RC22512 (W0206-141) June 26, 2002
Computer Science

IBM Research Report

A Label-switching Packet Forwarding Architecture for Multi-hop Wireless LANs

Arup Acharya, Archan Misra
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Sorav Bansal
IBM Solutions Research Center
Block 1, Indian Institute of Tech.
Hauz Khas, New Delhi 110016
India

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

A label-switching packet forwarding architecture for multi-hop wireless LANs

Arup Acharya
IBM TJ Watson Research Center
arup@us.ibm.com

Archan Misra
IBM TJ Watson Research Center
archan@us.ibm.com

Sorav Bansal
IBM India Research Labs
soravban@in.ibm.com

Abstract

A router in wired network typically requires multiple network interfaces to act as a router or a forwarding node. In a ad-hoc multi-hop wireless network on the other hand, any node with a wireless network interface card can operate as a router or a forwarding node, since it can receive a packet from a neighboring node, do a route lookup based on the packet's destination IP address and then transmit the packet to another neighboring node using the same wireless interface. This paper investigates a combined medium access and next-hop address lookup based on fixed length labels (instead of IP addresses), that allow the entire packet forwarding operation to be executed within the wireless NIC without the intervention of the host protocol stack. Medium access schemes to date, such as IEEE 802.11, have been designed implicitly for either receiving or transmitting a packet, but not for a forwarding operation, i.e. receiving a packet from an upstream node and then immediately transmitting the packet to a downstream node as an atomic channel access operation. This paper proposes a MAC protocol for packet forwarding in multi-hop wireless networks. The proposed protocol builds on the IEEE 802.11 DCF MAC using RTS/CTS and uses MPLS like labels in the control packets (RTS/CTS) to allow the forwarding node to determine the next hop node while contending for the channel. The throughput of this protocol is compared with 802.11 DCF MAC through simulation.

1 Introduction

Channel speeds for the IEEE 802.11 [ieee][wlan] family of standards continue to increase: while the recently proposed 802.11a operates at 54 Mbps, enhanced versions operating at speeds up to 108 Mbps are also under investigation. Such high-speed LAN standards are expected to further increase the popularity of wireless access to the backbone infrastructure and eventually lead to the deployment of *multi-hop*, *wireless* networks, where the wired backbone is reachable only via multiple wireless hops. Potential examples of this include *in-building* wireless networks in malls, hotels and apartment blocks, and *community* networks where rooftop antennas are used to create an ad-hoc wireless access infrastructure in specific residential communities.

In this paper, we propose an architecture for a *forwarding node* in a multi-hop wireless network that shifts the packet forwarding functionality away from the host processor to the wireless network interface card (NIC) by combining medium access control (MAC) for packet reception and subsequent transmission with address lookup in the interface card itself, using fixed-length addressing labels in the MAC control packets. The motivation for integrating medium access control with forwarding functionality arises out of one fundamental difference between wireless and wired networks:

> *In a wired network, a forwarding nodes typically*[1] *has at least two physical network interfaces, with the forwarding functionality consisting of receiving a packet over one physical interface and*

[1] Overlay networks could be created out of tunnels using single network interface cards.

> *subsequently sending it out over a second interface*[2]*. In contrast, a node N, with a single wireless interface, may act as a forwarding node by transmitting a packet to a node other than from which received the packet. In effect, N acts as an intermediary for two nodes that are each within the communication range of N but not directly within the range of each other*.

Accordingly, packet forwarding in the wireless environment does not typically imply the transfer of a packet between distinct interfaces on a single host. A conventional implementation of packet forwarding thus involves the reception of a packet on the wireless interface, transfer of the packet up the host's protocol stack to the IP layer where a routing lookup is used to determine the IP (and MAC) address of the next hop, and subsequent transmission of the packet using the same wireless interface to the MAC address of the next hop. The forwarding node is thus involved in two separate channel access attempts during the forwarding process: once to receive the packet and again to "forward" it. Moreover, the actual forwarding path involves two separate transfers of data between the memory on the network interface card (NIC) and the host's memory (accessed by the host software).

