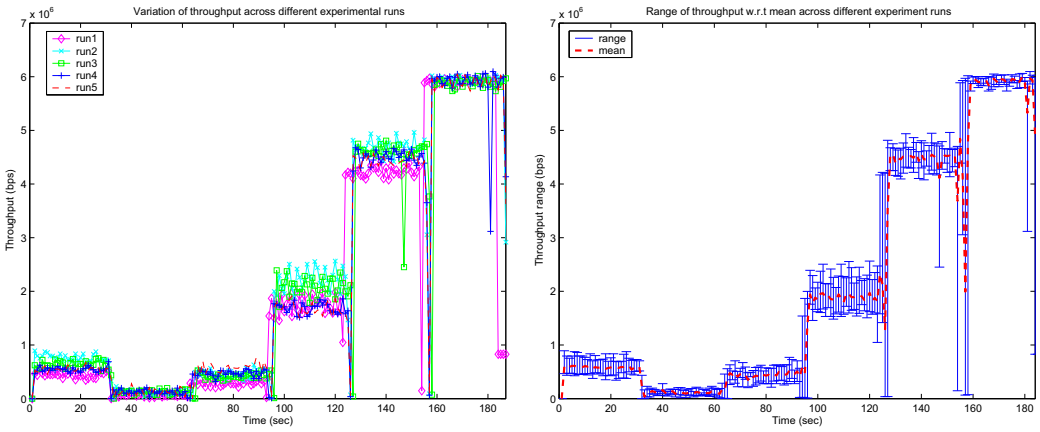



Figure 7.10: Experimental setup for testing temporal repeatability

Figures 7.11 show the throughput for each run of the same experiment and the maximum deviation of throughput amongst different experimental runs with respect to the mean throughput. It is seen that the differences are slightly greater when channel separation is 3 (partial channel overlap) corresponding to the time interval of 90-120 seconds. It is much smaller for the cases when channel separation is 0 (0 to 30 seconds) or greater than 4 (150-180). These cases correspond to LUT being on the same channel as the interferers or on an orthogonal channel respectively. Note that the concept of orthogonality is only valid for perfectly linear transmitters and receivers. Given the variability observed in these cards, it is unlikely that strict linearity will be achieved in these

low-cost devices and thus a power dependence of these results is expected and needs to be included in any calibration strategy.




(a) Throughput across multiple runs (b) Variation w.r.t mean throughput

Figure 7.11: Temporal repeatability results

**Spatial repeatability:** Another concern regarding testbed operation is whether different (symmetric) assignment of nodes for different experiment runs produces similar results. In order to study the effects of node positions on the outcome of the experiment, we performed a simple test across twelve different node topologies in the three basic arrangements as shown in Figure 7.12. In each run, we used four nodes: two senders and two receivers. The senders operated at 1 mW transmit power, on channel 1, using 1280 bytes UDP packets (40 packets/sec) for an offered load of 409.6 kbps per flow. Each experiment was conducted for 60 seconds. For each of the basic arrangements, we rotated the topology four times giving us a total of twelve experimental runs. Since, the two flows were also symmetric in terms of offered load, the total number of sample runs for the experiment was 12 topologies  2 flows = 24 samples runs.

In Figure 7.13a., we show the variations of throughput with respect to the mean taken over the entire duration of the experiment. For each second on the X-axis, we found the average throughput

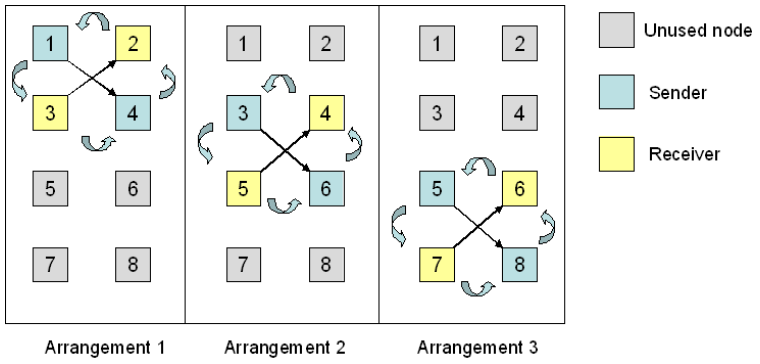
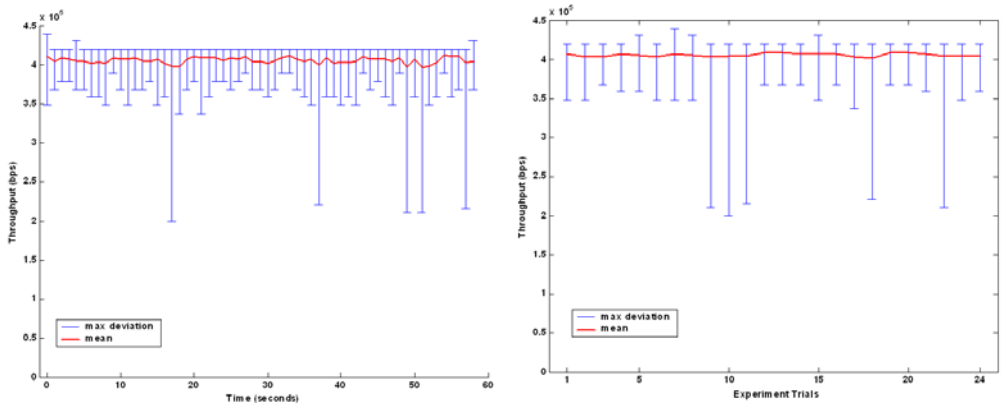



Figure 7.12: Experimental setup for testing spatial repeatability

and the maximum deviation from the average throughput using the 24 sample runs. Figure 7.13b. shows the results from a per experiment perspective. Here, we show the throughput averaged over a single experiment duration for each of the 24 different sample runs, and the maximum deviation from this mean.




(a) Throughput across multiple runs (b) Variation w.r.t. mean throughput

Figure 7.13: Spatial repeatability results

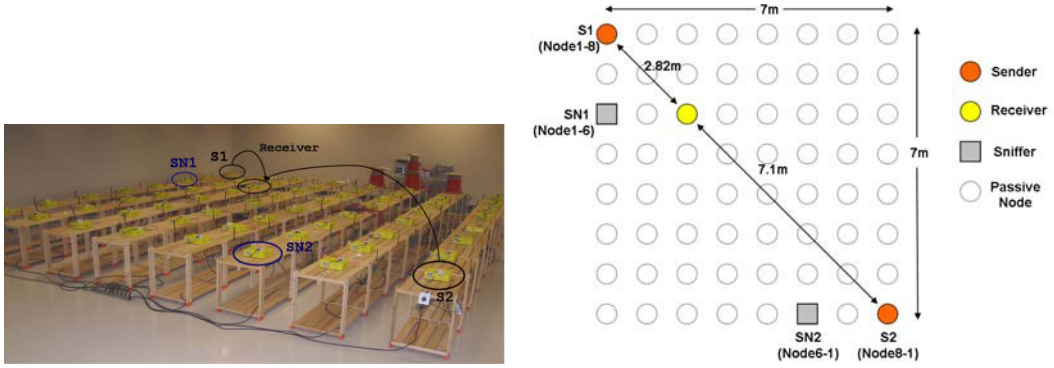
The primary observation we can make from these experiments is that over time periods of weeks, measurement variations associated with environmental changes or drift in the equipments is that they are demonstrably non-Gaussian. We have included the standard deviation for simplicity and

further work may be needed to understand the distribution better. However, these observed variations are still less than the initial differences between the cards. This gives us confidence that a calibration procedure describe earlier can be used to substantially improve the repeatability of the measurements, and thus the utility of the testbed. We next focus on hardware specific characteristics that can dominate the system performance in high density wireless network deployments such as the ORBIT testbed.

