ABSTRACT

Lee, Michael E. DCAP: A Multichannel Protocol for Single Interface 802.11 Wireless Mesh Networks. (Under the direction of Assistant Professor Mihail L. Sichitiu).

Wireless ad hoc networks are gaining popularity as quick and inexpensive methods of connecting computers. In particular, Wireless Mesh Networks (WMNs) are becoming a viable method of offering Internet access to entire neighborhoods. One reason WMNs are attractive is because of their use of inexpensive 802.11 wireless hardware. However, using 802.11 standard compliant hardware has a major limitation: the 802.11 standard does not support the use of multiple channels in the same network. Because of this limitation, wireless 802.11 networks are not able to achieve the traffic throughput possible when utilizing all the available channels.

To increase the throughput in WMNs, this paper proposes a novel protocol allowing the use of multiple channels with a single wireless 802.11 interface. This protocol, Distributed Channel-switching Accessory Protocol, or DCAP, requires no modifications to the 802.11 MAC layer. DCAP defines the methods wireless nodes use to send and receive traffic across multiple channels with a single wireless interface. The key concept of our approach is Home Channels. A node's Home Channel is the only wireless channel on which the node receives data. By requiring the sender to change to the receiver's Home Channel to transmit a packet, all nodes know on which channel to transmit each packet. Once on the receiver's Home Channel, the transmission of the packet follows the 802.11 standard.

In this work, DCAP is implemented in the ns-2 event simulator to evaluate its performance. DCAP is implemented as a separate protocol immediately above the 802.11 MAC layer in ns-2. The implementation of DCAP makes no modifications to the 802.11 MAC protocol. A series of performance evaluation tests are performed to compare DCAP's performance against a single channel 802.11 network. These tests compare the throughput of the two networks in a variety of different network traffic conditions and network topologies. The simulation results show the network with the DCAP protocol achieves significantly higher throughputs than the single channel 802.11 network (up to four times in a network with three channels).

DCAP: A Multichannel Protocol for Single Interface 802.11 Wireless Mesh Networks

by

Michael E. Lee

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Computer Networking - Electrical and Computer Engineering

Raleigh

2006

Approved By:

Dr. Wenye Wang

Dr. Alexander G. Dean

Dr. Mihail L. Sichitiu
Chair of Advisory Committee

To my wife Angela…

Biography

Michael Lee was born and grew up in Arkansas. He completed his early education in Russellville, AR, and graduated from Russellville High in 1999. He graduated from the University of Arkansas at Fayetteville, AR in May 2003 with a bachelor's degree in Computer Engineering and a minor in Mathematics. He then began full time work toward the Master of Science degree in Computer Networking at North Carolina State University in Raleigh, NC.

Acknowledgements

I would like to thank my advisor, Dr. Mihail Sichitiu, for his patient guidance and support. This thesis would not have been possible without his constructive criticism and suggestions. I am grateful to my advisory committee members, Dr. Wenye Wang and Dr. Alexander G. Dean, for their suggestions and comments on my work. I also appreciate my collegues in the WALAN Research Group, especially Jangeun Jun, for all their suggestions.

I would like to thank my parents, David and Becky Lee, for their support throughout my educational journey. Most of all, I want to thank my wife, Angela, for her patience and support during my research.

Contents

List of Tables **vii**

List of Figures **viii**

List of Abbreviations **ix**

1 Introduction **1**
1.1 Related Work . . . 3
1.1.1 Multi-Radio, Custom MAC . . . 3
1.1.2 Multi-Radio, 802.11 MAC . . . 5
1.1.3 Single-Radio, Custom MAC . . . 6
1.1.4 Single-Radio, 802.11 MAC . . . 7
1.2 Contribution . . . 9

2 DCAP Design **10**
2.1 Overview . . . 10
2.2 Receiving Packets . . . 12
2.3 Sending Packets . . . 12
2.4 Listen Time (T_L) . . . 15
2.5 Determining a Receiver's Channel . . . 18
2.6 Channel Table . . . 19
2.7 Neighbor Discovery (T_N) . . . 20
2.8 Gateway Node . . . 21
2.9 Home Channel Assignment . . . 21
2.9.1 Hyacinth Channel Assignment Algorithm . . . 23
2.9.2 Differences Between Channel Assignment Algorithms . . . 24
2.9.3 DCAP's Modified Channel Assignment Algorithm . . . 25

3 Implementation **27**
3.1 Test One: Two Nodes . . . 29
3.2 Test Two: Three Nodes . . . 31
3.3 Test Three: Multihop Network . . . 32

3.4 Test Four: Three Channels 37

4 Performance Evaluation **40**
4.1 Dependency on Channel Request Packet Timeout (T_Q) 42
4.2 Dependency on Listen Time (T_L) 45
4.3 Dependency on Home Channel Assignment Interval (T_C) 45
4.4 Dependency on Neighbor Discovery Interval (T_N) 47
4.5 Dependency on Channel Table Purge Interval (T_P) 49
4.6 Dependency on Channel Load (Poisson Interarrival Time) 51
4.7 Dependency on Number of Channels 55
4.8 Dependency on Number of Nodes 57
4.9 Dependency on Distance Between Nodes 61
4.10 Dependency on Node Positions 63

5 Conclusion and Future Work **64**

Bibliography **67**

A Validation Tests **71**

List of Tables

4.1 Performance Evaluation Base Scenario 41

List of Figures

1.1 Wireless Mesh Network 2

2.1 DCAP is Implemented Between the IP and the MAC Layers 11
2.2 Node A Sending a Packet to Node B 13
2.3 Sending a Broadcast Packet on Multiple Channels 14
2.4 Sending on Busy Channel 15
2.5 A Value of T_L to Small Leading to Transmission Failures From Node A to B When B Also Sends to C 17
2.6 Hop Distance Between Neighbors 22

3.1 Ns-2 Implementation of the Gateway Node 28
3.2 Test One: Two Nodes 30
3.3 Subtest 1.3 Throughput 30
3.4 Test Two: Three Nodes 31
3.5 Subtest 2.3 Throughput 32
3.6 Test Three: Multihop Network 33
3.7 Subtest 3.1 Throughput 34
3.8 Subtest 3.2 Throughput 35
3.9 Subtest 3.3 Throughput 36
3.10 Subtest 3.4 Throughput 38
3.11 Test Four: Three Channels 38
3.12 Subtest 4.3 Throughput 39

4.1 Performance Evaluation Network 41
4.2 Dependency of Throughput and Delay as a Function of Channel Request Packet Timeout T_Q 43
4.3 Dependency of Throughput and Delay as a Function of Listen Time T_L . . 44
4.4 Dependency of Throughput and Delay as a Function of Home Channel Selection Interval T_C 46
4.5 Dependency of Throughput and Delay as a Function of Neighbor Discovery Interval T_N 48

4.6 Dependency of Throughput and Delay as a Function of Channel Table Purge Interval T_P 50
4.7 Dependency of Throughput and Delay as a Function of Offered Load 52
4.8 Dependency of Throughput and Delay as a Function of HTTP traffic on Offered Load 54
4.9 Dependency of Throughput and Delay as a Function of Number of Channels 56
4.10 Dependency of Throughput and Delay as a Function of the Number of Nodes in the Network 58
4.11 Dependency of Throughput and Delay as a Function of the Distance Between Nodes 60
4.12 Dependency of Throughput and Delay as a Function of Node Positions . . . 62

A.1 Subtest 1.1 Throughput 71
A.2 Subtest 1.2 Throughput 72
A.3 Subtest 1.3 Throughput 72
A.4 Subtest 2.1 Throughput 73
A.5 Subtest 2.2 Throughput 73
A.6 Subtest 2.3 Throughput 74
A.7 Subtest 3.1 Throughput 74
A.8 Subtest 3.2 Throughput 75
A.9 Subtest 3.3 Throughput 75
A.10 Subtest 3.4 Throughput 76
A.11 Subtest 3.5 Throughput 76
A.12 Subtest 3.6 Throughput 77
A.13 Subtest 4.1 Throughput 77
A.14 Subtest 4.2 Throughput 78
A.15 Subtest 4.3 Throughput 78

List of Abbreviations

- 802.11 - IEEE 802.11 Wireless Standard
- ACK - Acknowledgment
- AODV - Ad Hoc On-Demand Distance Vector Routing
- AP - Access Point
- CBR - Constant Bitrate
- CTS - Clear to Send
- DCAP - Distributed Channel-switching Accessory Protocol
- DIFS - Distributed Inter Frame Space
- FHSS - Frequency-Hopping Spread Spectrum
- GPS - Global Positioning System
- HTTP - Hyper Text Transfer Protocol
- IP - Internet Protocol
- MAC - Media Access Control
- NAV - Network Allocation Vector
- PDR - Packet Delivery Ratio
- RTS - Request to Send
- TCP - Transport Control Protocol
- TDMA - Time Division Multiple Access
- WMN - Wireless Mesh Network

Chapter 1

Introduction

Wireless networks based on the IEEE 802.11 standard [14] have become a commonplace in aspects of our daily lives over the past five years. This popularity is due to the inexpensive 802.11 hardware and the easy deployment of wireless networks. Although these networks are common, they typically operate in infrastructure mode. In this mode, a wireless node connects directly to an Access Point (AP). All wireless traffic is received by the AP and forwarded to the correct destination. Using the APs, a wireless node is able to send and receive packets from other networks such as the Internet. This type of network is called a single-hop with infrastructure network since all wireless nodes must be within transmission range of the AP.

Wireless networks are also capable of operating in ad hoc mode. In ad hoc networks, the wireless nodes are not required to be in radio range of an AP. Instead, traffic from wireless nodes is forwarded by the other wireless nodes to reach the destination. Ad hoc networks are beneficial because they can create a network in places where it is difficult or nearly impossible to create a wired network. Ad hoc networks are gaining popularity for use in connecting neighborhoods to the Internet without the expense of laying the wires necessary for wired networks [7]. This type of network, known as a Wireless Mesh Network (WMN), is an ad hoc network that is connected to a wired network (see Figure 1.1). WMNs can be used to connect wireless nodes with the Internet through special wireless nodes called gateways. The gateways in a WMN connect the wireless ad hoc network to the wired network.

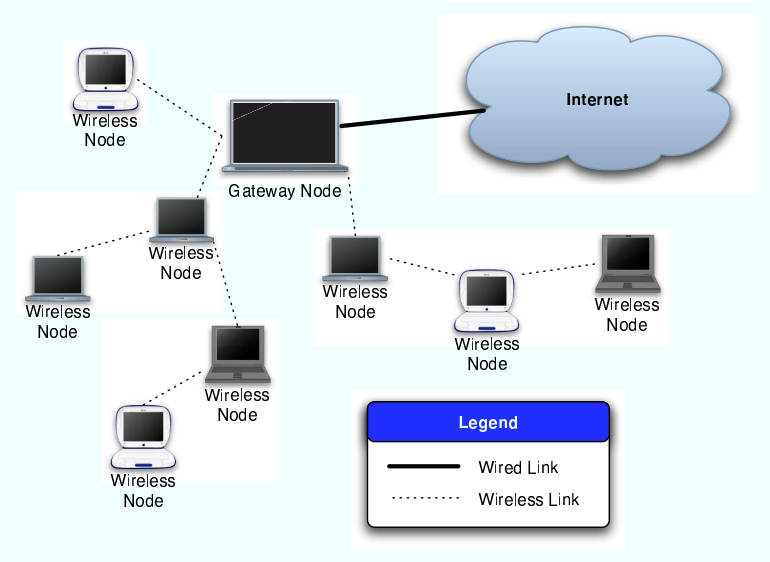

Figure 1.1: Wireless Mesh Network

Although WMNs allow for the creation of a network without any physical wires, their throughput is less than that of a single-hop with infrastructure network. This decrease in throughput is the result of more wireless nodes competing for the wireless channel. Nodes outside the transmission range of a gateway node must have their traffic forwarded by nodes closer to the gateway. This results in more transmissions per packet in an WMN than in a single-hop with infrastructure network. In an ad hoc network, as the distance between the sender and receiver increases, the throughput decreases due to interference from wireless neighbors [7]. This is caused by a wireless node having to compete with its neighbors for exclusive access of the channel before sending.

To further compound the throughput problem in a WMN, the network's traffic will concentrate around the gateway nodes. All traffic must go through a gateway node, since most traffic in a WMN is going from the wireless nodes to the wired network (or vice versa). This increase in traffic flowing through the gateway nodes causes an increase in competition for the channel around the gateway nodes, and decreases the throughput in the network.

Throughput in a WMN can be increased by increasing the bandwidth available on the channel or by using multiple channels. The 802.11b/g standards provide three

non-overlapping channels for wireless nodes to use, and the 802.11a standard provides 12 non-overlapping channels. According to the 802.11 standard, each of these channels has a fixed amount of bandwidth. By following the 802.11 standard, a network is able to utilize the inexpensive 802.11 hardware instead of expensive, custom hardware. Although multiple channels exist in the 802.11 standards, most ad hoc networks operate only on a single channel. A method of increasing the throughput in a multi-channel network by utilizing 802.11 hardware will result in higher throughput while still following the 802.11 standard. By utilizing multiple channels, the interference between neighbors will also be reduced [3]. The following section describes several different methods currently available for increasing the throughput in WMNs by utilizing multiple channels.

1.1 Related Work

There are two methods of utilizing multiple channels in wireless mesh networks: multi-radio and single-radio. Multi-radio solutions utilize multiple radios on each wireless node to achieve greater throughput in wireless networks with multiple channels. Single-radio solutions, however, attempt to increase the throughput in a wireless network with using only a single wireless radio. Although multi-radio solutions are more expensive to implement, they typically achieve a higher throughput than single-radio solutions.

Multi-channel wireless solutions can be further divided into methods that use an unmodified 802.11 MAC protocol and methods that use custom MAC protocols. The advantage of using the unmodified 802.11 MAC protocol is that existing 802.11 wireless hardware can be used to implement the solution. Since 802.11 wireless hardware has become inexpensive due to mass production, using unmodified 802.11 wireless hardware would decrease the cost of deployment compared to custom MAC solutions. The following sections describe existing solutions in each of these categories.

1.1.1 Multi-Radio, Custom MAC

Most methods of utilizing multiple channels in a WMN involve using multiple wireless interfaces with a custom MAC protocol. In these methods, each interface is located on a separate channel. A trivial solution is to place one wireless interface on each available

channel. Although this method would allow the use of all channels in the network, it does not scale well for networks with large numbers of channels. A network with 20 channels would require 20 wireless interfaces in each node. Another method is to use only one wireless interface to send packets while using multiple wireless radios to receive packets [9]. In this protocol each node has a wireless radio listening on each interface for incoming packets. When a node sends a packet, it will randomly choose an idle interface to send the packet, giving priority to the previously used interface. Since all nodes are listening on every channel, the receiver will be able to receive a packet sent on any channel. Although this solution can utilize all channels in the network, it requires a seperate wireless radio listening on each channel in the network.

Most methods use N wireless interfaces in a network with M available channels where $N \leq M$. This allows a more scalable solution while still gaining the benefits by using multiple interfaces. In these networks with fewer wireless interfaces than available channels, a method of assigning channels to interfaces is needed to utilize the multiple channels while maintaining the network's connectivity. Several static and dynamic algorithms have been developed that assign a pair of interfaces within transmission range to a channel [8]. The static algorithms are centralized and do not allow the channels to be reassigned to other nodes to accommodate changes in traffic patterns. Although the dynamic algorithms allow channel reassignment to accommodate traffic pattern changes, they require time synchronization and contain overhead for coordinating the channel reassignment.

Several other custom protocols use multiple interfaces to increase the throughput in a WMN by designating one channel as a control channel [19], [6], [20]. These protocols use the control channel to reserve a data channel for the exclusive use of transmitting a single packet. In a network with M available channels, this allows $M-1$ simultaneous data transmissions. These protocols exchange messages similar to Request to Send (RTS) and Clear to Send (CTS) messages on the control channel to reserve a pre-selected idle data channel for exclusive use during the transmission of the packet. This method guarantees that there is no channel contention on any channel except the control channel.

All nodes keep one wireless interface on the control channel and use the remaining wireless interfaces to send data packets on the other channels. When a node sends a RTS-type message on the control channel, the message will specify the data channel the packet will be sent on and how long the transmission will take. All nodes that receive the RTS-type message will update the Network Allocation Vector (NAV) of the data channel to indicate

it is busy for the duration of the transmission. This protocol requires information, including the NAV, to be stored for each channel.

The primary disadvantage of these protocols is the bottleneck the control channel creates. In order to send any packet, a node must send the RTS-type message on the control channel. When many nodes are trying to transmit at once, there will be heavy competition for access to the control channel. This competition can lead to under-utilization of the data channels as a result of collisions on the control channel. In cases when there are a few large packets to be transmitted, the control channel's bandwidth will be wasted due to infrequent RTS-type and CTS-type messages, while the data channels' bandwidth will be entirely consumed by the large data payloads.

1.1.2 Multi-Radio, 802.11 MAC

Another method of utilizing multiple channels in a WMN is to use multiple 802.11 wireless interfaces. This is less expensive and easier to deploy than using custom interfaces; however, the 802.11 MAC protocol only works on a single channel. The Hyacinth network accommodates this limitation by assigning each 802.11 wireless interface to operate on a specific channel [12], [10], [11]. Two variations of the Hyacinth network exist. One performs the channel assignment using a centralized method, and the other performs the channel assignment using a distributed method.

The Hyacinth network can perform static channel allocation by sending link load estimations to a centralized location [12]. The centralized location will perform two tasks: interface-to-channel binding and neighbor-to-interface binding. These tasks are different because a node has fewer interfaces than neighbors, so the centralized location must ensure the network maintains connectivity when performing the neighbor-to-interface binding. The centralized location also has to maximize throughput in the network by carefully performing the interface-to-channel binding to minimize the amount of interference.

A centralized approach to channel allocation does not scale well for large networks, so the Hyacinth network was modified to allow dynamic channel allocation [10], [11]. Dynamic channel assignment is performed by the nodes sending link load estimates to the node's neighbors. Before a node assigns any of its interfaces to a channel, it must first create a tree out of the network topology. The tree will contain the gateway node as the root. Every node divides its interfaces into two sets: UP-NICs and DOWN-NICs. Each of a

node's UP-NIC corresponds to a DOWN-NIC of the node's parent. Nodes can only assign channels to their DOWN-NICs. A node's UP-NICs must always stay on the same channel as the parent's corresponding DOWN-NIC, so the node's parent determines the channel of UP-NICs. This method of assigning channels allows the Hyacinth network to efficiently use multiple channels.

The Hyacinth network achieves a higher throughput by using multiple 802.11 interfaces. This requirement of multiple interfaces is the primary disadvantage of the Hyacinth network. Multiple interfaces are more expensive and consume more power than a single interface. Using a single wireless interface to harness multiple channels for increased throughput results in a lower cost of deployment. The following two sections describe methods of using a single wireless interface to send and receive packets on multiple channels.

1.1.3 Single-Radio, Custom MAC

Several protocols use a single wireless interface with a custom MAC to utilize multiple channels for increased throughput. These protocols work by dividing time into slots, called Time Division Multiple Access (TDMA). TDMA requires clock synchronization between nodes. The Multi-channel MAC (MMAC) protocol breaks time up into beacon intervals [13]. At the beginning of each beacon interval nodes negotiate which channel they will use for the remainder of the beacon interval. The Group Allocation Multihop Multiple Access (GAMMA) protocol also uses TDMA to reserve channels, but it reserves channels for only a single slot instead of for the entire interval.

The Hop Reservation Multiple Access (HRMA) protocol uses TDMA to reserve time slots for each transmitting node; however, it also uses slow frequency-hopping spread spectrum (FHSS) when sending packets [15]. In slow FHSS, a node will change channels according to a set schedule. HRMA uses FHSS to decrease the effects of noise on the channels by limiting the time nodes spend on each channel.

Another use of slow FHSS is performed by Channel-Hopping Multiple Access (CHMA) and Receiver-Initiated Channel-Hopping with Dual Polling (RICH-DP) [16], [17]. In both of these protocols, a common channel sequence is used by all nodes. When two nodes need to send to one another, they will exchange RTS and CTS frames. A successful RTS/CTS exchange will reserve that channel for use by the two nodes. While the two nodes complete the transmission on the reserved channel, the remaining nodes will continue with

the hop sequence by switching to the next channel in the sequence. This allows other nodes to reserve other channels and transmit while the original two nodes are still transmitting.

Although all of these single-radio, custom MAC solutions achieve greater throughput than a single channel 802.11 network, they all require clock synchronization. Clock synchronization results in more network traffic or specialized hardware to maintain the synchronization. By frequently communicating each node's current time, nodes can synchronize their clocks. However, this synchronization requires additional network traffic for the time exchange. The additional traffic decreases the number of other packets that can be sent in the network. Extra communication can be avoided by using specialized hardware. By using more accurate hardware clocks on each wireless node, synchronization can be maintained longer. Using Global Positioning System (GPS) hardware, also allows nodes to synchronize their clocks. However, these specialized hardware solutions result in more expensive wireless nodes. The additional traffic or specialized hardware prevent protocols requiring clock synchronization from scaling well to large networks. Also, most of these protocols have synchronization or contention time slots that introduce overhead regardless of how many nodes need to transmit.

1.1.4 Single-Radio, 802.11 MAC

Using a single wireless interface results in less energy consumption, and using a 802.11 interface results in easier deployment due to more commonly found and inexpensive hardware. However, utilizing multiple channels using a single, unmodified 802.11 wireless interface is difficult because the 802.11 standard does not support utilizing multiple channels in the same network. The Slotted Seeded Channel Hopping (SSCH) protocol is able to use unmodified 802.11 hardware by implementing the SSCH protocol immediately above the 802.11 hardware [18]. Although SSCH does not modify the 802.11 hardware, it does change the behavior of the 802.11 standard by reimplementing some of the 802.11 MAC protocol's functionality like retransmission of RTS frames.

SSCH uses channel hopping (identical to FHSS) to utilize the multiple interfaces. SSCH uses a different channel hopping schedule for each node. Nodes are made aware of each other's channel hopping schedules by periodically propagating their channel hopping schedules to their neighbors. A node propagates its channel hopping schedule by broadcasting its schedule and current offset on the current channel. Only the nodes currently on

that channel will receive the updated information. SSCH maintains packets in per-neighbor FIFO queues. When a node changes to a channel, it determines which receivers it believes are currently on the channel by examining its copies of their channel hopping schedules. It sends any queued outgoing packets for all neighbors it believes to be on the current channel until it changes to the next channel in its schedule.

Because SSCH stores all outgoing packets in per-neighbor FIFO queues and only transmits to neighbors on the current channel, SSCH introduces an additional delay when sending packets. Before being sent, outgoing packets must wait until the node changes to a channel where it believes the receiver is present. Because a receiving node might not be on the current channel (even through its channel hopping schedule indicates it should be there), SSCH must keep a copy of all packets until it realizes they have been successfully received. SSCH does not drop any packets until all packets to that destination have been dropped for the duration of an entire cycle through its channel hopping schedule. It must wait through the entire cycle before dropping any packets to make sure the receiver's channel hopping schedule does not overlap with its own.

Because nodes are frequently on different channels, broadcast packets transmitted on any one slot are likely to reach only a small subset of the nodes that are within transmission range. SSCH addresses this issue by periodically retransmitting broadcast packets so they statistically are received by a significant number of the nodes in transmission range. SSCH relies on upper layer protocols to ensure the destination receives the proper broadcast packets since it does not guarantee a broadcast packet reaches all its neighbors.

Since SSCH uses channel hopping, clock synchronization is needed. Although SSCH will work when all nodes' clocks in the network are not properly synchronized, its throughput will be dramatically less. Simulations of SSCH showed a significant decrease in throughput when clocks drifted more than 100 microseconds apart. Also, SSCH only works well when all nodes know each other's channel hopping schedules and current offsets. If a sender does not have this information accurate, it will not be able to determine which channel the receiver currently occupies to send packets to it. Since channel hopping schedule updates are performed in periodic broadcast packets, neighbors are not guaranteed to receive these updates.

