8-1-2005

DMesh: Incorporating Practical Directional Antennas in Multi-Channel Wireless Mesh Networks

Saumitra M. Das

Himabindu Pucha

Dimitrios Koutsonikolas

Y. Charlie Hu

Dimitrios Peroulis

DMesh: Incorporating Practical Directional Antennas in Multi-Channel Wireless Mesh Networks

Saumitra M. Das
Himabindu Pucha
Dimitrios Koutsonikolas
Y. Charlie Hu
Dimitrios Peroulis

TR-ECE 05-16
August 2005

School of Electrical
and Computer Engineering
Purdue University
West Lafayette, IN 47907-2035

Contents

Abstract

Wireless mesh networks (WMNs) have been proposed as an effective solution for ubiquitous last-mile broadband access. Three key factors that affect the usability of WMNs are high throughput, cost-effectiveness and ease of deployability. Recent research has focused on increasing WMN throughput assuming the use of multiple radios equipped with omnidirectional antennas accompanied with channel assignment to enable frequency separation *between contending transmissions. Compared to omni antennas, directional antennas offer* spatial separation *between contending transmissions and have the potential to further enhance the throughput of WMNs.*

In this paper, we propose DMesh, a WMN architecture that combines spatial separation from directional antennas with frequency separation from orthogonal channels to improve the throughput of WMNs. An important requirement in DMesh is to accomplish this throughput improvement without inhibiting the other two key WMN requirements: cost-effectiveness and ease of deployability. The high cost of smart *beamforming directional antennas and their form factor make it difficult to achieve these two requirements. Thus, in DMesh, we focus our effort on incorporating* practical *directional antennas that are widely and cheaply available (e.g. patch and yagi). The key challenge in DMesh is to exploit spatial separation from such practical directional antennas despite their lack of electronic steerability and interference nulling as well as the presence of significant sidelobes and backlobes.*

In this paper, we study how such practical directional antennas can improve the throughput of a WMN. Central to our architecture is a distributed, directional channel assignment algorithm for mesh routers that effectively exploits the spatial and frequency separation opportunities in a DMesh network. Simulation results show that DMesh improves the throughput of WMNs by up to 231% and reduces packet delay drastically compared to a multi-radio multi-channel omni network. A DMesh implementation in our 802.11b WMN testbed using commercially available practical directional antennas provides TCP throughput gains ranging from 31% to 57%.

Keywords*: Mesh networks, directional antennas, testbed, multi-channel, system design*

1 Introduction

WMNs are characterized by static mesh routers connected by wireless links to each other, each providing connectivity to end hosts. The most significant application of such networks is to provide broadband access where wired infrastructure is difficult or economically infeasible to deploy. Three key factors that affect the usability of WMNs are high throughput, cost-effectiveness and ease of deployability. These characteristics allow WMNs to support a large number of users while being competitive with other access services.

There have been several significant efforts on improving the throughput of WMNs [1, 2]. These works aim to improve throughput using multiple radios that utilize multiple channels (available in the IEEE 802.11a/b/g standards) to separate the contending transmissions in the frequency domain. However, these works assume the use of omnidirectional antennas at mesh routers, where a transmission on a given channel requires all other nodes in range to remain silent or use alternative channels. Thus, although multiple channels can separate the transmissions in the frequency domain, the extent of such separation is potentially limited by the number of available channels: (1) There exists a bound on the number of channels possible due to spectrum regulation. For example, 802.11a has 12 while 802.11b has only 3 non-overlapping channels. (2) Further, the number of available channels could potentially be even lower due to stricter spectrum regulation in some countries or channels being set aside to support other communication (e.g. among mesh routers, users and mesh routers, or other networks). (3) Finally, significant co-channel interference exists even among the non-overlapping channels [2] which limits the extent of frequency separation among contending transmissions. Due to these limitations, there is a need to find other means of separating contending transmissions to improve the throughput of WMNs.

Compared to omni antennas, directional antennas offer *spatial separation* between contending transmissions and have the potential to further enhance the throughput of WMNs. In this paper, we propose DMesh, a WMN architecture that combines spatial separation from using directional antennas with frequency separation from using orthogonal channels to improve the throughput of WMNs. An important requirement for DMesh is to accomplish this throughput improvement without inhibiting the other two key WMN requirements: cost-effectiveness and ease of deployability. The high cost of *smart* beamforming directional antennas and their form factor make it difficult to achieve these two requirements. Thus, in DMesh, we focus our effort on incorporating *practical* directional antennas that are widely and cheaply available for under $50 each (e.g. patch and yagi). The key challenge in DMesh is to exploit spatial separation from such practical directional antennas despite their lack of electronic steerability and interference nulling as well as the presence of significant sidelobes and backlobes. Further, enabling DMesh poses new design challenges, as it requires specialized protocols to enable routing, and channel assignment to exploit directionality. Central to our architecture is a distributed, directional channel assignment algorithm for mesh routers that exploits directional antennas and multiple channels to separate the contending transmissions in both *spatial* and *frequency* domains. Effectively, DMesh enables two degrees of separation between contending transmissions.

This paper makes the following contributions: (1) We propose and evaluate a WMN architecture that exploits both directional antennas for spatial separation and multiple orthogonal channels for frequency separation to provide significantly increased throughput; (2) We focus on cost-effective and deployable techniques that can easily be incorporated to extend current single-interface networks such as Roofnet [3]; (3) We describe and evaluate a distributed routing protocol along with associated directional channel assignment algorithms to fully exploit the proposed architecture, using a realistic and detailed antenna model; (4) We evaluate the proposed architecture and compare it to previous solutions that use omnidirectional antennas in a detailed simulator; (5) We validate the performance of our architecture using experiments in a mesh network testbed with the same antennas modeled in the simulations.

Experimental results show that compared to a multi-radio multi-channel network using omni antennas, a DMesh network increases the average per source throughput from 128% up to 231% and provides up to 10 times smaller packet delays in an 802.11a network with 50 mesh routers. The throughput improvement from DMesh is more pronounced when the number of available channels is limited. When only 6 instead of 12 channels in an 802.11a network are used, the throughput using DMesh is improved by up to 176% with significantly lower delay in comparison to a multi-channel omni network. DMesh also provides TCP throughput gains between 30% and 68% in 802.11a and 35% in 802.11b networks. We implemented DMesh in our wireless mesh network testbed using commercially available practical directional antennas. Our testbed evaluation shows that DMesh increases TCP throughput in 802.11b networks between 31% and 57% compared to a multi-channel omni network.

The rest of the paper is organized as follows. In Section 2, we describe DMesh, our architecture for a wireless mesh network with directional antennas and multiple channels. Our evaluation methodology and results are detailed in Sections 3 and 4. An evaluation of the performance of DMesh in a real world testbed is described in Section 5. Section 6 summarizes the related work and Section 7 concludes the paper.

2 DMesh: A Directional Wireless Mesh Network

In this section, we first describe the architecture of DMesh followed by descriptions of the associated physical tree formation, routing protocol and distributed directional channel assignment.

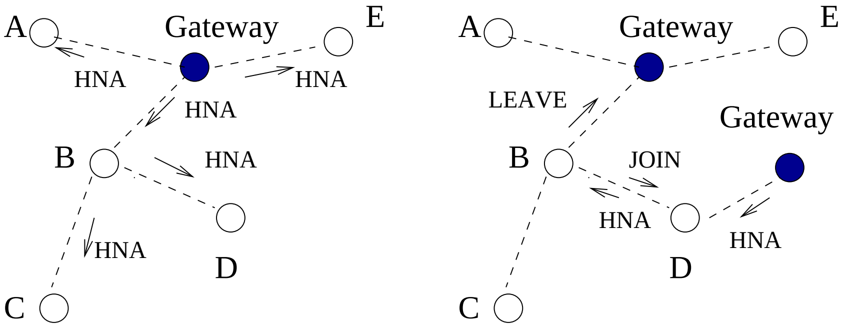
We consider a typical single-channel, single-interface WMN deployment with omnidirectional antennas in which mesh routers are placed on rooftops of subscribers [3] or other infrastructure (e.g. streetlights) and are interconnected via 802.11 links. DMesh is then used to enhance the performance of this existing network. To enable DMesh, we assume that each mesh router can have up to k additional interfaces (multiple radios) each with its own practical directional antenna. The practical directional antennas used in DMesh are non-steerable and they always point to the direction toward which they were manually placed during the network deployment. Since mesh routers are not highly form factor-constrained like other wireless devices (e.g. PDAs), the use of multiple radios and antennas is feasible. In fact, multi-radio mesh routers are already commercially available [4]. Given this DMesh architecture, the omnidirectional interface of each mesh router is always available for providing robust connectivity if the directional neighbors fail. Further, this omni interface is also used as a CONTROL interface.

The common application scenario for WMNs is likely to be Internet access in which most traffic will flow to and from gateway nodes which have wired connections to the Internet. These gateways will typically be deployed sparsely in the network due to their higher cost. The number and placement of these gateways are likely to be based on the area of deployment and availability of resources. Alternatively, in some scenarios it may be possible to place such gateways intelligently [5]. We focus on this application scenario in the rest of the paper. To support this application scenario, we propose to build high throughput routing trees rooted at the gateways. DMesh consists of three main stages to construct such trees: (1) physical tree formation to provide gateway connectivity using the directional antennas; (2) routing state creation and maintenance to correctly deliver packets along the tree; and (3) channel assignment to separate spatially contending transmissions whenever possible to further increase the throughput.

OLSR [6] is currently used as the multi-hop routing protocol in our single-radio single-channel omnidirectional mesh network testbed (described in Section 5). Thus, DMesh extends the OLSR protocol to aid in physical formation of trees using practical directional antennas, set up and maintain correspond-

Interface	Type	PIP
`wlan0`	OMNI	—
`wlan1`	DIR	1.0.0.5
`ath0`	DIR	2.0.0.7
`ath1`	DIR	3.0.0.3

(a) Interface Configuration



(b) Tree Construction

(c) Multiple Gateways

Figure 1: Physical tree formation in DMesh.

ing routing state, and perform channel assignment. We call this extended protocol DOLSR (*Directional OLSR*).

2.1 Physical Tree Formation

The physical tree formation takes place as follows. When each mesh node boots up, the DOLSR daemon reads the configuration file (*olsr.conf*) to identify the interface attached to an omni antenna and starts running in the single-channel single-interface mode. The configuration file depicted in Figure 1(a) contains a list of interfaces, along with the type of antenna attached to well as a PIP (pointing IP address) for each interface. The use of PIP is explained later in this section. In this mode, the node starts listening for HNA (*Host and Network Association*) messages on its omni interface. Nodes that have Internet access (i.e. a default route to a gateway or the gateway itself) advertise their connectivity over their omni interface using periodic HNA packets. The joining node picks the best next hop node based on metrics measured to nodes from which it receives HNA messages. The metric could be of various types, for example, hop count, remaining path bandwidth or a measurement based metric (e.g. ETX [7]). After the new node joins, it continues to receive HNA messages to reevaluate its choice of a next hop.

To incorporate directional antennas into this existing network, we modified the HNA message to also advertise whether a node is willing to host directional interfaces for its neighbors. A node can host directional interfaces after it itself has Internet connectivity through a directional interface or is the gateway node. In addition, the node should have free interfaces to connect to new children. For example, in Figure 1(b), the gateway node bootstraps DMesh by advertising its willingness to host directional interfaces in its HNA packets. The one-hop neighbors of the gateway (A, B and E) receive these HNA packets.