A key component of our proposed architecture for a forwarding node in multi-hop wireless network is an efficient medium access protocol for packet forwarding, i.e., the definition of an *atomic* channel access scheme that pipelines the reception of a packet from an upstream node and the subsequent transmission to the downstream node. To exploit this cut-through capability of the MAC layer, the NIC must also be capable of determining the identity of the next-hop node without invoking a lookup of the routing tables resident in the host protocol stack. Such NIC-resident lookups can be achieved by the use of a label-switching mechanism, such as MPLS [mpls], with a separate label-distribution algorithm such as LDP [ldp] being used to distribute levels to appropriately reflect the traffic routes. This allows packet forwarding to be confined *entirely to the NIC*, which matches the label of an incoming packet with an entry in the data structure to determine the MAC address of the next hop node and the label to be used for that hop. Our integrated MAC design thus eliminates the overheads associated with the functions of IP route lookup and the movement of the packet between the interface card and the host protocol stack.

Our current focus is only on static wireless multi-hop networks; while node mobility is indeed a feature of such networks, such mobility predominantly impacts the routing protocols. We do not propose any new routing protocol and assume that a suitable ad-hoc routing protocol, such as DSR [dsr] or AODV [aodv] is available to set up the appropriate routing tables at each node. Label distribution is achieved through a separate label distribution protocol [ldp] or by integrating label distribution with routing information which is deferred for a future paper.

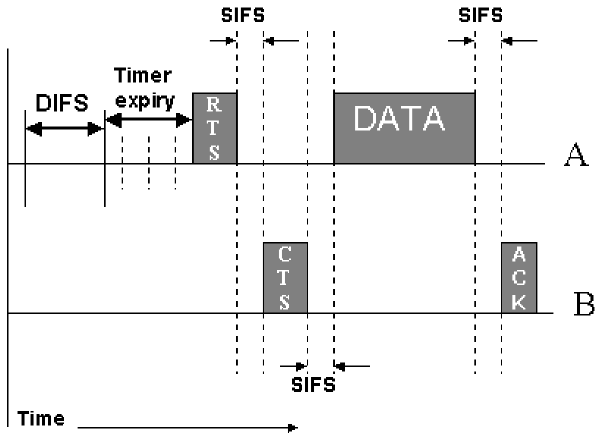
As related work, the use of MPLS (or labels) for providing fast and efficient packet forwarding in wireless environments has not been extensively reported in literature. [jabbari] proposes the use of MPLS [mpls] [ldp] to support packet routing and handoff in wireless cellular networks and the use of label merging to accommodate multiple links between a mobile node and the cellular infrastructure. To the best of our knowledge, there appears to be no prior public work in the area of devising MAC algorithms for providing label-based forwarding in multi-hop wireless networks.

The rest of this paper is organized as follows. In section 2, we introduce the notion of label switching and its application to a multi-hop wireless network using the standard 802.11 MAC. In section 3, we present DCMA protocol, based on the 802.11 MAC, that is specifically designed for efficient and low-overhead packet forwarding operation in wireless networks. The following section presents our simulation results and compares them with a 802.11 based wireless network. The last section is a discussion and summary of our future work.

[2] In high-end routers/switches, the packet is transferred from one interface to another via a dedicated switching fabric, while in software based routers, the packet is processed by the host CPU (e.g. route lookup) between packet reception on one interface and subsequent transmission on another.

2 Problem Definition

The 802.11 MAC is designed to provide shared access to the wireless medium in two basic modes: the Point Coordination Function (PCF) mode, which involves access control regulated by a unique master node, and the *Distributed Coordination Function (DCF),* which involves a purely distributed mechanism for contention resolution. The DCF mode is commonly employed in multi-hop ad-hoc networks, where each node essentially acts a peer to all nodes within its transmission range. Unicast communication in the DCF mode involves a 4-way handshake mechanism (shown in Figure 1) between a data sender node *A* and the corresponding recipient node *B* to both avoid collisions and verify reliable packet forwarding:



802.11 DCF MAC

Figure 1

1.The RTS (request-to-send), sent by node A, specifies a time interval T_{RTS} that includes B's response through a CTS (clear-to-send), followed by data transmission by A and time to send an ACK from A to B[3]. This is in effect informs anyone within A's neighborhood that the medium is "reserved" for the duration T_{RTS} .