To avoid modifications to the 802.11 hardware, SSCH must circumvent some of the 802.11 hardware functionality. To maintain control over transmission attempts, the 802.11 hardware is set to buffer no more than one packet at a time and to attempt to transmit only

a single RTS frame for each packet. SSCH requires these modifications because it handles packet retransmissions in a significantly different manner. Although SSCH uses a single 802.11 interface, it requires clock and schedule synchronization between nodes, introduces delays while sending packets, and does not guarantee the reception of broadcast packets to all nodes in transmission range.

1.2 Contribution

The contribution of this work is the design and performance evaluation of a single interface, 802.11, multi-channel protocol named DCAP (pronounced 'd-cap'). DCAP (Distributed Channel-switching Accessory Protocol) is a new protocol to augment, without modification, the 802.11 MAC protocol without requiring clock synchronization. Using DCAP, a WMN is able to achieve more network throughput than is possible with current 802.11 networks using only a single channel. This work demonstrates this increased throughput with a series of validation tests and performance evaluations.

The remainder of this paper is organized into the following sections:

- Chapter 2 - Describes the DCAP protocol and how it sends and receives on multiple channels.
- Chapter 3 - Provides modifications to the ns-2 simulator that implements DCAP.
- Chapter 4 - Evaluates the performance of the proposed protocol.
- Chapter 5 - Concludes the paper and describes future research that could result from this work.

Chapter 2

DCAP Design

2.1 Overview

DCAP (Distributed Channel-switching Accessory Protocol) is a protocol to utilize multiple channels of a 802.11 wireless network to maximize the network's capacity. DCAP defines the methods wireless nodes use to switch channels, receive packets, and send packets. It allows a wireless node with a single 802.11 wireless interface to utilize multiple channels to achieve better throughput than networks using only a single channel. DCAP distributes the network's traffic load across all available channels in the network. By distributing this load, DCAP can achieve better throughput than a network utilizing only a single channel.

DCAP allows multiple, simultaneous communications to occur between wireless nodes inside transmission range of one another. Since each node can receive packets on a different channel, multiple nodes within transmission range of each other can be sending packets on different channels at the same time. Although only a single transmission can occur within transmission range on each channel, using multiple channels allows multiple transmissions to occur simultaneously. This ability to use multiple transmissions to occur allows DCAP to achieve a higher throughput than is possible in a network using only a single channel.

DCAP achieves this better throughput without making any modifications to the 802.11 MAC protocol. DCAP operates immediately under the IP protocol inside the link

layer (see Figure 2.1). It intercepts packets going to the 802.11 MAC protocol [14] and, based on the destination MAC address, determines which channel is the receiver's Home Channel. It then switches to this channel before passing the packet to the 802.11 MAC. Because of this separation from the MAC protocol, DCAP will work with any 802.11 network. DCAP will also work with any routing protocols since it works under the IP protocol.

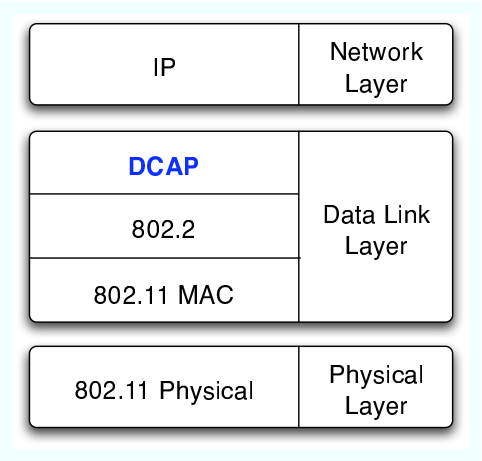

Figure 2.1: DCAP is Implemented Between the IP and the MAC Layers

DCAP achieves a much higher throughput by each node determining a "Home Channel" that they will receive packets on. A node will only receive packets on its Home Channel, and will always reside on this channel when idle. In order to send a packet, the sender must change to the Home Channel of the receiving node so both nodes will be on the same channel. By using this method to send and receive, the network maintains connectivity while taking advantage of multiple channels.

The following sections describe in detail how DCAP works in conjunction with the 802.11 MAC protocol to send and receive packets on multiple channels. These sections also explain the information obtained and used by DCAP, and how nodes determine their Home Channels.

2.2 Receiving Packets

A node will stay on its Home Channel unless it is currently sending a packet. While on its Home Channel, a node will listen to the channel so it can receive incoming packets. When the node sends a packet, it will switch to the receiver's Home Channel to send the packet, but it will return immediately to its Home Channel to listen for incoming packets. When a node returns to its Home Channel, it will remain there for T_L time to receiving incoming packets sent on its Home Channel (see Section 2.3). Whenever a node detects a channel is busy, it will set its backoff timer according to the 802.11 standard. It will then return to its Home Channel until the timer expires so it can receive any incoming packets sent during this time.

For example, when a sender, node A, wants to send a packet to node B, it will change to node B's Home Channel (see Figure 2.2). If node A is sending a unicast packet, it will send an RTS frame to node B. Node B will respond by transmitting a CTS to node A. Node A will then send the data, and node B will send an ACK. This entire exchange of information is done on the receiver's (B's) Home Channel. Once node A has received the ACK, it will then switch back to its Home Channel and remain there for T_L time. Node B will continue to listen on its Home Channel for other incoming packets.

By always staying on its Home Channel when not sending, a node is available to receive incoming packets for the maximum amount of time. Nodes are always able to determine which channel to send to a neighbor because nodes only receive packets on their Home Channels. The only information a node must have to send a packet is the receiver's Home Channel. The Home Channel assignment algorithm, discussed in Section 2.9, is the method nodes use to select their Home Channel.

2.3 Sending Packets

To send a packet, DCAP must change to the Home Channel of the receiver. However, it must first identify the receiver. When a node sends a packet, DCAP will intercept the packet from the IP protocol. After intercepting the packet, DCAP examines the packet's MAC address. If the MAC address indicates the packet is a broadcast packet, then DCAP must broadcast the packet on every channel (see Figure 2.3). DCAP transmits broadcast packets identical to the 802.11 standard. However, DCAP must repeat the process several

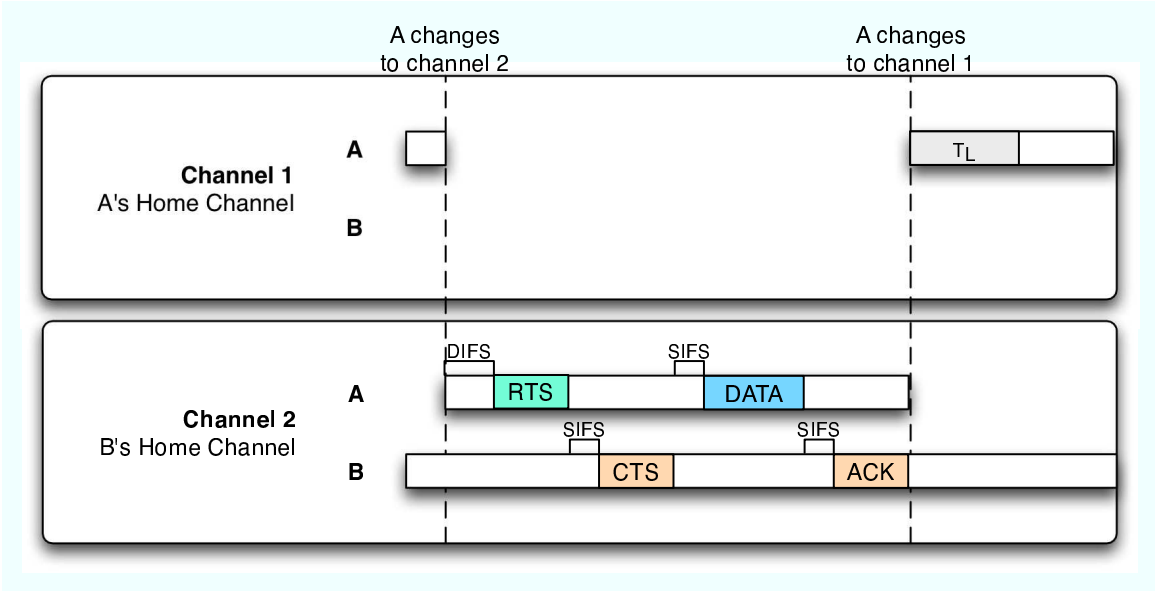

Figure 2.2: Node A Sending a Packet to Node B

times so it can send the packet on each channel which is the major source of overhead in DCAP. To send a broadcast packet, DCAP starts broadcasting the packets on its Home Channel then continues broadcasting the packet on each channel sequentially in increasing order. When a node changes to the channel, it will wait for Distributed Inter Frame Space (DIFS) + backoff time before sending the broadcast packet. After the packet has been successfully sent, DCAP will change to the next channel, wait for DIFS + backoff time, and repeat the transmission. Once DCAP has broadcast the packet on every channel, it returns to its Home Channel to listen for incoming packets.

If, after intercepting a packet from the IP protocol, DCAP determines from the packet's MAC address that the packet is a unicast packet, it will only transmit the packet on the receiver's Home Channel. The Channel Table (described in detail in Section 2.6) contains the MAC address and Home Channel of each node in transmission range. If the node does not find the receiver's Home Channel in its Channel Table, then it will need to send a Channel Request Packet to determine the receiver's Home Channel (see Section 2.5). However, if DCAP's Channel Table contains the Home Channel of the receiver, DCAP will change to the receiver's Home Channel and send the packet. Once on the receiver's Home

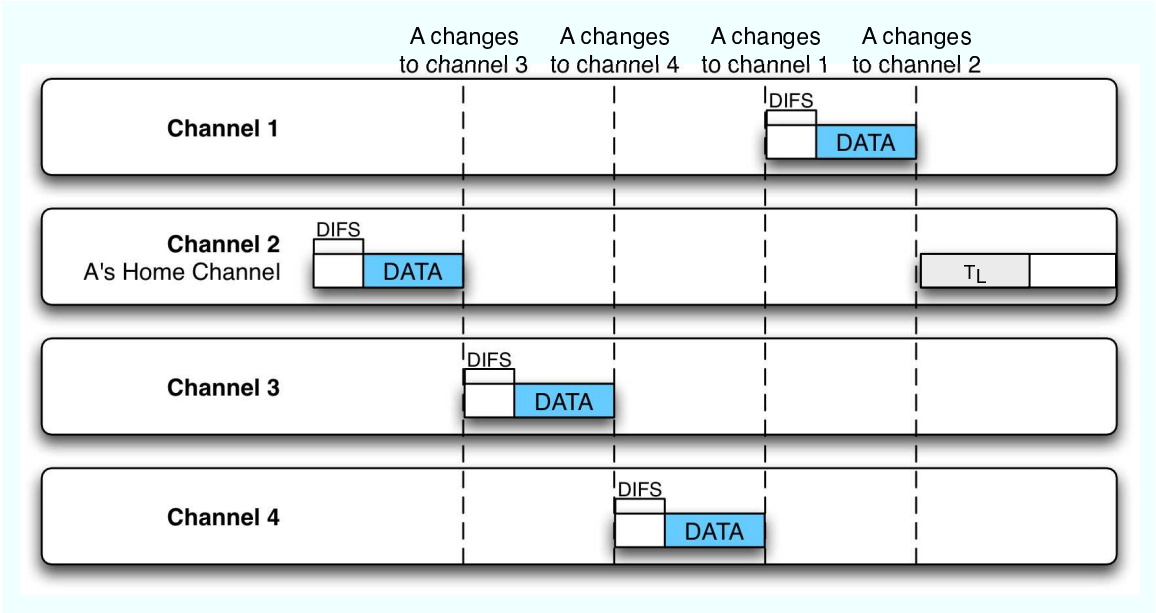

Figure 2.3: Sending a Broadcast Packet on Multiple Channels

Channel, DCAP transmits the packet identical to the 802.11 standard.

After the node changes to the receiver's Home Channel, it will determine if the channel is idle. If it is, then the node will wait DIFS time before sending the RTS packet (see Figure 2.2). The node will continue to follow the 802.11 standard while receiving the CTS, sending the data, and receiving the ACK. After the node has finished receiving the ACK, it will return to its Home Channel to listen for incoming packets. The node will listen on its Home Channel for T_L seconds so it can receive incoming packets before leaving its Home Channel again (see Figure 2.2). Determining the T_L time is explained in more detail in Section 2.6. If a node changes to the receiver's Home Channel and detects that the node is busy, it will set its backoff timer according to the 802.11 standard. It will then return to its own Home Channel to listen for incoming packets while waiting. When the backoff timer expires, it will change to the receiver's Home Channel again to send (see Figure 2.4). If the channel is detected as busy again, it will set its backoff timer and return to its Home Channel again. Only when it detects the receiver's Home Channel as idle, will it wait the DIFS time and send the packet.

Staying on its own Home Channel during the time a node is backing off allows

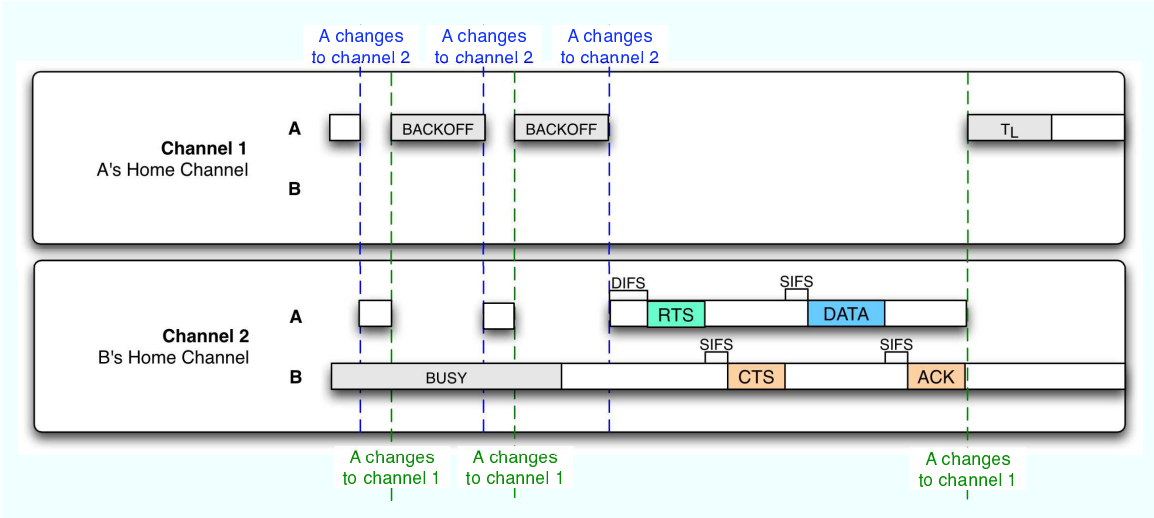

Figure 2.4: Sending on Busy Channel

the node to receive incoming packets during the duration of the backoff timer. If a node is backing off on its Home Channel and it receives an incoming packet, it will pause its backoff timer and receive the packet according to the 802.11 standard. After the reception is complete, it will resume its backoff timer and continue to listen on its Home Channel. After the backoff timer expires, it will then switch to the receiver's Home Channel to transmit the packet.

2.4 Listen Time (T_L)

Since a node must change to the receiver's Home Channel to send a packet, it might leave its Home Channel for the duration of the transmission. Ideally, its Home Channel will be different than the receiver's to insure minimal interference (see Section 2.9). If its Home Channel is different than the receiver's Home Channel, it will have to leave its Home Channel to send the packet. If a node has many packets to send, it will frequently leave its Home Channel. During the time a node is sending packets, it will not be on its Home Channel to receive incoming packets.

If a node were to immediately leave its Home Channel when it has a packet to send, it may never remain on its Home Channel long enough to receive any incoming packets. To

insure a node stays on its Home Channel long enough to receive incoming packets, DCAP will delay sending a new packet, with a receiver on another channel, to the 802.11 MAC protocol for T_L seconds. This T_L time must be long enough for the receiving node to receive an RTS frame from a sender before it changes from its Home Channel. Once the T_L time has expired, DCAP will send down the next packet to the 802.11 MAC protocol. By delaying the packet from being received by the 802.11 MAC protocol and returning to the node's Home Channel, DCAP forces the 802.11 MAC to receive incoming packets. The 802.11 MAC protocol will receive incoming packets since it will be unaware that it has any packets to send. If the next packet to be sent has a destination node on the sender's Home Channel, then DCAP does not wait the T_L time before sending the packet.

The T_L value also reduces the number of packets dropped because a node is unable to reach the receiver. If the sender sends a packet on the receiver's Home Channel while the receiver is on another channel, then the sender's transmission will fail. However, the sender may detect an idle channel because no other nodes are transmitting on the receiver's Home Channel (see Figure 2.5). This problem is called the Absent Receiver Problem because the receiver is not on its Home Channel to receive the packet. In this case, the RTS transmission will time out, and the sender will set its backoff timer and return to its Home Channel. After the backoff time expires, the sender will return to the receiver's Home Channel to transmit the RTS frame again.

In this scenario, the sender is unable to set it's NAV to wait until the receiver's transmission is complete, since it does not hear the receiver's transmission. The sender must continue to send the RTS frames on the receiver's Home Channel until the receiver responds with a CTS frame. This process follows the 802.11 standard for retransmitting RTS frames. As defined in the 802.11 standard, the retransmission of RTS frames will end after seven retransmissions. At this point, the packet will be dropped and the next packet sent. By using a sufficiently large value for T_L, DCAP can ensure that a node stays on its Home Channel long enough to receive the RTS frame so the number of RTS frame retransmissions can be decreased or eliminated.

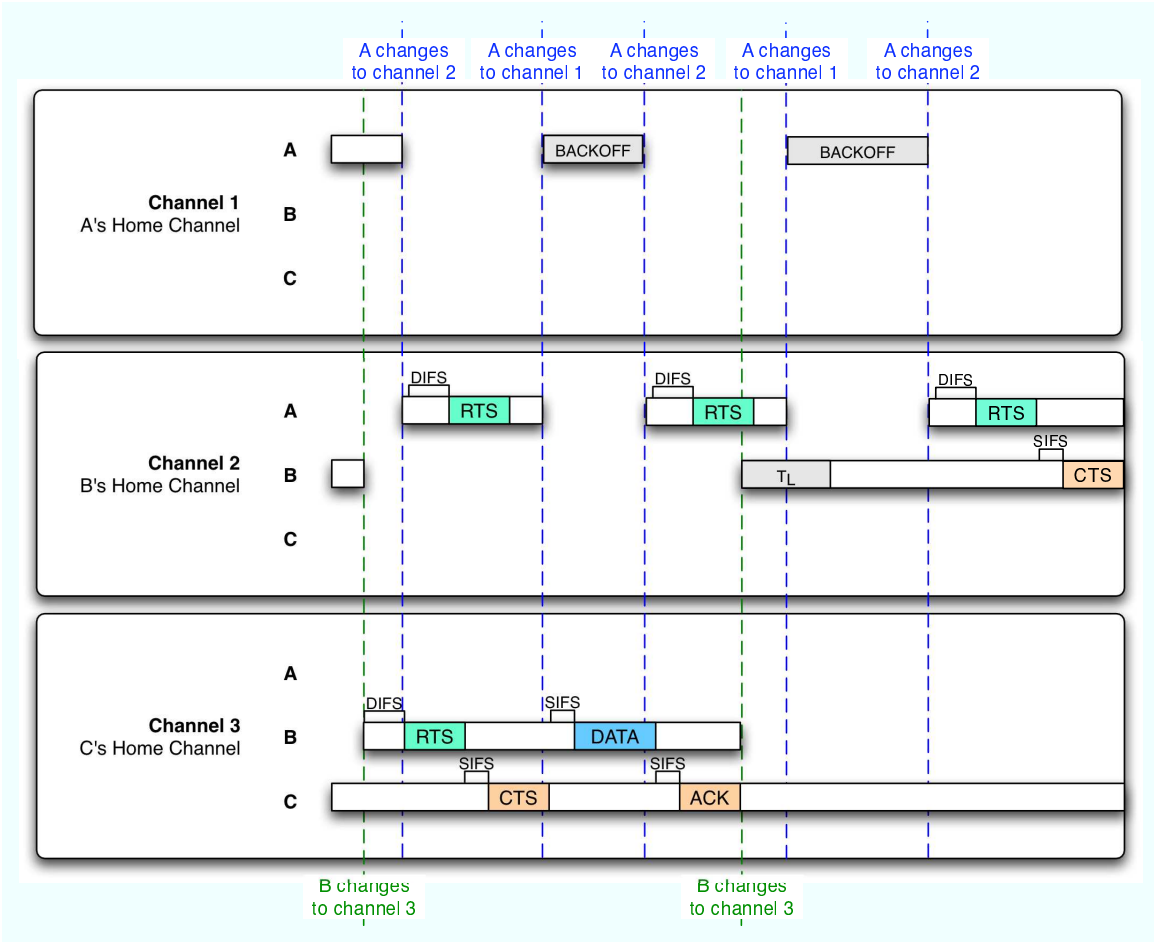

Figure 2.5: A Value of T_L to Small Leading to Transmission Failures From Node A to B When B Also Sends to C

2.5 Determining a Receiver's Channel

When DCAP intercepts a unicast packet, it must look up the receiver's MAC address in its Channel Table. If the Channel Table contains the receiver's Home Channel, then it will switch to that channel to send the packet (see Section 2.3). However, if the Channel Table does not contain the receiver's Home Channel, the sender must request it from its neighbors. The sender requests a node's Home Channel by broadcasting a Channel Request Packet. The Channel Request Packet contains the MAC address of the host whose Home Channel is needed. The Channel Request Packet is broadcasted in the same manner as other broadcast packets (see Section 2.3). The node will transmit the Channel Request Packet on all channels starting with its Home Channel and continuing sequentially in increasing order (see Figure 2.3).

When a node receives a Channel Request Packet, it will examine the requested MAC address. If the receiving node does not know the Home Channel of the requested MAC address, it will ignore the Channel Request Packet. However, if the node knows the Home Channel of the requested MAC address (even if its MAC address is not the requested one), it will send a Channel Reply Packet back to the sender. The Channel Reply Packet contains the MAC address of the requested node and the node's Home Channel. Any node hearing the Channel Reply Packet will update their Channel Table with the requested node's MAC address and Home Channel. After the original sender receives the Channel Reply Packet and records the requested node's MAC address and Home Channel, it will transmit the original packet to the receiver.

Since the Channel Request Packet is a broadcast packet, the receiving nodes will not acknowledge the successful reception of the packet. Packet collisions or radio interference can cause the broadcast packet to become corrupted and discarded by the receivers. To ensure that the neighbors successfully receive the Channel Request Packets, the sender will set a retransmission timer, T_Q. Once the T_Q timer has expired, the sender will retransmit the Channel Request Packet on all channels. When the sender receives a Channel Reply Packet or a Home Channel Packet that resolves its channel request, it will cancel the timer and proceed to send the original packet. However, if the request has not been resolved after three failed Channel Request Packet transmissions, the node will cancel the T_Q timer and drop the original packet.

During the process of determining the receiver's Home Channel, the sender will

store the original packet in DCAP's packet queue. This packet queue contains all the packets whose receiver's Home Channels are not known. Whenever a node receives a Home Channel Packet (see Section 2.6) or Channel Reply Packet, it will check it's DCAP queue for any packets it is able to send. When it determines it can send a packet from the queue, it will send the packet down to the 802.11 MAC protocol when the MAC protocol indicates it is available to send another packet. This process ensures that packets in the DCAP queue are sent before other packets.

2.6 Channel Table

Each node maintains a table of MAC addresses and Home Channels. This table, called the Channel Table, is used to resolve nodes' MAC addresses with their Home Channels when sending packets. The Channel Table contains information about each of the node's one-hop and two-hop neighbors. Each entry in the Channel Table contains the following information:

- MAC address
- Home Channel
- time of the last successful transmission
- reported channel load
- hop distance

The MAC address is used as the index into the table and identifies the information as belonging to that neighbor. The Home Channel is the channel on which the neighbor receives packets on (see Section 2.2). The time of the last successful transmission is used to purge old entries from the Channel Table. The reported channel load and the hop distance are used in the distributed channel assignment algorithm (see Section 2.9).