When DOLSR receives HNA packets on the omni interface with willingness indicated, it also evaluates the best potential *directional* next hop node (PARENT) using a suitable metric. For example, throughput measurements using the omni interface can be used to drive the PARENT selection. This choice of PARENT is then reported, directional antennas on the PARENT node and CHILD node (joining node) are installed pointing at each other, and corresponding entries in the *olsr.conf* on the PARENT and CHILD nodes are modified by offline means to include information about the new directional interface. Specifically, the PARENT node marks the interface as *DIR* and stores the IP address of the CHILD node's interface it is pointing to and vica-versa. This IP address is referred to as the PIP and only packets received from the PIP configured for a particular interface are passed to the DOLSR protocol. The PIP is required to filter out packets received from the sidelobes from other nodes that the antenna is not pointing at. Figure 1(a) depicts the *olsr.conf* file on the gateway after A, B and E have installed directional antennas toward the gateway. Directional interfaces are initially configured to operate on a default channel to enable connectivity. Note that in a cooperative network such as RoofNet, a CHILD could purchase antennas for its PARENT node that is willing to host an interface. On the other hand, in a commercial WMN, the ISP takes care of the installation.

After installation is complete and the updated configuration file is read in by the DOLSR daemon, the PARENT node sends out READY packets on the newly installed directional interface. The READY message acts as a trigger for the routing protocol on the CHILD node to set up the routing state (explained in the next section). Once the routing state is set up, the CHILD node also starts indicating willingness to host directional interfaces and more nodes can now join the tree. For example, in Figure 1(b), nodes C and D install directional antennas towards node B. In this way, nodes install at least one directional antenna on one of their interfaces pointing to a PARENT node, and a tree will be physically formed toward the gateway. In this paper, we evaluate our architecture over many different topologies generated by simulating the above tree formation model.

2.2 Routing Protocol

The *physical* placement of directional antennas in DMesh naturally forms a tree structure. On top of this physical tree, DOLSR now sets up forwarding entries to route packets along the tree. The routing state is set up as follows. Let T_a be any PARENT node. As mentioned earlier, T_a periodically broadcasts

READY messages on its newly installed directional interface. Only the corresponding CHILD node will receive the READY message due to filtering based on the PIP. On receiving READY messages from T_a, the CHILD node sends a JOIN message to T_a. T_a uses the IP address received from the JOIN packet to set up a forwarding entry toward the joining node. The joining node uses the PARENT's IP address (from the READY message) as a default route and inserts this information into its routing table. Once this exchange is completed, the joining node becomes part of the tree and sends its own HNA packets on the omni interface and can potentially become a PARENT for other nodes. Note that nodes that have the gateway as a PARENT node are denoted SUPERPARENTs (e.g. Nodes A, B and E in Figure 1(b)).

Once the JOIN packet is received, a ROUTE_SETUP message initiated by T_a is multicast along the tree simply by having each node recursively send the message to each of their active interfaces. The multicast proceeds until the gateway is reached since the gateway does not rebroadcast the message to the other SUPERPARENTs. The ROUTE_SETUP packet contains the IP address of the interface that the joining node used in the JOIN packet. This information is used to set up forwarding table entries in the routing tables of all nodes in the subtree including the gateway, to reach the newly joined node. This information also enables peer-to-peer routing (between mesh routers): a mesh router can route to all mesh routers in its own subtree rooted at its SUPERPARENT, and it can route to the mesh routers that are offsprings of other SUPERPARENTs by going through the gateway. A node only needs to consult its forwarding table to determine the next hop for any packet. A packet that is not matched with any forwarding entry is sent on the default path to the gateway, i.e. to the node's PARENT.

Switching Parents In certain situations, a node may be required to switch parents, e.g. if a better next-hop is detected through metric measurements or when a new gateway is installed. For example, when a new gateway is installed, each gateway starts its own HNA, and to the mesh routers, this is treated like any other HNA: if the metric advertised from a different gateway is lower than the one the node is currently using, the node will find that to be a better choice. For example in Figure 1(c), when B changes its PARENT to D, a LEAVE message is multicast by B similar to the ROUTE_SETUP message to invalidate old routing entries. When a node changes its PARENT, the subtree rooted at that node also moves to the new parent. The new PARENT multicasts a ROUTE_SETUP message to refresh the routing entries at all nodes in the subtree of its SUPERPARENT. In this manner, the routing state is maintained despite physical topology changes. For simplicity of evaluation, in this paper, we consider only a single gateway.

An internal node reattaching to a new PARENT can cause cycles to form. To eliminate this, we restrict the choice of the new PARENT to a node that is not part of a node's own subtree as follows: every node keeps track of nodes in the subtree rooted at itself and does not choose any of these nodes as parents, and this is accomplished by each node storing a list of nodes that sent ROUTE_SETUP messages through it. Also, in our architecture, multi-homing or multiple parents are not supported and each node has only one PARENT.

Failure Handling Note that in DMesh many issues such as simultaneous joining and oscillations are rare since only a designated node (with matching PIP) can attach to an interface. If a node does not receive any READY messages from its directional interfaces, it defaults to using the omnidirectional network which always remains as a backup for robust connectivity.

In summary, DOLSR aids in the formation of the physical DMesh tree as well as sets up and maintains routing state in the presence of physical topology changes.

2.3 Distributed Directional Channel Assignment

Once the routing state is set up, one degree of separation has been achieved since more parallel transmissions can now occur in the network due to spatial separation. However, practical directional antennas cannot provide complete spatial isolation due to the presence of *sidelobes* and *backlobes*. Therefore, DMesh also uses directional channel assignment to achieve the second degree of separation by separating these spatially contending transmissions in the frequency domain to further increase the throughput of the mesh network.

The directional channel assignment is performed by a PARENT for its CHILD node when the node joins. As mentioned earlier, both interfaces are initially configured on the same default channel to enable connectivity. As depicted in Figure 2, the PARENT selects a new channel using a channel assignment scheme and sends an ASSIGN message to the CHILD. Both PARENT and CHILD then use *iwconfig* to set their respective directional interfaces into the selected channel. Following this, bidirectional communication can occur. The channel assignment is periodically re-evaluated every 300 seconds.

2.3.1 Channel Assignment Schemes

In the routing tree, we denote each PARENT → CHILD or CHILD → PARENT communication as a flow. Let the network contain f one-hop flows and C channels. The *directional channel assignment*

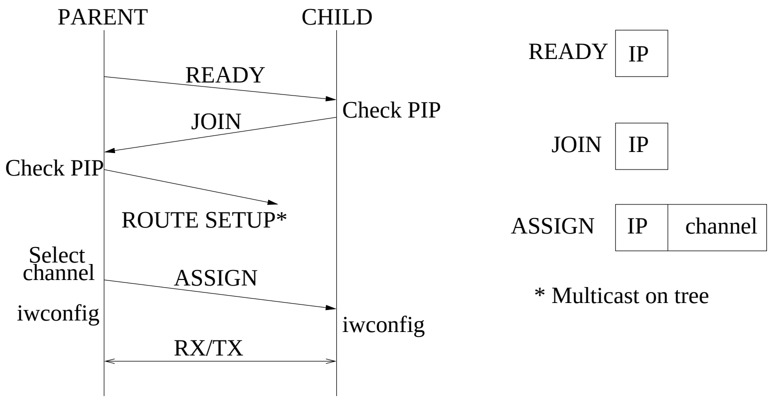
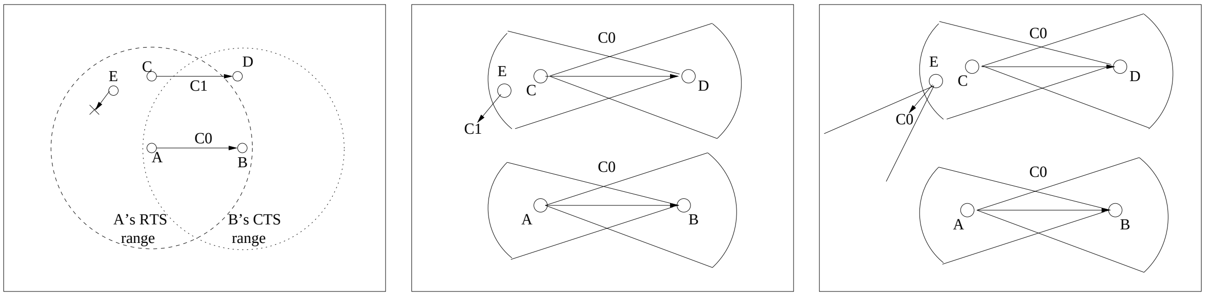


Figure 2: Time sequence of join and channel assignment operations in DMesh.

problem is to assign to each flow f_i a channel C_i taking into account directional interference, such that the available capacity to each flow is greater than or equal to the average bandwidth requirement for that flow. From a graph theoretic standpoint, channel assignment corresponds to some constrained coloring of a graph $T'(V, E)$ [8]. If $T(V, E)$ corresponds to the routing tree constructed to the gateway, then $T'(V, E) = T(V, E) \cup E'$, where E' consists of a set of edges such that edge $(u, v) \in E'$ if u and v, which are vertices of T, interfere with each other directionally. In an optimal algorithm, edges in T' can be assigned channels such that if any two edges are assigned the same channel, there should be no other edge from the transmitter of one edge to the receiver of the other edge. However, our network has additional constraints such as a finite number of channels, a finite number of interfaces, directionality-based capacity constraints and interference phenomena due to sidelobes and backlobes that are difficult to model. These constraints are complex to capture in an edge-coloring formulation. We also show that the directional channel assignment problem is NP-hard (see Appendix). Therefore, in this paper, our focus is to design a distributed algorithm that obtains a heuristic assignment which provides good performance and is easy to implement and deploy. In the following, we discuss our two omnidirectional and four directional channel assignment schemes using an example. Consider the scenario in Figure 3(a) where there are five mesh routers (A-E) and two flows A→B and C→D. Assume that E has a flow terminating at some other node in the direction indicated.

Omni / No Channel Assignment (ONOCA) In this scheme, omnidirectional antennas are used without multi-channel techniques. If A transmits to B omnidirectionally, then the channel is reserved in the entire region due to the RTS and CTS range of A and B as shown in Figure 3(a). Thus, only one flow can proceed. In this network there is no spatial or frequency separation possible between contending transmissions in a single collision domain.



(a) Omnidirectional with OCA (b) Directional with C-DCA (c) Directional with A-DCA

Figure 3: Channel assignment schemes.

Omni / Channel Assignment (OCA) In this scheme, omnidirectional antennas are used with multi-channel techniques. Each node first attempts to pick a unique unused channel for communicating to its PARENT. If the node cannot obtain an unused channel, it reuses the least loaded channels among those that have been selected by nodes in its interference range. This algorithm is similar to the one described in a recent work [1] and is implemented in a distributed manner by periodic exchange of channel usage information and least loaded channel selection. A least loaded channel is defined to be the one being used by the lowest number of flows. If the number of flows using a pair of channels are the same, then the traffic transmitted over the channels over the last information exchange period is used to determine the least loaded channel. Thus, CA exploits the frequency separation offered by multiple channels to allow contending transmissions in the same collision domain. If we apply this to Figure 3(a), assuming 2 channels exist in the network, both flows A→B and C→D can proceed using different channels C_0 and C_1. However, E still cannot transmit, since it is in the contention range of the other two flows. E realizes this through exchange of channel usage information and picks the least loaded channel in its vicinity.

Directional / No Channel Assignment (DNOCA) In this scheme, directional antennas are used without multi-channel techniques. Under this scheme, flows A→B and C→D can proceed since their *cone of interference* (the area defined by the sector of the transmitting nodes) is separated spatially. However, E cannot transmit since it lies in the cone of interference of flow C→D that uses the only available channel C_0.

Directional / Channel Assignment (DCA) In DCA, directional antennas are used with multi-channel techniques, and we aim to separate spatially (directionally) contending transmissions in the frequency domain. Two transmissions are defined to be *directionally* contending if either one of the nodes originating the transmission lies in the cone of interference of the other.