## 7.4 Physical layer capture effect: Mitigating throughput unfairness

An interesting observation related to the radio hardware used in the ORBIT testbed is *physical layer capture effect* (PLC). In this case, in the event of a collision between two frames at a receiver, the hardware is capable of detecting and decoding the packet with a stronger signal strength. This effect has been observed with multiple wireless NICs based on different chipsets (Atheros and Prism), occurring even in small setups (about 10m separation) with line-of-sight communications and is not usually modeled correctly in existing simulation tools (as shown in [105]). In this work, we experimentally investigated the physical layer capture effect in off-the-shelf 802.11 network cards and confirmed that it reduces throughput fairness of traffic flows. We then studied the feasibility of using several PHY and MAC layer approaches to mitigate the disproportionate allocation of throughput in capture dominated scenarios namely transmit power control, retransmission limits, CWmin adjustment, TxOp adjustment, and AIFS control.

In order to detect the presence of physical layer capture phenomenon, we used the node positions as seen in Figure 7.14(a). The sender, S1 is closer to the receiver, R as compared to sender, S2 with the ratio of distances to the receiver 1:2.5 for senders, S1 and S2 respectively. These senders send packets of different payload lengths (256 bytes, 512 bytes and 1024 bytes) to the receiver at an offered load of 7 Mbps over an 802.11b link. The PHY rates were adapted based on the default




(a) Setup (b) Layout

Figure 7.14: Experiment setup to study capture effect

rate adaptation algorithm implemented in the driver. One sniffer near each sender (as shown in Figure 7.14(b)) was chosen such that the signal strength or RSSI of packets received from this sender is higher than that of frames received from any other sender. The reasoning behind this placement is that a sniffer is also a regular radio receiver susceptible to the capture phenomenon.

### 7.4.1 Detecting capture effect

The primary difference between our technique and the one proposed in [105] is the use of a feature provided by Atheros cards - a station can perform "live monitoring"[1] and observe WLAN traffic while still being synchronized with the rest of the stations in the network. This implies that the logs from each of the sniffers do not have to be explicitly "synchronized"; they can be merged directly based on the hardware timestamp of each received frame. We used *tcpdump* on the sniffers and processed the collected information using *awk* scripts.

Figure 7.15 shows a snapshot from one of our traces that demonstrates the capture phenomenon.

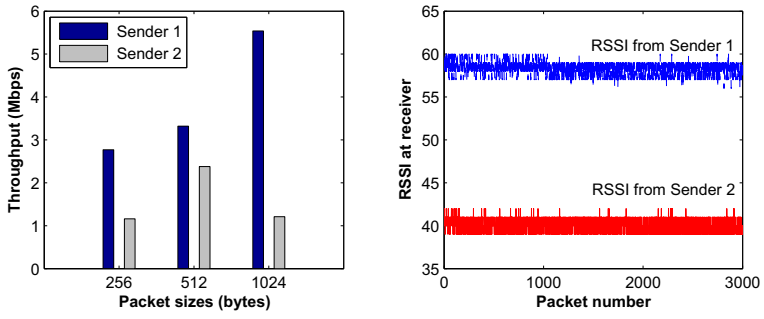
[1]The driver provides a separate virtual network interface, called ath0raw, which can be used to send/ receive frames directly to/from the card from user-space (bypassing the driver state machine). This interface can be enabled using the commands: *sysctl -w dev.ath0.rawdev=1; ifconfig ath0raw up;*

| Time | Frame Type | Frame Size | Source IP Address | Destination IP Address | Seq. No. |
|---|---|---|---|---|---|
| 737856416 | Data | 1088 | 192.168.1.8 | 192.168.3.6 | 476 |
| 737856532 | Ack | 14 | | 192.168.8.1 | |
| **737857611** | **Data** | **1088** | **192.168.8.1** | **192.168.3.6** | **726** |
| **737857612** | **Data** | **1088** | **192.168.1.8** | **192.168.3.6** | **477** |
| **737857729** | **Ack** | **14** | | **192.168.1.8** | |
| 737858633 | Data | 1088 | 192.168.1.8 | 192.168.3.6 | 478 |
| 737858749 | Ack | 14 | | 192.168.1.8 | |

Figure 7.15: Collision detection - the highlighted rows represent collision and subsequent capture. The two frames are received 1$\mu$s apart but an acknowledgement is sent to the stronger sender.

From these merged traces, we can see that frames collided because they picked the same time slot for transmission and an 802.11 acknowledgement was sent back for one of the senders implying that the stronger frame was correctly decoded.

Using the same experimental settings as described earlier, we measured the throughput unfairness caused by PLC. We used the Iperf traffic generator [89] to generate UDP traffic at each transmitter. Each sender uses an offered load of 8 Mbps and we monitor the flow throughputs for different packet sizes. We used packet sizes of 256, 512 and 1024 bytes for this experiment. For each test, both senders used the same CWmin (default set to 31).




(a) (b)

As seen in Fig 7.4.1 (a), there is significant unfairness in the throughputs of sender S1 and S2 at the receiver. Unfairness is higher for the larger packet sizes (1024 bytes). The observed RSSIs of

each sender plotted in Fig 7.4.1 (b) show that S1 is received almost 20 RSSI units stronger than S2.

### 7.4.2 Mitigating the throughput unfairness using cross layer techniques

In order to restore fairness caused by PLC, we experimentally evaluate various approaches that span both PHY layer as well as MAC layer adjustments. In particular, we look at the following knobs and their effectiveness in restoring fairness.

- Transmission power control (Physical Layer)
- Retransmissions (MAC)
- 802.11e QoS Parameters
  - CWmin (MAC) (default = 31)
  - TxOP (MAC) (default = 1 packet per attempt)
  - AIFS (MAC) (default = DIFS)

**Transmit power control:** The first approach to mitigate unfairness is to reduce the transmit power of the sender whose signal strength is stronger at the receiver. We configure the transmit power of the stronger sender from 60mW (18dBm) down to 1 mW (0 dBm) with two intermediate power levels of 30 mW (14.7 dBm) and 15 mW (11.7 dBm)[2].

As seen in Figure 7.16, transmit power control at the stronger sender reduces the gap between the two flow throughputs as well as the signal strength difference at the receiver from the two senders. A possible explanation is that since the difference in RSSI (and hence SNR) at the receiver from the two senders is lower, the probability of capture of the stronger sender is reduced. This results in an improvement in throughput for the weaker sender. However, using transmit power control alone, we

[2]For the MadWifi driver, we write to a file in the /proc directory (*/proc/sys/dev/ath0/txpowlimit;*) using the *echo* command. The values are from 1 to 100 in milliwatts which translate to 0 to 18dBm (clamped at 18 dBm).

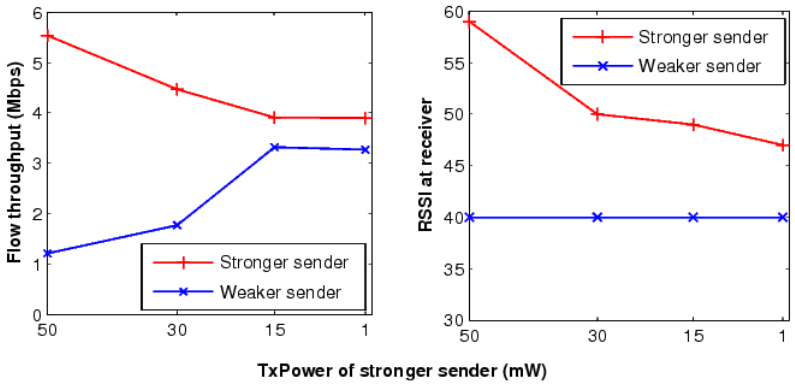



Figure 7.16: Throughput distribution and RSSI at the receiver with transmission power control at the stronger sender

were unable to restore fairness between the flows because of the limited dynamic range of allowable power level settings. Typically, most of the current hardware devices available off the shelf do not allow power levels below 1 mW or 0 dBm. Additionally, there at the time of the experiments there was no hardware support for per-packet transmit power adaptation and only certain discrete power levels are allowed, thereby limiting the granularity of control.