The hop distance to a node is the number of hops it takes a packet to reach the node. This includes the receiver and all nodes that forward the packet towards the receiver. Every wireless node keeps entries in its Channel Table of all nodes within a two-hop distance. This information is used during the channel assignment algorithm. Only those entries with

a hop distance of one-hop are used to determine Home Channels when sending packets since a node can only directly communicate with nodes with a one-hop distance.

Since information in a node's Channel Table can become inaccurate over time, a node must purge old entries occasionally. A node determines which entries are accurate by using the time of the last successful transmission. After every successful transmission to a host, the sender updates the Channel Table entry corresponding to that host with the current time. A node also updates its Channel Table when it receives a Home Channel Packet (see Section 2.7). To purge old entries, every T_P seconds the time of the Channel Table's entries are examined and all entries that have a time older than T_P seconds are removed. This ensures that the Channel Table contains accurate information.

2.7 Neighbor Discovery (T_N)

Determining a receiver's Home Channel after DCAP intercepts a packet to send introduces a small delay. To reduce this delay, DCAP provides a method of proactive Home Channel discovery through the exchange of Home Channel Packets. These packets are exchanged every T_N seconds. Home Channel Packets are also used in the Home Channel assignment algorithm (see Section 2.9). To proactively discover neighbors, a node will periodically send Home Channel Packets. Home Channel Packets include a node's MAC address and its Home Channel. The Home Channel Packet also includes the MAC addresses and Home Channels of all of the node's neighbors within transmission range (one-hop neighbors).

When a node receives a Home Channel Packet, it will update its Channel Table with the sender's MAC address and Home Channel (see Section 2.6). It will also update the Channel Table with all of the MAC addresses and Home Channels of the sender's one-hop neighbors. When the node updates its Channel Table with the sender's information, it will update the record's time of last successful transmission. A node does not update this time when updating records of the sender's one-hop neighbors. A node only updates the time value of the sender because, by sending the Home Channel Packet, the sender verifies the information is still accurate. Information about the sender's one-hop neighbors might not be accurate, so the one-hop neighbor's time of last successful transmission should not be updated.

A node will broadcast a Home Channel Packet every T_N seconds. The periodic broadcasts of Home Channel Packets allows nodes to maintain accurate information about their neighbors and reduces the delay caused by sending Channel Request Packets to resolve a node's Home Channel. DCAP does not require this neighbor discovery ability to function, since it will obtain the a receiver's Home Channel as needed by sending Channel Requests Packets. Using a small value of T_N will introduce more overhead as nodes frequently broadcast Home Channel Packets unnecessarily. Using a larger value of T_N will reduce this overhead but introduce the overhead and delay associated with Channel Request Packets. The neighbor discovery ability can be disabled completely with a T_N value of zero. Performance results of the T_N value are discussed in Section 4.4.

2.8 Gateway Node

The gateway node is a special wireless node in the wireless mesh network. Since most traffic in a wireless mesh network is coming from or going to the Internet or wired network, all this traffic will flow through the gateway node. To handle the higher traffic demands, the gateway node is constructed differently than the other wireless nodes. The gateway node will have a separate wireless 802.11 interface for each channel used in the network. This allows the gateway to simultaneously carry on several communications at once by utilizing the different channels. However, only the single gateway node uses multiple wireless interfaces. Since the gateway node has a separate wireless interface for each channel, its wireless interfaces do not change channels to send packets. Instead, the gateway node sends the packets to the correct wireless 802.11 interface based on the Home Channel of the packet's MAC address.

2.9 Home Channel Assignment

Nodes receive all incoming packets on their Home Channel and send packets to their neighbors on their neighbor's Home Channel. To minimize interference between neighboring nodes, a node must select a Home Channel different from all nodes in interference range. The interference range is assumed to be twice the reception range. Thus, all communication by nodes up to two-hops away will cause interference at a node (see Figure 2.6).

In addition, a node will be able to detect the presence of communication from nodes up to two-hops away. Although a node cannot receive the communication from nodes farther than one-hop away, the node will be able to detect that the channel is currently busy if the node is two-hops away. Since nodes over two-hops away do not cause interference, a node can select the same Home Channel as a node that is three-hops away. However, since nodes two-hops and closer cause interference, a node must select a different Home Channel from all nodes up to two-hops away.

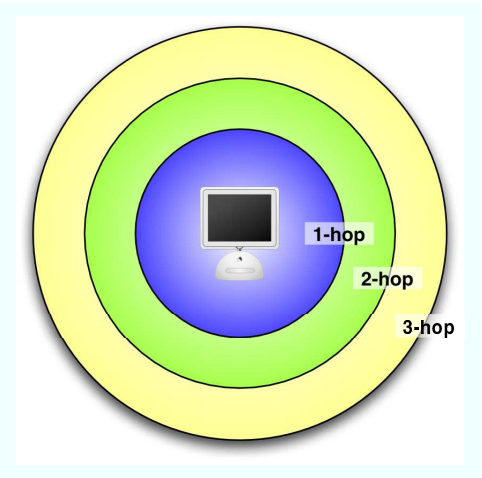

Figure 2.6: Hop Distance Between Neighbors

The channel assignment algorithm that DCAP uses is based on the algorithm used in the Hyacinth architecture [1]. Hyacinth utilizes multiple channels in a wireless mesh network by using multiple 802.11 interfaces per node. They use the channel assignment algorithm to determine the channels that each of a node's interfaces should remain on. Although they use multiple network interfaces, their algorithm can be modified for DCAP's channel assignment requirements. DCAP uses a modified version of their algorithm to select which channel a node should designate as its Home Channel. The remainder of this section will be divided into three parts. First, the Hyacinth channel assignment algorithm will be described. Next, the differences between the Hyacinth channel assignment algorithm and the modified algorithm used by DCAP will be covered. Last, DCAP's modified channel assignment algorithm will be described in detail.

2.9.1 Hyacinth Channel Assignment Algorithm

In the Hyacinth network, multiple channels are used by equipping all wireless nodes with multiple interfaces. Unlike DCAP, the wireless nodes do not change channels to send packets. Instead, a wireless node can only send packets to those neighbors that have a wireless interface located on the same channel as the node's interface. A wireless node with two interfaces can only use two channels. Network connectivity issues arise since a network might have more available channels than its nodes have interfaces. For the Hyacinth network, proper channel assignment for each interface is important to maintain network connectivity. Channel assignment is also done to increase network throughput, so a node attempts to select the channel with the most available bandwidth.

The Hyacinth network has a single gateway node and multiple wireless nodes. Before the channel assignment algorithm is run, the nodes arrange themselves in a tree with the gateway node as the root. Multiple gateway nodes can be used, but each will result in a separate tree. Building the tree allows several of Hyacinth's algorithms to be run such that they understand the path and the number of hops to the gateway.

The Hyacinth channel assignment algorithm divides the channel assignment problem into two subproblems: neighbor-interface binding and interface-channel binding. In neighbor-interface binding, a node divides its wireless interfaces into UP-NICs and DOWN-NICs. UP-NICs are used to communicate with a node's parents, while DOWN-NICs are used to communicate with a node's children. Most nodes attempt to split the available interfaces equally between DOWN-NICs and UP-NICs. However, nodes further from the gateway will assign fewer interfaces as UP-NICs to help aggregate traffic.

The primary reason behind the distinction between UP-NICs and DOWN-NICs is to prevent channel changes from propagating throughout the network. A node selecting a new channel for one of its interfaces might cause all of the node's neighbors to select new channels for their interfaces. Also, to maintain network connectivity, a node must verify that both it and its neighbors have interfaces on the same channel. To solve these problems, a node can only assign a new channel to one of its DOWN-NICs. Its UP-NICs will always stay on the channel that the node's parent determines.

For the second subproblem, interface-channel binding, the channel for each interface will be determined. The channel assignment of a node's UP-NICs is done by the node's parent. A node's DOWN-NICs are assigned channels based on the usage of the channels

within interference range. Interference can be caused by any of a node's k-hop neighbors, where k is the ratio of interference range to transmission range. Thus, each node must periodically exchange individual usage information with all of its $(k+1)$-hop neighbors to create a valid estimate of the channel's usage.

A node must communicate with all $(k+1)$-hop neighbors because it will be making a channel assignment decision for itself as well as all of its children. Since all nodes k hops away from its children can cause interference, it must use the channel usage information of its children's k-hop neighbors (its $(k+1)$-hop neighbors). A node's channel usage is determined by the amount of data a node transmits. The channel usage estimate is computed by the weighted combination of the sum of all interference neighbor's channel usage and the number of nodes on the channel.

Once a node has computed the channel usage of all channels, it will then select a new channel for its DOWN-NICs. A node will select a new channel for only one of its DOWN-NICs every T_C time units. Channel assignment priority is given to nodes closer to the gateway since they need more bandwidth to forward traffic. During the assignment, the node will exclude all channels containing interference neighbors with higher priority. This restriction allows nodes closer to the gateway to choose the channels with the least usage. Once a node has determined its new channel, it will send a message including its new channel to its children so they can reassign their corresponding UP-NICs. The node will then send its new channel usage information to each of its $(k+1)$-hop neighbors to ensure they know the load of the new channel.

2.9.2 Differences Between Channel Assignment Algorithms

Since the Hyacinth channel assignment algorithm is used in a network with multiple wireless interfaces on each node, it had to be modified to be used in DCAP. Only minimal modifications were required to apply the Hyacinth algorithm to a wireless nodes with a single interface. The primary modification is the elimination of the UP-NICs. Since DCAP dynamically changes channels to send packets, it does not have the connectivity problems present in the Hyacinth network. Also, since DCAP uses only a single interface per wireless node, the number of DOWN-NICs is always one. These two changes eliminates the neighbor-interface binding subproblem solved by the Hyacinth network.

DCAP only requires minimal modifications to the interface-channel binding sub-

problem. Like Hyacinth, DCAP makes channel assignments based on channel usage. However, unlike Hyacinth, DCAP uses the amount of data a node receives for the channel usage computation instead of the amount of data transmitted. Since, in DCAP, a node transmits on many channels, but always receives only on one, basing the channel usage on the amount of data receives is more appropriate. DCAP bases the channel usage only on unicast packets since broadcast packets are sent on all channels.

Since DCAP does not have access to a network topology tree to determine parents, children, and distance to the gateway, DCAP handles channels of interference neighbors slightly differently. Instead of excluding interference neighbors closer to the gateway, DCAP places priority on channels with no interference neighbors. In addition to selecting channels with no interference neighbor closer to the gateway, this method also selects channels with no other nodes that might cause interference. Since selecting a channel with no interference neighbor present might be impossible, DCAP does not require their selection. Instead, DCAP will prioritize it by selecting a channel that has more channel usage with no interference neighbors over a channel with less channel usage with interference neighbors.

The last modification to the Hyacinth channel assignment algorithm DCAP makes is to reduce the number of neighbors it must exchange information with. Since DCAP is only making the channel assignment decision for itself, it only needs to base the channel assignment decision on the channel usage of its k-hop neighbors instead of its $(k+1)$-hop neighbors.

2.9.3 DCAP's Modified Channel Assignment Algorithm

DCAP's channel assignment algorithm uses the modifications to the Hyacinth algorithm described in Section 2.9.2. DCAP's channel assignment algorithm is run every $T_C + X$ seconds where X is a random number between 0 and 0.1. The random value is used to ensure that two nodes are not likely to change Home Channels at the same time. If two nodes were to change Home Channels at the same time, they might both independently decide to change their Home Channel to the same new channel. This would result in more contention on that channel and could decrease the throughput in the network. Every $T_C + X$ seconds, each node decides which channel has the least load and designates this channel as its new Home Channel. Priority is given to channels that do not contain interference neighbors.

The channel assignment algorithm determines the load of each channel by summing the channel utilization reported by all nodes on that channel. Each node calculates the channel utilization by determining the total size of all unicast packets received in each $T_C + X$ time interval. Only unicast packets are used, because broadcast packets are sent on all channels and received by all nodes. The load calculation is shown in Equation 2.1. The load on channel i is the sum of the channel utilization of all nodes j that have a Home Channel of i. The channel utilization of j is the total size in bytes (B) of all unicast packets received by node j since node j last changed its Home Channel ($T_C + X$). The result of Equation 2.1 is the traffic load on each channel in the network. This value is used to select the least loaded channel to designate as a node's Home Channel.

$$\text{channel load}_i = \sum \left[\frac{B}{T_C + X} \right]_j, \forall \text{ nodes } j \text{ with Home Channel } i \tag{2.1}$$

Once a node has determined the load of each channel, it will choose the channel with the least load. Priority will be given to channels with no one-hop or two-hop neighbors to reduce the amount of interference. A node will select the channel with the least load that contains no one-hop or two-hop neighbors. If all channels contain at least one one-hop or two-hop neighbor, then the node will chose the channel with the least load that contains only two-hop neighbors. If no channels exist that contain only two-hop neighbors, then the node will select the least loaded channel out of all available channels. This priority system is used to reduce the amount of interference caused by nodes one-hop or two-hops away transmitting on the same channel.

Chapter 3

Implementation

The performance of DCAP was studied through a series of validation tests and performance evaluations. The validation tests were done to verify that DCAP performed as expected and to examine any overheads that DCAP introduced. The performance evaluation (discussed in Chapter 4) examines DCAP's performance in real-world scenarios.

The performance of DCAP was studied using ns-2 simulations. Ns-2 is a discrete event simulator supporting simulation of wireless networks, including the 802.11 MAC protocol [2]. Ns-2 was modified to allow wireless nodes to change channels during the simulation. DCAP was implemented in ns-2 by inserting the new protocol between the IP protocol and 802.11 MAC Protocol. Ns-2 does not support wireless nodes with multiple network interfaces, so the gateway node was implemented as a wired node connected to a separate wireless node for each interface on the gateway (see Figure 3.1). The gateway's interface nodes were connected to the wired node with network links with zero latency and a transmission rate of 100 gigabits per second. The result emulates a single gateway node with multiple interfaces since the network links introduce insignificant delay.

Ns-2's AODV (Ad hoc On-Demand Distance Vector Routing) routing protocol only supports a pure wireless ad hoc network, and does not support routing between wireless and wired nodes. Other routing methods are required to allow wireless nodes to determine proper routes to the wired nodes. To overcome this limitation, Ali Hamidian's AODV+ module was used as the routing protocol [5]. Ali Hamidian modified ns-2's AODV protocol to support routing between wired and wireless nodes. Using this modification, wireless

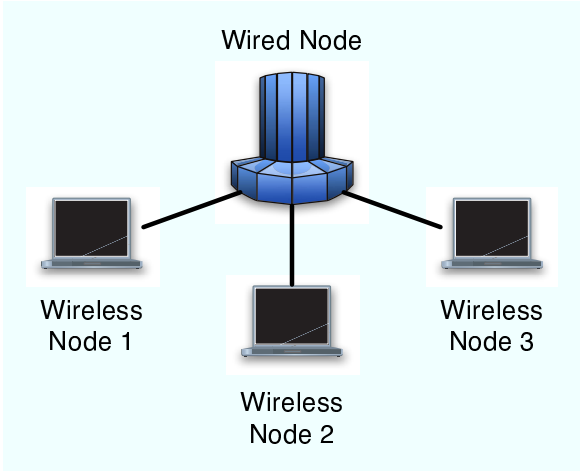

Figure 3.1: Ns-2 Implementation of the Gateway Node

nodes in both the DCAP and non-DCAP networks can resolve routes to and from the wired node portion of the gateway node.

To prove the correctness of the modifications and determine DCAP's performance, a series of validation tests were performed. These tests computed the network throughput of various network topologies to compare the performance of DCAP against a single-channel network (non-DCAP). These tests used the same network topology and traffic patterns between the DCAP and non-DCAP networks to make this comparison. All DCAP variables were set to specific values and the nodes were prevented from selecting new Home Channels. Nodes were precisely placed before the beginning of the simulation and there was no node movement during the simulation. Chapter 4 discusses the impact of the DCAP variables on the network throughput.

The number of channels utilized in the network, the number of nodes in the network, and the number of nodes sending and receiving from the gateway are different for each test; however, the distance between the nodes and the DCAP variables are all the same. All nodes are located 200 meters apart so they are within transmission range. Three nodes located in a straight line, however, are not all within transmission range since the 400 meter distance between the first and third is beyond the reach of the node's radio. The DCAP values T_C, T_N, and T_P were all set to values beyond the end of the test to disable their occurrence except once at the beginning of the test. T_Q was set to three seconds, and

T_L was set to the time it would take to send and receive the RTS and CTS frames (0.67 milliseconds).

Each node's Home Channel is manually set to achieve an optimal distribution of channels, and nodes were prevented from selecting a new Home Channel. At the beginning of each test, nodes sent Home Channel Packets to determine their neighbors and sent a single data packet to allow AODV to setup the routes. After 50 seconds, selected nodes start to send or receive data from the gateway. These data traffic flows were constant bitrate (CBR) traffic, and lasted for 110 seconds. The start and duration of the data traffic was constant for all tests, but the traffic rate was varied for each test. The results of each test are shown in the following sections as graphs of the network throughput as a function of the CBR traffic rate. The following sections describe the different test scenarios and discuss their results. Selected graphs of each test's throughput are shown below, but all the graphs of the validation results are shown in Appendix A.

3.1 Test One: Two Nodes

A network of two nodes was designed with one gateway node with a single interface and one wireless node (see Figure 3.2). The two nodes were located 200 meters apart, and the network had two available channels. Three subtests were performed (1.1, 1.2, and 1.3) containing different traffic patterns. In each of these tests, the throughput of the network with DCAP closely matches the throughput of the network without DCAP. In subtest 1.1, all traffic was coming from the wireless node and being received by the gateway. In subtest 1.2, all traffic was going from the gateway to the wireless node. In the last subtest, 1.3, traffic was going in both directions.

The results of test one show the throughput of the network was minimally impacted by the addition of DCAP. The throughput of the network was predicted to be nearly identical since the introduction of DCAP introduces no substantial overhead. Since the two nodes are located on the same channel (the wireless node's Home Channel), the nodes will not perform any channel switching. As shown in Figure 3.3 for subtest 1.3, both the throughput in the DCAP and non-DCAP networks are identical throughout the graph.

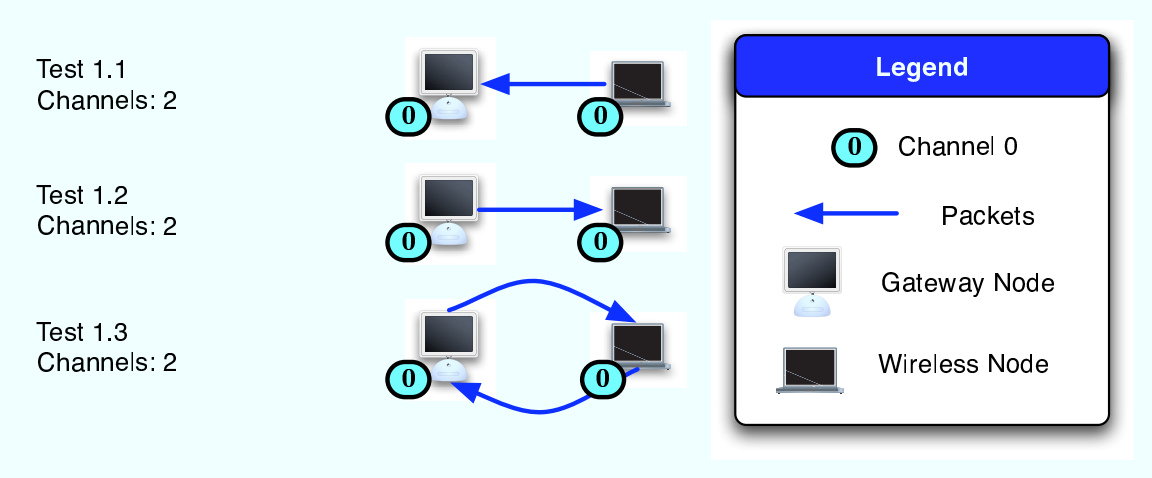

Figure 3.2: Test One: Two Nodes

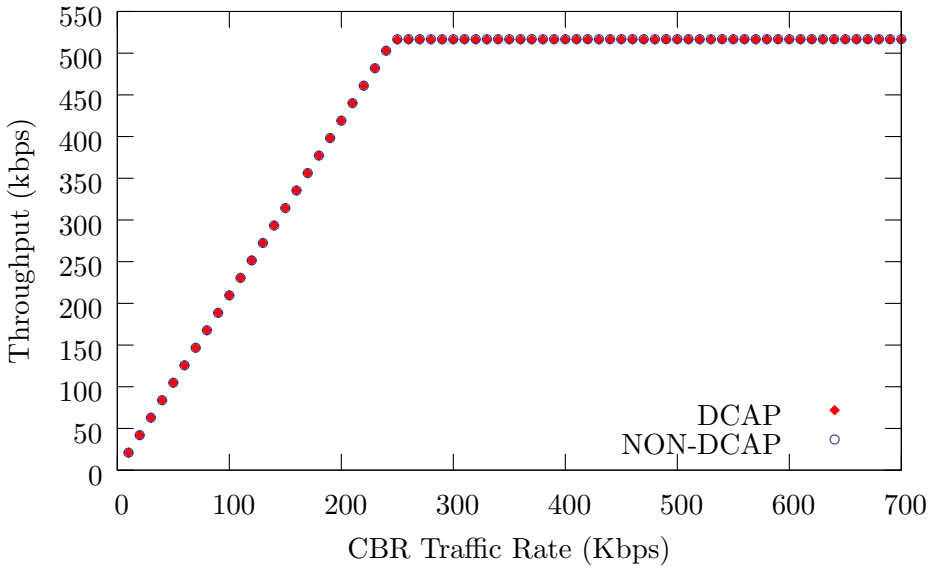

Figure 3.3: Subtest 1.3 Throughput

3.2 Test Two: Three Nodes

In test two, a network of three nodes was designed with all nodes in a line so the two wireless nodes were outside of each other's transmission range (see Figure 3.4). One gateway node containing two network interfaces was located in between the wireless nodes. Each wireless node had its own Home Channel, and one gateway interface was located on each of these channels. The network contained only two available channels. Each of the three subtests performed (2.1, 2.2, and 2.3) had different traffic patterns. The first subtest, 2.1, contained traffic going to the gateway node from the two wireless nodes. In subtest 2.2, the traffic was going to the wireless nodes from the gateway node, and subtest 2.3 had traffic flowing in both directions.