We propose and evaluate three directional channel assignment schemes. The first scheme is *Conservative DCA* (C-DCA). In C-DCA, given a set of channels C, a node X assigns a channel to a CHILD in two stages: (1) It first attempts to find a free channel C_i. A free channel for node X is defined as one that is not used by any node whose cone of interference contains X or its CHILD, i.e., no node in the vicinity of X and its CHILD is directionally contending on channel C_i. (2) If no such channel exists, the node X selects a channel C_i that is least loaded. If we use C-DCA in Figure 3(b), E will select to transmit using C_1, effectively insulating its transmission from its directionally contending transmissions. C-DCA is conservative because it always assigns a new channel to a transmission that lies in the cone of interference of another transmission. The C-DCA heuristic requires exchange of channel usage information in a local area to allow for selection of channels. For example, node E needs to receive channel usage information from C and D and perform geometric calculations to realize that it is in the cone of interference of D while executing the C-DCA algorithm. Calculating the cone of interference requires knowledge of the positions of C and D which is easily available in a static mesh network (e.g. through a localization scheme or GPS). In a mesh network with no mobility, it is even relatively simple to just encode the location of the mesh router when deploying or installing it, and no GPS hardware needs to be permanently installed in the mesh router. We use this one-time encoding method in our testbed. Other deployed testbeds such as RoofNet also maintain the GPS coordinates of their mesh routers.

Noe that C-DCA effectively takes into account the interference from sidelobes on a receiving node's antenna by assuming that the antenna can potentially receive interference from any direction (not just from the main lobe). In other words, C-DCA assumes the possibility of interference if a node lies in the transmit cone of neighboring nodes regardless of the orientation of the node's own antenna. For example, E could potentially use C_0 if its flow is not directed toward C or D. However, this would make E prone to interference through power received on its sidelobes on C_0. C-DCA does not allow the selection of C_0 to avoid such interference from sidelobes.

In addition, C-DCA takes into account the interference from the transmission power from a neighboring node antenna's sidelobes. Since C-DCA assumes a *flat-topped* transmit cone to make the geometric calculations simpler and applicable to any antenna, it can only anticipate interference caused by transmission through a main lobe of a neighboring node's antenna. To take into account the power from the sidelobes of a neighboring node's antenna, we incorporate measurement-based enhancements in the channel assignment as follows: Each node overhears transmissions on all its directional interfaces and

Table 1: Architecture choices in a multi-radio mesh network

Antenna	Channel Assignment
Omni	ONOCA, OCA
Directional	DNOCA, A-DCA, M-DCA, C-DCA

reception of data on an interface from any node whose IP address is different from the PIP for that interface indicates interference on the channel being used on that interface.

Our second assignment scheme, *Aggressive DCA* (A-DCA), assigns channels similar to C-DCA but with one important difference: X is considered to be in the cone of interference of another node Y, only if both X and Y lie in each other's cone of interference. Thus, in Figure 3(b), if E transmits in the direction shown, although E lies in the cone of interference of D, D does not lie in the cone of interference of E, and thus node E using A-DCA will reuse channel C_0. Thus, A-DCA is more aggressive in identifying opportunities where channels can to be reused based on directionality and thus results in a reduced channel usage. This can result in more interference (e.g. if E has sidelobes it will receive interference power from D). To accommodate interference from other transmissions, A-DCA uses a "guard angle" to control the aggressiveness of channel assignment as follows: the beamwidth of all transmit and receive cones during the geometric interference calculations are increased by guard angle degrees. Thus, a higher guard angle makes A-DCA more conservative when reusing channels while a lower guard angle makes A-DCA aggressive in reusing channels. Since the cone of interference is larger, nodes close to the transmission are likely to be separated in frequency. A typical guard angle value is 30°. Note that A-DCA does not use measurement-based enhancements, since that would limit its aggressiveness.

Finally, we evaluate a third assignment scheme, *Measurement-based DCA* (M-DCA), which uses only measurements instead of geometry to infer interference between two nodes. M-DCA works similarly to the measurement-based enhancement in C-DCA, where each node overhears transmissions in all its directional interfaces, and uses this information to discover interfering nodes.

Table I summarizes the various WMN architecture choices considered in this paper. Using omnidirectional antennas, we can apply OCA and ONOCA schemes, whereas using directional antennas we can apply DNOCA, A-DCA, M-DCA and C-DCA.

2.3.2 Distributed Algorithm for OCA/C-DCA/A-DCA/M-DCA

In order to select a channel in a distributed manner using local knowledge, a node needs to know the channel usage in its vicinity. This channel usage information is locally exchanged since the channel can be reused at a sufficient distance. Thus each node periodically (every 60 seconds) uses the omni CONTROL interface to broadcast its own channel usage information. The broadcast uses sequence numbers to suppress duplicates and is rebroadcasted for two hops in order to cover the interference range of that node which is typically twice that of the communication range. Note that in DMesh, the reachability of any node using the directional antennas is kept similar to the reachability achieved if the node were using an omnidirectional antenna by adjusting the transmission power on the directional interfaces. Note that the directional interfaces cannot be used to broadcast channel usage information since they will not cover all possible nodes that need to receive this information.

In OCA/C-DCA/A-DCA, each node broadcasts a channel vector C that specifies whether it is using each channel C_i, and a rate vector R that specifies a time average of the amount of traffic it has transmitted on each channel C_i over the past 60 seconds. For A-DCA/C-DCA, nodes initially exchange their position information and cache it for future use since the mesh routers are static. In our testbed, this position information is encoded using GPS during installation. However, other localization techniques could be easily used for this purpose as well. For A-DCA/C-DCA, a destination vector D indicating the destination being communicated with on each channel C_i is included for directionality information. The final channel map maintained at every node is constructed from the individual state vectors ($< C, R, D >$) received from neighboring nodes and is depicted in Figure 4. In addition, each CHILD also sends its PARENT a list of channels (BCL) computed from its own channel map which contains all channels being used in the CHILD's neighborhood. Once a node collects the channel usage information from its interfering neighbors (which may be more than 1 hop away), it uses the assignment algorithms to assign channels to a new CHILD or periodically (every 300 seconds) reevaluate the channel assignments for its current children.

In summary, for all the schemes[1], each node obtains a channel usage map which stores a list of node identifiers along with their associated channel and rate vectors. For A-DCA and C-DCA, a destination vector is also obtained. For M-DCA, there is no need of exchanging information between nodes. Each

[1] The pseudo-code for A-DCA, C-DCA and OCA are presented in the Appendix for the interested reader.

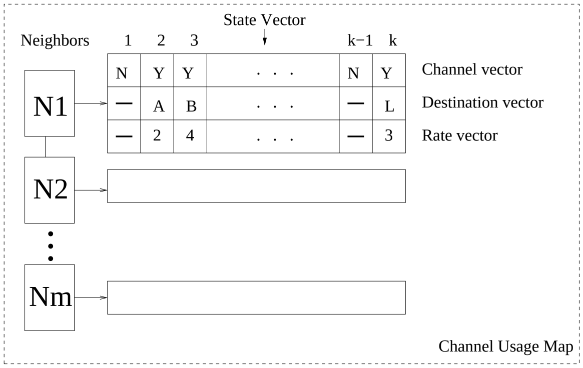


Figure 4: Structure of the channel map information stored at every mesh router to enable directional channel assignment. The channel map contains the rate vector R, channel vector C and destination vector D for each neighboring node. Each vector has a length equal to the number of available channels.

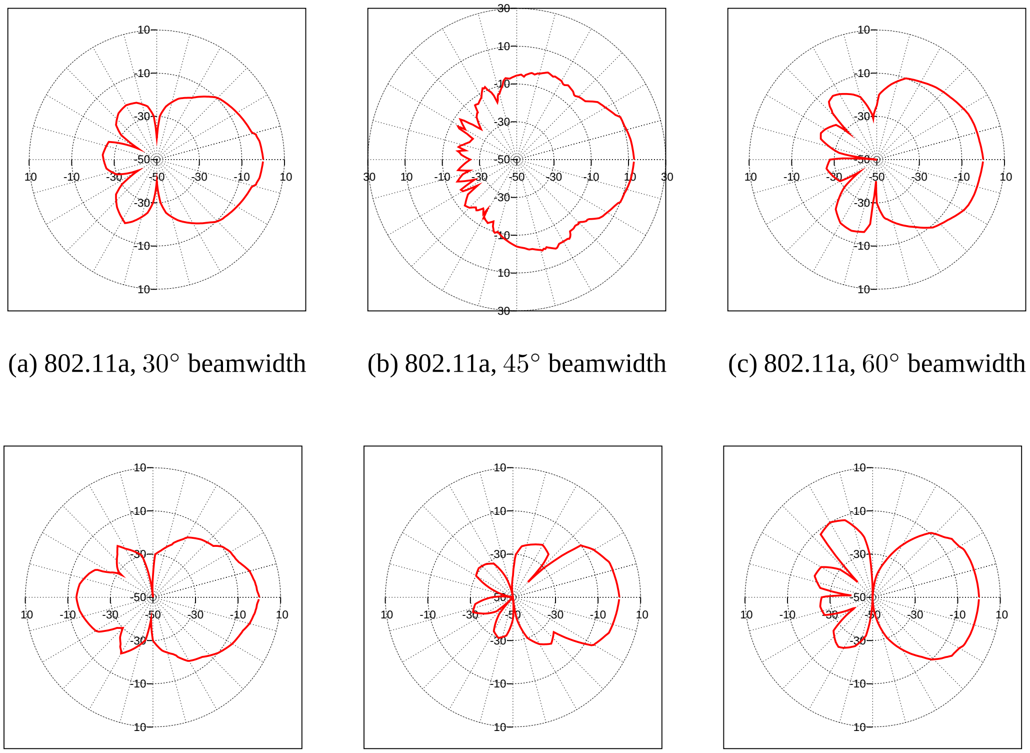
node monitors each of its directional interfaces for arrival of data from any node other than the child assigned to it. Reception of data from a non-child node indicates interference on the channel being used on that interface. This interfering node is then added to the channel usage map.

3 Experimental Methodology

We use the QualNet simulator [9] to evaluate DMesh. QualNet has been widely used to study directional antennas and multi-channel networks [10, 11, 12]. We model a node with multiple interfaces with support for dynamic channel assignment on each interface.

MAC, Physical Layer and Antennas We used IEEE 802.11a and 802.11b with autorate fallback. The simulator models OFDM and DSSS for 802.11a and 802.11b, respectively. Both MAC layers were verified to produce close to theoretically maximum throughput [13] and also model multi-rate operation based on inter-node distances. The two-ray path loss propagation model is used. We used real patterns for the directional antennas that faithfully model sidelobes. The patterns were taken from datasheets of commercially and cheaply available directional antennas from [14] and [15], and are shown in Figure 5. In fact, our testbed evaluation uses one of the antennas we model in the simulation.

Topology and Traffic We simulated a static mesh network of 50 nodes placed randomly in an area of 1000m x 1000m. Each node is assumed to have 1 omni interface, 1 directional interface to connect to a PARENT and 2 additional directional interfaces to support up to 2 children. The gateway is assumed to be at the center of the area. In order to isolate the effect of increased spatial reuse obtained from directional antennas from the range enhancement that also results, the reachability of any node using the



(a) 802.11a, 30° beamwidth

(b) 802.11a, 45° beamwidth

(c) 802.11a, 60° beamwidth

(d) 802.11b, 30° beamwidth

(e) 802.11b, 45° beamwidth

(f) 802.11b, 60° beamwidth

Figure 5: Antenna patterns used in the paper. Reproduced from manufacturer datasheets of commercially available practical directional antennas [14, 15].

directional antenna is kept similar to the reachability achieved if the node were using an omnidirectional antenna by adjusting the transmission power. We vary the number of sources and evaluate using both UDP and TCP traffic. All the sources communicate with the gateway node to simulate an Internet access pattern. The rate for each source is picked randomly from 0 to $Rate_{max}$. $Rate_{max}$ is varied from 100 Kbps to 5 Mbps. A packet size of 1500 bytes is chosen.