2.The CTS, sent by node B, specifying the time interval T_{CTS} during which A is permitted to send this data--in 802.11, the interval specified in the CTS is equal to the transmission times of the data and the ACK. The CTS informs all neighbors of B that the channel is reserved for the duration T_{CTS}.

3.The data itself, sent by node A, during the slot reserved for it by the CTS--this data transfer phase immediately follows the reception of the CTS. Note that the data transmission interval is typically larger that than control message transmission times (CTS/RTS/ACK). The max data frame that can be sent is 2346 bytes, while the RTS, CTS and ACK control frames are 20 , 14 and 14 bytes respectively.

4.The final data ACK, sent by node B, indicating successful reception--this ACK is sent after the end of the transmission of data by A.

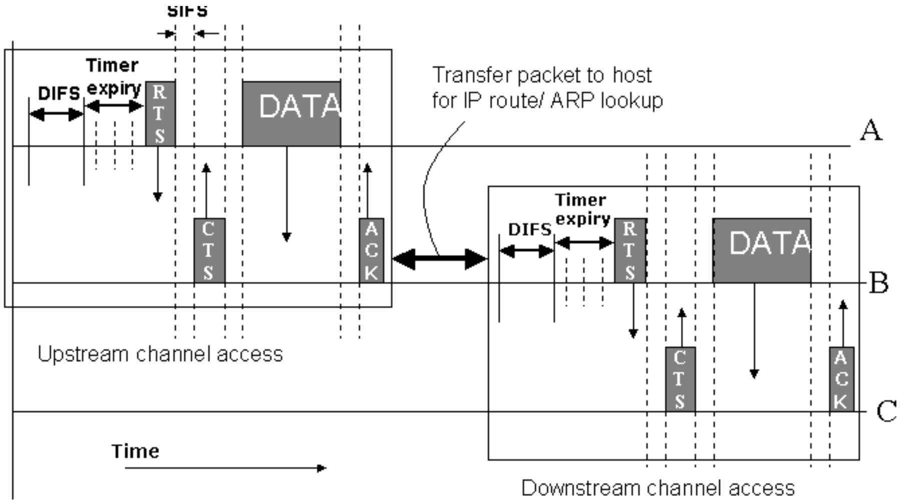
For contention resolution, 802.11 uses a timer-based exponential back-off scheme, as follows. Prior to transmitting a packet, a node senses the channel for a period equal to DIFS (Distributed Inter-Frame Space). If the channel is busy, the node selects a random back-off time in the range (0, Congestion Window) (specified in terms of slots). The backoff timer is decremented whenever the channel is free; the node

[3] The interval also includes short inter-frame spacing (SIFS) periods between the RTS and CTS, CTS and DATA, DATA and ACK. For the rest of the paper, a SIFS period is implied when referring to such transmissions even if not stated explicitly.

makes a fresh attempt at sending an RTS packet upon the expiration of the timer. Upon failure of the RTS packet (no CTS packet is received), the congestion window is doubled and a random timer is chosen from the new window. In addition to the backoff timer, the 802.11 MAC requires every transmission of an RTS packet to be preceded by a channel sense for a DIFS duration, thereby reducing the probability of collision with an ongoing transmission. Each 802.11 is also maintains a Network Allocation Vector (*NAV*) that monitors the state of the channel. Whenever the node overhears a control packet (RTS or CTS) transmitted by a neighboring node (to some other node), it updates its NAV appropriately to reflect the duration of the corresponding 4-way data exchange.