**Adjusting MAC retry limit:** Due to PLC, the weaker station has to retry packets that collided and were dropped by the receiver. According to the 802.11 standard, this station doubles its contention window for each unsuccessful attempt and defers until the CW counts down to zero. This significantly reduces the amount of data traffic that the station can send.

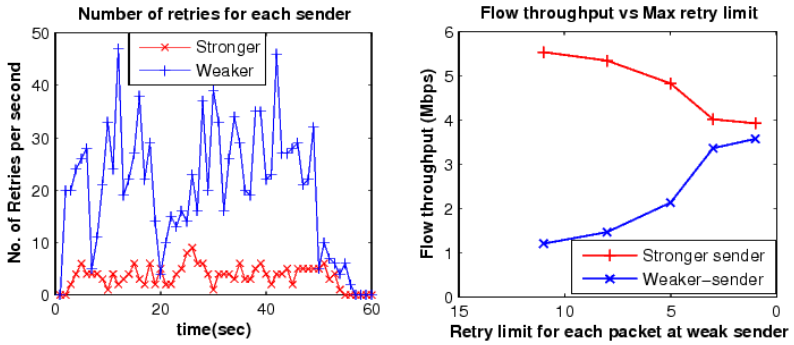



Figure 7.17: Number of retransmission attempts per second at each sender during the experiment duration. The unfairness reduced with the number of max. retries for the weaker sender

In our experiments, we measured the cumulative number of retries by each sender (reported per second) over the entire experiment duration. As seen in Figure 7.17(a), the weaker sender encounters four times more retransmissions than the stronger sender, on average. We varied the maximum number of transmission attempts per packet at the weaker sender from the default setting of eleven to one (no retries) based on driver changes. As seen in Figure 7.17(b), as the retry limit is decreased, the contention window stops growing after reaching the allowed retries. Thus, the weaker sender saves time on the backoff before attempting to transmit the next packet. As a result, it is able to attempt the delivery of the next packet resulting in a higher UDP throughput. This trend is seen for all the packet sizes that we studied.

**Adjusting minimum contention window size**

The basic idea behind adapting the minimum contention window is to increase the likelihood of channel access for the weaker sender (based on the probabilistic assumption that the weaker sender will, on the average, select earlier slots than the stronger one). We tried to set arbitrary values for the CWmin values that were not powers of two, however our observation was that the firmware rounds it off to the next higher power of two, thereby restricting our dynamic range of control.

In Figure 7.18, the numbers in the brackets represent the tuple (CWminSS, CWminWS) where SS and WS imply the strong sender and the weak sender respectively. For each packet size, reducing the CWmin of the weaker sender increases its share of throughput. This is seen for the setting (31,15) in each case. However, reducing CWmin further tends to overcorrect the unfairness as seen in the (31,7) case for each packet size. We also increased the CWmin for the stronger sender to 63 while keeping the default CWmin for the weaker sender. This is represented by the (63, 31) column for each packet size. Even though the flow throughputs are more proportionate for this setting, we see that it results in a reduction in the overall system throughput because of inefficient use of the channel.

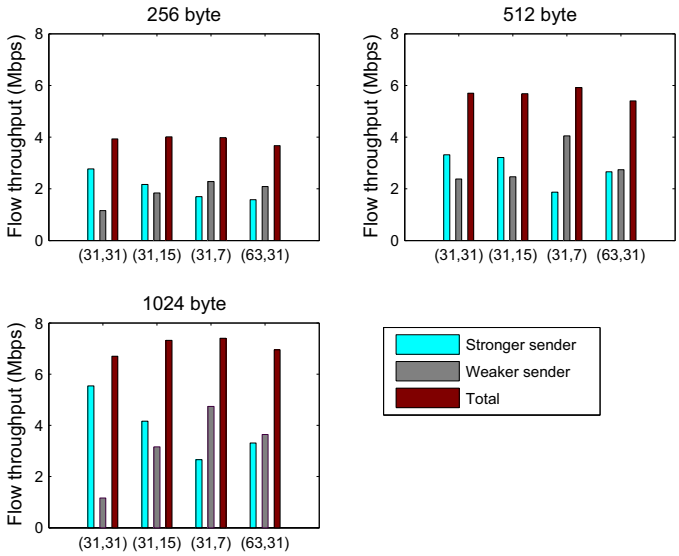



Figure 7.18: Flow throughputs for different packet sizes with different CWmin combinations

**Adjusting TxOp**

The IEEE 802.11e standard [26] provides TxOp (Transmission Opportunity in units of $\mu$seconds) for each class of service. This allows stations to send more than one packet separated by SIFS during their channel accesses instead of having to contend for the medium for every packet. By default, the transmit opportunity is set to one packet per channel access. Under ideal conditions, the two flows should contend equally for the channel and gain equal amounts of time to transmit data. However, in the event of collisions, capture and retransmissions, this time sharing of the channel is disproportionate. In order to rectify the problem, we varied the TxOp parameter for the weaker sender roughly in units of time required to transmit one packet of the given size. We only present the results for the 1024 byte packets due to space limitations.

The total transmission time for a 1024 byte packet (with additional 28 byte MAC header + 8 byte SAP/SNAP header + 20 byte IP header + 8 byte UDP header) using the short preamble option is around 911 $\mu$seconds. Also, the station has to wait for DIFS interval and an additional deferral time before it can send the first packet. In our experiments, we used normalized TxOp of 2 and 3

packets per channel access for the weaker sender (corresponding to 2 ms and 3 ms respectively).

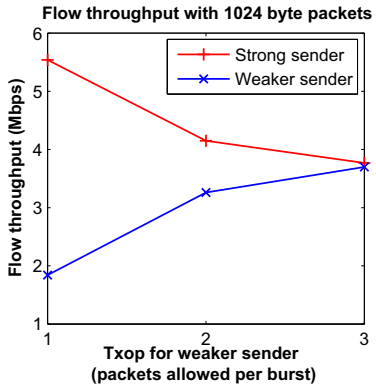



Figure 7.19: Flow throughputs as a function of TxOp for weaker sender

In [106], the authors have reported a linear relationship between throughput and TxOp. However, in our capture dominated environment, we found that the throughput increases much slower beyond TxOp = 2. As shown in Figure 7.19, by setting TxOp = 3 packets per channel access for the weaker sender, we restored throughput fairness. To gain further accuracy, we propose that the proportion of time spent by each flow on the channel should be measured and the TxOp of the weaker sender should be appropriately chosen to balance this ratio.

**Adjusting AIFS**

AIFS (Arbitration inter-frame spacing) is equivalent to DIFS in the 802.11b standard and represents the minimum mandatory spacing between two frames in addition to the deferral time. Prior to sending a packet, each station waits a fixed interval plus an additional randomly chosen interval from (0, CW). By decreasing the AIFS for the weaker sender, we prioritize its transmissions over those of the stronger sender and thus reduce the number of collisions.In our experiments, we varied the AIFS for the strong sender as shown in the Figure 7.20. For the 1024 and 512 byte packet sizes, AIFS values around DIFS +12 slot times for the stronger sender resulted in fair throughput allocation. For the smaller packet size (256 bytes), this balance was achieved further away at AIFS values of around DIFS+17 slot times for the strong sender.