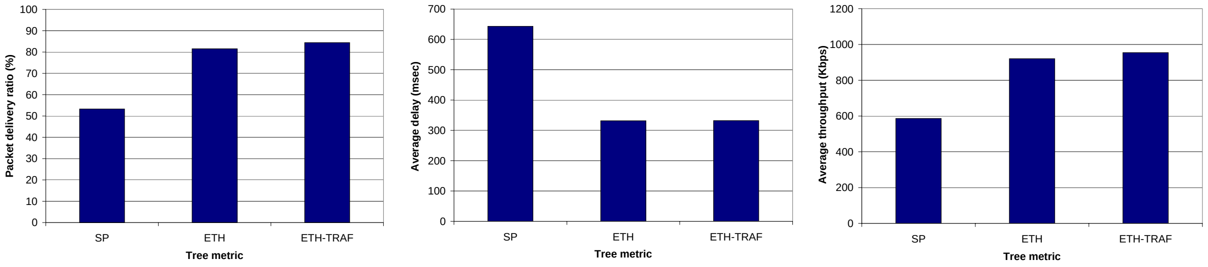
Metrics We use the following three metrics to evaluate the performance: (1) *Packet delivery ratio* (PDR) — the ratio of the overall number of successfully received data packets at the gateway to the number of data packets sent; (2) *Delay* — the average time between transmission and reception of data packets. This metric accounts for all possible delays caused by queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer times; (3) *Average source throughput* — the total number of bytes successfully transferred from each source during the simulation duration, divided by the simulation duration and averaged over all sources. This meteric is similar to PDR, except it also gives an indication of how much raw bandwidth is achieved. For all schemes, control overhead occurs only in exchange of channel usage information, and is negligible compared to the overall data traffic. Therefore, in the results below, due to lack of space, we omit the control overhead for all schemes.

4 Performance Evaluation

In this section, we first compare the impact of tree construction metrics. We then evaluate the performance of various WMN architectures. Following that, we perform controlled experiments by varying other parameters such as the number of available channels to study the tradeoffs involved in architecting a practical DMesh network.

4.1 Impact of Routing Tree Construction Metric

In this section, we evaluate the impact of routing tree construction metrics. The tree construction metric is particularly important for omni networks since there exist a large number of possible paths from a node to the gateway. Therefore, to fairly compare DMesh with an omni network, it is important to choose a good metric for the omni network. In contrast, DMesh has directed links between nodes and a tree is naturally formed by *physical* placement of antennas, which reduces the number of available paths and consequently the importance of highly tuned tree metrics. Thus, in this section, we focus on



(a) Packet Delivery Ratio (%) (b) Average Delay (msec) (c) Average Throughput (Kbps)

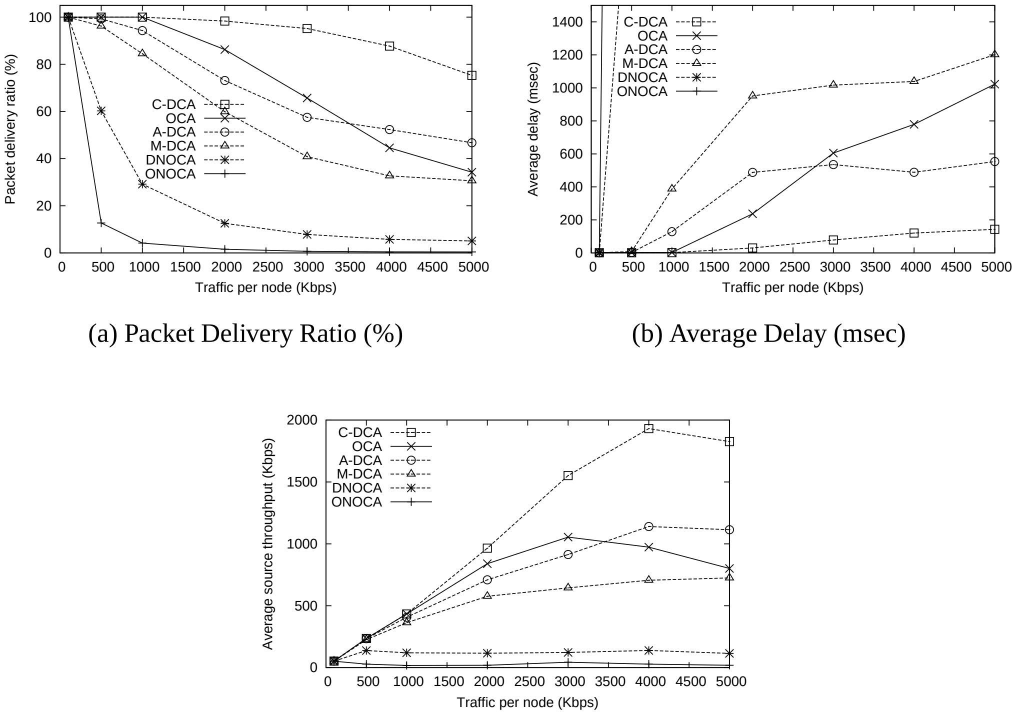
Figure 6: Different metrics for tree construction.

choosing a good metric for an omni network to fairly compare it to DMesh. Note that a tree construction metric is still useful in certain situations in DMesh when multiple parents connected on directional links advertise Internet connectivity.

We evaluate three different metrics: SP (shortest gateway hop count), ETH (expected throughput), and ETH-TRAF (expected path throughput with traffic profiles). In SP, each node chooses a PARENT node that advertises the lowest hop count in its HNA. In ETH, each nodes probes the throughput (e.g. using the *netperf* tool) to its potential PARENTs (from whom it receives an HNA) and selects the PARENT with the highest throughput. In ETH-TRAF, a node selects the parent based on the expected throughput of the whole path to the gateway from that parent, taking into account the traffic requirement of each node that affects the path. Note that ETH-TRAF is significantly more complex than the other two tree metrics. We evaluated the performance of OCA over these different tree metrics over 25 different network topologies with 12 non-overlapping channels. The performance results are depicted in Figure 6.

Figure 6 shows that SP performs poorly because it does not take into account the high rates possible with a longer path (in hops) as ETH does. Interestingly, there is not much difference between the other two schemes, ETH and ETH-TRAF. This implies that adding traffic profiling does not improve the performance significantly.

Hence, we find that the ETH metric is effective in constructing routing trees. It is based on simple throughput measurements between neighbors and does not require traffic profiling. In the rest of the paper, we use the ETH metric for all the evaluated architectures.



(a) Packet Delivery Ratio (%)

(b) Average Delay (msec)

(c) Average Throughput (Kbps)

Figure 7: Comparison of different mesh network architectures.

4.2 Overall Performance Comparison

In this section, we evaluate the performance gain from the different WMN architectures listed in Table I. The beamwidth of the directional antenna is assumed to be 45°.

Figure 7 shows the performance of all the above mesh network architectures in terms of the packet delivery ratio, delay, and average source throughput. As expected, the PDR, delay and throughput of ONOCA are the worst among all architectures. This is because ONOCA does not exploit any degrees of separation and is thus unable to support the simultaneous traffic load in the network.

DNOCA achieves better performance than ONOCA. DNOCA exploits spatial separation to support simultaneous transmissions and thus achieves higher throughput than ONOCA. At 500 Kbps traffic, the PDR of DNOCA is 48% better than that of ONOCA. However, spatial separation *alone* is not sufficient to support high throughput. The PDR of DNOCA drops drastically as traffic increases beyond 500 Kbps. This shows that practical directional antennas cannot be effectively used as complete spatial isolators. Two factors contribute to this poor performance: (1) The beamwidth of a practical directional antenna cannot be made arbitrarily small in order to completely separate all contending transmissions. (2) The presence of sidelobes limits the usability of multiple practical directional antennas at a node since one antenna will be able to hear the other due to reception from sidelobes. Thus, any pair of interfaces on the same node will always contend with each other. We also observed the sidelobe problem in our DMesh testbed and found that the problem was not mitigated even with greater physical separation between the antennas of a pair of interfaces.

OCA also achieves better performance than ONOCA. OCA exploits frequency separation to enable higher throughput than ONOCA. At 500 Kbps traffic, OCA achieves 88% higher PDR than ONOCA. However, separation by frequency alone is not sufficient for achieving the best performance. The PDR of OCA drops to only 40% and the delay drastically worsens as the traffic increases to 5 Mbps. An important factor contributing to this degradation is that there is an upper bound on the number of distinct orthogonal channels available which limits the ability to completely isolate all contending transmissions by frequency. For example, an examination of the channel assignment in this scenario shows that OCA shares many channels among multiple flows. Moreover, as the traffic increases, the per channel load increases, thereby increasing the contention among flows and thus the throughput drops. Thus, although OCA delivers a fraction of the packets, these packets are delivered after significant amount of queuing and buffering leading to high average delays. Note that another significant factor that could restrict the performance of OCA is inter-channel interference. Although the simulation evaluation of OCA assumes no inter-channel interference[2], in our DMesh testbed, we observed that two interfaces maximally separated by orthogonal channels fail to achieve simultaneous throughput. In summary, the observed performance of both DNOCA and OCA suggests that an additional degree of separation can further improve the throughput.

Another interesting observation from the above experiments is that OCA significantly outperforms DNOCA. Although both schemes have a single degree of separation, separation by frequency alone is more effective than separation by space alone from using practical directional antennas. This effect was also observed during our testbed experiments. This is primarily because any pair of interfaces on a single node, when connected to practical directional antennas can always hear each other (due to sidelobes and backlobes) and it is difficult to separate them physically far enough apart from each other to obtain simultaneous throughput. On the other hand, any pair of interfaces on a single node separated

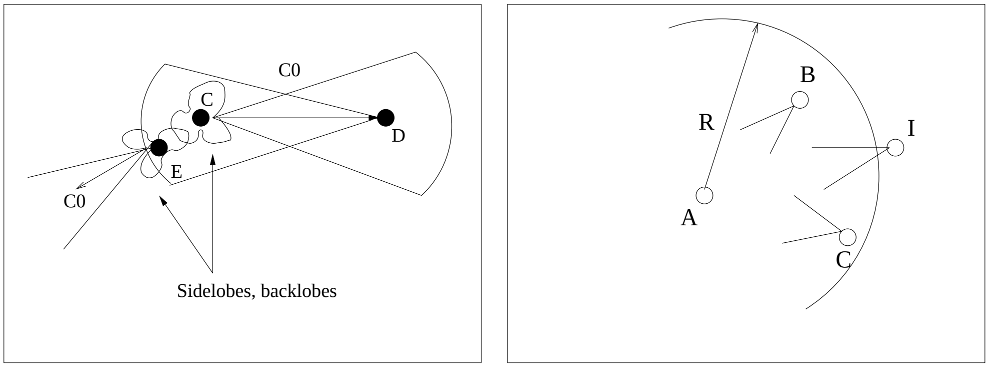
[2] Qualnet does not simulate inter-channel interference and the code where this feature can be implemented is not open-source. Only binaries are provided.

by frequency try to filter out each others signal. Although this filtering is also not perfect, the separation achieved between contending signals from this technique is superior to the separation achieved using a pair of practical directional antennas.