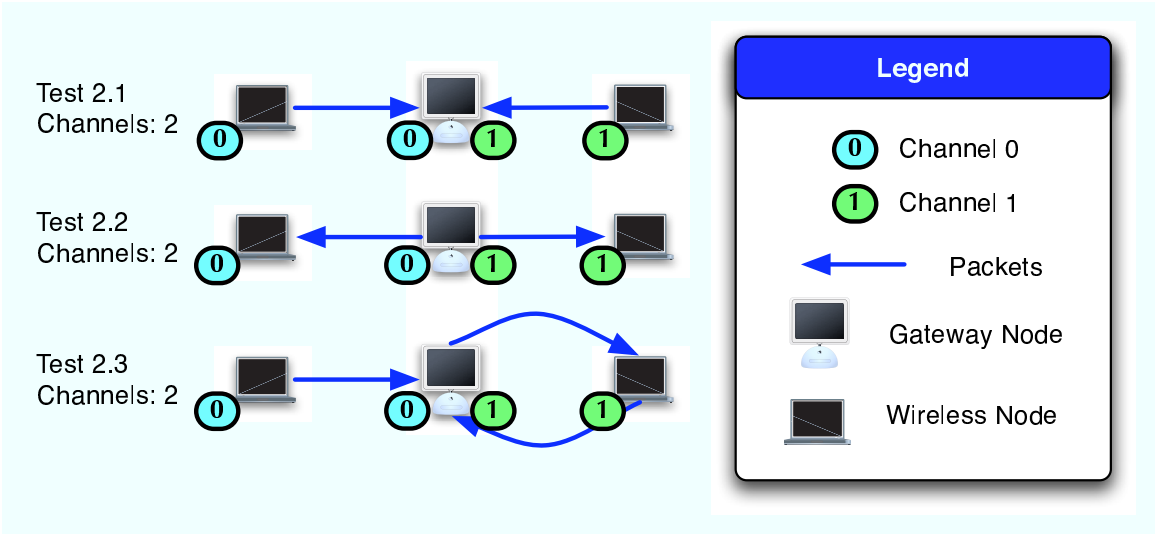

Figure 3.4: Test Two: Three Nodes

Since the DCAP network utilized two channels, DCAP's predicted throughput was slightly less than twice that of non-DCAP. As the graph (see Figure 3.5) for subtest 2.3 shows, the throughput in the DCAP network approaches twice the throughput of the non-DCAP network, as expected. The throughput of the DCAP and non-DCAP networks is identical as the offered load increases until the non-DCAP saturates the channel around 7Mbps. At that point, the non-DCAP throughput remains constant while the DCAP throughput continues to increase at the same rate. When the DCAP throughput

reaches around 10.7Mbps, it increases at a slower rate until it levels off at around 14Mbps. The DCAP throughput's rate decreases around 10.7Mbps because the channel becomes saturated at this point. Since there is twice as much traffic on channel one as channel zero, channel one will reach capacity before channel zero. At 14Mbps, both channels are at capacity, so DCAP's throughput remains constant. Since subtests 2.1 and 2.2 contain the same traffic on both channels, their throughput increases at the same rate until around 14Mbps.

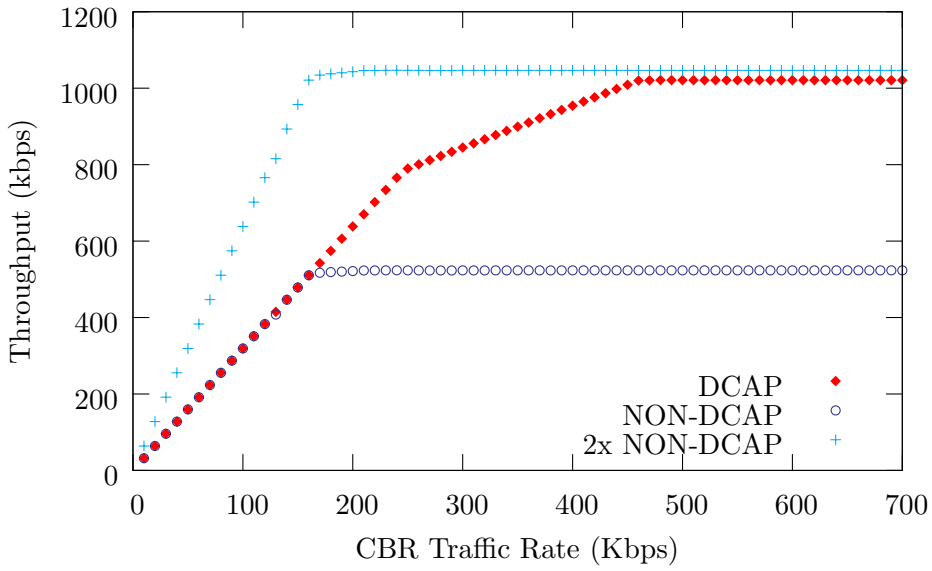

Figure 3.5: Subtest 2.3 Throughput

3.3 Test Three: Multihop Network

A network of three nodes was designed with two wireless nodes and a gateway node in a line (see 3.6). Unlike test two, test three's gateway node is not between the two wireless nodes. Since all nodes are 200 meters apart, only one wireless node is within transmission range of the gateway. The other wireless node is two hops away from the gateway, so it must send all its traffic through the middle node. The network has two available channels, and each wireless node has a separate Home Channel. The gateway node has interfaces on

each channel, but only uses one to communicate with the middle node. Six subtests were performed (3.1, 3.2, 3.3, 3.4, 3.5, and 3.6) containing different traffic patterns going to and from the gateway from and to each wireless node as shown in Figure 3.6.

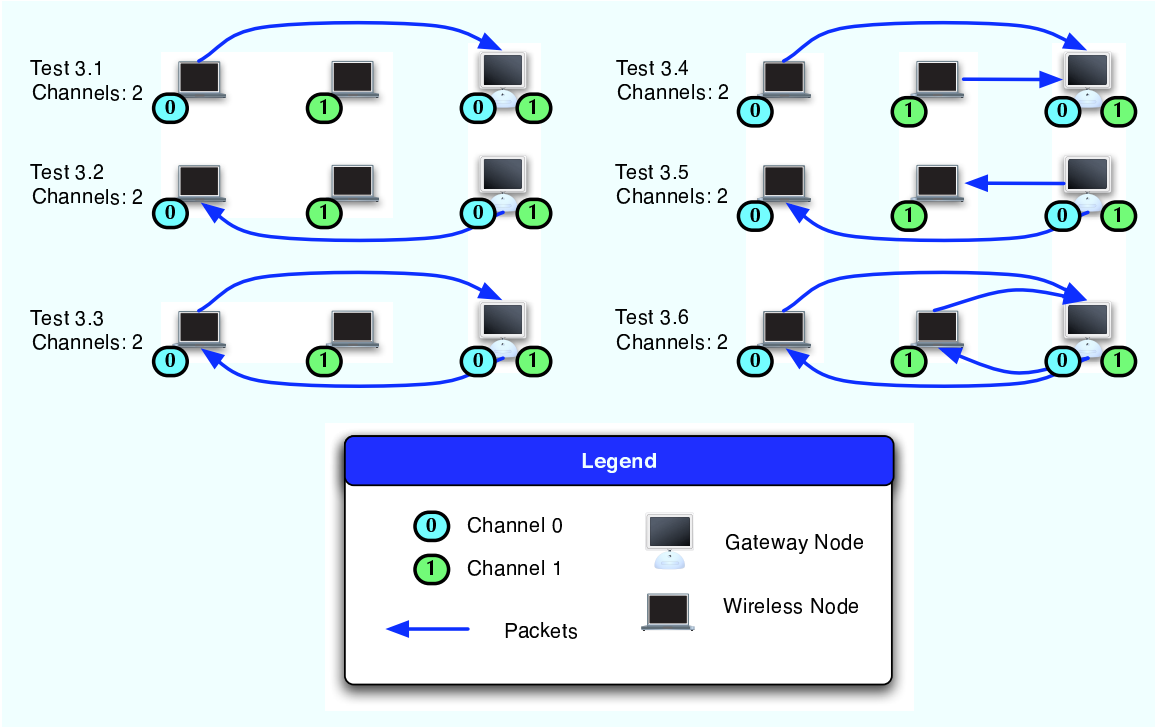

Figure 3.6: Test Three: Multihop Network

Although DCAP uses two channels, all traffic must flow through node two. Since node two must forward all of node one's traffic to the gateway, all traffic from nodes one and two must be transmitted by node two. This limitation eliminates the advantage of using multiple channels in this particular network since the throughput is limited to the capacity of the channel node two transmits on. Because of this limitation, the throughput of the DCAP network is predicted to be close to the throughput of the non-DCAP network. Since in some subtests node two is changing channels to send the traffic, the throughput for the DCAP network in some subtests is predicted to be less than that of the throughput in the non-DCAP network.

In subtest 3.1, node one changes to channel one to send to node two. When node one changes channels it can detect if channel one is busy and backoff until it becomes

idle. Node two does not change channels to send to the gateway node. Since only node one changes channels and detects when channel one is busy, the throughput of the DCAP network is expected to be very close to the throughput of the non-DCAP network. The graph in Figure 3.7 for subtest 3.1 shows that this is a valid prediction.

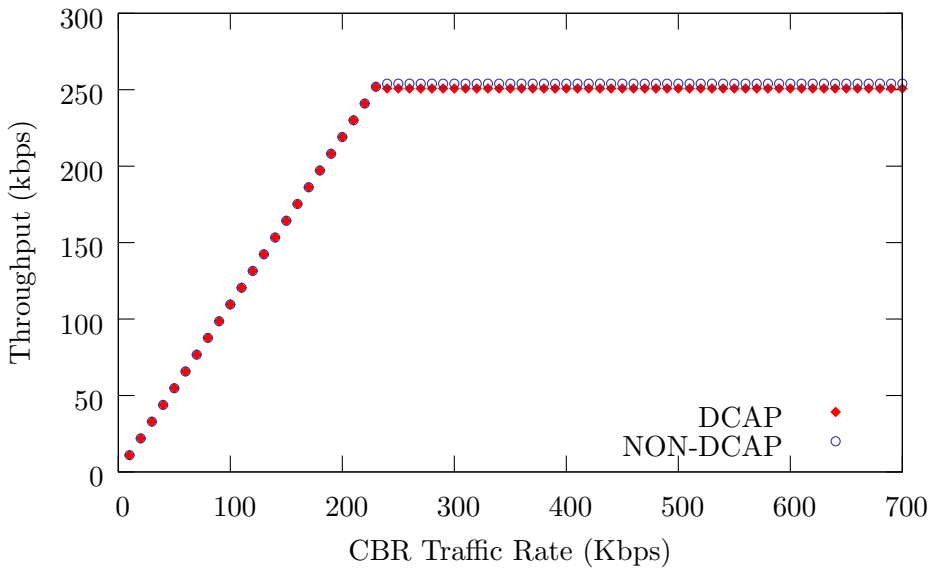

Figure 3.7: Subtest 3.1 Throughput

In subtest 3.2, the gateway node sends traffic relayed by node two to node one. This subtest can result in a reduced throughput since the gateway node cannot detect when it can send to node two. Since node two changes to channel 0 to send to node one, the gateway might send packets on an empty channel (the Absent Receiver Problem). If the gateway sends a packet on an idle but empty channel, it will timeout and then backoff before attempting again. This issue is illustrated in Figure 2.5 and described in Section 2.4.

As shown in graph in Figure 3.8, the throughput in the DCAP network levels off before the the throughput in the non-DCAP. This smaller throughput value was expected since the gateway might be sending packets on an empty channel. The lower throughput in the DCAP network is caused because the T_L value is not large enough, and the gateway node is unable to send packets to node two. The gateway's transmissions fail to reach node two, because node two is not on its Home Channel (it is currently sending packets to node one). The gateway's transmissions will fail, timeout, and backoff. Since the backoff time is

exponential, it can cause a large drop in that node's throughput. Since the only traffic in the network is from the gateway to node one, this drop in throughput will consistently show in the network's throughput. This same issue is the cause of the difference in throughput in subtest 3.5.

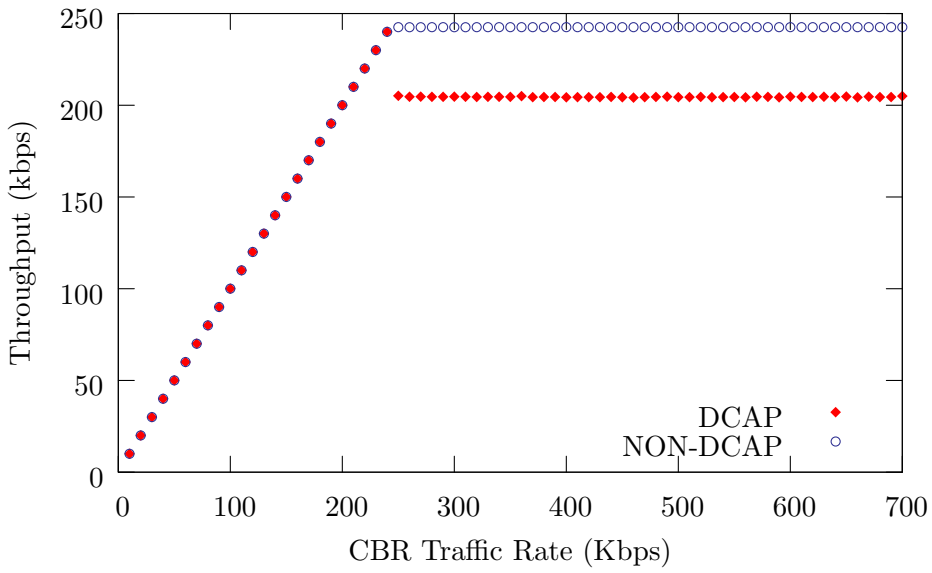

Figure 3.8: Subtest 3.2 Throughput

Subtest 3.3 contains the same traffic present in subtests 3.1 and 3.2, so it will have the same problem described in subtest 3.2. However, unlike in subtest 3.2, subtest 3.3 also has traffic going from node two to node one. This second flow should reduce the impact of the Absent Receiver Problem. As a result of the combined traffic, the throughput of DCAP was predicted to be close to non-DCAP's throughput. Node two sends traffic to both the gateway and node one. Because of this, it must change channels to send some packets; therefore, the throughput values are expected to fluctuate. Figure 3.9 shows the throughput graph for subtest 3.3. As shown in this graph, the throughput of the DCAP network is very close to the throughput of the non-DCAP network as expected. Also as expected, the throughput of the DCAP network fluctuates. These same traffic patterns and results are also present in the throughput graph for subtest 3.6.

The DCAP throughput is occasionally higher than the non-DCAP throughput in both subsection 3.3 and 3.6. This behavior was unexpected, but valid. The few cases in

which the DCAP throughput was higher than the non-DCAP's throughput are caused by nodes detecting that the channel is idle more often than in the non-DCAP network. When a node detects the channel is idle before it sends, it does not set its backoff timer because there is no contention on the channel. Since communication between nodes one and two is done on a different channel than communication between node two and the gateway, a node will be more likely to detect an idle channel in DCAP than in non-DCAP.

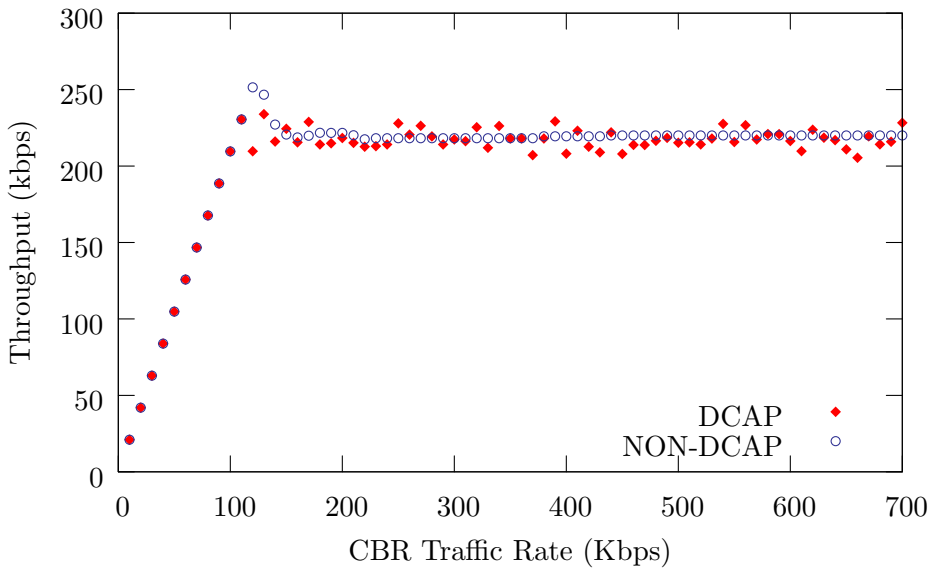

Figure 3.9: Subtest 3.3 Throughput

Subtest 3.4 contains traffic going from nodes one and two to the gateway node. In the DCAP network, node one must change to channel one before sending packets to node two. Since node one must change channels before sending packets to node two, the throughput in the DCAP network was predicted to be less than the throughput in the non-DCAP network. Although this same scenario exists in subtest 3.1, the difference in throughputs was predicted to be worse since there is the additional traffic between node two and the gateway node.

Because all traffic coming from node one must be forwarded by node two, node one's traffic will compete with node two's traffic. In both the DCAP and non-DCAP networks, more of node two's traffic will arrive at the gateway since node one's traffic must go through node two. In both networks, the throughput will increase at the same rate until

the throughput reaches around 4.5 Mbps. During this time all traffic from nodes one and two reach the gateway. After this point, the traffic in the DCAP network remains constant while the traffic in the non-DCAP network increase at a slower rate. During this time the traffic from node two competes with the traffic from node one. When node one wins this competition, it will reduce the throughput since it must be transmitted twice. When node two's traffic wins the competition, it will achieve a higher throughput since it is only transmitted once. This combination results in the throughput increasing at a slower rate. Once non-DCAP's throughput reaches 6.2Mbps, it remains constant.

The difference between the two graphs (see Figure 3.10) is the result of the traffic from node one competing for the channel. In the non-DCAP network, node one will detect the channel as busy when it attempts to send and will set its backoff timer. However, in the DCAP network, node one will change to channel one and assume the network is idle. If it doesn't hear any communication, it will send after the DIFS time. Because node one sets its backoff timer less in the DCAP network than in the non-DCAP network, it will successfully send more packets to node two in the DCAP network than it can in the non-DCAP network. Although this results in more successful transmissions by node one, it also results in more competition between node one's traffic and node two's traffic. The result of this competition is reduced throughput since all packets sent by node one must be transmitted twice. This competition causes the throughput in the DCAP network to remain constant while the throughput in the non-DCAP network increases at a slower rate.

3.4 Test Four: Three Channels

A network of six wireless nodes are placed in a multihop configuration around the gateway node (see 3.11). Each node is located 200 meters apart with three nodes in transmission range of the gateway and three nodes two hops away from the gateway node. All nodes' Home Channels were assigned on three different channels. Each of the three nodes within transmission range of the gateway node had a different Home Channel. All nodes two hops away from the gateway were assigned a separate Home Channel than their one-hop neighbors. Three subtests were performed (4.1, 4.2, and 4.3) containing different traffic patterns.

Since the network uses three channels, the throughput in the DCAP network

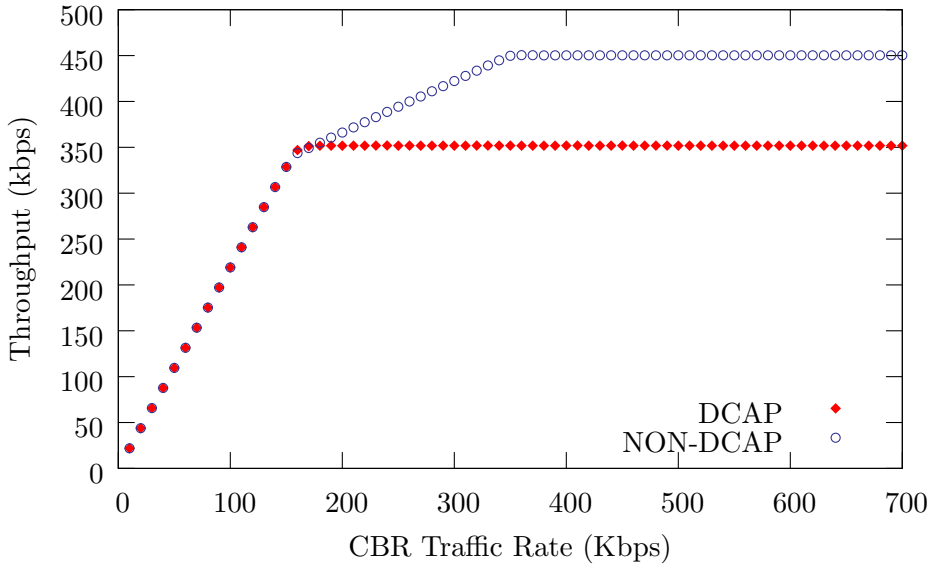

Figure 3.10: Subtest 3.4 Throughput

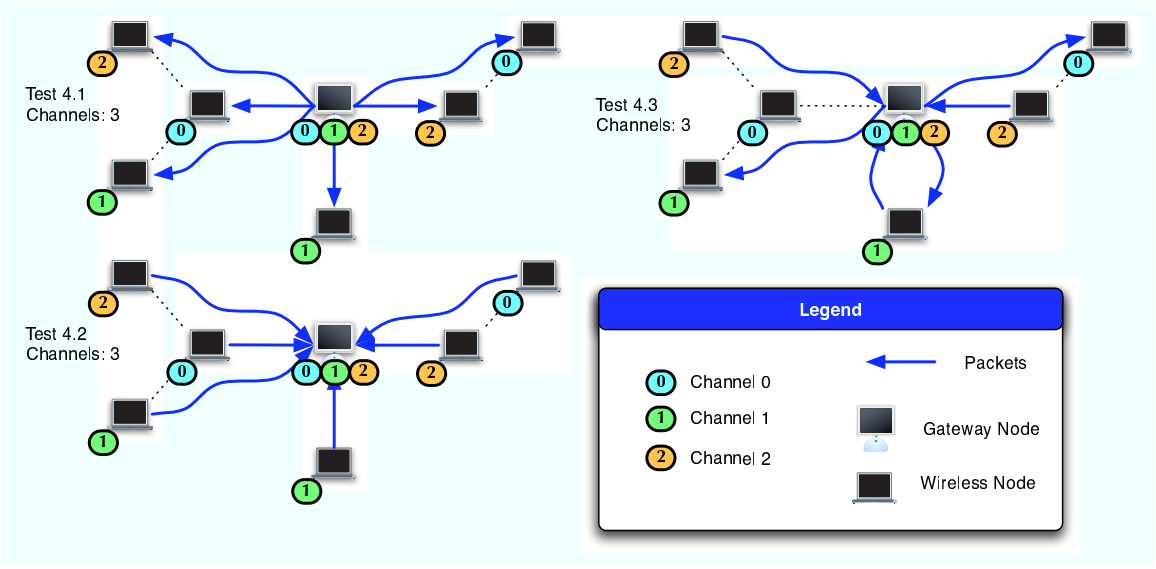

Figure 3.11: Test Four: Three Channels

was predicted to be close to three times the throughput in the non-DCAP network. The throughput in the DCAP network was expected to be slightly less than three times the non-DCAP throughput due changing channels and the other issues present in test three. As illustrated in the subtest 4.3 throughput graph (see Figure 3.12), the throughput of the DCAP network comes close to three times the throughput of the non-DCAP network. The rate the throughput increases slows several times in the graph due to multiple traffic flows competing for the channels similar to test three. The throughput also fluctuates slightly in subtest 4.3. This fluctuation is the result of changing channels similar to subtests 3.3 and 3.6. The results of all three subtests in test four were exactly what was predicted.

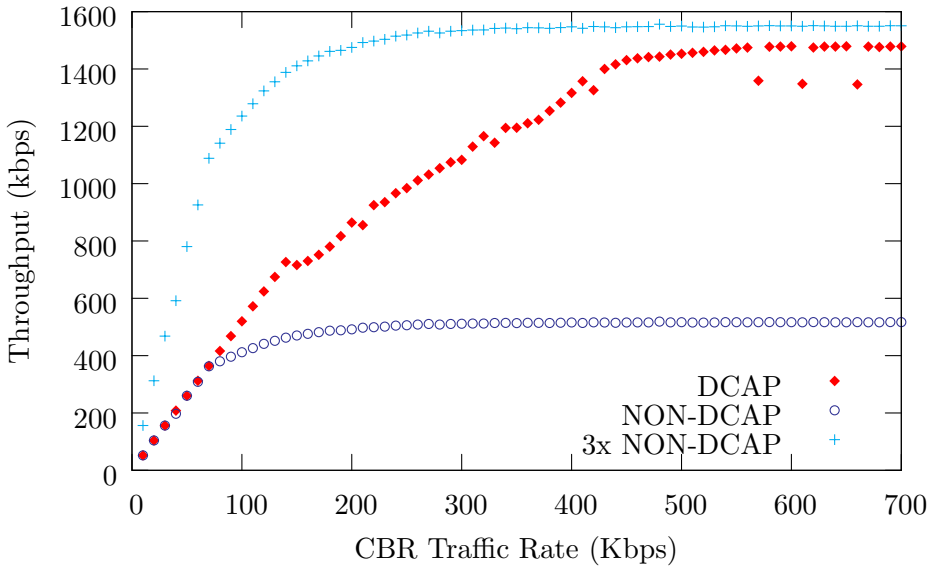

Figure 3.12: Subtest 4.3 Throughput

The results of all of the validation tests confirmed the predicted throughput, and the results validated the ns-2 implementation of DCAP. The validation tests also show that DCAP does achieve a higher throughput that a single channel network. In the next chapter, DCAP's performance will be evaluated in more realistic scenarios to determine how DCAP will perform with other traffic patterns and network topologies. The next chapter will also analyze the best values for all of DCAP's parameters.