3.1 Forwarding Operation in 802.11 MAC

We now discuss the overheads associated with a forwarding operation when using the 802.11 MAC in a multi-hop wireless environment. The terms **upstream** node, **forwarding** node and **downstream** node are defined as follows : the upstream node sends a data packet to the forwarding node; from the upstream node's perspective, the forwarding node is the next hop neighbor on the path to the packet's final destination. Similarly, the downstream node is the next hop neighbor for the forwarding node towards the packet's destination; upon receiving the packet from the upstream node, the forwarding node will subsequently send it to the downstream node. A routing protocol executes in the background to setup per-hop routing tables.



802.11 Based packet forwarding

Figure 2: Multi-hop forwarding in 802.11 MAC

Consider the case shown in Figure 2, where A is the upstream node, B is the forwarding node and C the downstream node. After the IP lookup function in host A determines that B is the next hop of the DATA packet, the packet is transferred to A's NIC. The MAC implementation on A's NIC then performs a 4-way handshake (including any backoff timer-based countdown that may be needed to gain access to the channel) to forward the packet to B's NIC. Note that node C is guaranteed to remain silent during the DATA and ACK portions of this packet transfer, since the CTS from B to A effectively updates its NAV and blocks any concurrent transmission attempt. At B, the packet is transferred to the main memory from the NIC, and the host CPU is notified (e.g. via interrupts) for further processing of the packet by the IP protocol stack running on the host CPU. The host software (IP protocol stack) would typically queue up the packet in a transmission queue and select packets for transmission based on a scheduling algorithm (typically, FIFO). When this packet reaches the head of the queue, the same steps as those executed at A, would be taken, e.g. perform lookups to determine the IP address and then the MAC address of the next hop (C), insert the MAC-layer header (corresponding to next hop C) and transfer the packet to the NIC. This packet is now treated as an independent data transfer between the nodes B and C; accordingly, B performs the usual backoff timer countdown before initiating an RTS-CTS-DATA-ACK exchange with C. Once this handshake is successfully completed, the packet is received by C's NIC, at which point the whole forwarding process is repeated. As with the initial data transfer (from A to B), the NAV of node A is blocked (by the RTS sent by B) for the entire duration of the 4-way exchange between B and C.

3 Use of MPLS labels

On inspecting the entire forwarding process between the upstream node and the downstream node, we observe that considerable performance gains may be expected if the forwarding node's (B's) NIC is able to directly redirect the packet received from the upstream node A back onto the channel towards the downstream node C. There are two separate enhancements necessary to achieve this pipelined functionality:

a) B's NIC must be capable of resolving the identity of the downstream node (and it's MAC address) directly without resorting to an IP lookup in the host kernel.

b) The MAC protocol must allow B to instantaneously initiate the downstream transfer (B to C) immediately upon completion of the transfer from A to B.