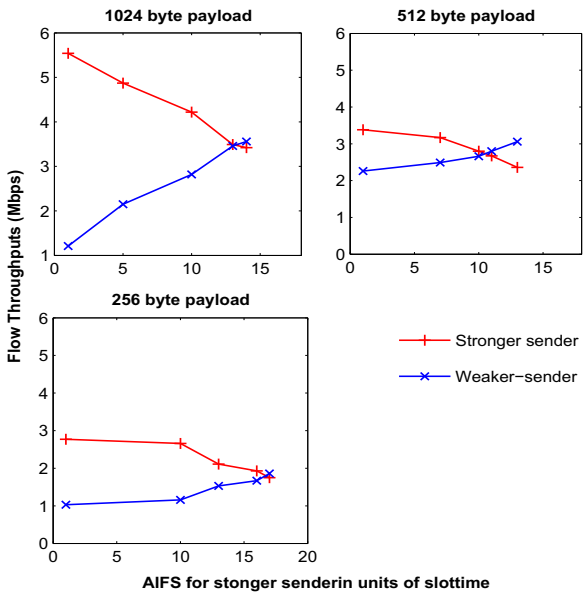



Figure 7.20: Effect of AIFS on the flow throughputs

**Summary of observations and comparison of each approach**

We summarize our observations for each adaptation mechanism and also compare throughput fairness achieved by each approach. Our findings suggest that

- Reducing the transmission power of the strong sender may achieve fairness; however the adjustment is limited by the discrete power levels allowed by the underlying hardware device.
- Reducing the number of retransmissions of the weaker sender helps; this may be useful for applications that are tolerant to packet loss.
- Increasing CWmin of stronger sender may be better than reducing CWmin of weaker sender due to reduced number of collisions in the former case. However, CWmin control is restricted to 10 settings (strictly in powers of 2) and hence we cannot achieve fine grained control.
- Increasing TxOp for the weaker sender allows increased number of packet transmissions per channel access. Also, it allows finer granularity of control as compared to the previous approaches.

- Increasing AIFS for the stronger sender achieves the desired throughput fairness due to reduced number of collisions.

Table 7.1 summarizes the flow throughput before and after each adjustment. It can be seen that TxOp control provides the greatest benefit in terms of mitigating unfairness. This initial study is used to motivate joint adaptation for further improvements and is described next.

Table 7.1: Fairness achieved by each method

| **Method** | **Flow 1 Throughput (Mbps)** | **Flow 2 Throughput (Mbps)** |
|---|---|---|
| Default | 5.54 | 1.21 |
| TxPower | 3.9 | 3.27 |
| Retries | 3.93 | 3.58 |
| CWmin | 3.31 | 3.64 |
| TxOp | 3.77 | 3.7 |
| AIFS | 3.49 | 3.46 |

**Multiple flows and joint adaptation**

Based on our observations, we studied the fairness behavior of five different flows chosen such that two out of the five senders had a significantly weaker RSSI at the receiver (approx. 20 units lesser than the stronger stations). Each sender always had a packet to transmit. We used 802.11b channel 1 with fixed rate setting of 11 Mbps. The RSSI and throughput distribution for each flow are shown in Figure 7.21. We present only the case for 512 byte packets.

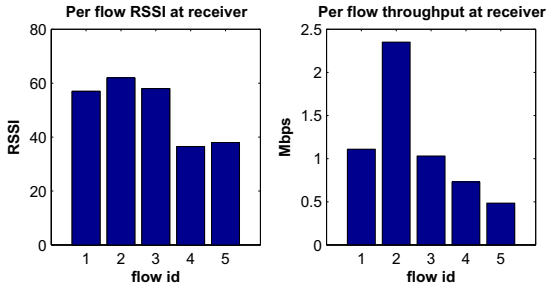



Figure 7.21: Per-flow RSSI and throughput at receiver. The first three flows are received approx. 20 RSSI units stronger than the last two flows

It can be seen that the flows with sufficiently higher RSSI always get a much higher proportion of the total throughput; where as the weaker senders suffer due to repeated collisions. The maximum throughput imbalance was as high as 5x. Based on our earlier observations, we employed a two step heuristic approach to mitigate this unfairness:

1. We increased the TxOp for flow 4 (around 2 packets per channel access) and flow 5 (around 3 packets per channel access) based on their respective flow throughputs. Default settings were used for all other parameters. This was done to give weaker senders an opportunity to send additional packets during their successful channel accesses. Figure 7.22a shows that the throughputs of these flows improve as compared to the default case.

2. Flow 2 still has a higher throughput compared to the other flows. We additionally adjust the AIFS of this flow to DIFS + 10 slot times. After step 2, the flow throughputs are more balanced as seen in Figure 7.22(b).

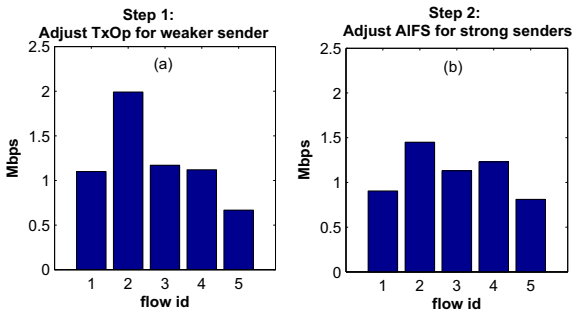



Figure 7.22: Per Flow throughput distribution after (TxOp, AIFS) correction

We quantify the effective fairness gain in terms of Jain's fairness index [107]. The index, F, is calculated as $F = \frac{(\sum_i x_i)^2}{n \times \sum_i {x_i}^2}$ where $x_i$ is the individual flow throughput and n is the total number of flows. An index value equal to one is considered to be perfectly fair.

Table 7.2 evaluates the gains of our approach w.r.t. the default case with no adaptation.

Table 7.2: Fairness comparison

| Scheme | Fairness Index |
|---|---|
| Default (no adaptation) | 0.7584 |
| Step 1 (Adjust TxOp) | 0.8877 |
| Step 2 (Step 1 + Adjust AIFS) | 0.9588 |

### 7.4.3 Conclusions and Future Work

In this work, we have experimentally verified the physical layer capture effect in 802.11 network cards as reported by earlier work. We address the related throughput fairness issue by evaluating several PHY and MAC layer options and their effectiveness in restoring fairness. Based on our observations, we apply a heuristic correction method (combined AIFS and TxOp) that yields an improvement of 25% in throughput fairness as compared to default settings. This can be further extended by developing efficient algorithms for capture detection as well as restoring fairness using a combination of frame level analysis from distributed sniffers and flow specific feedback from the receiver.

## 7.5 Towards federation of heterogeneous network testbeds

We are currently working on expanding the ORBIT control and experiment management infrastructure to enable conducting experiments that span heterogeneous networks (both wired and wireless). This is expected to provide the community with heterogeneous networking research infrastructure, to experiment with the interaction and integration of different types of networks and to test the performance of various networking protocols in realistic environments. This has led to the GENI (Global Environment for Network Innovations) effort, supported by NSF, which aims at creating a global infrastructure for conducting networking experiments across diverse substrates such as wired (local and wide-area), wireless, sensor and cellular networks [108]. In particular, the focus of this

work is to address the challenges involved in federating such diverse networking testbeds (PlanetLab and ORBIT in particular) and present models for building such an experiment infrastructure. We also identify the critical problem of simultaneous experimentation in networks involving the wireless medium and describe approaches towards solving it.

### 7.5.1 Integration of Wired and Wireless Experimental Networks

Both ORBIT and Planetlab were designed to meet very different experimental research requirements. More specifically,

- **PlanetLab** is based on "service oriented" network architecture and provides the users with the ability to run long term experiments. This experimental testbed runs on top of the Internet as an overlay thereby giving researchers access to (1) a large set of geographically distributed machines; (2) a realistic network substrate that experiences congestion, failures, and diverse link behaviors; and (3) the potential for a realistic client workload. Planetlab uses a central slice creation and management utility, Planetlab Central (PLC) for adding, renewing and managing existing slices on different nodes

- **ORBIT** Radio Grid is a multi-user wireless experimental research testbed that allows "sequential" short-term access to the radio grid resources for repeatable experimentation. Scheduling is done so that users have exclusive access to the grid during their assigned time slot. Exclusive access to the grid ensures no RF interference from other experiments. Experimental control software (NodeHandler) as well as an integrated measurement collection framework (OML) facilitate the definition, execution of experiments and also enable the collection of experimental results at run-time.