Chapter 4

Performance Evaluation

The performance of DCAP was examined by implementing a wireless network in the ns-2 simulator. Different traffic patterns, network topologies, and DCAP parameters were analyzed to determine DCAP's performance compared to a single channel 802.11 wireless network (non-DCAP). For each scenario, the traffic throughput, Packet Delivery Ratio (PDR), and end-to-end delay of the network are examined. For the evaluation, a base case is used with the parameters shown in Table 4.1. For convenience, the base case results are marked with a hollow circle in each graph in this chapter. To determine the effect of each of these parameters on DCAP's performance, they are varied one at a time.

The network was created as a 5x5 grid of wireless nodes 200 meters apart with a base station in the center (see Figure 4.1). Each node's manual routing table was manually set to eliminate the transient behavior induced by the AODV's route setup. The route packets take from each wireless node to the gateway is shown by the arrows in Figure 4.1. The route from the gateway to the wireless nodes is the reverse of the arrows. The network has three available channels, and the gateway has a single interface on each channel. Traffic was 500 byte packets sent according to the Poisson distribution with an interarrival time of 0.15 seconds.

DCAP's parameters are shown in Table 4.1. The remainder of this chapter is divided into several sections, each analyzing the performance of DCAP while varying a single traffic pattern, network topology, or DCAP parameter.

Before the Poisson traffic begins, nodes send Home Channel Packets to each other

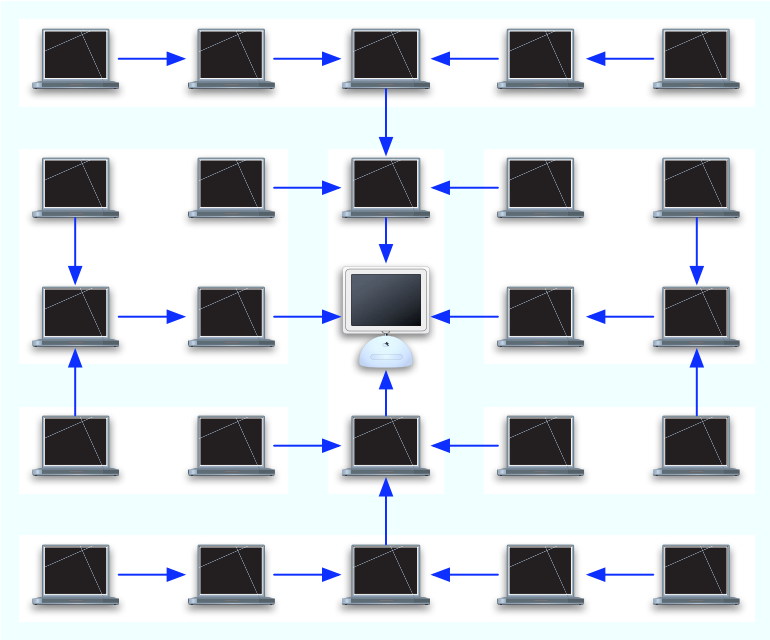

Figure 4.1: Performance Evaluation Network

Table 4.1: Performance Evaluation Base Scenario

DCAP Parameters	
Channel Request Timeout (T_Q)	0.1 seconds
Listen Time (T_L)	0.67 milliseconds
Interval Between Home Channel Selection (T_C)	100 seconds
Interval Between Neighbor Discovery (T_N)	100 seconds
Interval Between Channel Table Purge (T_P)	300 seconds
Network and Traffic Conditions	
Channel Load (Poisson Interarrival Time)	0.15 seconds
Number of Channels	3
Grid Size	5 x 5
Distance Between Nodes	200 meters
Random Node Placement	No Randomness

to determine their neighbors every 2 seconds ($T_N = 2$). After 10 seconds of neighbor discovery, the T_N value is changed to the base scenario value ($T_N = 100$). After the 10 second neighbor discovery phase, every wireless node send and receives a single packet with the gateway. These initial exchanges are performed to eliminate the transient behavior induced by resolving ARP requests. After this exchange, the Poisson traffic starts and lasts for 500 seconds. Only the data payloads of the Poisson traffic is counted in the throughput calculation. DCAP packets such as the Channel Request Packet, Channel Reply Packet, and Home Channel Packet are not counted in the throughput calculation.

4.1 Dependency on Channel Request Packet Timeout (T_Q)

In this section, the timeout value for Channel Request Packets T_Q is varied. Every time a node sends a Channel Request Packet, it sets a timer for T_Q seconds. After the timer expires, it will retransmit the Channel Request Packet (see Section 2.5). The results of the throughput and delay for different T_Q values are shown in Figure 4.2.

Since the majority of the neighbor discovery is done during the first 10 seconds of the simulation, the T_Q value was expected to have little effect on the throughput. The results of the graph verify this prediction. The throughput of the DCAP network remains fairly constant regardless of the T_Q value.

The delay was also expected to be fairly constant because few Channel Request Packets are sent after the first 10 seconds. The delay was expected to be higher for very small and very large values of T_Q. At small T_Q values, the unnecessary retransmissions of Channel Request Packets can cause a slight delay in sending packets. If the timer expires faster than the first reply can be received, then a second Channel Request Packet will be sent. Since neighbors will respond to both Channel Request Packets, this will increase the amount of traffic in the network and decrease the network's throughput. Also with large T_Q values, senders will wait too long before sending another Channel Request Packet. If the neighbors know the receiver's Home Channel or the Channel Request Packet was not received, then waiting additional time introduces delay.

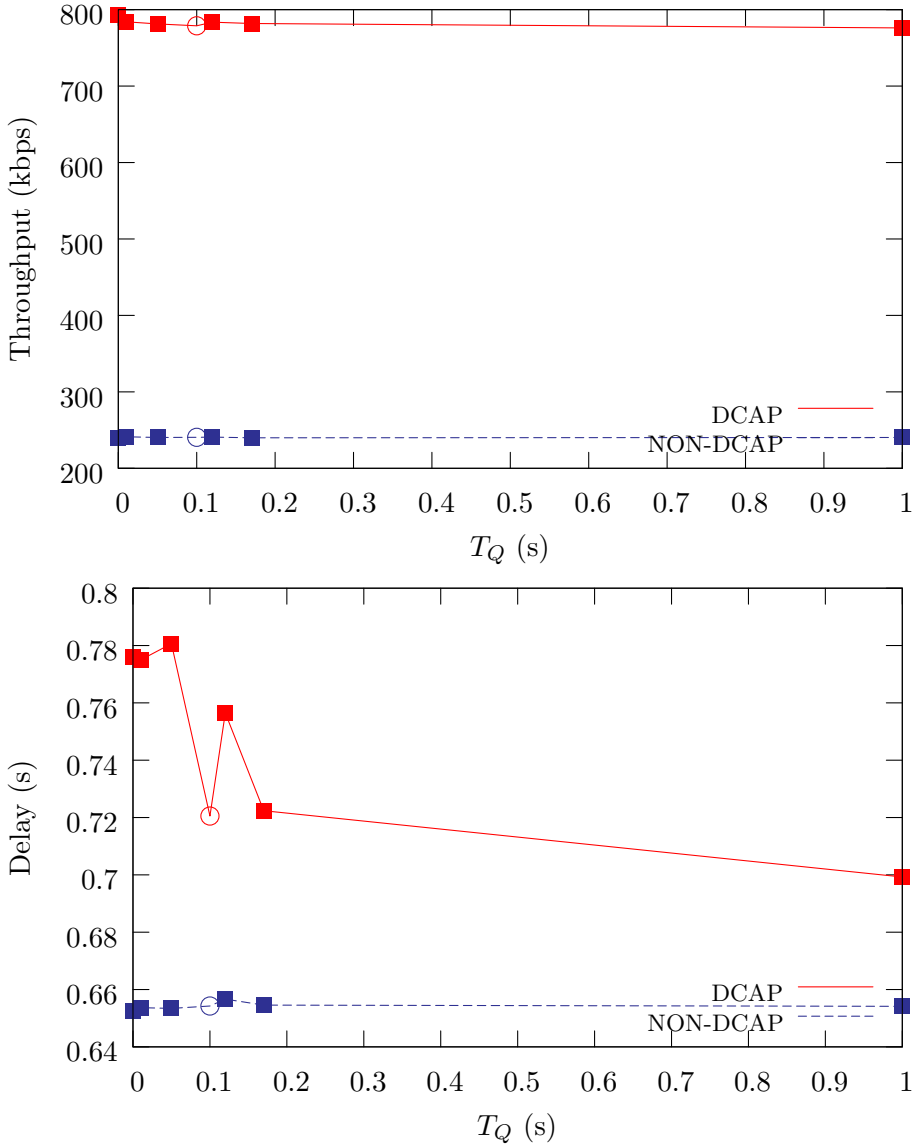

Figure 4.2: Dependency of Throughput and Delay as a Function of Channel Request Packet Timeout T_Q

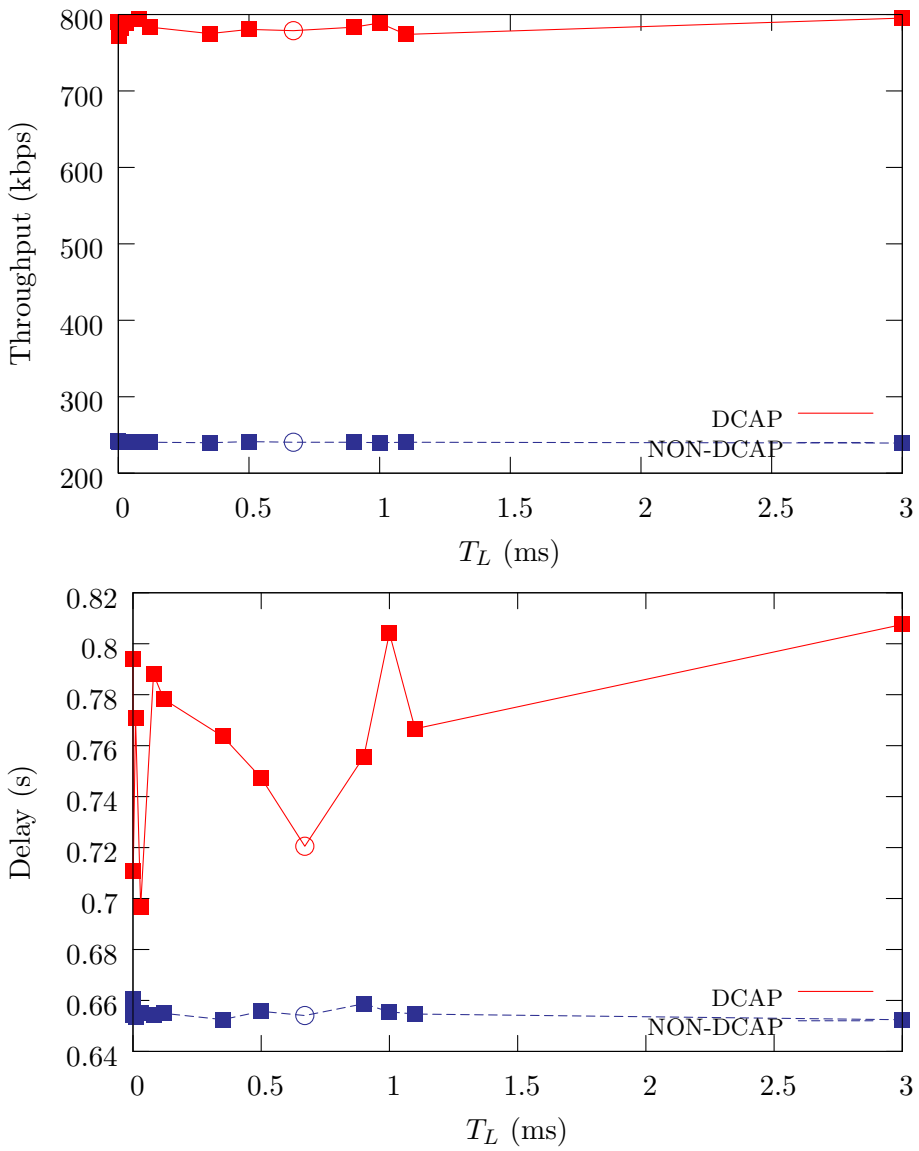

Figure 4.3: Dependency of Throughput and Delay as a Function of Listen Time T_L

4.2 Dependency on Listen Time (T_L)

In this section, the time a sender must remain on its Home Channel after sending a unicast packet on another channel is varied. After sending every unicast packet on a channel other than the node's Home Channel, a node must wait T_L milliseconds before it can leave its Home Channel (see Section 2.2). The results of the throughput and delay for different T_L values are shown in Figure 4.3.

The throughput was expected to be less for very small values of T_L. Since a node will not remain on its Home Channel long enough to receive packets, small T_L values will decrease the network's throughput. The throughput was also expected to be less for very large values of T_L since a node will be waiting unnecessarily for incoming packets. As expected, very small and very large values of T_L resulted in decreased throughput. From the performance results, the optimal value for T_L is 0.67 milliseconds. This value provides the best balance between waiting on the Home Channel for incoming packets and changing packets to send to other nodes.

The delay in the DCAP network was expected to increase as the value of T_L increases. At very small T_L values, the delay was also expected to be high. The sender will not be able to send packets to the receiver if the receiver is constantly leaving its Home Channel to send other packets. This results in an increased delay caused by waiting to send packets to the receiver. For very large T_L values, nodes will remain on their Home Channels longer and not be able to send packets. This increased time on their Home Channel will cause additional delay.

4.3 Dependency on Home Channel Assignment Interval (T_C)

In this section, the interval for determining a new Home Channel is varied. Every T_C seconds, a node will re-evaluate the channel usage on all channels and select a new Home Channel (see Section 2.9.3). The results of the throughput and delay for different T_C values are shown in Figure 4.4.

The throughput was expected to be less for both large and small values of T_C. At small values, nodes are selecting new Home Channels frequently which will increase the amount of Home Channel Packets being sent. This increased traffic will cause a decrease in the network's throughput. Larger values of T_C will have an even greater impact on the

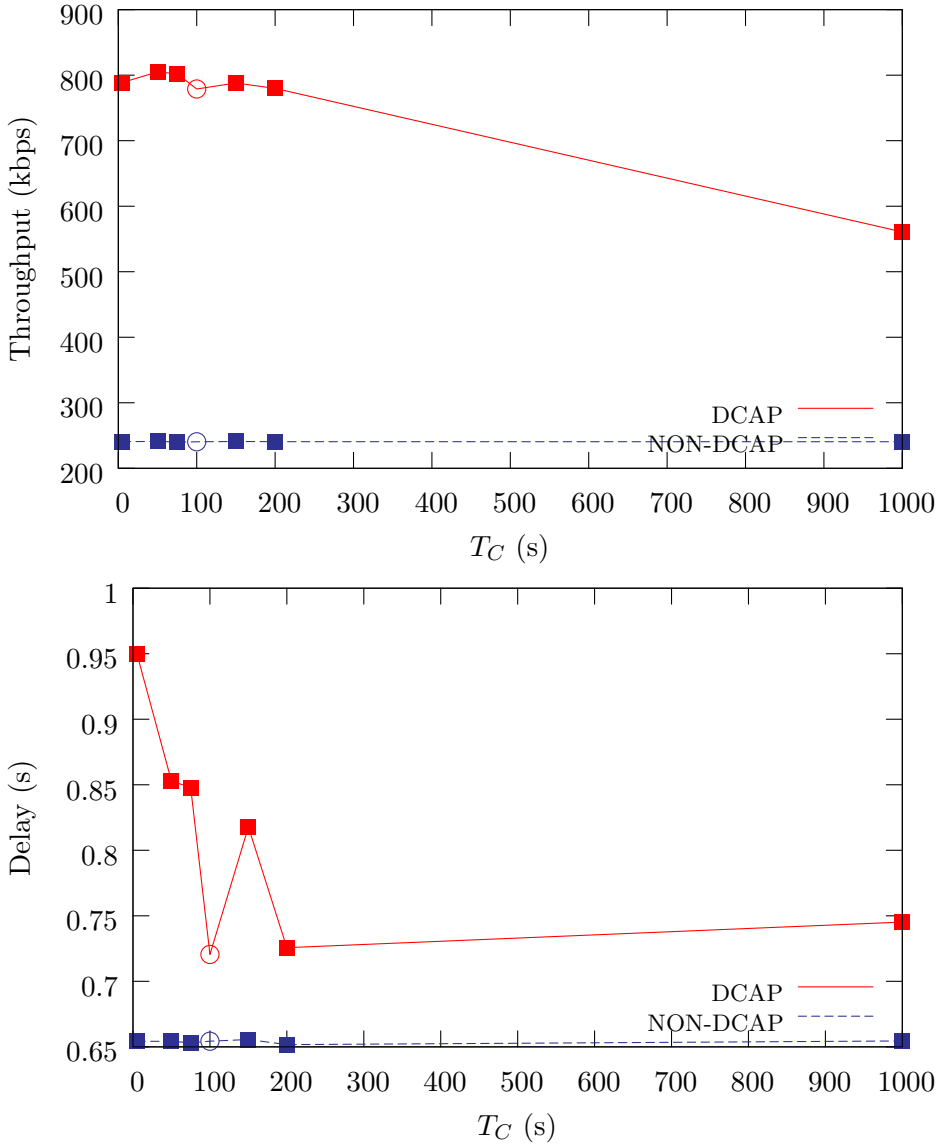

Figure 4.4: Dependency of Throughput and Delay as a Function of Home Channel Selection Interval T_C

throughput than the small values. As the T_C value increases, nodes will more often stay on busy channels since they select new Home Channels less often. This will cause some channels to be under utilized while other channels become saturated. This inefficient use of the channels will cause the throughput to decrease.

The simulation results verified these expectations. The throughput is slightly less for values under $T_C = 100$. As the T_C values increase, the throughput dramatically decreases. From the simulation results, a value of $T_C = 100$ is the optimal Home Channel Selection Interval.

The delay of the network is expected to be larger for both very small and very large values of T_C. At smaller values of T_C, there will be more Home Channel selections, and more Home Channel Packets will be sent which will introduce a greater delay in the traffic. Less impact is expected for larger values of T_C. Since there is more traffic on some channels, there will be a slightly longer delay on those channels. However, since the other channels contain less traffic, delays will be decreased on those channels. The combination of the larger and smaller delays was expected to result in a smaller increase in delay at large values of T_C. As the simulation results show, the delay is much larger for very small values of T_C, but only slightly larger for very large values of T_C. These findings were as expected.

4.4 Dependency on Neighbor Discovery Interval (T_N)

In this section, the interval for proactively discovering neighbors is varied. Every T_N seconds, a node will broadcast a Home Channel Packet on all channels. The Home Channel Packet contains the Home Channel and channel usage of the sender and all the sender's one-hop neighbors (see Section 2.7). The results of the throughput and delay for different T_N values are shown in Figure 4.5.

Since the majority of the neighbor discovery is done during the first 10 seconds of the simulation, the T_N values were expected to have little effect on the throughput. At smaller values of T_N, the throughput was expected to be less since there are more Home Channel Packets being sent. However, the simulation results show a different trend. The results show an increase in throughput for smaller T_N values. Upon investigating, the slight increase in throughput is caused by the increased awareness of neighbors Home Channels. By sending broadcast Home Channel Packets more often to update neighbors of the node's

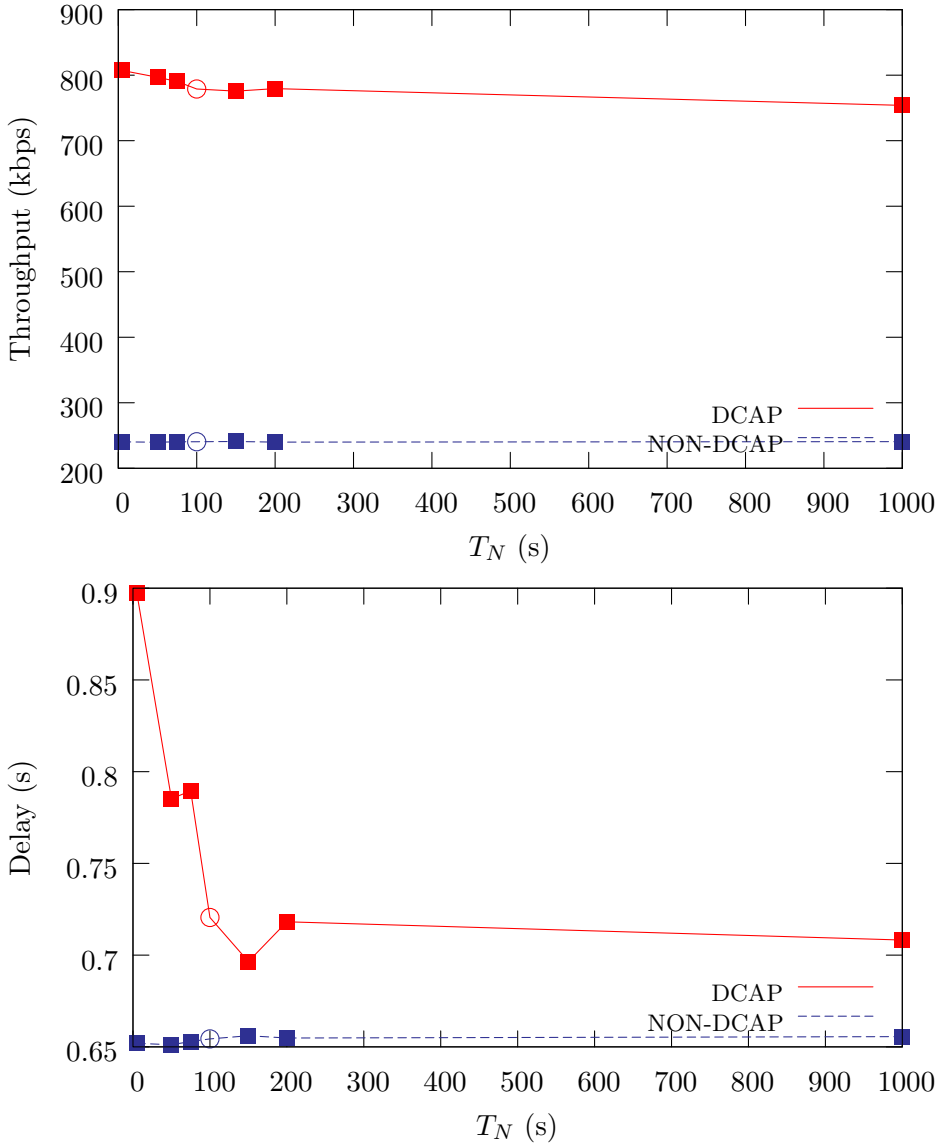

Figure 4.5: Dependency of Throughput and Delay as a Function of Neighbor Discovery Interval T_N

current Home Channel, Home Channels of neighbor nodes, and channel load, other nodes have more accurate information in their Channel Table. This results in fewer Channel Request Packets and better Home Channel selection decisions.

The delay was expected to be significantly larger at lower values of T_N. The simulation results verify this assumption. The results show the delay is much larger at smaller T_N values, but it also shows an increase in delay for larger T_N values. This increased delay for larger T_N values is caused by the increased traffic caused by less accurate Channel Table information. At larger T_N values, the Channel Table will only be updated when neighbors select a new Home Channel. These infrequent updates will cause poorer Home Channel selections resulting in increased delay.

4.5 Dependency on Channel Table Purge Interval (T_P)

In this section, the interval for purging old Channel Table entries is varied. Every T_P seconds, a node will remove all entries older than T_P seconds. Purging the Channel Table in this manner ensures that all information is no older than T_P seconds (see Section 2.6). The results of the throughput and delay for different T_P values are shown in Figure 4.6.

