On-demand channel switching for multi-channel wireless MAC protocols

Priyank Porwal and Maria Papadopouli
Department of Computer Science
University of North Carolina at Chapel Hill
{porwal, maria}@cs.unc.edu

***Abstract* – We propose and analyze the on-demand channel switching (ODC), a novel broadcast-based medium access control (MAC) protocol for *ad-hoc* wireless networks with *multiple channels* at each host. The ODC performs an on-demand, dynamic, channel selection based on the traffic conditions of the channels and communication pattern of each host. A host stays on a channel as long as its traffic share on that channel is above a certain threshold, below which it switches to another channel. It aims on reducing the unnecessary receives and channel switches, while still being able to send and receive legitimate traffic. This can enhance the bandwidth utilization, reduce packet delays, and increase energy savings. ODC does not require any specialized hardware. We evaluate the ODC with extensive simulations and compare it with the IEEE 802.11 and another related MAC protocol (MMAC). We found that the flow distribution among hosts has a great impact on its performance. For example, in networks in which a host communicates with only one other host at a given time, there is an up to 180% increase in the aggregate throughput compared to the IEEE 802.11, and a 15-35% increase compared to the MMAC. On the other hand, networks with hosts with multiple flows to different hosts simultaneously tend to favor the IEEE 802.11 over the ODC and MMAC.**

I. Introduction

The IEEE 802.11 [1] has been widely deployed to provide wireless LAN access. Its MAC protocol is responsible for coordinating access to the shared radio channel. The IEEE 802.11 employs a CSMA-CA based approach along with virtual carrier sensing and channel reservation techniques. In CSMA-CA, the channel is always sensed before any transmission. If the channel is sensed busy, the transmitter waits for it to become idle and goes through a random back-off period before retrying. The random back-off period ensures fairness among transmitting hosts and also reduces the probability of collisions. In addition, transmitting and receiving hosts exchange *request-to-send* (RTS) and *clear-to-send* (CTS) control frames to reserve the channel before transmitting data packets. This effectively performs a virtual carrier sensing at the receiver. Successful exchange of these control frames also reserves the channel temporarily for the data transmission. Although, the IEEE 802.11 standard provides multiple channels, its MAC layer is designed for sharing a single channel. It performs well under low-load conditions, but under heavy-load conditions, it causes bandwidth under-utilization due to collisions and back-offs. These problems motivate the spatial reuse of the wireless medium. Transmission power control, directional antennae, and multiple channels are the most popular techniques for spatial reuse.

We propose the ODC, a novel IEEE 802.11-based MAC protocol for multi-channel wireless ad hoc LANs. It does not require any specialized hardware and allows hosts with a single transceiver to utilize multiple channels by *dynamically* switching from one channel to another. The decision of a host to switch is based on the *traffic conditions* of the channels, its *communication pattern* with other hosts, and its traffic share. A host performs passive measurements to estimate the traffic conditions of all channels and its traffic share on its local channel. It remains on a channel as long as its traffic share exceeds a certain threshold, below which it switches to another channel. With channel switching, the ODC aims to increase the throughput and reduce collisions, back-off periods, and unnecessary traffic. This can have a dominant impact on the energy consumed by wireless network interfaces.

The main contribution of this paper is the design of this novel MAC protocol and its performance analysis. It introduces a novel paradigm, namely, the autonomous manner with which devices monitor channels and decide to switch. We analyze the performance of the ODC via simulations and compare it with the IEEE 802.11 and another multi-channel MAC protocol, the MMAC [4]. Depending on the communication pattern of the hosts, their performance can vary dramatically. We distinguish various communication patterns based on the traffic model and flow distribution among hosts to simulate networks that could potentially use such protocols. For example, networks in which a host communicates with only one other host at a given time (one-to-one schemes) are good candidates for multi-channel protocols, whereas single-channel protocols perform better in networks where hosts have multiple flows to other hosts simultaneously (such as star or clique schemes). We envision several applications that operate in a wireless LAN and use the one-to-one communication paradigm (such as devices in an office or car that form a personal area network and share/synchronize owner's information/files, collaborators that communicate, share a presentation or play a game). In the one-to-one scheme, the ODC exhibits an up to 180% increase in the aggregate throughput compared to the IEEE 802.11 and a 15-35% increase compared to the MMAC at peak load. On the other hand, star and clique schemes tend to favor the IEEE 802.11, although the difference in their performance is less prominent.

The rest of the paper is organized as follows. Section II discusses related work on multi-channel MAC protocols and Section III describes the ODC. We present the simulation models and performance analysis results in Sections IV and V, respectively. Section VI summarizes our main conclusions and future work plans.

II. Related work

Several researchers have proposed MAC protocols that use multiple channels. We classify these approaches into different categories depending on the channel assignment and usage, and the availability of multiple transceivers. We distinguish two main approaches regarding the classification based on the channel assignment. The first dedicates a channel for control packets and uses the remaining channels for data packets, whereas the latter treats all channels identically. Based on the availability of multiple transceivers, there are two main trends, namely, the multiple-transceivers with one transceiver per channel, and the use of a common transceiver for all channels. Unlike the multi-transceiver case, the common transceiver operates on a single channel at any given point of time. [1] In the following paragraphs, we briefly discuss these multiple-channel MAC protocols and their main results. Table 1 summarizes their classification.