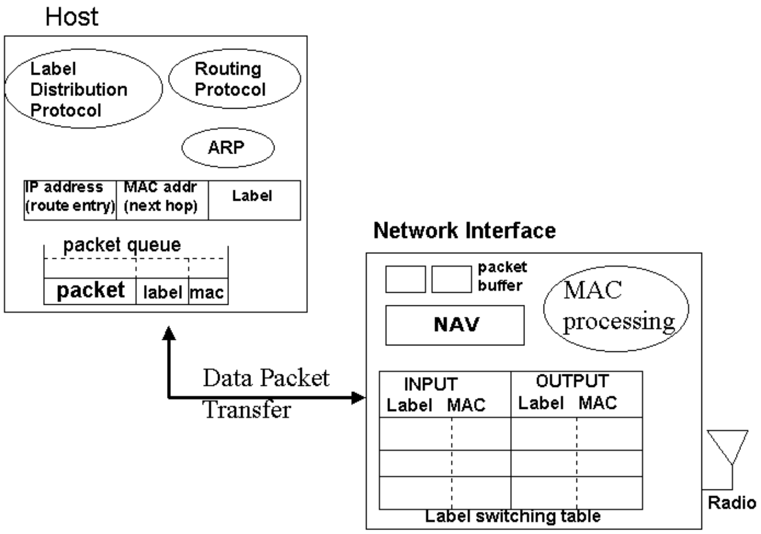


Figure 3: Host and NIC components for packet forwarding using labels

In this section, we explain how task(a) can be achieved through the use of MPLS-based labels. The network interface card is enhanced to store a label switching table, consisting of an incoming MAC address, an incoming label , an outgoing MAC address and a outgoing label. Labels are associated with routes or destinations, i.e. all entries in the label switching table that refer to the same route, will share the same outgoing MAC address (of the next hop) and outgoing label. For example, let an entry in the switching table of B be <A, L_{AB} , C, L_{BC} >. The interpretation of this entry is that any packet received at B from A with a label L_{AB} will use C as the next downstream hop with a label L_{BC}.[4] The combination of the outgoing label L_{BC} and the MAC address of the next hop node C, essentially defines a specific route to a destination, say Z. If B has another neighbor, say D, which uses B to reach Z as well, then there will a corresponding entry in the label switching table <D, L_{DB} , C, L_{BC} >. The number of distinct outgoing labels is equal to the number of destinations in the network. It should be noted that each label is unique only to a single hop, and the same label may be re-used by different nodes of the network.

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

When a packet is forwarded by node B, the incoming label will be replaced by the outgoing label. In the cut-through MAC protocols described below, we do not need to label the DATA packets per se, but instead carry the labels on the control packets such as RTS or ACK packet. This is possible because the MAC protocol reserves a time duration (via control packets) during which a forwarding node can expect to receive a DATA packet. Thus, if the control packet carries the label, then the forwarding node can decide the next hop MAC address and the outgoing label, and no label needs to be carried with the DATA packet.

The label switching table is populated by a label distribution protocol running at the host in conjunction with a routing protocol. Though in this paper, we have not explored the design of such a combined protocol, we believe it is fairly straightforward to piggyback labels with route updates or run a separate label distribution protocol, as is the case for wired networks, e.g. [mpls] [ldp]. As a result of the routing and label distribution protocols, along with ARP, the host maintains a queue of packets waiting to be moved to the wireless interface card for transmission onto the wireless channel. Each packet is associated with a (outgoing) label and the MAC address of the next hop node. A packet is placed in the queue when (a) the host generates a packet or (b) when a cut-through terminates at this node, either because the cut-through could not be extended beyond this node because the channel access for the next hop was not successful, or because this node is the final destination of the packet. Prior to inserting each packet in the queue, the host does a IP lookup using the packet's destination IP to determine the packet's next hop node. In addition, the ARP cache is inspected to determine the MAC address of the next hop node, and a route (destination IP) to label mapping table is used to determine the (outgoing) label. Then packets are handed over to the NIC one at a time, along with the outgoing label and next-hop's MAC address. In general, the NIC does not need to maintain a packet queue; the packet buffer shown in the figure above is use to hold a packet awaiting channel access, and to buffer a packet while it is in the process of being forwarded. Packets that are successfully forwarded need to be buffered only between reception (from an upstream) node and immediate transmission to the downstream node. If the forwarding fails, i.e. the cut-through did not succeed, or an ACK was not received for DATA transmission, the packet is sent to the host and inserted at the back of the queue.