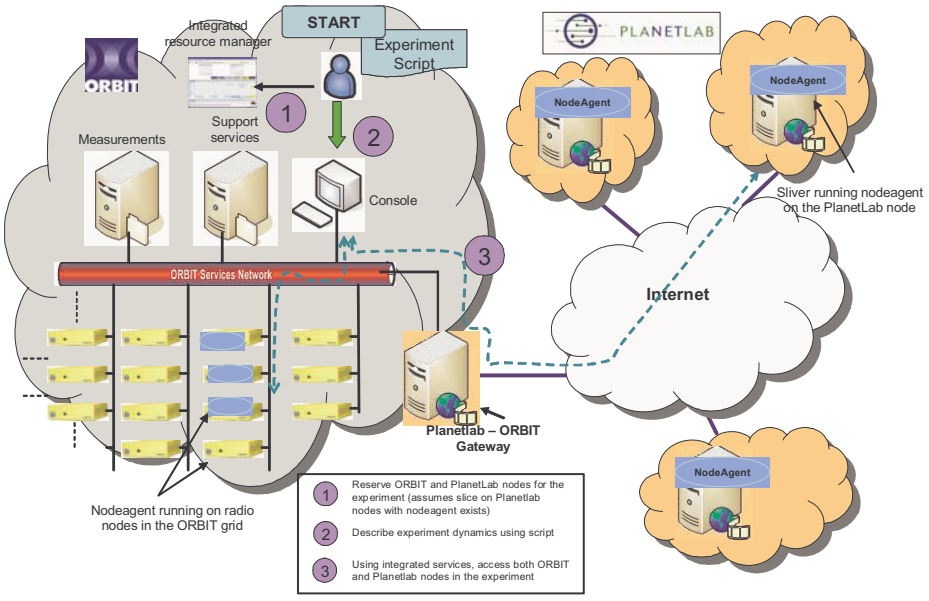



Figure 7.23: Outline of integration model (1) in which ORBIT users can add Planetlab nodes to their experiment using the concept of an *ORBIT slice*

In order to conduct experiments that encompass both these testbeds, we look at the following usage models.

**PlanetLab access for ORBIT users:** The first model is intended to serve ORBIT wireless network experimenters who want to augment their experiments by adding wired network features without major changes to their code. Figure 7.23 presents a conceptual view of this model.

The above model can be implemented by extending the ORBIT NodeAgent to Planetlab nodes, using a modified version of ORBIT NodeHandler for experiment definition, code download and execution. This version of NodeHandler will support NodeAgents running in Planetlab slivers as well as data collection from nodes spread out over the Internet. The design enhancements to NodeAgent and NodeHandler required on the ORBIT side include:

- *An extended addressing scheme* to include Planetlab nodes. We intend to address the Planetlab nodes as though they were part of the ORBIT network and have the local DNS map requests

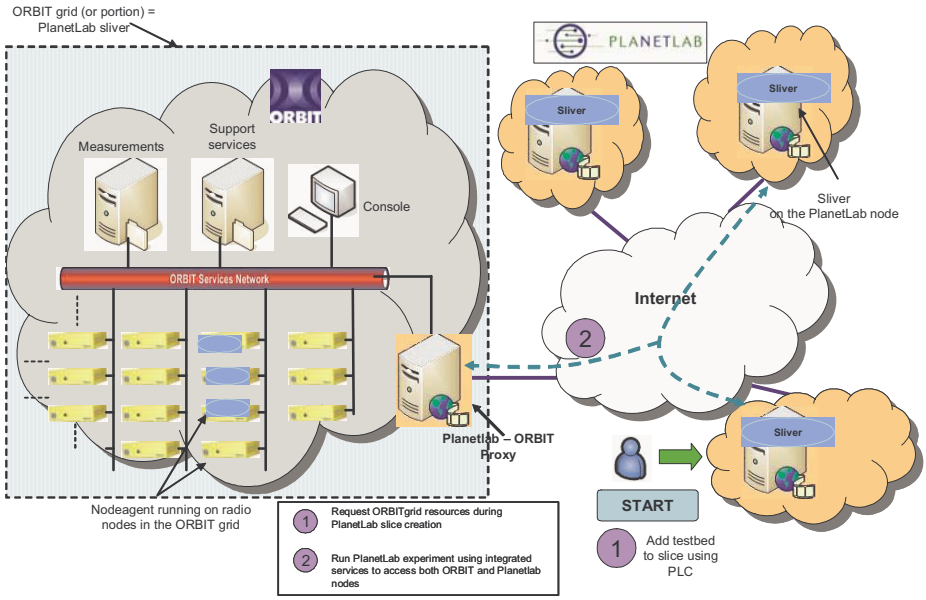



Figure 7.24: Outline of integration model in which Planetlab users get scheduled access to the entire ORBIT radio grid using the concept of an *ORBIT sliver*

for Planetlab nodes to their respective public domain names (for e.g. nodeP1.orbit-lab.org will map to Planetlab-1.cs.princeton.edu).

- *An extended communication layer* - currently commands to nodes in an experiment are sent using reliable multicast. For Planetlab nodes on the Internet, these commands will need to be tunneled using reliable unicast since multicast support on the routers in the path cannot be assumed. NodeHandler will need to perform this function of communicating with the Planetlab nodes in each experiment.

- *Caching mechanisms to assist measurements collection* - if a Planetlab node goes down during an experiment, mechanisms will be needed to extract measurements related to the previously running ORBIT experiment.

**ORBIT access for PlanetLab users:**

The second model (Figure 7.24) would provide a Planetlab-ORBIT gateway as a node that users of a PL slice can access when they want to include emulated wireless edge networks in their experiments. The PL-ORBIT GW will provide abstractions for setup, control and measurement on a specified wireless topology using a modified version of NodeHandler as the interface software. This approach does not involve major changes to either PL or ORBIT but would require the development of a Planetlab proxy module which accesses ORBIT grid services to provide a familiar experimental environment to PL users. PL users will be able to use either their own preferred tools (such as pssh [109]) or the NodeHandler/NodeAgent framework.

The integration requires extensions to the Planetlab resource specification model to include topology and some other wireless-specific features. Note that this Planetlab-ORBIT gateway would also benefit from future virtualization capabilities to be developed for ORBIT bridging the gap between the Planetlab slice model and ORBIT's single-user model. Using the virtualized ORBIT testbed, a Planetlab slice could extend to include individual ORBIT nodes while maintaining the level of isolation as offered by PlanetLab. This will facilitate experimentation over a heterogeneous substrate with both wired and wireless components providing an invaluable networking research infrastructure for the scientific community.

# References


[1] IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical layer (PHY) Specifications, June 1999.

[2] B. Crow, I. Widjaja, J. Kim, and P. Sakai. IEEE 802.11 wireless local area networks. *IEEE Communications Magazine*, Sept 1999.

[3] Bianchi G., Fratta L., and Oliveri M. Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs. *Proceedings of the IEEE Portable and Mobile Radio Communications (PIMRC)*, pages 392–396, Oct 1996.

[4] Bianchi G. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Area in Comm*, pages 535–547, March 2000.

[5] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. Macaw: a media access protocol for wireless lan's. In *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, pages 212–225, New York, NY, USA, 1994. ACM Press.

[6] S. Xu and T. Saadawi. Does IEEE 802.11 MAC protocol work well in multihop wireless ad-hoc networks? In *IEEE Communications Magazine*, June 2001.

[7] J. Li, C. Blake, D. DeCouto, H. Lee, and R. Morris. Capacity of adhoc wireless networks. In *ACM International Conference on Mobile Computing and Networking*, Aug 2001.

[8] Charles E. Perkins and Pravin Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for mobile computers. In *SIGCOMM*, pages 234–244, 1994.

[9] C. Perkins and E. Belding-Royer. Ad hoc On-Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA*, pages 90–100, February 1999.