Research/protocol	**Reserved control channel?**	**Num. of transceivers**	**Channel selection**
Li *et al.* [5]	Yes	Single	Explicit negotiation
DPC [6]	Yes	Multiple	Explicit negotiation
Nasipuri *et al.* [7]	No	Multiple	*Soft channel reservation* (last used channel), Random
RBCS [8]	Yes	Multiple	Explicit negotiation based on interference at receiver
Chaudhuri *et al.* [9]	No	Single	*Home channel* (default based on MAC address)
MMAC [4]	Yes	Single	Explicit negotiation, Least selected channel, Random
ODC	No	Single	Arrival and departure broadcasts, Cyclic

Table 1: Classification of different multiple-channel MAC protocols

[1] Recently, manufacturers, such as Engim [10] and D-Link [11], have launched access points (AP) that use multiple channels simultaneously and claim to provide high bandwidth wireless networks.

Li *et al.* [5] assume that each node has a single half-duplex transceiver and that the number of available channels is the same as the number of nodes in the network. One of the channels is reserved as the common access control channel and all others are used as traffic channels. The traffic channel reservations for data packet transmissions take place on the control channel through RTS/CTS exchanges. They analyzed the performance of the modified IEEE 802.11 by simulating two network topologies, namely, the line and grid topologies. In the line topology, all nodes are in a single straight line, whereas in the grid topology, nodes are placed at the corners of a grid with hexagonal cells. Compared with the per-node throughput of the single-channel case, there is an increase of 47.8% in line, and 139% to 163% in grid topologies. The end-to-end delay improves marginally for line networks but becomes significant for grid networks.

The dynamic private channel protocol (DPC) [6] assumes hosts have multiple transceivers and are thus capable of accessing multiple channels at the same time. Each host has as many transceivers as the number of channels. Like [5], one channel is reserved as the control channel, on which potential senders and receivers negotiate the channel they will use for their communication from. The RTS and *reply-to-RTS* exchange happens on the control channel, whereas the CTS and data/ACK exchanges on the traffic channels. Since all hosts listens to the control channel continuously, they know the status of all channels. Their simulations show that the channel utilization, *i.e.*, ratio of throughput to offered load, increases as the number of channels increases up to 4, but drops beyond 4 channels. This degradation is due to the blocking of flows that occurs when a sender and receiver have already agreed to use different channels with other hosts.

Nasipuri *et al.* [7] propose a protocol where each host can concurrently listen to each channel. They introduce the *soft channel reservation*, where a transmitter tries to reuse the channel of its last successful data transmission. They simulate the protocol with soft channel reservation and random channel assignment. They consider the single-user and multi-user scenarios. In the single-user case, hosts receive only one packet at a time, while in multi-user, hosts receive multiple packets at the same time, each on a different channel. They show that the throughput is significantly higher in the multi-user case and the *soft channel reservation* performs better than random channel selection. However, a large number of channels causes an unacceptably high transmission time.

Jain *et al*. [8] propose the *receiver-based channel selection* (RBCS). Like [5], they have a control channel for channel negotiation using RTS/CTS exchange, and multiple data channels for data and ACK frames. The clearest channel is chosen based on the interference power sensed at the receiver. They assume that hosts can carrier sense on all the channels for incoming transmissions. However, at a given time, only one packet can be transmitted on any channel. Their simulations show an up to 50% improvement in throughput over the IEEE 802.11 for stationary networks. The improvement is more significant under heavy load conditions, when the contention for channels is high. In mobile networks, they observe higher average packet delays.

Chaudhuri *et al.* [9] propose a simple rendezvous-based protocol in which each host is associated with a particular channel, i.e., its *home channel*, based on its MAC address. Every host waits for all incoming traffic on this home channel. A transmitting host switches temporarily to their receiver's home channel, and returns to its own home channel after completing the transmission. They show that the proportional throughput gain of their protocol *vs*. the IEEE 802.11 decreases with the increase in the ratio of number of transmitting hosts to total hosts.

The MMAC, proposed by Vaidya *et al*. [4], most closely relates to our work. They use the *ad-hoc traffic indication message* (ATIM) window and a control channel for channel negotiation. Each host maintains a *preferred channel list* (PCL) with a high, medium, or low preference attached to every channel. At the start of an ATIM window, all hosts switch to the common control channel. During the ATIM window, senders and receivers negotiate channels using a three-way handshake messaging that indicates also their respective PCLs. At the end of an ATIM window, all hosts switch to their respective selected channel and begin normal RTS-CTS-data-ACK cycles for data transmission. The control channel is also used for data transmission outside the ATIM window. They show a significant improvement over the IEEE 802.11, both in terms of aggregate throughput and average packet delay. However, the MMAC favors scenarios where hosts are involved in only one flow at a time. When hosts are involved in multiple flows, MMAC performs poorly due to the *head of line blocking* problem (i.e., a sender cannot transmit to a receiver because they are on different channels).

Compared to these approaches, the ODC assumes all channels to be identical in all respects, with no notion of control channel. In addition, our protocol does not require multiple transceivers or any other specialized hardware. It is designed for hosts equipped with the most commercially viable wireless network interfaces, which have a single half-duplex transceiver. Section V closely compares the ODC with the IEEE 802.11 and MMAC.

III. The ODC protocol

The ODC allows a host to dynamically switch to different channels depending on their traffic condition, its communication pattern and bandwidth utilization at its current channel. It extends the basic IEEE 802.11 functionality by a simple host discovery process and some new message types. However, it does not support its PSM mode (e.g., does not have any ATIM concept). Unlike MMAC, we do not assume that channel switch occurs instantaneously. Because of the time and energy overhead associated to channel switch, we try to reduce the number of channel switches. A host remains on its current channel until its condition becomes unacceptable or a potential receiver is on a different channel. In either case, the host switches to an appropriately selected channel after broadcasting its departure announcement. After the channel switch, it broadcasts its arrival announcement on its new channel. The following sections describe the ODC in more detail.