C-DCA fully exploits two degrees of separation offered by directional antennas and multiple channels to maximally isolate contending transmissions. The PDR and throughput of C-DCA are the highest and the delay the lowest among all schemes across all traffic scenarios. In particular, the throughput gains achieved in comparison to OCA are up to 128% with up to 10 times lower delay. On one hand, C-DCA avoids the sidelobe problems with multiple interfaces in DNOCA through frequency separation. On the other hand, C-DCA reduces the high channel load problem in OCA through spatial separation. Finally, C-DCA can account for interference from sidelobes of nearby transmitting antennas through measurements.

As described in Section 2.3.1, C-DCA is conservative in reusing channels. In contrast, the A-DCA algorithm aggressively exploits spatial separation to reuse channels which can potentially improve the throughput. However, the results show that A-DCA does not offer significant benefits. The performance of A-DCA is always lower than that of C-DCA and in fact comparable to that of OCA. In fact, A-DCA outperforms OCA only when the traffic exceeds 3Mbps when the average channel load in OCA increases significantly. The performance degradation of A-DCA is because A-DCA assumes a flat-topped pattern during geometrical calculations in channel assignment. This pattern has only one main lobe, and the power is kept very low for all other angles outside the main lobe. Since our simulations use real antenna patterns (i.e. with sidelobes and backlobes causing interference), while A-DCA optimistically assumes no interference from sidelobes in order to maximize channel reuse, it does not always assign non-interfering channels correctly. An example of this is shown in Figure 8(a) in which node C using A-DCA reuses channel C_0 to transmit to node D since the nearby node E is directed away from C. However, sidelobes and backlobes in the antennas on C and E using the same channel C_0 can cause interference, degrading the throughput.

Finally, to evaluate the accuracy of geometrical techniques in C-DCA/A-DCA, we compared them with a pure measurement-based channel assignment scheme, M-DCA. M-DCA accounts for sidelobes since it uses real measurements to infer interference between two nodes, instead of geometry. Surprisingly, Figure 7 shows that M-DCA achieves worse performance than A-DCA and OCA. The reason for this is that in M-DCA, each node checks if it interferes only with nodes from which it can *receive* pack-



(a) Sidelobe interference in A-DCA

(b) M-DCA estimation problem

Figure 8: Channel assignment issues in A-DCA and M-DCA.

ets, that is only from nodes in its communication range, and not in its range of interference. However, even nodes in twice the communication range may interfere because of carrier sensing. For example, in Figure 8(b), node A can assign channels to reduce interference from nodes B and C within its communication range R. However, node A cannot identify node I as an interferer (due to it not being able to receive packets from I) and is likely to assign the same channel node I is using, resulting in interference. Such inaccuracies in channel assignment significantly degrade the performance of M-DCA. Note that such far away nodes could be identified using using two-hop packet exchanges and interference estimated using geometric calculations. However, the objective of M-DCA is to study the effectiveness of a pure measurement-based algorithm.

In summary, exploiting two degrees of separation by integrating practical directional antennas with channel assignment significantly improves the performance of WMNs. Further, a combination of measurement-based and geometric techniques to assess interference for channel assignment, as in C-DCA, provides the best performance.

4.3 Impact of Guard Angle on A-DCA

As discussed in Section 2.3.1, A-DCA exploits directionality aggressively to minimally use channels. However, the results in the previous section showed that the aggressiveness of A-DCA can also hurt its performance. The aggressiveness of A-DCA can be controlled by means of the guard angle (GA). To evaluate whether we can tune the aggressiveness of A-DCA to achieve better performance, we performed simulations with varying GA. The effect of GA is tabulated in Table II. As GA is increased, the scheme

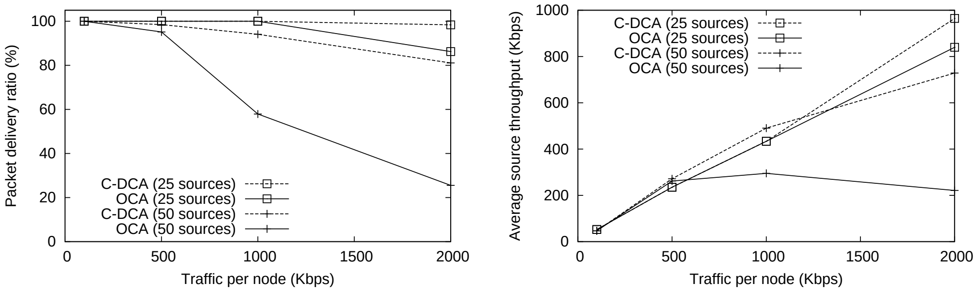
Table 2: Impact of Guard Angle (GA) on A-DCA performance. Maximum rate per source node is 2 Mbps.

Scheme	PDR	Delay	Throughput	Channels
	(%)	(msec)	(Kbps)	
A-DCA (GA)				
10	49.08	992.76	463.405	6
20	50.57	969.52	475.647	8
30	73.14	488.08	709.609	9
40	98.94	60.19	969.438	11
C-DCA	98.35	29.81	964.736	11

becomes less aggressive and the performance is closer to C-DCA. Lower aggressiveness improves the PDR, delay and throughput of the A-DCA scheme. However, an increase in GA increases the number of channels employed in assignment. As seen in the results, a GA of 40° has to be used for A-DCA to give performance similar to C-DCA. But with this large GA, A-DCA uses the same number of channels as C-DCA; in other words it is not aggressive anymore in reusing channels. Thus, we conclude that highly aggressive reuse of channels by exploiting fine-grained spatial separation is not feasible with practical directional antennas. The effectiveness of A-DCA with smart antennas that can tightly control the sidelobes is a focus of our future work.

4.4 Impact of Traffic Sources

In this section, we evaluate the impact of increasing the number of traffic sources on the performance of OCA and C-DCA (the best performing schemes for the omni and directional cases, respectively). Increasing the sources increases the need for separation of contending transmissions. For this experiment, all 50 nodes were sources of traffic. Figure 9 shows the results in terms of PDR and throughput for different values of maximum traffic. For comparison, we replot the results for 25 sources from the previous section. We make the following observations. First, when the number of sources increases, both the PDR and the throughput drop, and this is more pronounced for larger maximum traffic. This is obvious, since a larger number of sources increases the interference between contending nodes. Second, the gap between C-DCA and OCA becomes larger. For example, when the maximum traffic is 2Mbps, the difference between the PDR of the two schemes is about 14% with 25 sources, but increases to 224% when all 50 nodes are sources. Also, the difference in throughput is about 15% with 25 sources, but in-



(a) Packet Delivery Ratio (%)

(b) Average Throughput (Kbps)

Figure 9: Impact of increasing traffic sources. All mesh routers are sources of traffic.

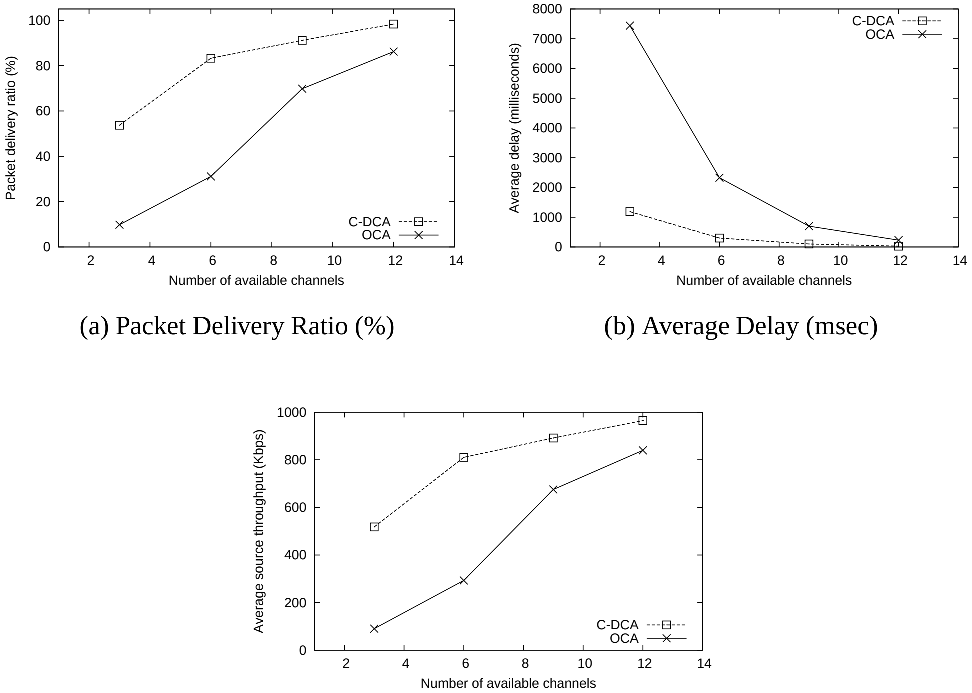
creases to 231% when all 50 nodes are sources. This is because when the number of sources increases, the total traffic in the network increases, and the channels become loaded faster. Hence, OCA has to reuse loaded channels, and this results in performance degradation. On the other hand, C-DCA can still exploit the second degree of separation, achieving larger PDR and higher throughput.

In summary, increasing the number of sources increases the difference in gain between the two schemes. The gain from combining spatial and frequency separation is more pronounced in heavily loaded networks.

4.5 Impact of Available Physical Channels

In this section, we study the impact of the number of available channels on the performance of WMNs. As the number of available channels is reduced, both OCA and C-DCA have to resort to reusing channels. This increases the average contention per channel.

Figure 10 depicts the performance of C-DCA and OCA, as the number of available channels is varied from 3 to 12. Two observations can be made from the results. First, both OCA and C-DCA experience performance degradation as the number of available channels are reduced, since both OCA and C-DCA have to resort to reusing channels. This increases the average contention per channel and worsens performance. Second, C-DCA performance is less affected than OCA as the available channels are reduced. This is because OCA performance is directly dependent on the number of channels as it has only one degree of separation between contending transmissions. In contrast, C-DCA requires less than 12 channels (11 channels on average) to completely isolate contending flows in most of the scenarios we studied. Even when C-DCA has to reuse channels, the average load per channel is also lower. Thus, in comparison to the 12 channel case, the gains of C-DCA against OCA in PDR, throughput, and delay



(a) Packet Delivery Ratio (%)

(b) Average Delay (msec)

(c) Throughput (Kbps)

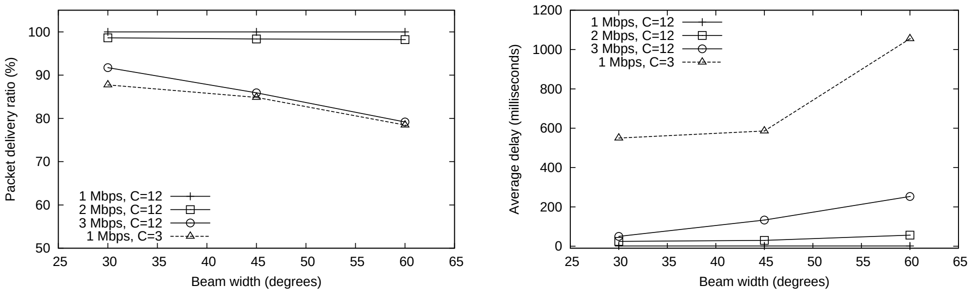
Figure 10: Impact of a finite number of available channels. Maximum traffic per node is 3 Mbps. Beamwidth used is 45 °.

increase to 173%, 176% and 676%, respectively, when the number of channels is reduced to 6. In summary, C-DCA is more effective at utilizing scarce spectrum resources than OCA.

4.6 Impact of Beamwidth

In this section, we study the impact of beamwidth on the performance of C-DCA. We vary the beamwidth from 30° to 60° in a network with 12 channels, and the maximum traffic per node from 1Mbps to 3 Mbps. We also repeat the experiment for a network with 3 channels and maximum traffic per node of 1 Mbps.