[10] David B Johnson and David A Maltz. Dynamic Source Routing in Adhoc Wireless Networks. In *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[11] Douglas S. J. De Couto, Daniel Aguayo, Benjamin A. Chambers, and Robert Morris. Performance of Multihop Wireless Networks: Shortest Path is Not Enough. In *Proceedings of the First Workshop on Hot Topics in Networks (HotNets-I)*, Princeton, New Jersey, October 2002. ACM SIGCOMM.

[12] Control based MANETs. `http://www.darpa.mil/sto/solicitations/cbmanet/briefs/CBMANET_Overview-%Ramming.pdf`.

[13] Ian F. Akyildiz, Xudong Wang, and Weilin Wang. Wireless mesh networks: a survey. *Computer Networks*, 47(4):445–487, March 2005.

[14] I. Chlamtac, M. Conti, and J. Liu. Mobile ad hoc networking: Imperatives and challenges. *Ad Hoc Network Journal*, 1 N1, Jan-Mar 2003.

[15] IEEE 802.11a Wireless Medium Access control (MAC) and Physical layer specifications: Medium Access Control (MAC) High-speed Physical Layer in the 5 GHz band, 1999.

[16] IEEE 802.11n Wireless Medium Access control (MAC) and Physical layer specifications: Medium Access Control (MAC) Further Higher-Speed Physical Layer Extension in the 2.4 GHz band, 2003.

[17] IEEE 802.11 Task Group N Mesh Networking Standard. `http://grouper.ieee.org/groups/802/11/Reports/tgn_update.htm`.

[18] T. Ho, R. Koetter, M. Medard, D. Karger, and M.Effros. The benefits of coding over routing in a randomized setting. In *International Symposium on Information Theory*, 2003.

[19] S. R. Li, R. W. Yeung, and N. Cai. Linear network coding. *In IEEE Transactions on Information Theory*, 2003.

[20] Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Médard, and Jon Crowcroft. XORs in the air: practical wireless network coding. In *Proceedings of the 2006 SIGCOMM conference on Applications, technologies, architectures, and protocols for computer communications*, pages 243–254, New York, NY, USA, 2006. ACM Press.

[21] Arup Acharya, Archan Misra, and Sorav Bansal. Design and analysis of a cooperative medium access scheme for high-performance wireless mesh networks. In *Proceedings of 1st IEEE International Conference on Broadband Networks*, 2004.

[22] Zhibin Wu and D. Raychaudhuri. D-LSMA: Distributed Link Scheduling Multiple Access protocol for QoS in ad-hoc networks. In *Global Telecommunications Conference, Globecom*, volume 3, pages 1670–1675, 2004.

[23] Xue Yang and N.H Vaidya. Explicit and implicit pipelining for wireless medium access control,. In *IEEE Vehicular Technology Conference, VTC 2003*, volume 3, pages 1427–1431, 2003.

[24] Yijun Li, Hongyi Wu, Dimitri Perkins, Nian-Feng Tzeng, and Magdy Bayoumi. MAC-SCC: Medium access control with a separate control channel for multihop wireless networks. In *Proceedings of ICDCSW'03*, May 2003.

[25] P. Kyasunur, J. Padhye, and P. Bahl. On the efficacy of separating control and data into different frequency bands. In *IEEE Broadnets*, October 2005.

[26] IEEE 802.11e Wireless Medium Access control (MAC) and Physical layer specifications: Medium Access Control (MAC) Quality of Service(QoS) Enhancements, 2004.

[27] M. Krunz, A. Muqattash, and S.J. Lee. Transmission Power Control in Wireless Adhoc Networks: Challenges, Solutions, and Open Issues. In *IEEE Network*, pages 8–14, Sept-Oct 2004.

[28] Jing Deng, B. Liang, and P. K. Varshney. Tuning the carrier sense range of IEEE 802.11 MAC. In *Proceedings of IEEE Global Telecommunications Conference - Wireless Communications, Networks, and Systems (Globecom'04)*, 2004.

[29] Romit Roy Choudhury, Xue Yang, Nitin H. Vaidya, and Ram Ramanathan. Using directional antennas for medium access control in ad hoc networks. In *MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 59–70, New York, NY, USA, 2002. ACM Press.

[30] N. Jain, S. Das, and A. Nasipuri. A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks. In *Proceedings of IEEE ICCCN*, 2001.

[31] Jungmin So and Nitin H. Vaidya. Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver. In *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 222–233, New York, NY, USA, 2004. ACM Press.

[32] Paramvir Bahl, Ranveer Chandra, and John Dunagan. SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking (MOBICOM)*, pages 216–230, New York, NY, USA, 2004. ACM Press.

[33] P. Kyasanur, Jungmin So, C. Chereddi, and N. H. Vaidya. Multichannel mesh networks: challenges and protocols. *Wireless Communications, IEEE*, 13(2):30–36, 2006.

[34] Atul Adya, Paramvir Bahl, Jitendra Padhye, Alec Wolman, and Lidong Zhou. A multi-radio unification protocol for ieee 802.11 wireless networks. In *BROADNETS '04: Proceedings of the First International Conference on Broadband Networks (BROADNETS'04)*, pages 344–354, Washington, DC, USA, 2004. IEEE Computer Society.

[35] Baruch Awerbuch, David Holmer, and Herbert Rubens. The medium time metric: High throughput route selection in multi-rate ad hoc wireless networks. In *Proceedings of WONS*, pages 251–268, 2004.

[36] Rohit Dube, Cynthia D. Rais, Kuang-Yeh Wang, and Satish K. Tripathi. Signal stability based adaptive routing (SSA) for ad-hoc mobile networks. Univ. of Maryland Institute for Advanced Computer Studies Report No. UMIACS-TR-96-34, 1996.

[37] Richard Draves, Jitendra Padhye, and Brian Zill. Comparison of routing metrics for static multi-hop wireless networks. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 133–144, New York, NY, USA, 2004. ACM Press.

[38] Anand Prabhu Subramanian, Milind M. Buddhikot, and Scott Miller. Interference aware routing in multi-radio wireless mesh networks,. In *Second International Workshop on Wireless Mesh Networks (WiMesh 2006), Reston, VA*, September 2006.

[39] Suli Zhao, Zhibin Wu, Arup Acharya, and Dipankar Raychaudhuri. Parma: A PHY/MAC aware routing metric for ad-hoc wireless networks with multi-rate radios. In *WOWMOM '05: Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM'05)*, pages 286–292, Washington, DC, USA, 2005. IEEE Computer Society.

[40] Yaling Yang, Jun Wang, and Robin Kravets. Designing routing metrics for mesh networks. In *EEE Workshop on Wireless Mesh Networks, WiMesh*, 2005.

[41] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 114–128, New York, NY, USA, 2004. ACM Press.

[42] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 134–146, New York, NY, USA, 2003. ACM Press.

[43] Qi Xue and Aura Ganz. Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks. *J. Parallel Distrib. Comput.*, 63(2):154–165, 2003.

[44] R. Gupta, Zhanfeng Jia, T. Tung, and J. Walrand. Interference-aware QoS routing (IQRouting) for ad-hoc networks. In *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, volume 5, page 6 pp, 2005.

[45] Chenxi Zhu and M. Scott Corson. QoS routing for mobile ad hoc networks. In *Proceedings of IEEE INFOCOM*, 2002.

[46] Ashish Raniwala, Kartik Gopalan, and Tzi cker Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 8(2):50–65, 2004.

[47] Ashish Raniwala and Tzi cker Chiueh. Architecture and Algorithms for an IEEE 802.11-based Multi-Channel Wireless Mesh Network. In *Proceedings of the IEEE INFOCOM*, volume 3, pages 2223–2234, March 2005.

[48] Tae-Suk Kim, Jennifer C. Hou, and Hyuk Lim. Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks. In *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 366–377, New York, NY, USA, 2006. ACM Press.

[49] Murali Kodialam and Thyaga Nandagopal. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 73–87, New York, NY, USA, 2005. ACM Press.