The throughput was expected to be fairly constant for larger values of T_P. At very large values, the throughput was expected to decrease due to inaccurate information in the Channel Table. For small values, however, the throughput was expected to decrease significantly. For values around zero seconds, the throughput was expected to be close to zero. Less frequent purges will have negligible impact on the throughput because the Channel Table information is updated more frequently than the purges. More frequent purges, however, will have a significant impact on the throughput. Once the Channel Table data is removed at smaller intervals than it is updated, the node will have to send more Channel Request Packets to determine the Home Channels of neighbors. Nodes will also have less accurate channel load information to make efficient Home Channel selection decisions. As the T_P value approaches zero, the nodes were expected to send at least one Channel Request Packet for every outgoing packet. These extra transmissions were expected to decrease the throughput to zero.

The simulation results verify the expectations for smaller values of T_P. As T_P

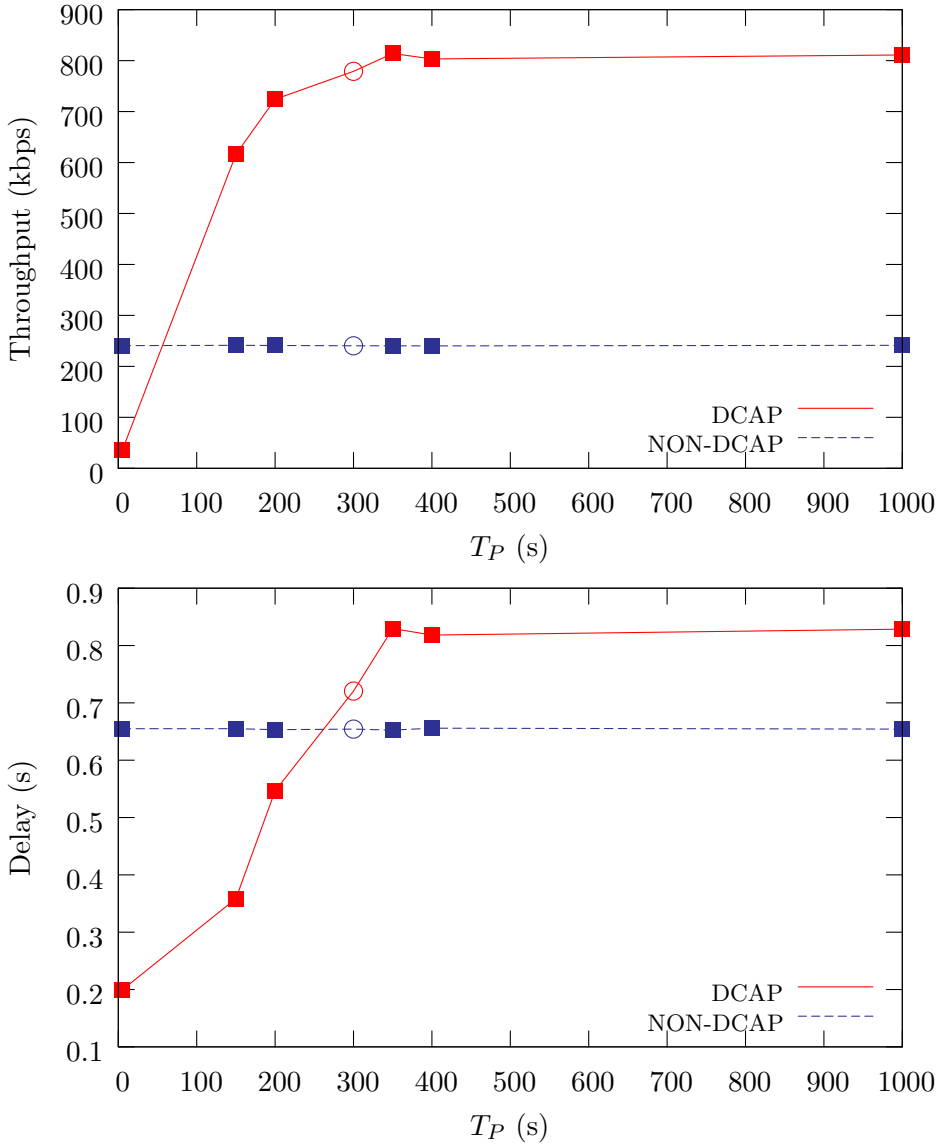

Figure 4.6: Dependency of Throughput and Delay as a Function of Channel Table Purge Interval T_P

approaches zero, the throughput also decreases close to zero. However, at large T_P values, the results show the throughput remains constant. It was predicted that the throughput would decrease at larger T_P values due to inaccurate information in the Channel Table. Upon investigation, the Channel Table entries never became inaccurate since all nodes selected a new Home Channel more frequently. When nodes select a new Home Channel, they send a Home Channel packet with current information. Nodes also send Home Channel packets every T_N seconds. These Home Channel Packets update the Channel Table, so its information never becomes invalid.

The delay was expected to be larger for very large values of T_P since the Channel Table was expected to contain inaccurate information. This inaccurate information would result in more Channel Request Packets and inefficient utilization of the available channels. The inefficient utilization would cause more contention on some channels resulting in more delay. For smaller values of T_P, the delay was expected to decrease. Although nodes will send more Channel Request Packets and make less efficient usage of available channels (both increasing the delay), there will be less transmissions in the network reducing the delay. Since nodes have less information in their Channel Table, they will have to send Channel Request Packets to obtain the information. Everytime a node broadcasts a Channel Request Packet, they will have to return on their home channel and wait for T_L seconds. If they do not receive a reply, they will have to wait an additional T_Q seconds before trying again. These increase the delay of sending packets, but also decrease the contention on the channels. The reduction of contention will cause a significantly less end-to-end delay.

4.6 Dependency on Channel Load (Poisson Interarrival Time)

In this section, the offered traffic load is varied. The traffic load is the number of packets per second transmitted by each node. The results of the throughput and delay for different traffic loads are shown in Figure 4.7.

As expected, the throughput in both the DCAP and non-DCAP networks increase as the traffic load increases. At lower traffic loads, DCAP performs similar to non-DCAP. It was expected that DCAP's throughput would be less for small traffic loads because of the overheads caused by broadcasting packets and remaining on the Home Channel for T_L seconds. However, as the results indicate, there is not a substantial difference in the

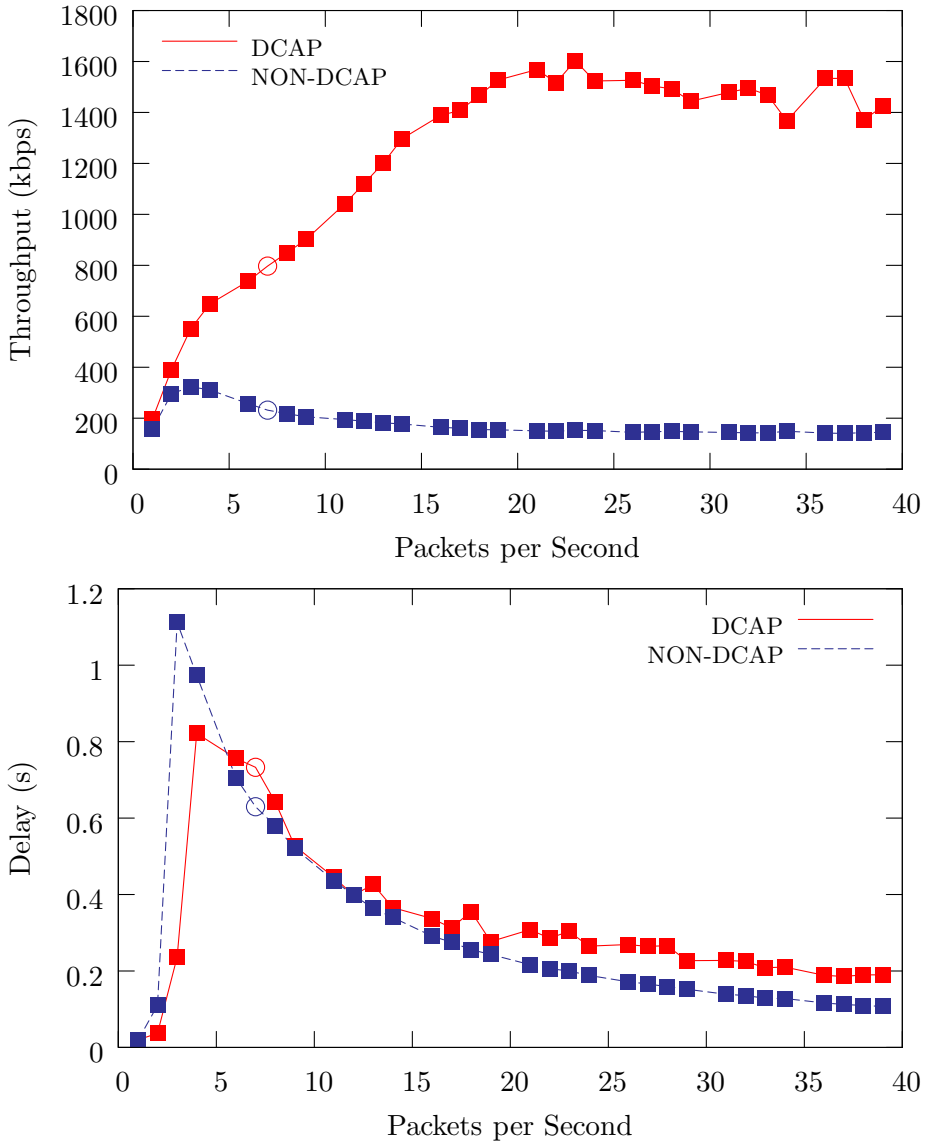

Figure 4.7: Dependency of Throughput and Delay as a Function of Offered Load

throughputs of the DCAP and non-DCAP networks at small traffic loads.

The throughput was expected to increase as the offered load increased. DCAP's throughput was expected to increase at the same rate as non-DCAP initially. However, DCAP was expected to continue to increase at this rate as non-DCAP's increase slowed to to increased contention on the channel. As expected, DCAP performs better than non-DCAP at larger offered loads since it is able to utilize the multiple channels to simultaneously carry out multiple communicates within transmission range.

The delay was expected to decrease as the traffic load increases. At very small traffic loads, the delay was expected to be small, but the delay was expected to increase as more packets contend for the channel. The delay in the DCAP network is larger than the non-DCAP network at small offered loads due to the overheads of channel switching and listening on the Home Channel for T_L seconds. DCAP's delay decreases at high traffic loads as the packets being forwarded across multiple nodes are dropped due to increased channel contention. Since the traffic from the nodes further from the gateway are dropped, only the packets from nodes close to the gateway are successfully received. These packets from the closer nodes will have less delay than the packets from further nodes, so the average delay will be less.

To further investigate the throughput advantages DCAP provides, a test was performed with Internet modeled traffic. The base case scenario was modified to use a ns-2 module provided by AT&T Research [4]. This module provides Hyper Text Transfer Protocol (HTTP) traffic over Transport Control Protocol (TCP) connections modeled to resemble realistic Internet traffic. The module simulates multiple user agents accessing and downloading content from web servers (see the Figure). Each user agent is modeled as a separate session. Each session includes multiple TCP connections, one for each web page requested. Each connection contains multiple HTTP requests from the user agent and their corresponding responses from the web server.

The simulation was performed by varying the number of sessions in the network while keeping all other parameters the same. The network was identical to the base case scenario except it only used two channels. When a session is created, a wireless node is selected at random from the network to act as the client (user agent). The gateway node always acts as the web server. The total throughput in the network is shown in Figure 4.8 as a function of the number of sessions.

As expected, the throughput increases with the number of connections. DCAP

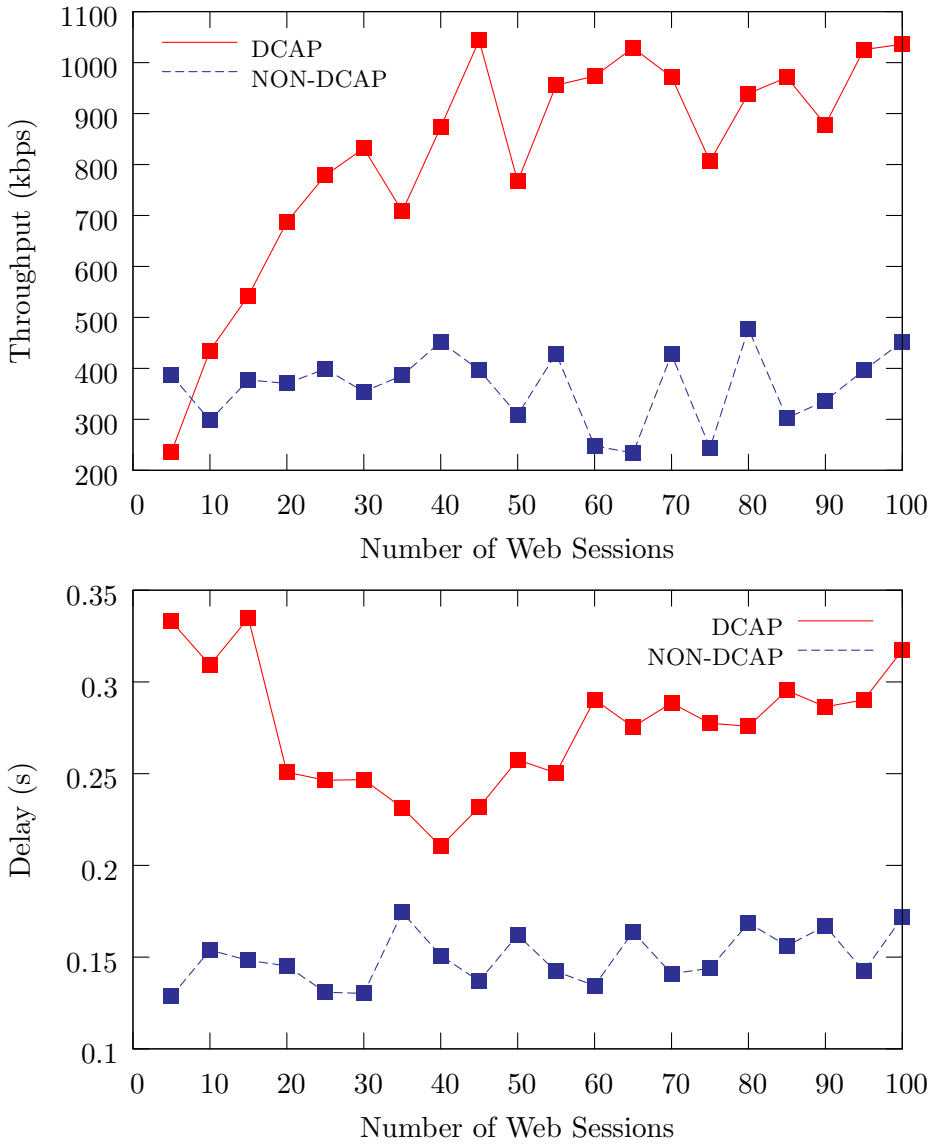

Figure 4.8: Dependency of Throughput and Delay as a Function of HTTP traffic on Offered Load

achieves higher throughputs at higher loads than non-DCAP. Since TCP retransmits packets when dropped, the throughput is expected to vary slightly. However, the simulation results show a significant variance in the throughput. This variance is caused by TCPs retransmission strategy and timeout that results in unstable behavior. The results suggest the need for a new transport protocol for multihop wireless networks. Since the size and number of the HTTP requests and responses can vary, the resulting throughput will vary. Despite this variance, the results show a clear trend demonstrating that the DCAP network achieves a higher throughput than the non-DCAP network.

4.7 Dependency on Number of Channels

In this section, the number of available channels in the network is varied. Since the non-DCAP network can only utilize a single channel, this variable has no effect on it. The results of the throughput and delay for different numbers of channels are shown in Figure 4.9.

The throughput in the DCAP network was expected to increase as the number of channels increased. DCAP will be able to spread the network traffic over more channels reducing the load on each channel. However, with very large numbers of channels, the throughput was expected to be less than with smaller numbers of channels due to the increased overhead of broadcasting packets. In a network of 20 channels, a node will have to send a broadcast packet on each of the 20 channels. This overhead increases as the number of available channels increases. With a very large number of channels, the overhead will have more of an impact than load balancing the traffic across the channels. Even at higher traffic rates, this overhead was expected to be greater than the benefit of load balancing.

The simulation results verify the expected results. As the number of channels decrease, the throughput also decreases. However, as the number of channels increase, there is a reduction in the rate the throughput increases. The results show a network with 20 channels actually has less throughput for all traffic loads than a network with four channels. This decrease in throughput indicates the overhead of broadcasting packets is greater than the benefit of load balancing the traffic on all 20 channels.

The delay was expected to decrease as the number of channels increases. Since the

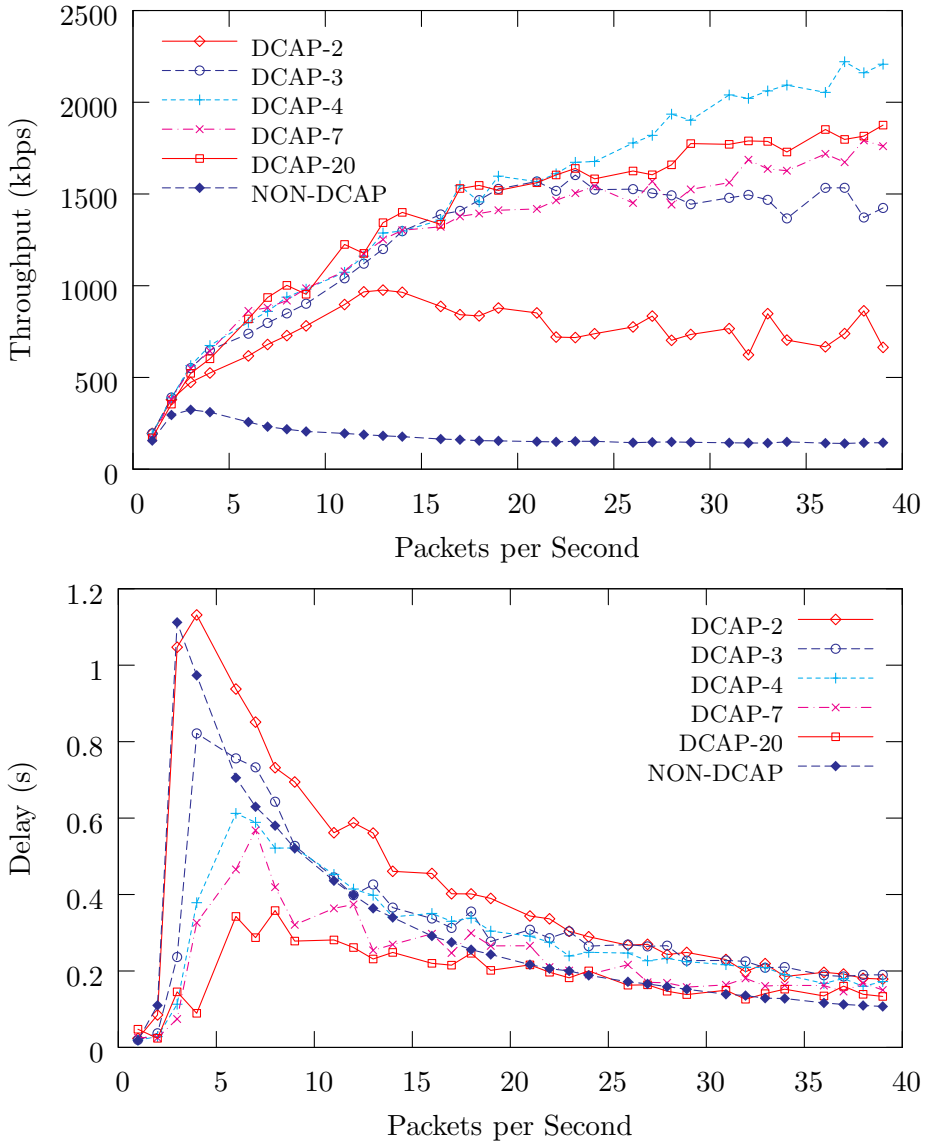

Figure 4.9: Dependency of Throughput and Delay as a Function of Number of Channels

same amount of traffic will be spread across more channels, there will be less contention on each channel. This reduction of contention will result in smaller delays. The results verify this prediction. At a greater number of channels, the delay decreases to be substantially less than the delay of non-DCAP.

4.8 Dependency on Number of Nodes

In this section, the number of nodes in the network is varied. The network is constructed as grid with the same number of nodes in each row and column. By varying the number of nodes in a row, the total number of nodes in the network are varied while maintaining the grid network topology. Each node in the network is both sending and receiving information from the gateway. Since the number of nodes increases, so does the number of nodes sending packets. To maintain the same amount of network traffic across the different test cases, the Poisson interarrival time is modified to adjust for smaller and larger numbers of nodes. The adjustment, shown in Equation 4.1, modifies the base scenario's interarrival time so the same amount of traffic will be sent in the network. In the equation, τ_b is the base scenario's interarrival time, r_b is the number of nodes in a row in the base scenario, and r is the number of nodes in a row in the test case. The results of the throughput and delay for different number of nodes in the network are shown in Figure 4.10.

$$\tau = \tau_b * \frac{r^2}{r_b^2} = 0.15 * \frac{r^2}{5^2} \tag{4.1}$$

The throughput of both networks was expected to decrease as the number of nodes in the network increases. Since all nodes remain 200 meters apart, the additional nodes in the network cause packets to travel across more nodes before they reach their destination. This traffic must be forwarded which causes more packet transmissions resulting in more channel contention. As the contention for the channels increases, the throughput will decrease. The throughput of the DCAP network was expected to be higher than the throughput of the non-DCAP network for all network sizes. Since DCAP spreads the traffic across multiple channels, there will be less contention on each channel resulting in a higher throughput.

The simulation results validate these predictions. The results show the throughput

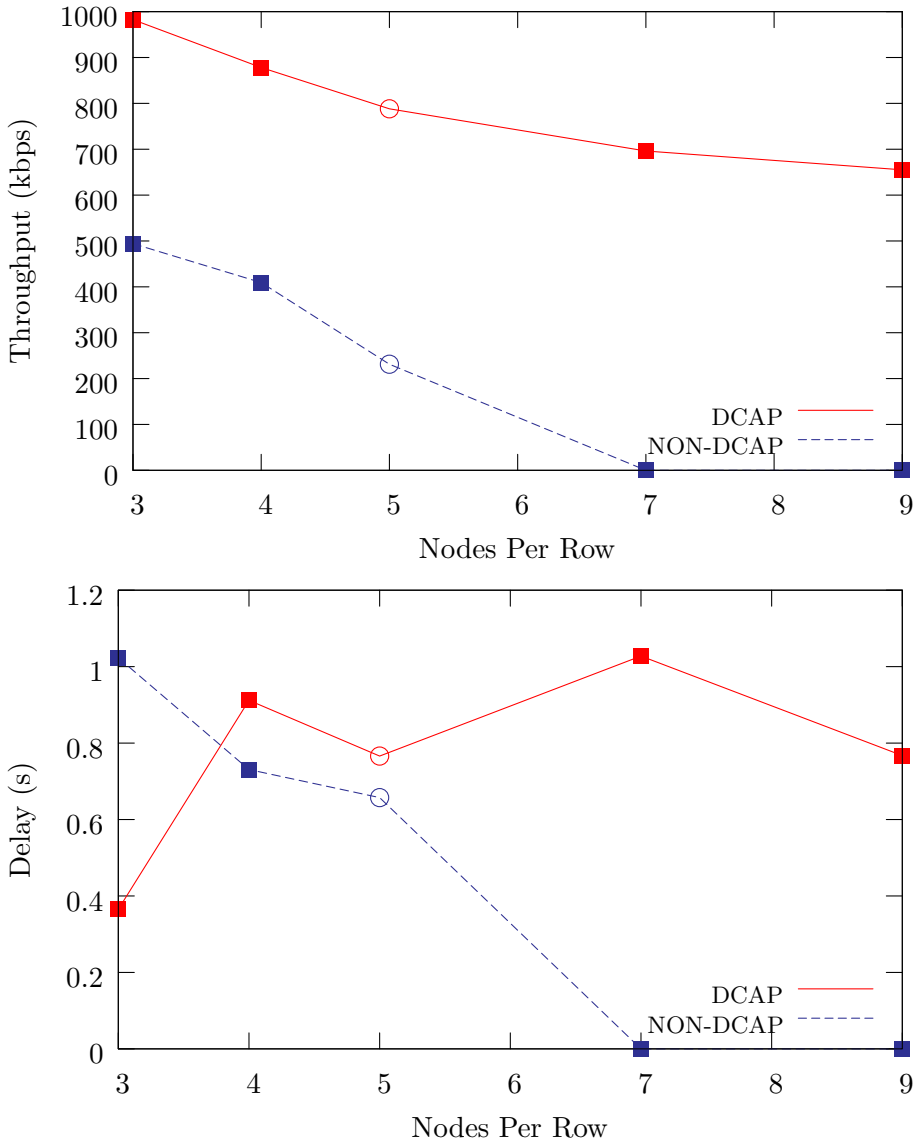

Figure 4.10: Dependency of Throughput and Delay as a Function of the Number of Nodes in the Network

of both networks decrease as the number of nodes increase. The results also show the throughput in the DCAP network is higher than non-DCAP's throughput for all network sizes. In a 7x7 grid network, non-DCAP has a zero throughput. This is caused because the channel contention is preventing packets from reaching their final destination. This low throughput was predicted because in a 5x5 grid network, only 18% of all packets reached their destination. By nearly doubling the number of nodes, the throughput was expected to be close to zero.