Figures 11(a) and 11(b) illustrate the effect of beamwidth on PDR and delay, respectively. In a network with 12 channels, C-DCA requires on average 11 channels to completely isolate contending flows. In this case, beamwidth has no effect on the performance of C-DCA. As shown in Figures 11(a) and 11(b), when the traffic is 1 or 2 Mbps, the PDR and the delay remain almost constant and close to 100% and below 50 msec, respectively. The beamwidth affects the performance, only when the traffic increases to 3 Mbps. With high traffic the performance drops, as we explained in Section 4.4, and in order to improve



(a) Packet Delivery Ratio (%)

(b) Average Delay (msec)

Figure 11: Impact of beamwidth on performance.

it, we can reduce the beamwidth to increase the spatial separation. In fact, as the beamwidth is reduced to 30°, the PDR of C-DCA increases to 92%.

Reducing the beamwidth is also useful, when the number of channels is limited. In this case, C-DCA in many cases cannot find free channels, and shares channels among contending flows, degrading the performance. In other words, when only a small number of channels are available, separation in the frequency domain is not enough, and increasing spatial separation is required to improve performance. For example, with a 60° beamwidth, when we reduce the number of channels from 12 to 3, the PDR drops from 100% to 78%, and the delay increases from 1 msec to 1 sec. By reducing the beamwidth to 45°, the PDR is increased by 13%, but the delay is reduced by almost 50%. However, further reduction of beamwidth to 30°, does not significantly improve the performance of C-DCA.

In summary, when the number of channels is reduced for a given number of nodes or the number of nodes in a collision domain is increased for a fixed number of channels, the spatial separation becomes insufficient. In such a scenario, lower beamwidth antennas offer the flexibility of increasing the spatial separation and thus improving the performance. The tradeoff is that the physical size of lower beamwidth practical directional antennas is larger than those with larger beamwidth.

4.7 802.11b Performance

In this section we study the performance of C-DCA in comparison to OCA in 802.11b networks. An 802.11b network uses different modulation schemes from 802.11a, and has significantly reduced bandwidths and only 3 orthogonal channels. On the other hand, the transmission ranges achievable are typically larger than 802.11a. 802.11b radios are widely used for constructing mesh networks and it is useful to understand the performance of our architecture in this scenario. The results of the experiment

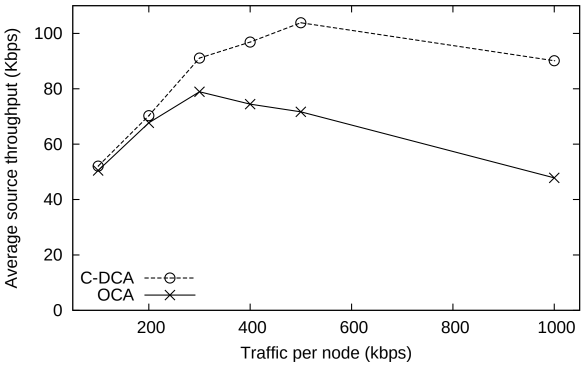


Figure 12: Performance in a 802.11b wireless mesh network.

are shown in Figure 12. In this figure, we show the average source throughput in Kbps for OCA and C-DCA.

First, as expected, the throughput for both schemes is significantly lower, compared to the throughput achieved in a 802.11a network (Figure 7(c)). Since there are only three orthogonal channels available, there is not enough frequency separation, and both schemes have to resort to reusing channels. For OCA, the frequency separation is not efficient with only three available channels, and the result is low throughput. One more factor that results in lower throughput in 802.11b, compared to 802.11a, is the larger transmission range, which can be typically achieved by the former. Because of larger range, the interference per node increases, since every node can listen to the transmissions of more neighbors.

Second, C-DCA still outperforms OCA, similar to in the 802.11a scenario. For OCA, the throughput drops quickly when the maximum traffic per source increases beyond 300 Kbps. For C-DCA, throughput starts dropping only after 500 Kbps. More importantly, the gap between the two schemes increases as the maximum traffic per node increases. For example, when the maximum traffic per node is 1 Mbps, C-DCA outperforms OCA by about 80%.

In summary, the results for 802.11b remain consistent with the results for 802.11a, showing that combining two degrees of separation increases the performance, compared to only frequency separation, even for 802.11b networks.

4.8 TCP Performance

In this section, we study how TCP flows perform using C-DCA in comparison to OCA. We study an 802.11b as well as an 802.11a network for this experiment. Figure 13 shows the percentage increase in TCP throughput for C-DCA compared to OCA, both for 802.11a and 802.11b. For the 802.11a scenario, we vary the number of sources as 15, 25, and 50; while for 802.11b we show the results for 25 sources.

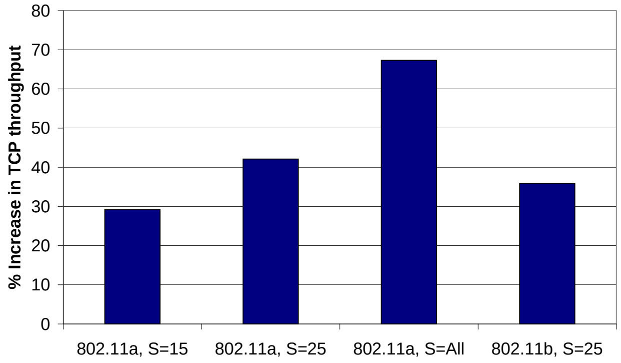


Figure 13: Increase in throughput of TCP flows from using C-DCA in comparison to OCA for 802.11b and 802.11a networks.

Each source initiates a TCP session for the entire duration of the simulation to the gateway with segment sizes of 1500 bytes.

In general, we observe that C-DCA increases the TCP throughput, compared to OCA. More specifically, for 802.11a, we observe that the gap in throughput between the two schemes increases with the number of sources. It is about 30% with 15 sources, and it reaches 68% when all 50 nodes are sources. For 802.11b and 25 sources, the gain from using C-DCA is about 35%. We showed in previous sections that delays are much higher in OCA, compared to in C-DCA. Given that TCP throughput is inversely proportional to RTT, such delays in OCA result in very low throughput for end users of the WMN. In addition, we found that the number of packet losses in OCA is higher than C-DCA, causing frequent TCP timeouts and consequently lower throughput.

5 Testbed Evaluation

In order to verify the usefulness and performance of the DMesh architecture, we incorporated practical directional antennas in MAP (Mesh@Purdue) [16], a wireless mesh network testbed at Purdue. We use C-DCA as the channel assignment algorithm for DMesh. We also compare DMesh with OMesh, the original MAP architecture with only omnidirectional antennas and running the OCA channel assignment algorithm.

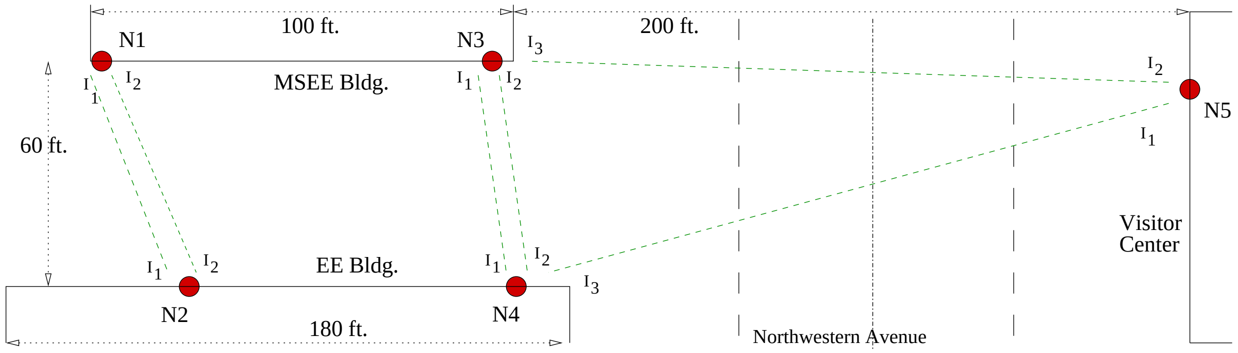


Figure 14: Purdue MAP testbed schematic (top view). Only outdoor mesh routers are depicted.

5.1 Setup

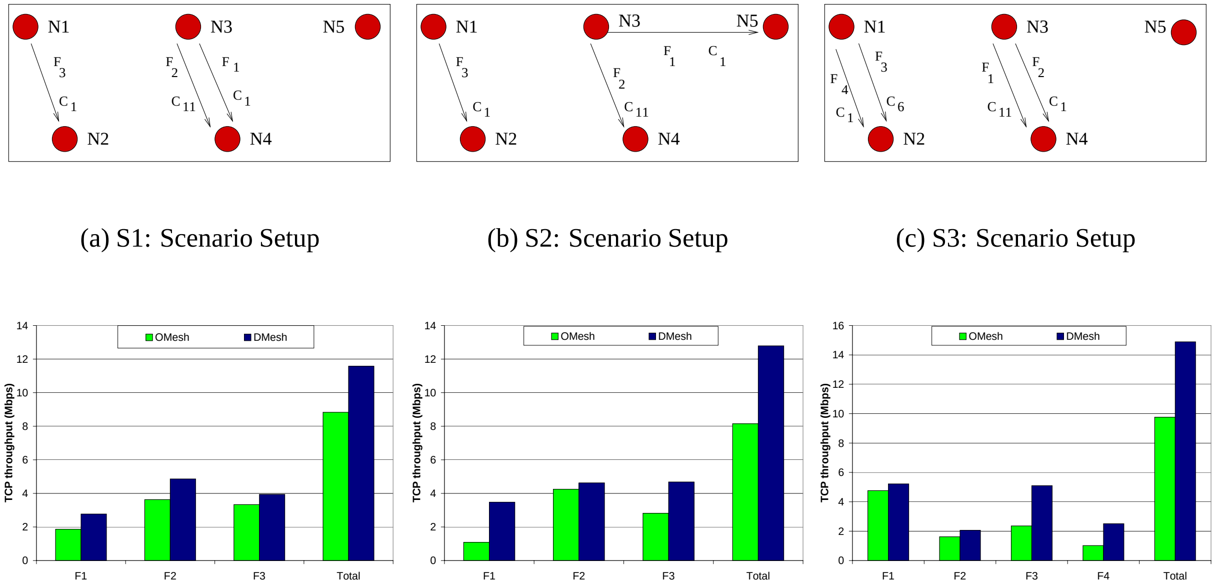
Our testbed consists of 16 wireless mesh routers (small form factor desktops) spread out across three buildings (EE, MSEE and VC (Visitor Center)). We initially deployed Atheros 5212 based 802.11a/b/g cards to evaluate DMesh on 802.11a. However we encountered issues with the Linux drivers for these cards (madwifi) which limited the performance of multiple simultaneous transfers. We thus switched to use Senao Engenius 2511 802.11b wireless cards. Each radio is attached to a 2dBi rubber duck omnidirectional antenna with a low loss pigtail to provide flexibility in antenna placement. Each mesh router runs Linux kernel 2.4.20-8 and the open-source *hostap* drivers are used to enable the wireless cards. IP addresses are statically assigned. The wireless cards we use can support a wide range of power settings (up to 200mW). As in the simulations, the power was adjusted when the cards were used with directional antennas to keep the range similar to the omnidirectional antennas. We used directional antennas in the outdoor part of our network to interconnect the 3 buildings together. The outdoor portion of our testbed is depicted in Figure 14. Two of the buildings (EE and MSEE) have 2 outdoor mesh routers each equipped with 2 or 3 radios while the VC building has a single outdoor mesh router with 2 radios. These outdoor nodes provide high bandwidth interconnection across the indoor networks in each building. We used directional antennas on these outdoor nodes and compared their performance to using omnidirectional antennas. The directional antennas are thex same as those modeled in the simulation. Specifically, we use a 802.11b 45° beamwidth antenna (model HG2412Y from [14]) with the pattern depicted in Figure 5(e).