[50] Kamal Jain, Jitendra Padhye, Venkata N. Padmanabhan, and Lili Qiu. Impact of interference on multi-hop wireless network performance. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 66–80, New York, NY, USA, 2003. ACM Press.

[51] Mansoor Alicherry, Randeep Bhatia, and Li (Erran) Li. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 58–72, New York, NY, USA, 2005. ACM Press.

[52] Ram Ramanathan. Challenges: a radically new architecture for next generation mobile ad hoc networks. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 132–139, New York, NY, USA, 2005. ACM Press.

[53] P. Gupta and P. R. Kumar. The capacity of wireless networks. *Information Theory, IEEE Transactions on*, 46(2):388–404, 2000.

[54] A. Iwata, C. Chiang, G. Pei, M. Gerla, and T. Chen. Scalable routing strategies for ad-hoc wireless networks. *IEEE Journal of Selected Areas in Communications*, August 1999.

[55] Ram Ramanathan and Martha Steenstrup. Hierarchically-organized, multihop mobile wireless networks for quality-of-service support. *Mobile Networking and Applications*, 3(1):101–119, 1998.

[56] B. Liu, Z. Liu, and D. Towsley. On the capacity of hybrid wireless networks. In *IEEE INFOCOM'03*, 2003.

[57] IEEE 802.11 Task Group S Mesh Networking Standard. `http://grouper.ieee.org/groups/802/11/Reports/tgs_update.htm`.

[58] L. Raju, S. Ganu, B. Anepu, I. Seskar, and D. Raychaudhuri. BOOST: A BOOtSTrapping mechanism for Self-Organizing Hierarchical Wireless Adhoc Networks. In *IEEE Sarnoff Symposium, Princeton*, April 2004.

[59] K. Fall. ns notes and documentation. *The VINT Project*, 2000.

[60] Channel overlap calculations for 802.11b networks. Cirond Technologies white paper.

[61] Matlab version 7.0. `http://www.mathworks.com`.

[62] Lalit Raju. Discovery in Self-Organizing Hierarchical Adhoc Wireless Networks. Masters Thesis, Department of Electrical Engineering, Rutgers University, 2004.

[63] ORBIT testbed, Open Access Research Testbed for Next-generation Wireless Networks. `http://www.orbit-lab.org`.

[64] Dipankar Raychaudhuri, Ivan Seskar, Maximilian Ott, Sachin Ganu, Kishore Ramachandran, Haris Kremo, Robert Siracusa, Hang Liu, and Manpreet Singh. Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols. In *IEEE Wireless Communications and Networking Conference*, March 2005.

[65] The Libnet Packet Construction Library. `http://www.packetfactory.net/libnet`.

[66] Libpcap Packet Capture Library. `http://www.tcpdump.org`.

[67] Libmac ORBIT Packet Capture Library. `http://www.orbit-lab.org/wiki/Documentation/Libmac`.

[68] S. Ganu and D. Raychaudhuri. Integrating Short Range ad-hoc radio technologies into Next Generation Wireless Networks. In *Proceedings of the International Conference and Exposition on Communications and Computing ICCC 2005*, February 2005.

[69] J. Robinson, K. Papagiannaki, C. Diot, X. Guo, and L. Krishnamurthy. Experimenting with a multi-radio mesh networking testbed. In *Proceedings of Workshop on Wireless Network Measurements, WiNMee*, 2005.

[70] T.S. Rappaport. *Wireless Communications: Principles and Practise*. Prentice Hall, 2002.

[71] The journey of a packet through the linux 2.4 stack network. `http://gnumonks.org/ftp/pub/doc/packet-journey-2.4.html`.

[72] W. Zhu, A. Maccabe, and R. Riesen. Identifying the sources of latency in a splintered protocol. Technical report, Department of Computer Science, University of New Mexico, Dec 2003.

[73] MPLS Label Switching Architecture. IETF RFC 3031.

[74] LDP specification. IETF RFC 3036.

[75] B. Jabbari, P. Papneja, and E. Dinan. Label switched packet transfer for wireless cellular networks. In *IEEE Wireless Communications and Networking Conference*, August 2000.

[76] X. Yang and N. Vaidya. A Wireless MAC Protocol using Implicit Pipelining. In *IEEE Transactions on Mobile Computing*, volume 5, pages 258–273, March 2006.

[77] X. Yang, N. Vaidya, and P. Ravichandran. Split-channel pipelined packet scheduling for wireless networks. *IEEE Transactions on Mobile Computing*, 5(3):240–257, 2006.

[78] A. Acharya, A. Misra, and S. Bansal. A label-switching packet forwarding architecture for multi-hop wireless lans. In *5th ACM International Workshop on Wireless Mobile Multimedia (WoWMom), in conjunction with MobiCom 2002*, Sept 2002.

[79] H. Liu and D. Raychaudhuri. Label switched multi-path forwarding in wireless ad-hoc networks. In *Third IEEE International Conference on Pervasive Computing and Communications, PERCOM*, pages 248–252, March 2005.

[80] Yu. Wang and Jie. Wu. Label routing protocol: a new cross-layer protocol for multi-hop ad hoc wireles networks. In *Proceedings of the International Conference on Mobile Adhoc and Sensor Systems Conference, MASS*, pages 139–145, November 2005.

[81] D. Raguin, M. Kubisch, H. Karl, and A. Wolisz. Queue-driven cut-through medium access in wireless ad hoc networks. In *IEEE Wireless Communications and Networking Conference*, March 2004.

[82] The CMU Monarch Project's Wireless and Mobility Extensions to NS, 1998.

[83] Agilent E4438C Vector Signal Generator Data Sheet. `http://cp.literature.agilent.com/litweb/pdf/5988-4039EN.pdf`.

[84] J. Bicket. Bit-rate selection in wireless networks. Master's thesis, MIT, 2005.

[85] A. Kamerman and L. Monteban. Wavelan: A high-performance wireless LAN for the unlicensed band. Technical Report 3, Bell Labs Technical Journal, 1997.

[86] Wireless Tools for Linux. `http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.htm`.

[87] Henrik Lundgren, Erik Nordstrom, and Christian Tschudin. Coping with communication gray zones in IEEE 802.11b based ad hoc networks. In *WOWMOM '02: Proceedings of the 5th ACM international workshop on Wireless mobile multimedia*, pages 49–55, New York, NY, USA, 2002. ACM Press.

[88] Weinschel Programmable Signal Attenuator. `http://www.weinschel.com/8310.htm`.

[89] Iperf Traffic Generator. `http://dast.nlanr.net/Projects/Iperf`.

[90] A. Acharya, S. Ganu, and A. Misra. DCMA: A label-switching MAC for efficient packet forwarding in multi-hop wireless networks. IEEE Journal of Selected Areas in Communications, Special Issue on Wireless Mesh Networks, November 2006.

[91] R. Draves, J. Padhye, and B. Zill. The architecture of the Link Quality Source Routing protocol. Technical Report MSR-TR-2004-57, Microsoft Research, 2004.

[92] Mathieu Lacage, Mohammad Hossein Manshaei, and Thierry Turletti. Ieee 802.11 rate adaptation: a practical approach. In *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 126–134, New York, NY, USA, 2004. ACM Press.

[93] Starsky H. Y. Wong, Songwu Lu, Hao Yang, and Vaduvur Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 146–157, New York, NY, USA, 2006. ACM Press.

[94] E. Carlson, C. Prehofer, C. Bettstetter, H. Karl, and A. Wolisz. A distributed end-to-end reservation protocol for IEEE 802.11-based wireless mesh networks. *IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Multi-Hop Wireless Mesh Networks*, 24:2018– 2027, November 2006.

[95] Zhibin Wu, Sachin Ganu, and D. Raychaudhuri. IRMA: Integrated Routing and MAC Scheduling in Multi-hop Wireless Mesh Networks,. In *Second IEEE Workshop on Wireless Mesh Networks, WiMesh 2006*, pages 109–118, 2006.