A. Message types

The ODC introduces two new messages to the existing IEEE 802.11 control messages to assist in host discovery, namely the departure (DEP) and arrival (ARR) announcements. Their destination address is the MAC broadcast address. The DEP message also includes the MAC address of the potential receiver. A host that plans to send some data packets, first broadcasts a DEP message, in which, it specifies the MAC address of the receiver. This MAC address field remains empty when a host broadcasts DEP to indicate that it plans to switch channels for reasons other than to communicate with a host.

The ARR message summarizes the traffic conditions that the sender observes on its previous channel during the last channel observation window (T_{ch}). It includes the total number of bytes transmitted, received, and overheard, and total number of hosts observed to be present on that channel. It also indicates the duration of these observations. The overheard traffic of a host is the one for which it is neither the source nor the destination. The information included in ARR can give a rough estimate of the traffic rate in that channel during that time period.

B. Traffic entries

Each time a host receives an ARR message from a host (e.g., host *h*), it maintains a traffic history of the sender (*h*)'s last channel. Specifically, it logs the entry <*c, r, t*>, where *c* is the host *h*'s last channel, *r* the traffic rate estimated from the information included in that announcement, and *t* the time when it was received.

C. Host discovery

Each participating host maintains a *channel vector*, in which it records the current channel of all other hosts. Any node that receives a DEP packet records the destination channel field as the current channel of the sender of the packet. Any node that receives an ARR packet records its own channel as the current channel of the sender of the packet. Whenever a host needs to send data, it looks up the destination's current channel in its channel vector, and switches to that channel, if required. A channel vector is not always complete or accurate. A host may not receive all the ARR and DEP due to either packet loss or announcements made on different channels.

D. Traffic rate estimation in different channels

The traffic estimation is based on ARR announcements and observations of the local host on the traffic of its current channel. Each host counts the total number of transmitted, received, and overheard bytes (B_{total}), total number of transmitted and received bytes (B_{trxrcv}), and number of hosts on its current channel (N_{hosts}) during the last T_{ch} s. Using this measurements, it estimates the traffic rate (e.g., r_L) of its current channel (e.g., c_L), and records this entry ($<c_L,r_L,t_L>$, where t_L is the time of this measurement), as in the case of ARR announcements. Each host maintains a channel history with all these entries. Considering all the traffic entries received during the last *channel history window* (H_{ch}), it computes the *mean traffic rate* of each channel which is the average of all traffic rates of that channel.

E. Channel switching process

When a host receives a DEP broadcast in which the *receiver field matches its own MAC address*, it learns that the source is switching to a new channel to communicate with it. The receiver then switches channel to follow the source. The switching process at a local host is also triggered each time the host *intends to transmit* or *has overheard a data packet*. Before each packet transmission, the sender looks up the channel of the receiver in its channel vector. If it differs from its own current channel, it decides to switch to that channel. The sender, then, sends an RTS to the receiver. After each RTS, the sender waits for a CTS from the receiver. If after R_1 RTS retransmissions, it has not received a CTS message from the receiver, the sender switches to the next channel. Similarly, on the new channel, if after R_2 RTS broadcasts, the sender still has not received any CTS from the receiver, it switches to the next channel. There, it rebroadcasts at most R_3 RTS messages. In the case of no CTS response, the sender drops the DATA packet and switches to its original channel. By next channel, we mean the next channel in cyclic order, *i.e.*, (current channel + 1) modulo N_{ch}, where N_{ch} is the number of channels.

Whenever a host overhears a packet, it checks the following conditions in this order. Only when all of them are satisfied, the host decides to switch to its next channel.

1. To prevent from frequent channel switches, the ODC requires hosts to remain continuously on the same channel for *at least* T_{ch} s.
2. There has been at least T_{ch} s since the last time these conditions were checked. If this condition is true, the host sets its arrival time on its current channel to the current time. It also updates the channel history of its current channel with all the traffic observations during the last T_{ch} time window.
3. The time since its last arrival on the new channel is at least $(N_{ch} + 1) * T_{ch}$ s. This condition prevents rapid channel scans, when all channels are equally loaded.
4. To prevent receivers from unnecessary channel switches, when they have a fair share of traffic on their current channel, the ODC computes the *traffic share* $S_{traffic}$ on its current channel. It considers the switch only when it is less than S_{thresh}.

$$S_{traffic} = \frac{B_{trace}}{2 * B_{total} / N_{hosts}}$$

5. The estimated traffic rate of the next channel is at least G_{thresh} % less than the estimated traffic rate on the current channel. As mentioned earlier, each host maintains entries with traffic related information for each channel in its channel history. To estimate the traffic of the next channel, it considers all such entries regarding that channel received during the last H_{ch} s.

When a host decides to switch channels, it broadcasts a DEP announcement at some randomly selected time during the next contention window. After the DEP broadcast, the host immediately switches to the new channel and broadcasts an ARR announcement at some randomly selected time during the next contention window.

IV. Simulation models

We used the ns-2 [2] network simulator with the CMU wireless extension [3] and simulated four scenarios as representatives of different classes of future ad-hoc wireless network applications. These classes are parameterized based on traffic characteristics and flow distribution schemes among hosts. More specifically, we consider a one-to-one scheme with constant bit rate (CBR) traffic, a one-to-one scheme with simulated speech traffic, a star scheme with CBR traffic simulating a sensor or personal area network, and a clique with extreme-value on/off traffic simulating a game (Figure 1).