4 Cut-through MAC protocol

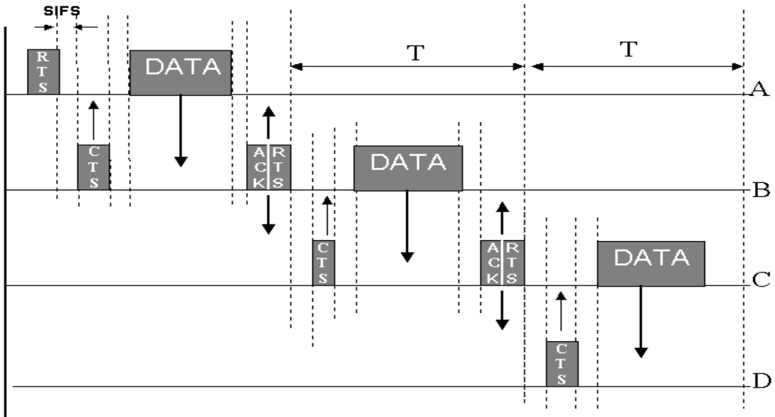
We have seen in the last section how the presence of label information in the "data" stream helps the forwarding node's NIC to correctly identify the identity (and the associated label) of the downstream node. Without additional enhancements at the MAC layer, such a packet would however need to be buffered at the NIC between the two separate channel accesses (depicted in Figure 2) until the channel is again acquired for transmission to the downstream node. The resulting latencies (which can be of the order of milliseconds or even seconds if multiple backoffs are involved) can effectively negate any performance benefits (in terms of latency or throughput) achieved by the elimination of the routing lookups. We now explain how our proposed extension to the 802.11 DCF channel access scheme is designed to allow the forwarding node to combine the two separate access channels depicted into a single "seamless" access.

Our proposed MAC scheme is based on enhancements to the IEEE 802.11 Distributed Coordination Function (DCF) mode of channel access and follows the associated 4-way handshake involving the exchange of RTS/CTS/DATA/ACK packets. We term this scheme as Data-driven Cut-through Multiple Access (DCMA). DCMA attempts to replace the two distinct channel accesses, upstream and downstream, with a combined access. The reservation for the downstream hop is attempted only after successfully receiving the DATA packet from the upstream node. The advantage is that a downstream reservation is made only after the upstream channel access has been granted *and* the packet reception from the upstream node is successful. Accordingly, DCMA combines the ACK (to the upstream node) with the RTS (to the downstream node) in a single ACK/RTS packet that is sent to the MAC broadcast address. The payload of the ACK/RTS packet, now contains the MAC address of the upstream node, and the MAC address of the downstream node. It also includes a label intended for use by the downstream node to figure its next hop. Since the downstream node (and all other neighboring nodes of the forwarding node) is assured to be silent till the completion of the ACK[5], piggybacking the RTS packet provides the forwarding node with preferential channel access for the downstream transmission. Cut-through in DCMA fails when the downstream node fails to respond to the ACK/(RTS) with a positive CTS; the forwarding node then simply queues the packet in the NIC queue and resumes normal 802.11 channel access.

Since DCMA has no notion of future reservations (all access attempts are for immediate transfer of DATA packets), it does not require any modifications or enhancements to the 802.11 NAV—a node simply stays quiet as long as it is aware of (contiguous) activity involving one or more of its neighbors. Any node that overhears an ACK/RTS not addressed to it merely increments the NAV by the time interval included in the ACK/RTS message; this NAV increment is also performed by the target of the ACK (the upstream node). The operation of DCMA can be understood by following the timing diagram provided in Figure 6. Assume that node A has a packet to send to node D. A[6] sends a RTS to B, which includes a label L_{AB} associated with the route to D. Assuming that its NAV is not busy for the proposed transmission duration, B replies with a CTS. B receives the DATA packet, and then sends a RTS/ACK control packet, with the ACK part addressed to A, and the RTS part addressed to C, along with a label L_{BC} . C's actions would be analogous to B, except that it uses the label L_{CD} in its RTS/ACK message.

T		
ACK Flag	MAC address	
RTS Flag	MAC address	(out) Label

Payload of ACK/RTS control packet



DATA DRIVEN CUT-THROUGH MAC

Figure 6

Label lookup : In DCMA, the label is carried in the RTS/ACK (or RTS). In principle, this label could also have been carried by the DATA field, since the label lookup (to find the downstream node) is not strictly necessary until after the DATA is received. However, by providing the label information in the RTS, we provide the forwarding node additional time to complete the lookup (in parallel with the DATA transfer from the upstream node). This should not be a problem, since the DATA duration is at least tens of μsecs (e.g., a 500 byte packet on 2Mbps channel takes 2 msecs).