[96] NSF Workshop on Network Research Testbeds, Chicago, il. `http://www.net.cs.umass.edu/testbed_workshop`.

[97] L. Peterson, T. Anderson, D. Culler, and T. Roscoe. A blueprint for introducing disruptive technology into the internet. In *First Workshop on Hot Topics in Networking (HotNets-I)*, 2002.

[98] Jing Lei, Roy Yates, Larry Greenstein, and Hang Liu. Wireless link snr mapping onto an indoor testbed. In *TRIDENTCOM '05: Proceedings of the First International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities (TRIDENTCOM'05)*, pages 130–135, Washington, DC, USA, 2005. IEEE Computer Society.

[99] S. Ganu, M. Ott, I. Seskar, D. Raychaudhuri, and S. Paul. Architecture and Framework for Supporting Open-Access Multi-user Wireless Experimentation. In *First International Conference on Communication System Software and Middleware, 2006. Comsware 2006*, pages 1–9, January 2006.

[100] Manpreet Singh, Maximilian Ott, Ivan Seskar, and Pandurang Kamat. ORBIT measurements framework and library (OML): Motivations, design, implementation, and features. In *Proceedings of the IEEE Tridentcom' 05)*, pages 146–152, Washington, DC, USA, 2005. IEEE Computer Society.

[101] XDR External Data Representation Standard. RFC 1832.

[102] Mike Hibler, Leigh Stoller, Jay Lepreau, Robert Ricci, and Chad Barb. Fast Scalable Disk imaging with Frisbee. In *Proceedings of USENIX Annual Technical Confernence*, June 2003.

[103] Sachin Ganu, Haris Kremo, Richard Howard, and Ivan Seskar. Addressing repeatability in wireless experiments using ORBIT testbed. In *Proceedings of the IEEE Tridentcom 2005*, pages 153–160, Washington, DC, USA, 2005. IEEE Computer Society.

[104] Agilent 89600 Vector Signal Analyzer Data Sheet. `http://cp.literature.agilent.com/litweb/pdf/5988-7811EN.pdf`.

[105] A. Kochut, A. Vasan, U. Shankar, and A. Agrawala. Sniffing out the correct Physical Layer Capture model in 802.11b. In *Proceedings of 12th IEEE International Conference on Network Protocols (ICNP'04)*, pages 252–261, May 2004.

[106] Anthony C. H. Ng, David Malone, and Douglas J. Leith. Experimental evaluation of TCP performance and fairness in an 802.11e test-bed. In *E-WIND '05: Proceeding of the 2005 ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis*, pages 17–22, New York, NY, USA, 2005. ACM Press.

[107] R. Jain, D. Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical report, DEC Tech Report TR-301, September 1984.

[108] NSF Global Environment for Network innovations. `http://www.geni.net`.

[109] PSSH tools. `http://www.theether.org/pssh/`.

# Curriculum Vita

## Sachin Ganu

### Education

**1994-1998** BE in Electronics Engineering; V.E.S. Institute of Technology, Mumbai University, Mumbai, India.

**1998-1999** MS in Electrical Engineering; Virginia Tech, Blacksburg, VA, USA.

**2002-2007** Ph.D. in Electrical Engineering; WINLAB, Rutgers University, NJ, USA.

### Industry

**2000-2001** Systems Engineer, Lockheed Martin Global Telecommunications, Clarksburg, MD, USA

**2003** Research Scientist, Avaya Research Labs, Basking Ridge, NJ, USA

**2005** Network Software Engineer, Intel Corporation, Hillsboro, OR, USA

**2007-** Member of Technical Staff, Aruba Networks, Sunnyvale, CA, USA.

### Publications

**2004** S. Ganu, P. Krishnan, A. S. Krishnakumar *Infrastructure based location estimation in WLAN networks*, Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), 2004, vol. 1, 21-25, pp. 465 470, Atlanta.

P. Krishnan, A. S. Krishnakumar, W. Ju, C. Mallows, S.Ganu, *A System for LEASE: Location Estimation Assisted by Stationary Emitters for Indoor RF Wireless Networks*, in the Proceedings of the IEEE Infocom, vol. 2, pp. 1001101, 2004.

L. Fazal, S. Ganu, M. Kappes, P. Krishnan, A.S. Krishnakumar, *Tackling Security Vulnerabilities in VPN-based Infrastructure Wireless Deployments*, in the Proceedings of the International Conference on Communication (ICC) 2004, vol. 1, pp 100-104, Paris.

S. Ganu, L. Raju, B. Anepu, S. Zhao, I. Seskar and D. Raychaudhuri *Architecture and Prototyping of an 802.11 Based Self Organizing Hierarchical Ad-hoc Wireless Network (SOHAN)*, in Proceedings of the International Symposium on Personal, Indoor and Mobile Radio Communications, (PIMRC) 2004, Barcelona, Vol. 2, pp. 880- 884. Also appears as a book chapter in Emerging Location Aware Broadband Wireless Ad Hoc Networks, Springer Publications, ISBN: 978-0-387-23070-2, pp 109-122.

L. Raju, S. Ganu, B. Anepu, I. Seskar and D. Raychaudhuri, *Bootstrapping Protocol for Self Organizing Hierarchical Ad-hoc Wireless Network*, Proceedings of the IEEE Sarnoff Symposium 2004, Princeton.

L. Raju, S. Ganu, B. Anepu, I. Seskar and D. Raychaudhuri, *Beacon-Assisted Discovery (BEAD) Protocol for Self-Organizing Hierarchical Ad-hoc Wireless Network*, Proceedings of the IEEE Global Telecommunications Conference, (Globecom) 2004, vol. 3, pp. 1676- 1680.

**2005** S. Ganu, H. Kremo, R. Howard and I. Seskar, *Addressing Repeatability in Wireless Experiments using the ORBIT Testbed*, Proceedings of the first IEEE Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (Tridentcom) 2005, pp 153 160, Trento, Italy.

D. Raychaudhuri, M. Ott, I. Seskar, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu and M. Singh, *Overview of the ORBIT Radio Grid Testbed for Evaluation of Next Generation Wireless Networks*, Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), 2005, Vol. 3, pp. 1664- 1669, New Orleans

Z. Wu, S. Ganu, I. Seskar and D. Raychaudhuri, *Experimental Investigation of Physical Layer Rate Control and Frequency Selection in 802.11 Based Ad-hoc Networks*, Workshop on Experimental Approaches to Wireless Network Design and Analysis, (EWIND), cohosted with ACM Sigcomm 2005, pp 41-45, Philadephia

S. Ganu and D. Raychaudhuri, *Integrating short-range ad-hoc radio technologies into Next Generation Wireless Networks*, in Proceedings of the International Conference and Symposium on Communications and Computing (ICCC), Kanpur, India, 2005

**2006** S. Ganu, K. Ramachandran, M. Gruteser, I. Seskar and J. Deng, *Methods for restoring MAC layer fairness in 802.11 networks with Physical Layer Capture*, Proceedings of the Second International Workshop on Multi-hop Ad-hoc Networks (REALMAN), colocated with ACM Mobihoc 2006, pp. 7-14, Florence.

Z. Wu, S. Ganu and D. Raychaudhuri, *IRMA: Integrated Routing and MAC Scheduling in Wireless Mesh Networks*, in Proceedings of the Second IEEE Workshop on Wireless Mesh Networks (WiMesh), 2006, pp 109-118, Reston, VA.

A. Acharya, S. Ganu, A. Misra, *DCMA: A Label Switching MAC for Efficient Packet Forwarding in Multihop Wireless Networks*, in IEEE JSAC Special Issue on Wireless Mesh Networks, Nov 2006, Volume 24, Issue 11, pp. 1995-2004.

**2007** D Rastogi, S. Ganu, Y. Zhang, W. Trappe and C. Graff, *A Comparative Study of AODV and OLSR on the ORBIT Testbed*, in Proceedings of MILCOM 2007, Orlando, FL.