We use 3 non-overlapping channels of 1 Mbps each, the two-ray ground radio propagation model, and Ad-hoc On-Demand Distance Vector (AODV) routing protocol. The nodes simulate the participant hosts and were placed randomly with a uniform distribution within a 100mx100m grid. They remained stationary throughout the simulations. Each simulation run lasts for 600 seconds. For the performance analysis plots, each data point is an average of 30 runs. We varied the number of nodes, types of traffic generators, and packet rate to study the performance of the ODC under different traffic load conditions. Table 2 summarizes the simulation parameters.

Parameter	Value	Parameter	Value
Number of nodes	2-32	Number of simulations	30
Node placement grid	100 m * 100 m	Simulation duration	600 s
Node placement pattern	Random	Traffic generation start time	0-1 s
Node mobility	Stationary	Traffic generation end time	600 s
Radio propagation model	Two-ray ground	Channel switching time (T_{switch})	100 µs
Radio transmission range	250 m	Traffic share threshold (S_{thresh})	80 %
Ad-hoc routing protocol	AODV	Traffic gain threshold (G_{thresh})	20 %
Number of channels (N_{ch})	3	Channel history window (H_{ch})	30 s
Bandwidth per channel (B_{ch})	1 Mbps	Channel traffic window (T_{ch})	3 s
		RTS retransmission limits (R_1, R_2, R_3)	7, 4, 3

Table 2: Simulation parameters with their values

For the aggregate throughput and average packet delay, we only use the successfully delivered packets. These values represent the overall network throughput and delay when all nodes and channels are considered. Efficiency index is computed by taking the ratio of data traffic and total traffic, on all channels combined for the entire simulation period. Since the IEEE 802.11 uses a single channel, its simulation results show the aggregate throughput obtained using a single channel. On the other hand, the ODC and MMAC use N_{ch} channels and hence their performance results correspond to a throughput aggregated over all N_{ch} channels. We discuss all the scenarios in more detail in the following subsections.

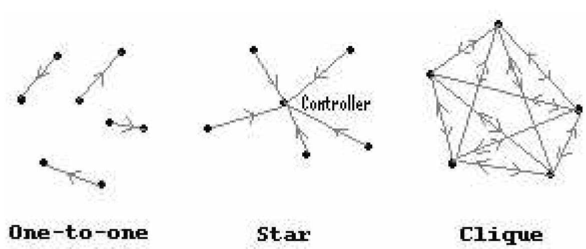

Figure 1: Distribution of flows across hosts

	Location	Scale
Packet size	120	36
Packet inter-arrival time	0.055	0.006

Table 3: Extreme-value parameters for game traffic

A. One-to-one flow distribution with CBR traffic

In one-to-one, each node is involved in a single flow, either as a source or destination. Half the nodes are selected as sources and the other half as destinations with one-to-one mapping between sources and destinations. Each source generates CBR traffic with a fixed packet size of 512B. We vary the number of nodes from 2 to 32 (in powers of 2), and packet arrival rate (pps) on each flow from 1 pps to 512 pps.

B. One-to-one flow distribution with simulated speech traffic

Each traffic source generates simulated speech traffic. The speech traffic follows an exponential distribution with alternating ON (talk spurt) and OFF (silence) periods of mean lengths 1.004s and 1.587s, respectively [13]. During the talk spurts, packets of fixed size, 512B are generated at a constant bit-rate. We vary the number of nodes from 2 to 32 (in powers of 2), and the packet arrival rate during talk spurts on each flow from 1 pps to 512 pps.

C. Star flow distribution with CBR traffic and central controller as in sensor or personal area networks

In star schemes, the central node is treated as a controller in a wireless LAN. All other nodes have a flow to the controller. The traffic sources on these nodes generate CBR traffic. We vary the number of nodes from 2 to 16 in powers of 2, and the packet arrival rates for each flow from 1 pps to 256 pps. This could correspond to a sensor or a personal network; the central node is a controller, server, or an AP and other nodes are sensors or devices of the personal area network.

D. Clique flow distribution with extreme-value on/off traffic as in peer-to-peer games

We also attempt to simulate a peer-to-peer game traffic where every player communicates with every other (e.g., the *Age of Empires)*. Farber [12] shows that the traffic between players in most common network games follows an *extreme-value distribution* [14], both in packet size and their inter-arrival time. Each node corresponds to a player. We vary the number of nodes from 2 to 8. Every node has a flow to every other node and its traffic follows the extreme-value distribution (Table 3 shows its parameters).

V. Performance analysis via simulations

A general observation is that one-to-one networks in which a host typically communicates with only one other host at a given time are good candidates for multi-channel protocols, whereas networks with hosts that simultaneously communicate extensively with several hosts (many-to-many) do not perform well. In one-to-one, pairs of sender and receiver can be evenly divided among the available channels, thus reducing the channel switches. Using the ODC and MMAC, the throughput is approximately 2.5-3 times of that of the IEEE 802.11, and the packet delays are relatively much smaller (Figures 2 and 3). On the other hand, star patterns follow an opposite trend (Figure 4). There, all hosts communicate with a central host. The ODC would perform better if all hosts stayed on the same channel. Multi-channel support and channel switches do not particularly help much. The IEEE 802.11 performs better in terms of throughput. In ODC and MMAC, the average packet delay is smaller than in IEEE 802.11, because channel switches reduce collisions on the channel of the central host. Yet, the gain in packet delays does not justify the loss in throughput. With clique-like patterns, channel changes may result in smaller throughput and longer packet delay (Figure 5). We discuss the results in more detail in the following paragraphs.