5.2 Evaluation

In this section, we demonstrate the advantages of the DMesh architecture using experiments on our testbed. We first discuss our experiences in engineering a multi-channel multi-radio network. We then compare the performance of DMesh and OMesh.

Exploiting multiple channels The first challenge in the evaluation was to obtain simultaneous throughput in the multi-channel omnidirectional scenario. As a baseline, we found that a single flow between any two nodes without any other interfering flows achieved a TCP throughput in the range of 4.5-5Mbps using *netperf*. We then ran two simultaneous flows on interfaces I_1 and I_2 on nodes N_1 to N_2 (Figure 14) on maximally separated channels (1 and 11). In this experiment, we found that either one radio achieved close to the full throughput in the single flow case (4.5 Mbps) with the other card being totally starved and achieving less than 0.5Mbps, or both radios achieved around 2-3Mbps each. Thus even though we used external antennas with pigtails, simultaneous throughput was not possible due to interference among the two radios. We then purchased 12 inch extension cables for each radio and further separated the antennas on both N_1 and N_2. This resulted in almost simultaneous throughput (with some loss) being achieved of around 4.3 Mbps each. Thus, antenna separation is critical to enabling multi-channel simultaneous throughput and is required on both the receiver and transmitter. In conclusion, to exploit even maximally frequency separated transmissions required antenna isolation. We also noticed that using power control to have a higher power on one interface adversely affected the other interface despite antenna separation. We thus use similar transmission powers on all interfaces of a single mesh router.

Evaluating the gain from DMesh We used 3 scenarios depicted in Figures 15(a), 15(b), 15(c), each in a different topology and traffic setting, to demonstrate the throughput gains from DMesh, compared to OMesh. In scenario S1, we assume that the number of channels allowed to be used is 2. This is because the Purdue Wireless LAN (Airlink) operates on the remaining orthogonal 802.11b channel in the area and we did not want to disrupt the other users' performance. There are three flows as shown in Figure 15(a). Both OMesh and DMesh automatically assign separate channels on N_3 based on the OCA and C-DCA algorithms. Node N_1 chooses channel 1 to assign to its flow F_3. The performance of the flows in this scenario for OMesh and DMesh are shown in Figure 15(d). The results show that for OMesh, the total throughput is 8.8 Mbps. This occurs because F_1 and F_3 contend with each other



(a) S1: Scenario Setup

(b) S2: Scenario Setup

(c) S3: Scenario Setup

(d) S1: 2 channels, 3 fbws

(e) S2: 2 channels, 3 fbws

(f) S3: 3 channels, 4 fbws

Figure 15: Testbed evaluation results. OMesh denotes a network with omnidirectional antennas running OCA while DMesh denotes a network with practical directional antennas running C-DCA.

causing a reduction in their individual throughputs. We also observe that unfairness exists, since F_3 achieves a higher fraction of the bandwidth than F1. Interestingly, the throughput of F_2 also reduces from its maximum individual throughput although it is on a separate channel. In contrast, using DMesh, F_2 achieves close its individual maximum throughput. The directivity reduces the inter-channel interference from other nodes. In addition F_1 and F_3 improve their throughputs due to increased spatial separation. Overall DMesh provides a 31% increase in TCP throughput. This result is close to the 35% gain observed overall in simulation results for TCP performance.

Note that although DMesh improves the throughput, the practical directional antennas do not completely isolate the transmissions spatially. For example, even in DMesh, F_1 and F_3 do not obtain their maximum individual throughput which is on average 4.5Mbps. Thus, the gain from DMesh can be attributed to a reduction in interference power levels since the interference between F_1 and F_3 is limited to the power received from their sidelobes. This reduced interference is enough to provide significant throughput gains to justify the use of such practical directional antennas in WMNs.

The pattern of the antenna used in this experiment has a very low gain in the backlobes in comparison to the sidelobes and the main lobe (Figure 5(e)). To verify whether further gains are possible if the back lobe is facing the interfering flow, we used scenario 2 (Figure 15(b)). Similar to the previous scenario, we again assume that only 2 channels are available. In this scenario, although F_1 and F_3 are still on the same channel, the antenna used for F_1 is facing the other way with its back lobes towards the antenna for F_3. The results in Figure 15(e) show that OMesh performs slightly worse than in scenario 1 and achieves an aggregate throughput gain of around 8Mbps. One possible explanation is that F_1 now operates over a longer distance (and thus lower SNR) and is starved by F_3 more in comparison to scenario 1. As expected, DMesh now performs better, providing a 57% TCP throughput gain over OMesh, since the transmitting antenna for F_1 only interferes using the backlobe with F_3. The receiving antenna for F_1 whose main lobe is towards F_3 could potentially interfere with F_3. However, the power received and thus the interference are lowered since this antenna is 380 ft away and only transmits short TCP ACK and MAC layer ACK packets. Note that despite the low gain backlobes, the transmitting antenna for F_1 is not totally isolated from the antennas involved in F_3. Nodes N1, N2, and N3 can still ping each other although with a packet loss rate of close to 50%.

In the third scenario (Figure 15(c)), we used all 3 802.11b orthogonal channels in experiments performed late at night so as not to interfere with Airlink users. There are 4 flows depicted in the figure. Again, we find that OMesh provides an aggregate throughput of less than 10Mbps (Figure 15(f)). Flows F_2 and F_4 contend since OMesh runs out of channels. Although DMesh also runs out of channels, it significantly reduces interference power between F_2 and F_4 and provides an aggregate throughput of close to 15Mbps resulting in a 50% TCP throughput gain.

In summary, the testbed evaluation with practical directional antennas shows that DMesh is a viable and useful architecture to improve the performance of wireless mesh network. Although practical directional antennas do not provide *complete* spatial isolation, the reduced interference power provides significant throughput gains. In particular, the reduced interference is useful for multiple-channel networks to obtain close to their maximum throughput potential. Costlier array-based antennas could potentially improve the spatial isolation with tightly controlled sidelobes and backlobes. However, such antennas are still not cost-effective. The cost-benefit tradeoffs in using more advanced antennas in comparison to cheaper versions is an interesting topic of our future research. As a side-note, our experiences show that using practical directional antennas was not inconvenient. Their sizes are reasonable and orienting the antennas perfectly towards each other was not critical to performance.

6 Related Work

Recently, many design issues of WMNs have been studied [17, 2, 18, 19] and many companies are offering products for deploying WMNs [20, 4]. The use of multiple radios/interfaces has been proposed previously as a means to increase the throughput of wireless networks [2, 18, 1, 21, 22]. Multiple beamformed antennas were previously proposed in a position paper [17] for the TAPs architecture. In contrast to DMesh, TAPs proposes to use significantly costlier smart beamforming antennas but may provide higher spatial reuse. We position DMesh as an architecture to quickly and cheaply deploy high throughput mesh networks with minimal cost using commodity antennas combined with channel assignment. The work in [23, 24] uses multiple directional antennas. However, their technique does not exploit frequency separation and is designed for situations where only a single channel is available.

Many MAC layer solutions [25, 26, 27, 28, 29] as well as routing layer solutions [30] have been proposed to exploit multiple channels in wireless networks. The work in [1, 18, 12, 22] applies multi-channel techniques to mesh networks. A recent work [1] proposes a WMN architecture with multiple interfaces and utilizes multiple channels to improve throughput. However, their architecture provides only one degree of separation (frequency) as it uses omnidirectional antennas. In contrast, DMesh combines cheap directional antennas with multiple channels to further improve the performance of WMNs. Two recent works [1, 31] study *joint* routing and channel assignment for multi-radio, multi-channel omnidirectional WMNs. In contrast to these approaches, DMesh performs directional channel assignment and decouples routing from channel assignment (similar to in [2]). The work in [2] computes routes for a network in which channel assignment has been previously done; while DMesh computes channel assignments for a routing tree that has been physically formed due to placement of antennas.

The use of beamforming antennas have been proposed for mobile ad hoc networks [32, 33, 11, 10, 34, 33]. Channel assignment has also been studied in the context of cellular networks [35]. Work has also been done on combining channel assignment and sectored separation of users in cellular networks (e.g. [36]). However, the multi-hop routing and non-existence of infrastructure devices such as base stations make WMNs very different from cellular networks. Many cellular network techniques cannot be directly applied to WMNs due to these different characteristics.

A number of issues, such as load balancing, gateway discovery, addressing, tree construction and fairness, arise in provisioning Internet access via multi-hop wireless networks. Issues related to addressing and gateway discovery are discussed in [37]. Fairness issues in WMNs are discussed in [19].

7 Conclusions and Future Work

In this paper, we proposed DMesh, a novel architecture for improving the performance of wireless mesh networks. Such an architecture exploits multi-radio, multi-channel nodes in the mesh network, where each interface is equipped with a practical directional antenna. We also proposed a distributed algorithm to perform routing and directional channel assignment in the DMesh architecture. By exploiting the spatial separation offered by directional antennas and the frequency separation offered by multiple non-overlapping channels, our proposed architecture allows more concurrent transmissions than an omnidirectional, multi-radio, multi-channel mesh network, and as a result achieves higher throughput. In particular, simulation results and evaluation on a mesh network testbed show that, compared to the omnidirectional, multi-radio, multi-channel mesh network, our proposed architecture improves packet delivery ratio and throughput and drastically lowers average per-packet delay. In conclusion, DMesh provides higher throughput for WMNs while remaining cost-effective and easy to deploy. DMesh can be used to naturally extend current widely deployed single radio mesh networks.

References

[1] A. Raniwala and T. Chiueh, “Architectures and algorithms for an IEEE 802.11-based multi-channel wireless mesh network,” in *Proc. of IEEE INFOCOM*, March 2005.

[2] R. Draves, J. Padhye, and B. Zill, “Routing in multi-radio, multi-hop wireless mesh networks,” in *Proc. of ACM MobiCom*, September 2004.

[3] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, “Link-level measurements from an 802.11b mesh network,” in *Proc. of SIGCOMM*, August 2004.

[4] Bel Air Networks, “http://www.belairnetworks.com.”

[5] R. Chandra, L. Qiu, K. Jain, and M. Mahdian, “Optimizing the placement of internet taps in wireless neighborhood networks,” in *Proc. of ICNP*, October 2004.

[6] T. Clausen, P. Jacquet, C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L.Viennot, “Optimized link state routing protocol (OLSR),” RFC 3626, October 2003. [Online]. Available: http://ietf.org/rfc/rfc3626.txt

[7] D. S. J. D. Couto, D. Aguayo, J. C. Bicket, and R. Morris, “A high-throughput path metric for multi-hop wireless routing.” in *Proc. of ACM MobiCom*, 2003.

[8] S. Ramanathan, “A unified framework and algorithm for channel assignment in wireless networks,” *Wirel. Netw.*, vol. 5, no. 2, 1999.

[9] QualNet, "http://www.scalable-networks.com."

[10] M. Takai, J. Martin, A. Ren, and R. Bagrodia, "Directional virtual carrier sensing for directional antennas in mobile ad hoc networks," in *Proc. of ACM MobiHoc*, June 2002.

[11] R. Choudhury, X. Yang, R. Ramanathan, and N.Vaidya, "Using directional antennas for medium access control in ad hoc networks," in *Proc. of ACM MobiCom*, September 2002.

[12] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," in *Proc. of ACM Mobicom*, September 2004.

[13] J. Jun, P. Peddabachagari, and M. Sichitiu, "Theoretical maximum throughput of ieee 802.11 and its applications," in *Proc. of NCA*, 2003.

[14] HyperLink-Technologies, "Home page http://www.hyperlinktech.com."

[15] MAXRAD, "Home page http://www.maxrad.com."

[16] M. Mesh@Purdue, "Home page http://www.engineering.purdue.edu/MESH."