As expected, the delay in the DCAP network was less for very small and very large networks. In very small networks, packets are forwarded less often before they reach their destination. The fewer number of nodes must forward a packet before it reaches its destination, the less the forwarding will introduce delay in the network. For very large networks, the delay was also expected to be only slightly less. Since the amount of traffic in the network remains the same, in larger networks, the traffic rate of each node is less. This decreased traffic rate was expected to reduce the local channel contention resulting in smaller delays. However, since packets must be forwarded by more nodes in the larger networks, this reduction in delay was expected to be small.

Since the non-DCAP network only uses a single channel, its delay was expected to to be higher for smaller networks. The simulation results validate this prediction. Since the amount of traffic in the network is the same, smaller networks will have more channel contention. With more contention for the channel, nodes will have to wait longer before being able to transmit packets resulting in larger delays. Since the DCAP network uses three channels, the delay in the non-DCAP network was expected to be close to three times the delay in the DCAP network. The simulation results show non-DCAP's delay greater than DCAP's by a factor of 2.83. As the number of nodes increase, the throughput in the non-DCAP network was expected to decrease. The simulation results show the delay in the non-DCAP network decreases as the network size increases. Since fewer of the packets in non-DCAP's network reach the destination, the delay of the successfully received packets is expected to be less. In very large networks the delay is zero since no packets are received.

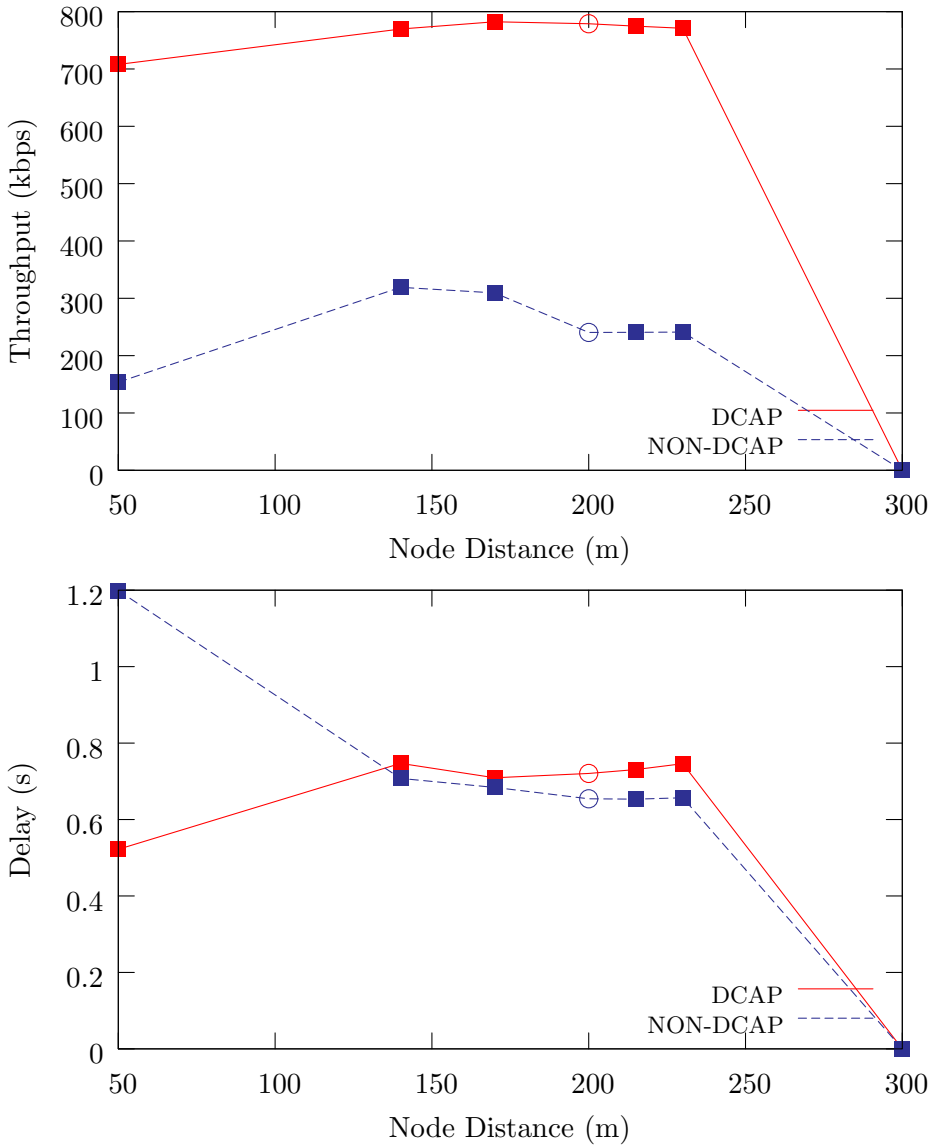

Figure 4.11: Dependency of Throughput and Delay as a Function of the Distance Between Nodes

4.9 Dependency on Distance Between Nodes

In this section, the distance between nodes is varied. At a distance of 10 meters between nodes, all nodes are with each other's transmission range. As the distance between the nodes increases, fewer nodes are within transmission range. At 130 meters, only the eight nodes adjacent to a node are with transmission range. Once the distance reaches 200 meters, only the four nodes in each cardinal direction are within transmission range. The results of the throughput and delay for distances between nodes are shown in Figure 4.11.

The throughput for both the DCAP and non-DCAP networks was expected to be the greatest with a distance of 130 to 200 meters between nodes. With very small distances between nodes, there is increased contention in the network as all nodes contend with each other. At distances beyond 200 meters, the network becomes disconnected since nodes are outside of each other's radio range. The simulation results confirm the expected results. The throughput is greatest around 130 to 200 meters in both the DCAP and non-DCAP networks. The non-DCAP network has a higher throughput at 130 meters and then the throughput decreases as the distances reaches 200 meters. The DCAP network does not show a noticeable difference in throughput between 130 and 200 meters.

The delay in the networks was expected to be larger for smaller distances. Since nodes are closer together, more nodes will be within transmission range and the contention for the channel will be higher. As the distance between nodes increase, there will be less contention for the channel, so there will be less delay. The results show this prediction was valid for the non-DCAP network, but not accurate for the DCAP network. The simulation shows the delay for shorter distances between nodes in the DCAP network is less than for distances around 130 to 200 meters. This smaller delay is caused by the reduction in contention due to using multiple channels. Since the DCAP network spreads the traffic across three channels, each channel's contention will be less than in the non-DCAP network. The less contention will result in less delay. As the distance in the DACP network increases, the delay also increases. This increase in delay is caused by the packets having to be transmitted over more than a single hop.

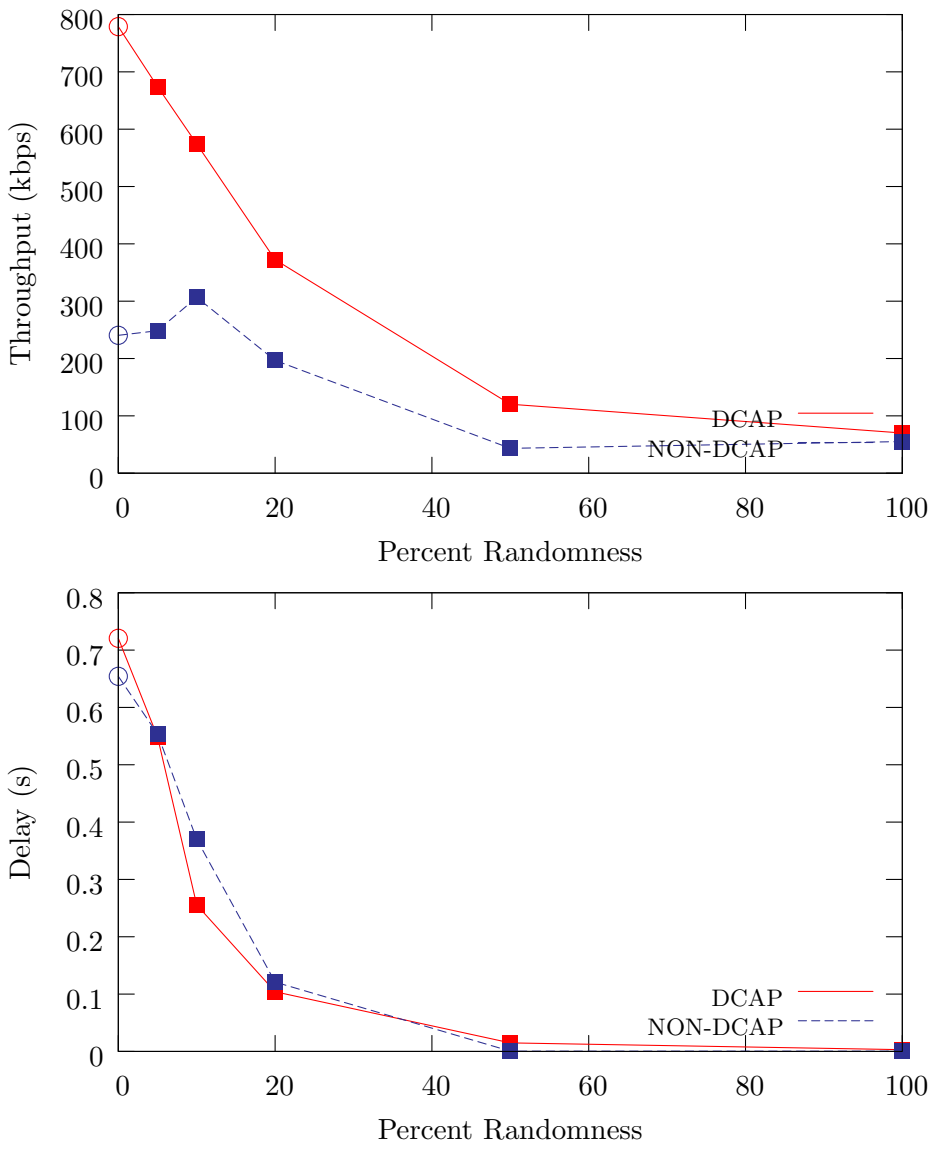

Figure 4.12: Dependency of Throughput and Delay as a Function of Node Positions

4.10 Dependency on Node Positions

In this section, the randomness in node placement is varied. At a value of 0%, all nodes are placed in specific locations to make a regular grid. As the randomness in the node positions increase, the network topology becomes less regular. At a value of 100%, any node has an equal probability of being placed anywhere in the network. In the simulation each node is placed in a specific x and y position in the network. The random node placement percentage is combined with the height and width of the network to determine the possible range of x and y positions a node can occupy.

Equation 4.3 shows the calculation for determining the x and y positions (x_{pos} and y_{pos}) of a node, where x_{grid} and y_{grid} are the node's original structured grid position, r is the random placement percentage, and x_{max} and y_{max} are the width and height of the network. The equation allows a node to be located in an area half the percentage of the network. This area is centered around the node's original structured grid position. For example, with a value of 50%, a node can be placed in an area 25% the size of the network.

$$x_{pos} = x_{grid} \pm (r * x_{max}) \tag{4.2}$$

$$y_{pos} = y_{grid} \pm (r * y_{max)} \tag{4.3}$$

The results of the randomness of node positions are shown in Figure 4.12.

The throughput was expected to decrease as the randomness in node positions increased. As nodes become closer together the contention for the channel will increase causing less throughput. Also, for large random placements, portions of the network could become disconnected because they are outside transmission range of all other nodes. The simulation results verify the expected results. The throughput of both the DCAP and non-DCAP networks decreases as the randomness in node position increase. DCAP seems to be more sensitive to random node positions than non-DCAP, especially at smaller values.

The delay in the network was expected to decrease since the throughput also decreased. More randomness in the node's positions will cause an increasingly likely disconnected network. This will decrease the contention on the channels and thus decrease the delay. Since delay is only computed on the successfully sent traffic, a disconnected network will have a smaller delay because of the reduction in traffic being sent.

Chapter 5

Conclusion and Future Work

In this paper, DCAP, a multi-channel protocol, is introduced. DCAP is a link layer protocol allowing wireless nodes with a single 802.11 interface to use multiple channels. Without modification of the 802.11 MAC protocol, DCAP provides the capabilities for nodes to send and receive packets using a single interface. DCAP is used in conjunction with the 802.11 MAC protocol to allow wireless nodes to send packets by changing to the receiver's wireless channel Unlike DCAP, other protocols supporting the use of multiple channels require multiple interfaces or a custom MAC protocol. DCAP allows easier and cheaper deployment by using only a single, inexpensive 802.11 interface.

DCAP allows wireless nodes to send and receive packets on multiple channels through the use of Home Channels. Each wireless node selects one channel as its Home Channel, and receives all packets on this channel. By each node selecting a Home Channel to receive incoming packets, nodes are able to determine on which channel to send outgoing packets. Since nodes only receive packets on their Home Channel, other nodes must change to the receiver's Home Channel to send packets. When nodes send broadcast packets, they must transmit the packet on every channel. Sending broadcast packets is the primary source of overhead in DCAP.

After sending a packet, a wireless node must remain on its Home Channel to listen for incoming packets before changing its channel again to send other packets. Remaining on its Home Channel for a small amount of time after every transmission ensures that a node will listen on its Home Channel even if it currently has packets to send on other channels.

This prevents the Absent Receiver Problem, a problem created when the sender repeatedly attempts to send packets to a receiver who is not on its Home Channel.

DCAP uses a channel assignment algorithm modified from the Hyacinth network's algorithm. DCAP's channel assignment algorithm provides each node a distributed method to assign the least used channel as its Home Channel. The channel assignment algorithm determines the current load of each channel by calculating the cumulative size of packets receives by each node. DCAP uses this channel load information to periodically select a new, less used Home Channel.

DCAP was implemented in the ns-2 simulator to determine DCAP's performance compared to a single channel 802.11 network (non-DCAP). Several tests with CBR traffic were performed to verify the implementation of DCAP and identify overheads DCAP introduces. The results of these tests show that DCAP achieves a much higher throughput than non-DCAP in single-hop and multihop networks.

DCAP's performance was compared against non-DCAP's performance in ns-2 simulations using Poisson traffic. These performance tests examined the effects of different DCAP parameter values, network topologies, and traffic rates. The performance results showed DCAP performed significantly better at higher traffic rates and larger networks than non-DCAP. The simulation results also show DCAP's performance increases as the number of available channels increase.

Simulations prove that a three channel network with 25 wireless nodes using DCAP has four times the throughput than the same network using a single channel without DCAP. Simulations also show DCAP's performance is greater with more channels.

Based on the simulation results, the DCAP protocol has been demonstrated to be a viable and beneficial option for increasing the throughput in current 802.11 wireless networks. By adding the DCAP protocol to an existing 802.11 network, it can use multiple channels to achieve higher throughput without requiring modifications any of the 802.11 hardware. Based on simulation results, adding DCAP to a 802.11b wireless network with three non-overlapping channels can result in four times the throughput than without DCAP.

In this paper, the benefits of the DCAP protocol were examined using ns-2 simulations. The DCAP protocol needs to be implemented in commercially available wireless devices and its performance evaluated against a single channel 802.11 network. Further research needs to be performed on DCAP in real network situations. DCAP needs to be implemented and its performance evaluated in a wireless testbed. Different network traf-

fic representative of WMNs needs to be performed in the testbed to determine DCAP's performance.

In addition to comparing DCAP's performance against a single channel 802.11 network in a wireless testbed, DCAP should be compared against other multi-channel protocols including the Hyacinth network [12], [10], [11]. The Hyacinth network uses multiple network interfaces to spread traffic over multiple channels. By comparing DCAP's performance against the Hyacinth network, the advantages and disadvantages of using multiple network cards can be analyzed.

The major overhead introduced by the DCAP protocol is caused by sending broadcast packets. Since most wireless ad hoc routing protocols rely heavily on broadcast communication, other routing protocols need to be evaluated or developed that would work better with DCAP. Using DCAP in conjunction with a routing protocol that minimizes the amount of broadcast packets will allow DCAP to reduce the broadcast overhead and achieve a higher throughput.

DCAP's performance on mobile nodes should also be examined. Since mobility can constantly change the topology of the network, it can affect DCAP's performance. Highly mobile wireless nodes can result in a larger number of broadcast packets sent by routing protocols. The increased number of broadcast packets can decrease DCAP's throughput. Using DCAP with routing protocols more resilient to mobility could allow DCAP to maintain a higher throughput than with routing protocols like AODV.

Bibliography

[1] Tzi-cker Chiueh Ashish Raniwala. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 2223 – 2234, 2005.

[2] Sandeep Bajaj, Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, Padma Haldar, Mark Handley, Ahmed Helmy, John Heidemann, Polly Huang, Satish Kumar, Steven McCanne, Reza Rejaie, Puneet Sharma, Kannan Varadhan, Ya Xu, Haobo Yu, and Daniel Zappala. Improving simulation for network research. Technical Report 99-702, University of Southern California, 1999.

[3] Pravin Bhagwat, Bhaskaran Raman, and Dheeraj Sanghi. Turning 802.11 inside-out. *SIGCOMM Comput. Commun. Rev.*, 34(1):33–38, 2004.

[4] Anja Feldmann, Anna C. Gilbert, Polly Huang, and Walter Willinger. Dynamics of IP traffic: A study of the role of variability and the impact of control. In *SIGCOMM*, pages 301–313, 1999.

[5] Ali Hamidian. A study of internet connectivity for mobile ad hoc networks in ns 2. Master's thesis, Lund Institute of Technology, Sweden, January 2003.

[6] N. Jain, S. Das, and A. Nasipuri. A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks, 2001.

[7] R. Karrer, A. Sabharwal, and E. Knightly. Enabling large-scale wireless broadband: the case for TAPs. *SIGCOMM Comput. Commun. Rev.*, 34(1):27–32, 2004.

[8] 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.

[9] A. Nasipuri and S. R. Das. A multichannel CSMA MAC protocol for mobile multihop networks. In *IEEE Wireless Communications and Networking Conference*, 1999.

[10] A. Raniwala and Tzi cker Chiueh. Evaluation of a wireless enterprise backbone network architecture. In *High Performance Interconnects, 2004. Proceedings. 12th Annual IEEE Symposium on*, pages 98–104, 2004.

[11] A. Raniwala and Tzi cker Chiueh. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 2223–2234, 2005.

[12] 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.

[13] 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.

[14] IEEE standard 802.11. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, 1999.

[15] Z. Tang and J. Garcia-Luna-Aceves. Hopreservation multiple access (HRMA) for multichannel packet radio networks. In *Proc. IEEE IC3N*, 1998.

[16] Asimakis Tyamaloukas and J. J. Garcia-Luna-Aceves. Channel-hopping multiple access. In *Proc. IEEE ICC 2000*, pages 415–419, 2000.

[17] Asimakis Tzamaloukas and J. J. Garcia-Luna-Aceves. A receiver-initiated collision-avoidance protocol for multi-channel networks. In *INFOCOM*, pages 189–198, 2001.

[18] J. Dunagan V. Bahl, R. Chandra. SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In *Proc. ACM Mobicom*, 2004.

[19] A. Leon-Garcia Wing-Chung Hung, K.L. Eddie Law. A dynamic multi-channel MAC for ad-hoc LAN. In *Proc. 21st Biennial Symposium on Communications*, pages pp.31–35, June 2-5 2002.

[20] Shih-Lin Wu, Chih-Yu Lin, Yu-Chee Tseng, and Jang-Laing Sheu. A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks. In *ISPAN '00: Proceedings of the 2000 International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '00)*, pages 232–237. IEEE Computer Society, 2000.