One-to-one flow distribution

In one-to-one patterns (Figures 2 and 3), the ODC exhibits 20-25% smaller average packet delay than the MMAC, irrespective of the network size. This is mainly because the MMAC reserves 20ms for the ATIM per beacon (typically 100ms). During an ATIM window, senders and receivers exchange channel negotiation messages and select the channels for data transmission. There is no data exchange during ATIM window. At the end of an ATIM window, all hosts switch to their respective selected channels and continue the data transmission with RTS-CTS-DATA-ACK cycles (for the remaining 80ms interval). This periodic hiatus in data transmission is the primary source of increased packet delays in the MMAC. However, the ODC and MMAC exhibit 65-70% smaller packet delays compared to the IEEE 802.11. This is due to the use of multiple channels, which reduces collisions and retransmissions.

In terms of throughput, the ODC and MMAC consistently perform better than the IEEE 802.11. This is due to the use of multiple channels, which essentially increases the total bandwidth by as many times as the

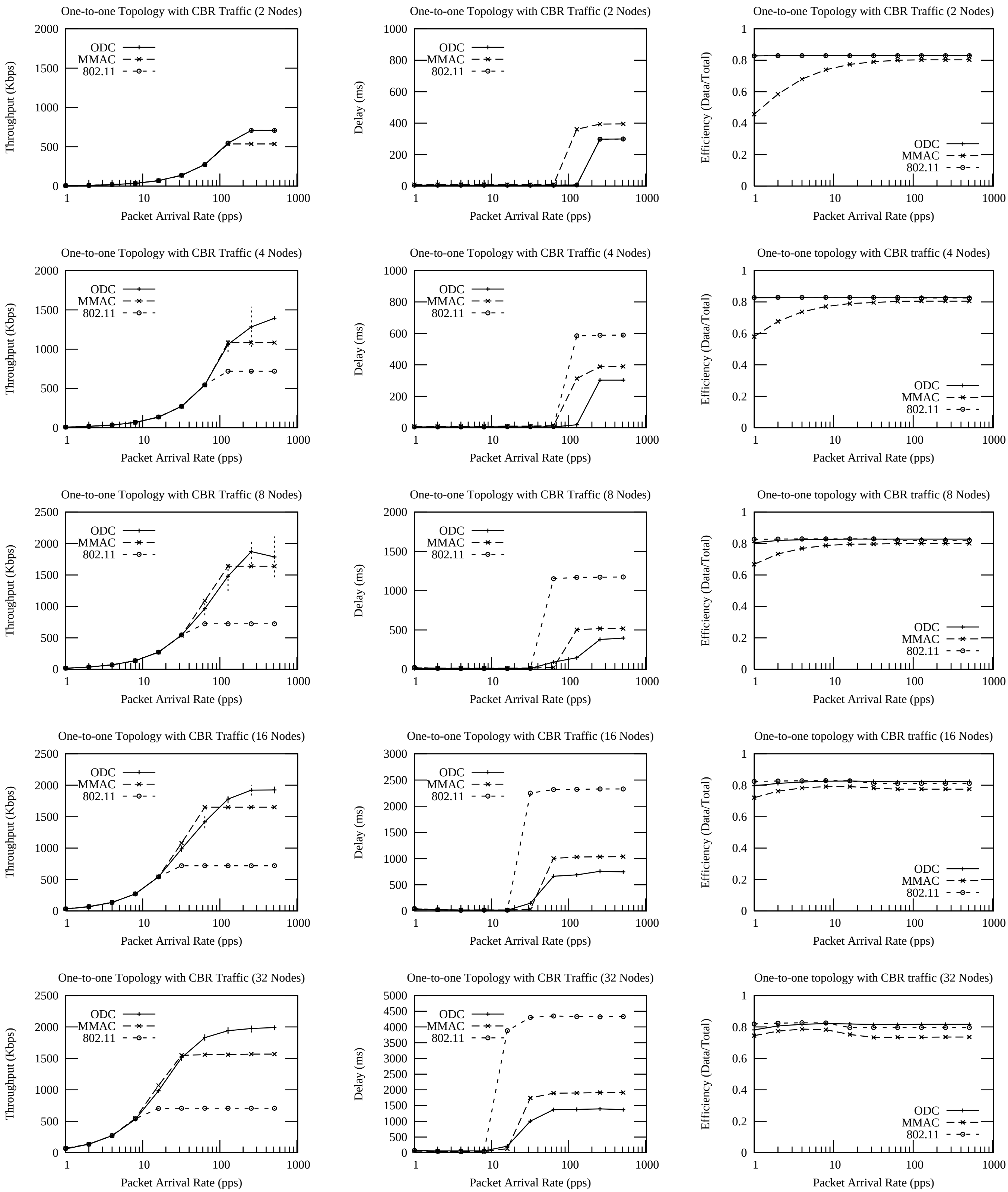

Figure 2: Aggregate throughput, average packet delay and efficiency for one-to-one flow distribution with CBR traffic

number of channels. The ODC provides higher aggregate throughput than the MMAC for small and large networks, and comparable aggregate throughput for medium size networks. This happens mainly because the 20ms ATIM window per 100ms beacon interval is too long for small networks. The data transfer takes place only during the remaining 80ms, thereby restricting throughput to 80% of the maximum achievable. Unlike the MMAC, the ODC does not have any concept of ATIM window or beacon interval. The timeline is contiguous,