[17] R. Karrer, A. Sabharwal, and E. Knightly, "Enabling large-scale wireless broadband: the case for TAPs," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 1, pp. 27–32, 2004.

[18] A. Raniwala, K. Gopalan, and T. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *ACM MC2R*, vol. 8, no. 2, April 2004.

[19] V. Gambiroza, B. Sadeghi, and E. W. Knightly, "End-to-end performance and fairness in multihop wireless backhaul networks," in *Proc. of ACM MobiCom*, 2004.

[20] Mesh Networks, "http://www.meshnetworks.com."

[21] P. Bahl, A. Adya, J. Padhye, and A. Wolman, "Reconsidering wireless systems with multiple radios," *ACM MC2R*, October 2004.

[22] P. Kyasanur and N. H. Vaidya, "Routing and interface assignment in multi-channel multi-interface wireless networks," in *Tech. Rep. UIUC*, October 2004.

[23] B. Raman and K. Chebrolu, "Revisiting mac design for an 802.11-based mesh network," in *Proc. of ACM HotNets*, November 2004.

[24] ——, "Design and evaluation of a new mac protocol for long-distance 802.11 mesh networks," in *Proc. of ACM Mobicom*, August 2005.

[25] Y. Liu and E. Knightly, "Opportunistic fair scheduling over multiple wireless channels," in *Proc. of IEEE INFOCOM*, 2003.

[26] J. So and N. Vaidya, "Multi-channel mac for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver," in *Proc. of ACM MobiHoc*, 2004.

[27] A. Tzamaloukas and J. J. Garcia-Luna-Aceves, "A receiver-initiated collision-avoidance protocol for multi-channel networks," in *Proc. of IEEE INFOCOM*, 2001.

[28] A. Nasipuri and S. Das, "A multichannel csma mac protocol for mobile multihop networks," in *Proc. of IEEE WCNC*, 1999.

[29] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu, “A new multi-channel mac protocol with on-demand channel assignment for multi-hop mobile ad hoc networks,” in *Proc. of I-SPAN*, 2000.

[30] J. So and N. H. Vaidya, “A routing protocol for utilizing multiple channels in multi-hop wireless networks with a single transceiver,” in *Tech. Rep. UIUC*, October 2004.

[31] M. Alicherry, R. Bhatia, and L. E. Li, “Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks,” in *Proc. of ACM Mobicom*, August 2005.

[32] Y. Ko and N. Vaidya, “Medium access control protocols using directional antennas in ad hoc networks,” in *Proc. of IEEE INFOCOM*, March 2000.

[33] R. Ramanathan, “On the performance of ad hoc networks with beamforming antennas,” in *Proc. of ACM MobiHoc*, October 2001.

[34] A. Nasipuri, J. Mandava, H. Manchala, and R. Hiromoto, “On-demand routing using directional antennas in mobile ad hoc networks,” in *Proc. of IEEE ICCCN*, October 2000.

[35] I. Katzela and M. Naghshineh, “Channel assignment schemes for cellular mobile telecommunication systems: a comprehensive survey,” in *IEEE Personal Communications*, June 1996.

[36] J.-L. Pan, S. S. Rappaport, and P. M. Djuric, “Multibeam cellular communication systems with dynamic channel assignment across multiple sectors,” *Wirel. Netw.*, vol. 5, no. 4, pp. 267–278, 1999.

[37] Y. Sun, E. Belding-Royer, and C. Perkins, “Internet connectivity for ad hoc mobile networks,” *Int. Journal Wir. Inf. Netw.*, 2002.

8 Appendix

8.1 Directional Channel Assignment is NP-hard

Here we show that the directional channel assignment problem stated in Section 2.3.1 is NP-hard. In [18], the authors prove that the omnidirectional channel assignment problem in a single collision domain network is NP-hard, by reducing the *Multiple Subset Sum Problem* to the channel assignment problem. In case of directional antennas, the channel assignment problem is same as in [18], except that the collision domain is smaller due to the presence of directional antennas. In other words, the single collision domain used in [18] can be broken into smaller collision domains. However, in each of these collision domains, we can still reduce the Multiple Subset Sum Problem to the channel assignment problem, following the same methodology as in the omnidirectional case. Since the Multiple Subset Sum Problem is NP-hard, the directional channel assignment problem is also NP-hard.

8.2 Pseudocode for Channel Assignment Algorithms

Table 3: Pseudo code for checking if a neighbor node is in sender's cone of transmission.

```
Algorithm CHECK_IN_CONE()
Input: POSITION OF SENDER $s$, POSITION OF RECEIVER $r$, POSITION OF NEIGHBOR NODE $i$

$\theta = atan((r.y - s.y)/(r.x - s.x))$   // calculate angle of receiver with respect to sender.
//   calculate sender's cone of transmission.
$\theta_u = \theta + (BEAMWIDTH/2)$
$\theta_l = \theta - (BEAMWIDTH/2)$
$\theta_i = atan((i.y - i.y)/(i.x - i.x))$   // calculate angle of neighbor node with respect to sender.
if ($\theta_l \leq \theta_i \leq \theta_u$)   // check if neighbor node is in sender's transmission cone.
    return TRUE
return FALSE   // neighbor node is not in sender's transmission cone.
```

Table 4: Pseudo code for computing bad channel list.

```
**Algorithm** COMPUTE_BCL()
Input: CHANNEL USAGE MAP map, CURRENT NODE node

for each node $i \in map$    // go through all nodes in interference range.
    for each destination $D_j \in map[i].D$    // go through all destinations in destination vector D.
        if (CHECK_IN_CONE($i$, $map[i].D_j$, $node$) == TRUE)    // if $node$ is in transmission cone of nbr. node $i$ towards node $D_j$.
            insert channel $map[i].C_j$ in BCL    // channel $C_j$ used by node $i$ while transmitting to node $D_j$ is bad for $node$.
return BCL
```

Table 5: Pseudo code for OCA. Selects a channel between current node and child node.

```
**Algorithm** OCA()
Input: CHANNEL USAGE MAP map, CURRENT NODE node, CHILD NODE child

$bcl_{final}$ = COMPUTE_BCL($map$, $node$) $\bigcup child.bcl$    // find bad channels - union of current node's and $child$'s bad channel list.
while(1)
    curr_channel = GetFreeChannel($bcl_{final}$, $node$)    // look for a channel not marked as bad.
    if (curr_channel == -1)    // could not find a free channel.
        select channel $c \in map$ such that $\forall c_i \in map\ load(c) < load(c_i)$    // assign the least loaded channel.
        assign $c$ to $child$    // send ASSIGN message to $child$ on CONTROL interface.
        set $c$ on $child.interface$    // use the selected channel on the interface pointing to the $child$.
        add channel $c$ to $bcl_{final}$    // update bad channel list.
        break
      // found a free channel - check if you can use it.
    usable = TRUE
    for each node $i \in map$
        if (node $i \in CHILDREN(node)\ \vee\ i == node$) continue
        if (curr_channel $\in map[i].C$)    // if node $i$ uses curr_channel.
            usable = FALSE    // node $i$ is an interferer for curr_channel - curr_channel cannot be used.
    if (usable == TRUE)
      // found a channel that can be used.
        assign curr_channel to $child$    // send ASSIGN message to $child$ on CONTROL interface.
        set curr_channel on $child.interface$    // use the selected channel on the interface pointing to the $child$.
        add curr_channel to $bcl_{final}$    // update bad channel list.
        break
```

Table 6: Pseudo code for C-DCA. Selects a channel between current node and child node.

```
**Algorithm** C-DCA()
Input: CHANNEL USAGE MAP map, CURRENT NODE node, CHILD NODE child

$bcl_{final}$ = COMPUTE_BCL($map$, $node$) $\bigcup child.bcl$   // find bad channels - union of current node's and $child$'s bad channel list.
$bcl_{final}$ = $bcl_{final} \bigcup bcl_{measurement}$   // add overheard transmission information to bad channel list.
while(1)
    curr_channel = GetFreeChannel($bcl_{final}$, $node$)   // look for a channel not marked as bad.
    if (curr_channel == -1)   // could not find a free channel.
        select channel $c \in map$ such that $\forall c_i \in map\ load(c) < load(c_i)$   // assign the least loaded channel.
        assign $c$ to $child$   // send ASSIGN message to $child$ on CONTROL interface.
        set $c$ on $child.interface$   // use the selected channel on the interface pointing to the $child$.
        add channel $c$ to $bcl_{final}$   // update bad channel list.
        break
    // found a free channel - check if you can use it.
    usable = TRUE
    for each node $i \in map$
        if (node $i \in CHILDREN(node)\ \vee\ i == node$) continue
        if (curr_channel $\in map[i].C$)   // if node $i$ uses curr_channel.
            // check if node $i$ is in sender's or receiver's transmission cone.
            if (CheckInCone($node$, $child$, $i$) == FALSE $\wedge$ CheckInCone($child$, $node$, $i$) == FALSE)
                continue   // node $i$ is not an interferer.
            usable = FALSE   // node $i$ is an interferer for curr_channel - you cannot use it.
    if (usable == TRUE)
    // found a channel that can be used.
        assign curr_channel to $child$   // send ASSIGN message to $child$ on CONTROL interface.
        set curr_channel on $child.interface$   // use the selected channel on the interface pointing to the $child$.
        add curr_channel to $bcl_{final}$   // update bad channel list.
        break
```

Table 7: Pseudo code for A-DCA. Selects a channel between current node and child node.

```
Algorithm A-DCA()
Input: CHANNEL USAGE MAP map, CURRENT NODE node, CHILD NODE child

bcl = COMPUTE_BCL($map$, $node$)    // find $node$'s bad channels
for each channel $c_i$ {   // go through all channels to see if you can use them.
    selected_channel = -1
    usable = TRUE
    if ($c_i \in$ bcl)    //channel not free for sender, see if you can still use it.
          // check if somebody who marked $node$ is conflicted if $node$ transmits.
        for each node $i \in map$ {
            for each node $j \in map[i].D$ {
                if (CheckInCone(i, j, node) == TRUE)    // if transmission $i \rightarrow j$ interferes with $node$.
                    if (CheckInCone($node$, $child$, $i$) == TRUE)    // node $i$ is in sender's transmission cone.
                        usable = FALSE
                        break
            }
            if (usable == FALSE)    // $node$ cannot use channel $c_i$.
                    break
        }
    if (usable == FALSE) continue channel conflict on sender's side - try another channel.
      // else clear from sender side, check on receiver side
    if ($c_i \in child.bcl$)    // channel not free for receiver, see if you can still use it.
          // check if somebody who marked $child$ is conflicted if $child$ transmits.
        for each node $i \in map$
            for each node $j \in map[i].D$ {
                if (CheckInCone(i, j, child) == TRUE)    // if transmission $i \rightarrow j$ interferes with $child$.
                    if (CheckInCone($child$, $node$, $i$) == TRUE)    // node $i$ is in receiver's transmission cone.
                        usable = FALSE
                        break
            }
            if (usable == FALSE)    // $node$ cannot use channel $c_i$.
                    break
        }
    if (usable == FALSE) continue    // channel conflict on receiver's side - try another channel.
    assign channel $c_i$ to $child$    // $node$ can use channel $c_i$.
    selected_channel = $c_i$
    break
}
if (selected_channel == -1)    // no channel has been assigned.
    select channel $c \in map$ such that $\forall c_i \in map\ load(c) < load(c_i)$    // assign the least loaded channel.
    selected_channel = $c$
assign selected_channel to $child$    // send ASSIGN message to $child$ on CONTROL interface.
set selected_channel on $child.interface$    // use the selected channel on the interface pointing to the $child$.
add selected_channel to bcl and to $child's$ bcl    // update bad channel list for sender and receiver.
```