APPENDICES

Appendix A

Validation Tests

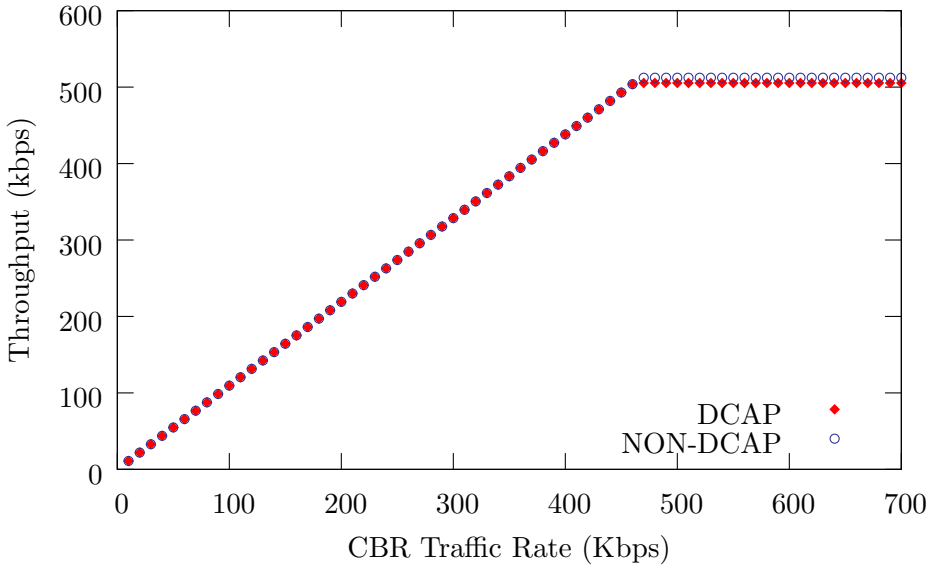

Figure A.1: Subtest 1.1 Throughput

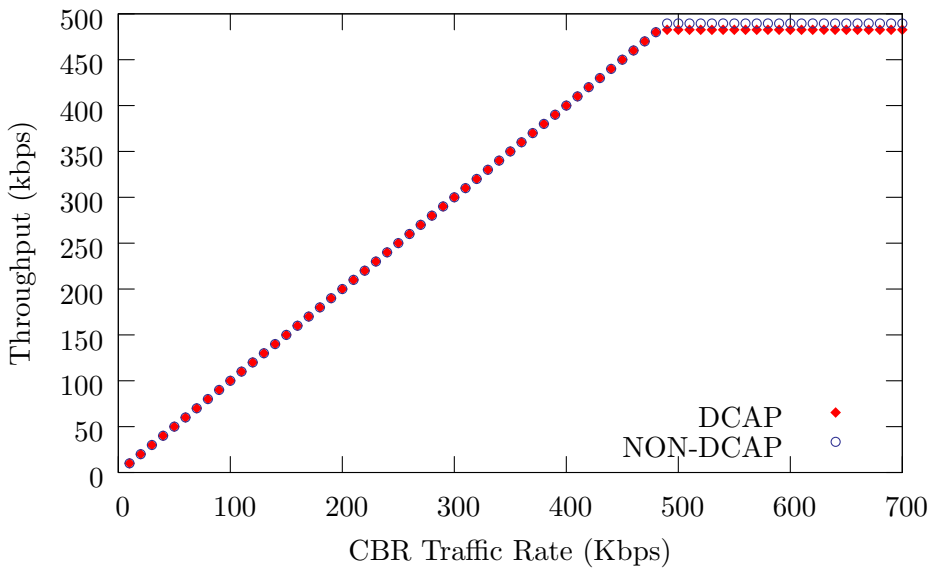

Figure A.2: Subtest 1.2 Throughput

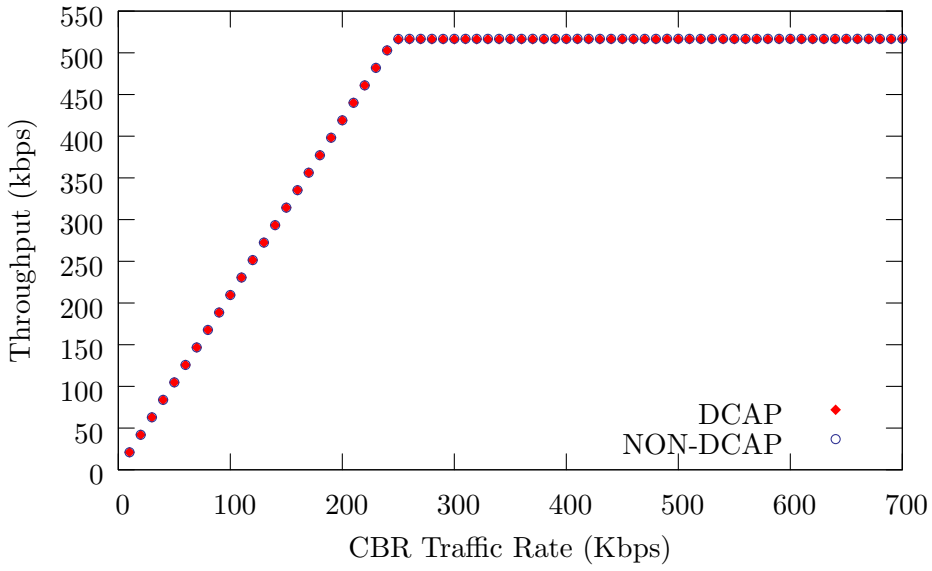

Figure A.3: Subtest 1.3 Throughput

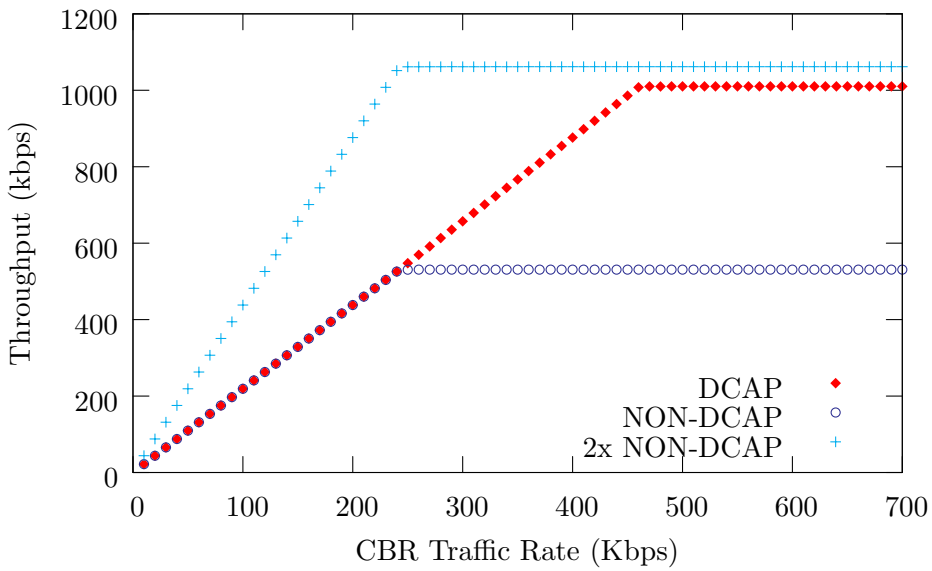

Figure A.4: Subtest 2.1 Throughput

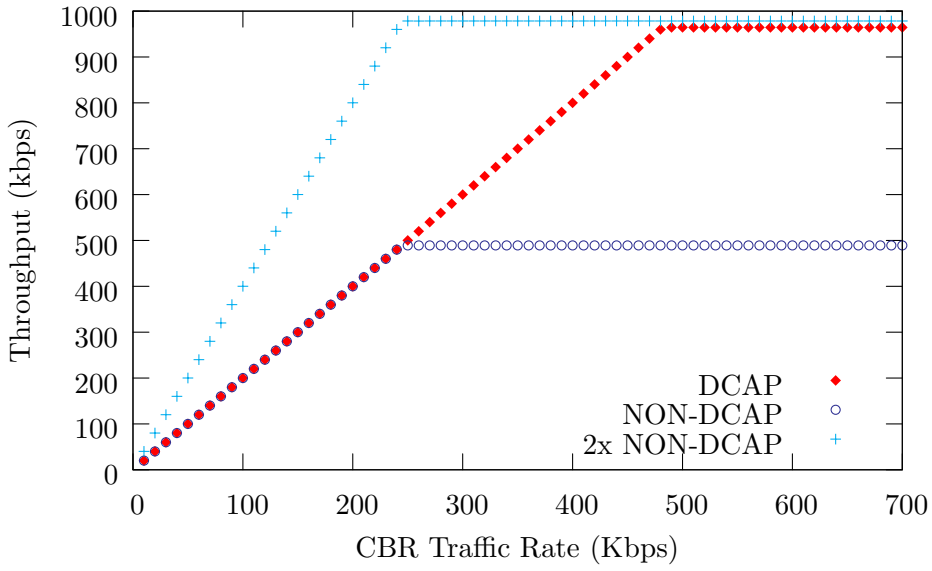

Figure A.5: Subtest 2.2 Throughput

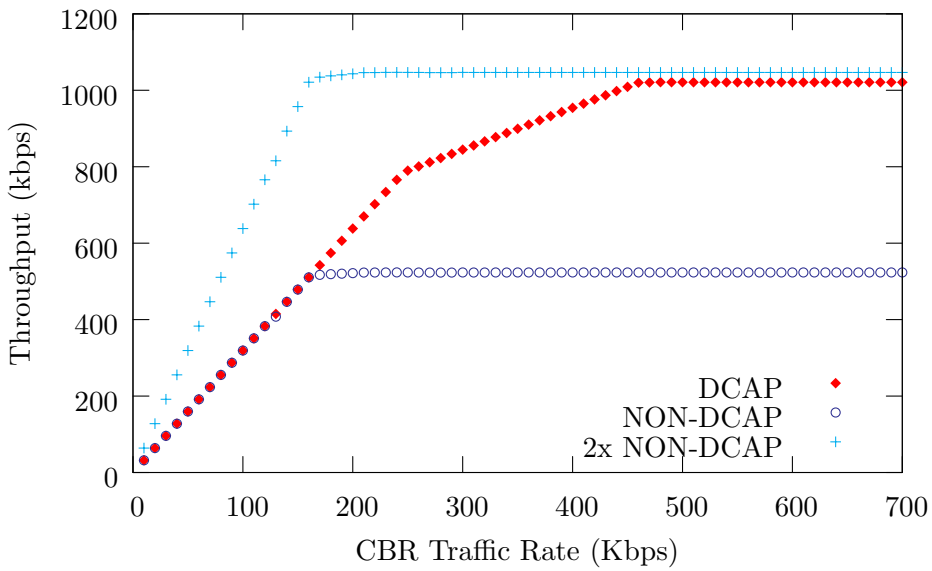

Figure A.6: Subtest 2.3 Throughput

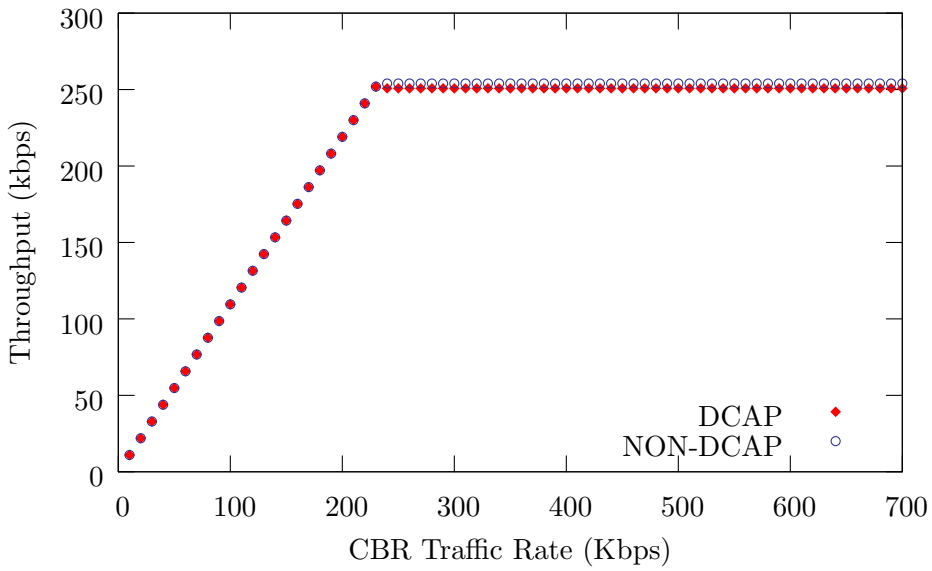

Figure A.7: Subtest 3.1 Throughput

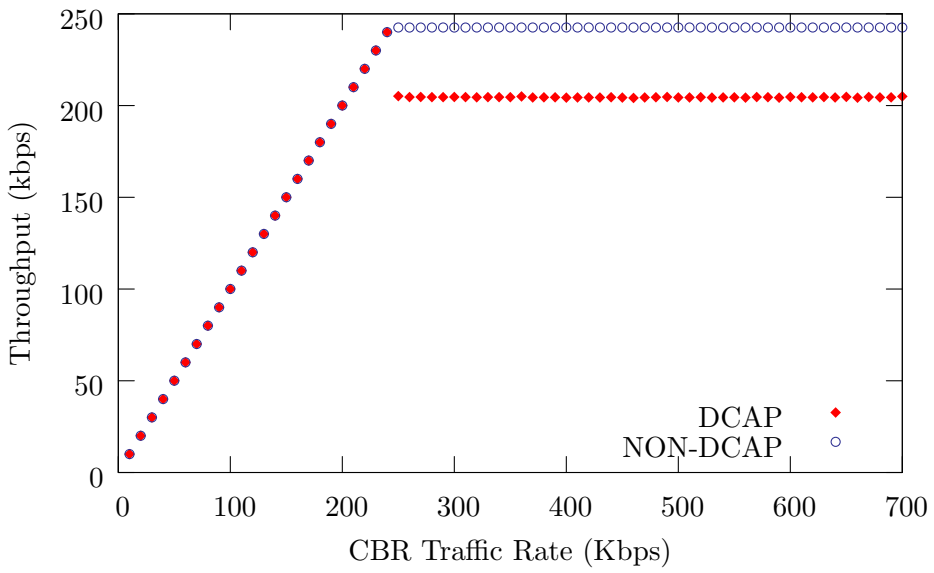

Figure A.8: Subtest 3.2 Throughput

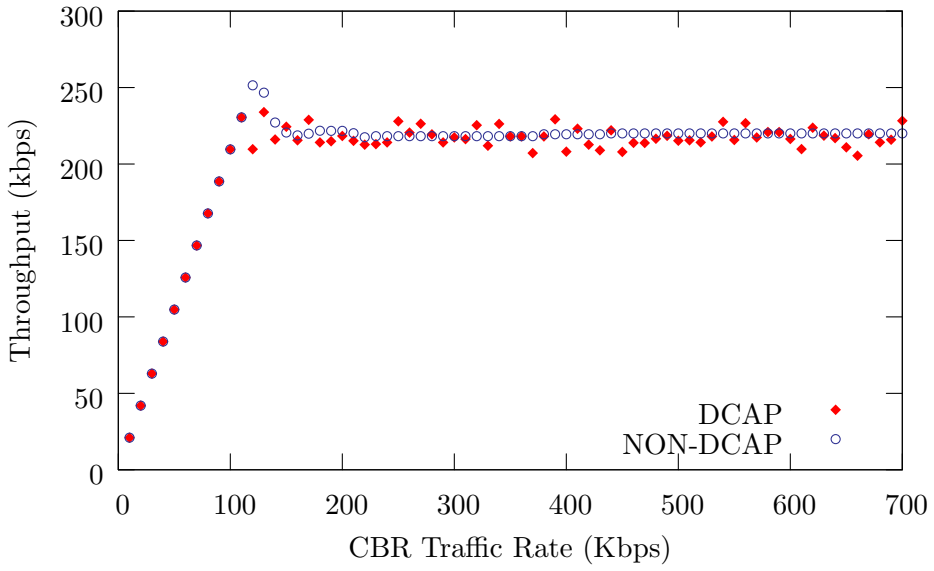

Figure A.9: Subtest 3.3 Throughput

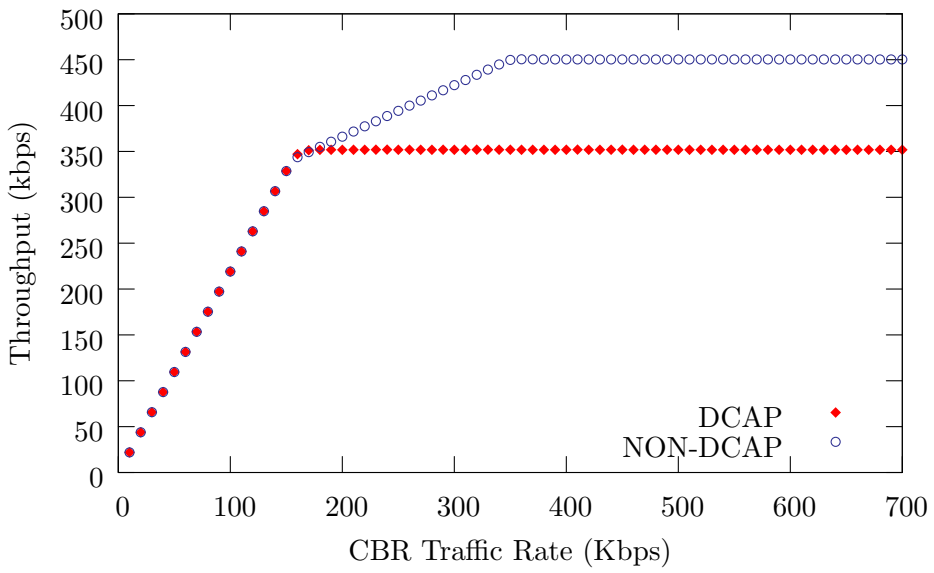

Figure A.10: Subtest 3.4 Throughput

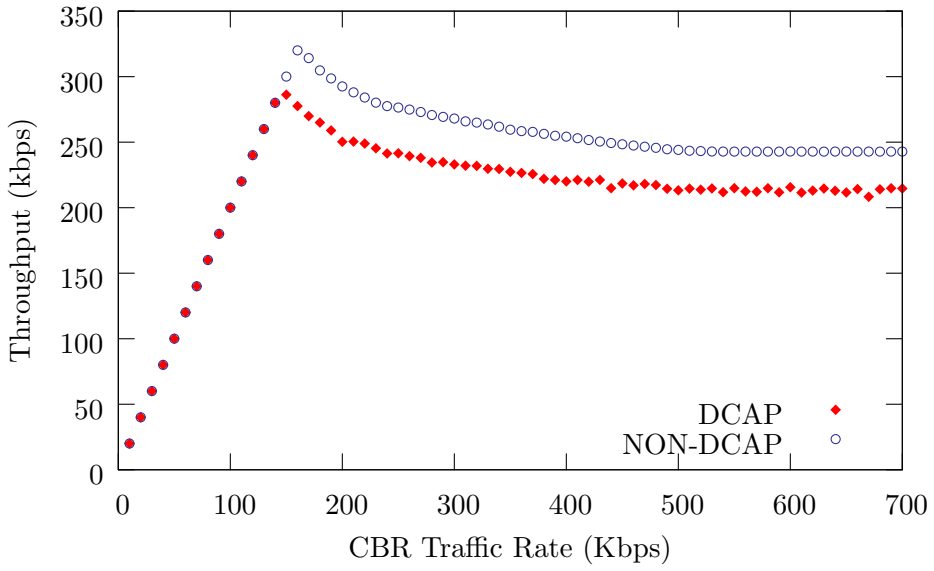

Figure A.11: Subtest 3.5 Throughput

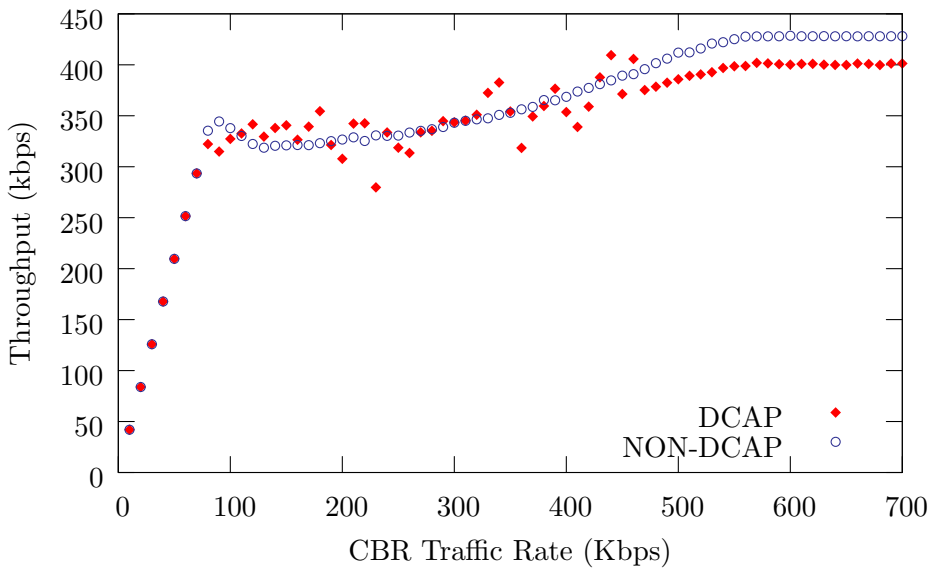

Figure A.12: Subtest 3.6 Throughput

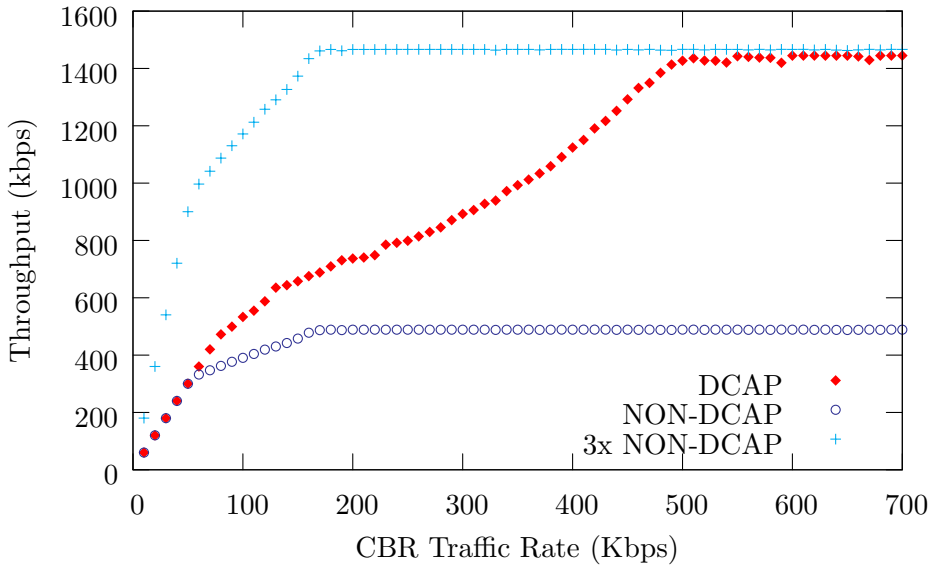

Figure A.13: Subtest 4.1 Throughput

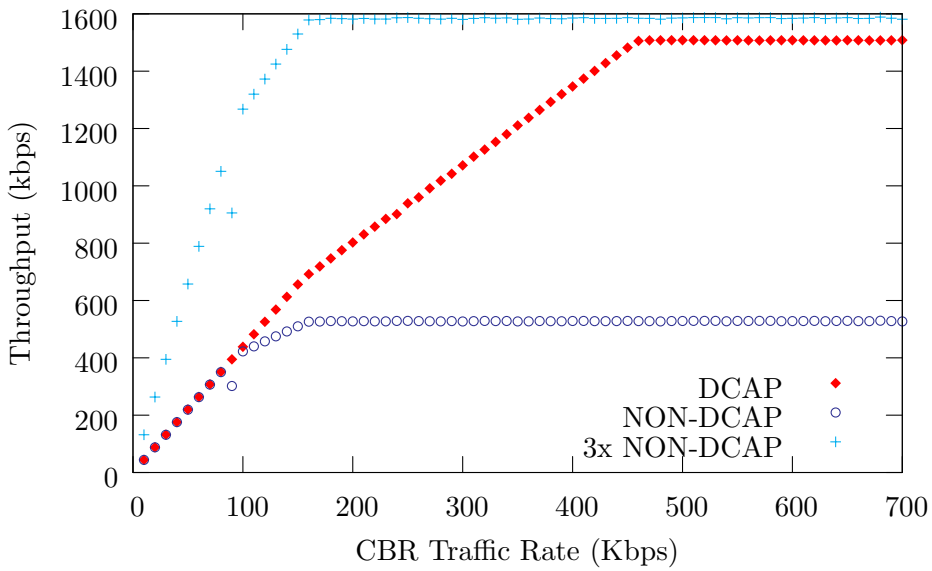

Figure A.14: Subtest 4.2 Throughput

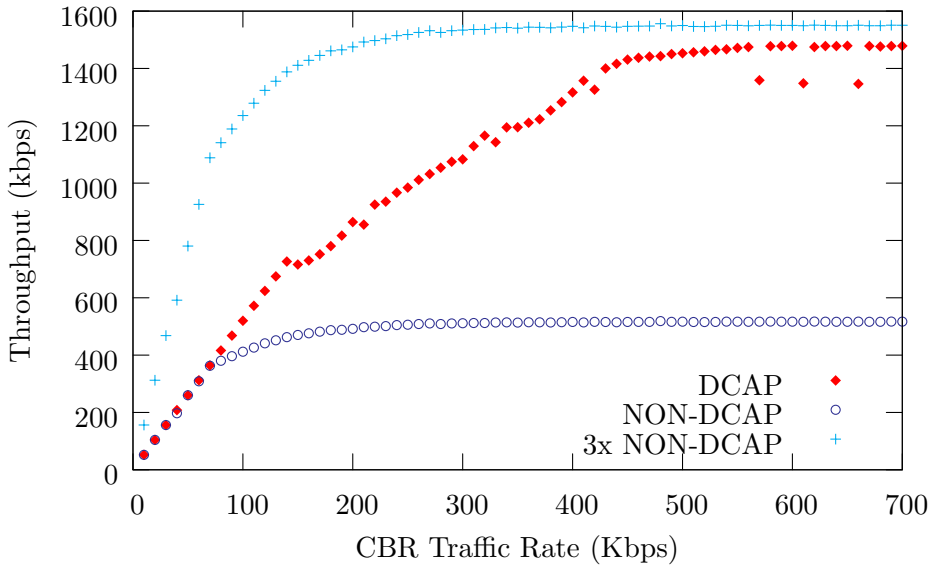

Figure A.15: Subtest 4.3 Throughput