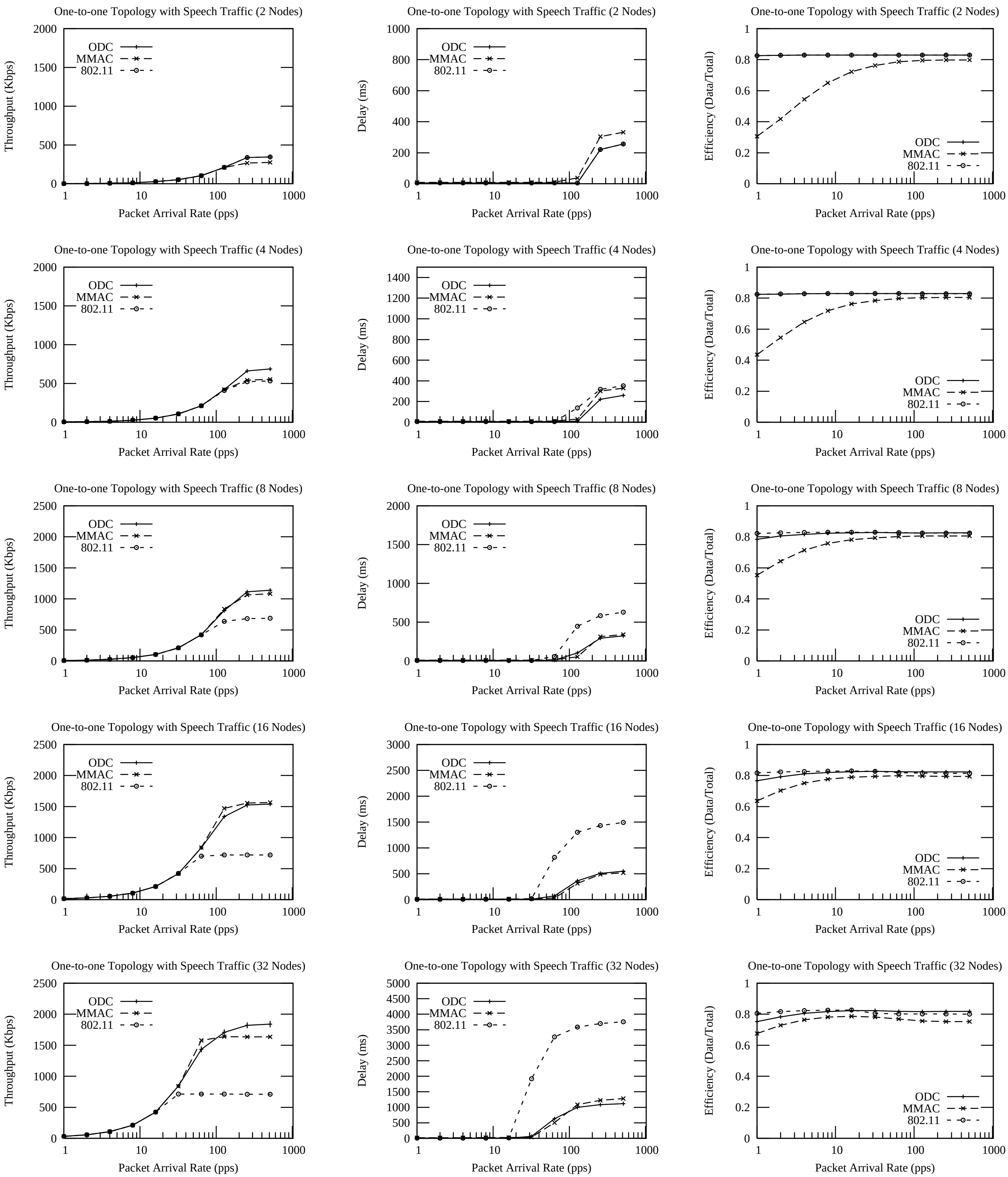

Figure 3: Aggregate throughput, average packet delay and efficiency index for one-to-one flow distribution with speech traffic

and data transfer as well as channel switches can happen anytime. In small-size networks (2-6 nodes), hosts can easily find each other and stay on their channels. This is particularly true for scenarios with one-to-one topology where each host is involved in only a single flow. In the two-node case, the ODC provides the same throughput as the IEEE 802.11 but higher than the MMAC. We see similar trends in 4-node networks (Figures 2 and 3).

Each channel negotiation during an ATIM window of the MMAC involves a three-way handshake to decide the channel. Each handshake cycle takes from 1-2.5 ms on average, including the interval between the last message of one cycle and the first message of the next cycle. [2]For a medium-size network (8 to 16 nodes), the ATIM window in MMAC is just sufficient for most communicating pairs of nodes to negotiate channels. Whereas, the ODC does not have any channel negotiation and senders and receivers can end up being on different channels. With such network sizes, the ODC is expected to perform comparable to the MMAC. In medium size networks, the ODC and MMAC perform similarly with respect to throughput (Figure 2 and 3).

In larger networks (with 32 nodes or more), the ODC performs better than the MMAC, because unlike in MMAC, nodes in ODC do not reset channel information of other nodes. The 20ms ATIM window of the MMAC is not sufficient for all communicating pairs to negotiate channels. In addition, there are more communicating pairs attempting to negotiate channels than that can be accommodated, which results in more collisions and greater delays. As a result, nodes unable to negotiate channels with their peers switch to randomly selected channels at the end of ATIM window. In ODC, nodes maintain channel information about their peers and do not randomly switch channels. As a result, the ODC exhibits higher aggregate throughput than the MMAC.

To estimate the protocol overhead, we define the *efficiency index (μ)* as the ratio between total data traffic and total traffic (including the control messages) on all channels during the entire simulation period. We contend that (1- μ) gives a close estimate of protocol overhead in terms unnecessary network traffic. The higher the protocol overhead, the higher is the wasted bandwidth and energy consumption. The ODC approximates the IEEE 802.11; whereas the MMAC has high protocol overhead, especially at low packet transmission rates (rightmost Figures 2 and 3). Thus the ODC would be more energy efficient than the MMAC in one-to-one network scenarios.

Star and clique flow distribution patterns

To analyze the performance of the ODC under even worse conditions, we simulated networks with star and clique flow distributions. Figures 4 and 5 show the performance of ODC, MMAC, and IEEE 802.11 with respect to throughput, packet delay, and protocol efficiency. As expected, single channel protocol like the IEEE 802.11 provide the maximum throughput in these scenarios. Since all nodes contend for the medium on the same channel, packet delays are higher using the IEEE 802.11.

In the star topology, we observe that the ODC provides higher throughput than the MMAC for all network sizes. This happens because in the ODC all nodes try to stay on the same channel as the central node. Whereas, in the MMAC nodes switch to other channels if they are not able to negotiate channels with the central node during the ATIM window. In addition, 20ms ATIM window causes further decrease in throughput in the MMAC.

[2] Observations from the simulation traces show that the number of three-way handshakes per ATIM window is 8-16, with an average of 11. On average 11 communicating pairs are able to negotiate channels during the ATIM window.

Average packet delay in case of the MMAC is better than that in the ODC because there are fewer collisions due to fewer transmitting nodes on the same channel as the central node compared to the ODC. The efficiency index plots show that the ODC introduces a small additional protocol overhead over the IEEE 802.11. The MMAC exhibits higher protocol overhead than the ODC and IEEE 802.11.

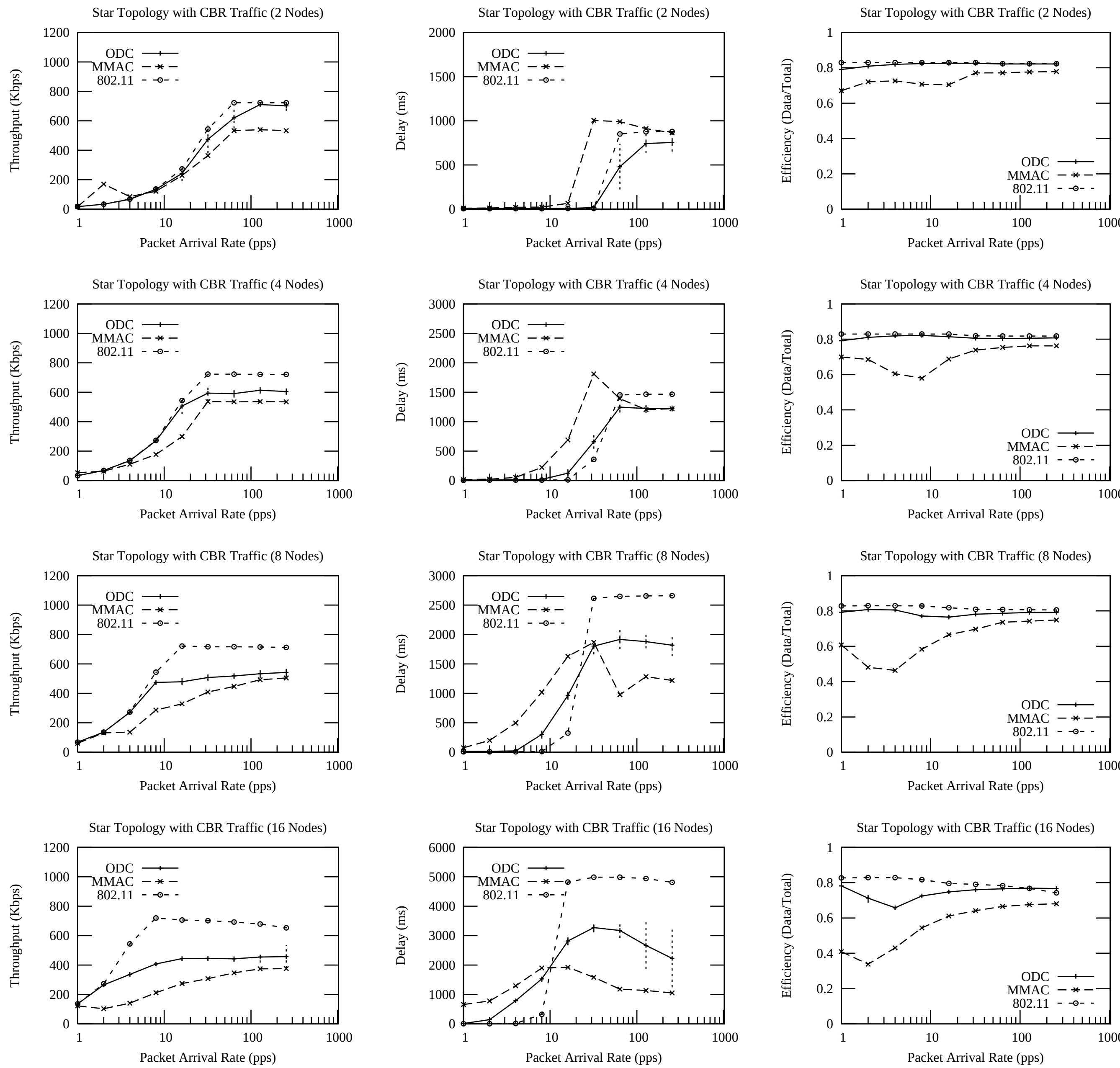

Figure 4: Aggregate throughput, average packet delay and efficiency index for star flow distribution with CBR traffic

In clique schemes, the ODC and MMAC perform similarly and the IEEE 802.11 outperforms them in terms of aggregate throughput and average packet delay (Figure 5).

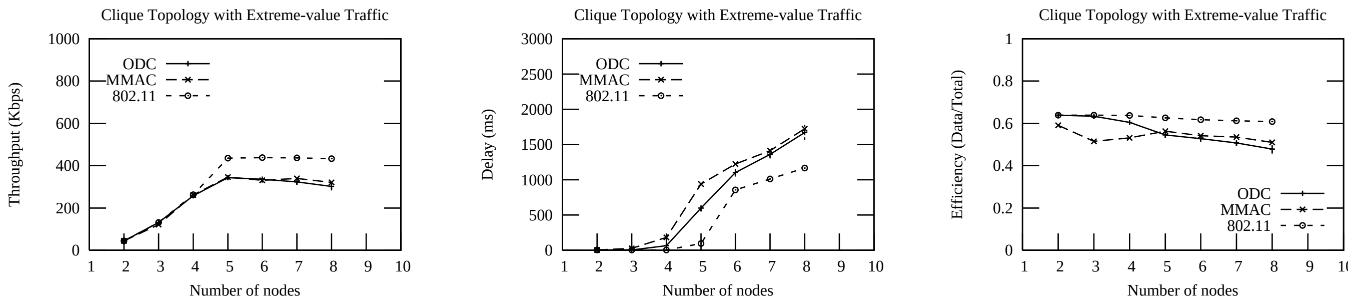

Figure 5: Aggregate throughput, average packet delay and efficiency index for clique flow distribution with extreme-value traffic

VI. Conclusions and future work

Channel switch in ODC takes place on a per packet basis. We plan to investigate how a host could make a more informed channel selection by looking ahead in the packet queue and rearranging the packets, before deciding to switch channels. This can also be coupled with an explicit power-savings mode for devices that choose to remain temporarily unavailable and trade the delay for an extended battery life.

As the simulation results indicate, the flow distribution across hosts has a dominant impact on the performance. The ODC enhances the bandwidth utilization and keeps the average delay and protocol overhead reduced in networks where a host typically communicates with a one other device at any given time. Another advantage of the ODC is that it does not require any specialized hardware and can be used as an IEEE 802.11 extension. In general, based on the characteristics of the system to be deployed, availability and cost of transceivers, switching overhead, its expected communication patterns, energy constraints, and delay requirements, a designer can choose a certain MAC protocol over another. We hope that this paper will stimulate further comparative analysis, as more applications are deployed using wireless LANs and new technologies emerge reshaping the wireless landscape.

References

[1] IEEE 802.11 Working Group, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", 1999.

[2] The Network Simulator, Version ns-2.1b9a, http://www.isi.edu/nsnam/ns/.

[3] CMU Monarch (Mobile Networking Architectures) Project, http://www.monarch.cs.cmu.edu/.

[4] J. So and N. H. Vaidya, "Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver", in Proceedings 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC), May 2004, Tokyo, Japan.

[5] Jiandong LI, Zygmunt J. Haas, Min Sheng, and Yanhui Chen, "Performance Evaluation of Modified IEEE 802.11 MAC for Multi-Channel Multi-Hop Ad Hoc Network", 17th International Conference on Advanced Information Networking and Applications (AINA), March 2003, Xi'an, China.

[6] Wing-Chung Hung, K.L. Eddie Law, A. Leon-Garcia, "A Dynamic Multi-Channel MAC for Ad-Hoc LAN", in Proceedings of the 21st Biennial Symposium on Communications, April 2002, Kingston, Canada.

[7] Asis Nasipuri and Samir R. Das, "A Multichannel CSMA MAC Protocol for Mobile Multihop Networks"; in Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), September, 1999, New Orleans.

[8] N. Jain and S. Das, "A Multichannel CSMA MAC Protocol with Receiver-Based Channel Selection for Multihop Wireless Networks", in Proceedings of the 9th Int. Conf. On Computer Communications and Networks (IC3N), October 2001.

[9] Santashil Pal Chaudhuri, Rajnish Kumar and Amit Kumar Saha, "A MAC protocol for Multi frequency Physical layer", Technical Report, January 2003, Rice University, Houston, TX.

[10] Engim's Multi-Channel WLAN Switching Engine chipset, Engim Inc., http://www.engim.com.

[11] D-Link Extreme G Wireless Networking Products, D-Link Corp., http://www.dlink.com.

[12] Johannes Farber, "Network game traffic modeling", Proceedings of the 1st workshop on Network and systems support for games, 2002, Bruanschweig, Germany.

[13] M. Benaissa, V. Lecuire, F. Lepage, A. Schaff, "Analysing End-to-End Packet Delay and Loss in mobile ad hoc networks for interactive audio applications", Workshop on Mobile Ad Hoc Networking and Computing, MADNET, 2003.

[14] Samuel Kotz, Saralees Nadarajah, "Extreme Value Distributions: Theory and Applications", ISBN 1-86094-224-5.