Due to the competition among different flows, it is possible that DCMA can fail to set up the "fast-path" (cut-through) forwarding at different points in the traffic path. Upon the failure of a cut-through attempt, DCMA reverts to the base 802.11 specification, aborting the cut-through attempt and using the exponential backoff to regulate subsequent access to the shared channel. The channel contention resolution of DCMA is same as that of 802.11, with a node remaining silent as long as any of its one-hop neighbors are either receiving or transmitting a data packet. Accordingly, this protocol do not suffer from any additional penalties, over and above those present in 802.11.

[5] This was discussed in section 2.

[6] We assume that, like CCMA, the initial IP address to label mapping is done by the host and the label to be used, MAC address of the next hop and the packet is moved over to wireless interface card.

5 Advantages of an integrated MAC and label-switching architecture

In the previous two sections, we outlined an enhancement to the 802.11 DCF MAC protocol that combined channel accesses on the upstream and the downstream hops. This was made possible by the use of labels in the RTS/ACK packets which allowed the forwarding node to select the downstream node at the NIC, without invoking participation from the host CPU for a IP route lookup. We proposed an architectural enhancement to the wireless NIC in the form of a label switching table which consisting of incoming and outgoing <label, MAC address> pairs. This enabled the NIC itself to decide the next hop of an incoming labeled packet. The key advantages of this approach are:

- Packets that can be label-switched do not interrupt the CPU for packet processing. This could lead to considerable power savings for example, if the node is used only for packet forwarding purposes (which can be accomplished entirely within the NIC). The CPU needs to wake up for processing route updates and changes in label mappings.

- Since the forwarding node makes an immediate attempt to grab the channel following successful reception of the packet from the upstream node, this could lead to better utilization of the wireless channel. This is so because subsequent to the upstream transmission, the forwarding node obtains preferential access to the channel instead of all its neighbors contending for access. In effect, whenever the forwarding node responds with a CTS to the upstream node's RTS, it has implicitly gained channel access for the downstream transmission (since a neighboring node must sense the channel to be free for a DIFS period following the ACK corresponding to the upstream transmission before it can contend for the channel)

- End-to-end latency for a packet will tend to be lower since the delay at each node will be lower due to cut-through channel access.

In the next section, we study the latency and throughput benefits of DCMA. Note that since the ns simulator does not have the capability to measure the delay in transferring a packet from the NIC to memory and subsequent host processing, this was not considered for the latency measurement. If it was possible to do so, we would expect a even lower latency for DCMA compared to 802.11 since packets that are entirely forwarded by the NIC would not incur this additional delay.

6 Simulation Results

We have implemented the DCMA access protocol as part of the ns-2.1b8 simulator [ns] with the CMU wireless extensions [monarch] and have conducted initial simulation studies to evaluate their performance characteristics relative to 802.11. As part of our studies, we focus on two metrics : a) the *throughput* improvement achieved by the cut-through protocols and b) the potential reduction in *end-to-end latency* due to the expedited MAC forwarding.

The parameters of the ns simulator are tuned to model the Lucent Wavelan card at a 2 Mbps data rate. The effective transmission range is 250 meters, and the interfering range is about 550 meters. All simulated data packets are preceded by an RTS/CTS exchange regardless of the size.

To measure the throughput, high packet rate sources were run over UDP. The packet rate at the source was kept high enough to ensure availability of queued packets at any point in the simulation. The throughput was measured by counting the number of received packets at the destination(s). We measured latency only for packets that were received at the receiver. The buffer size at each node was 50 packets. The routing tables were pre-configured with the shortest path routes to their respective destinations.

Two topologies were used for the simulation:

Chain : Distance between successive nodes in the chain is 250m. Traffic consists of a single flow of UDP packets sent from the leftmost node to the rightmost node.

Grid : We used a 4X4 grid where the traffic pattern consists of 4 vertical flows, one along each column and starting at the top row and terminating at the bottom row. Since the distance between neighboring nodes (vertical or horizontal) on the grid is set to 250 meters, this topology provides an example where one or more nodes lie within the interfering distance of another node (diagonal nodes are less than ~425 meters away and hence within the interference range of 550m).

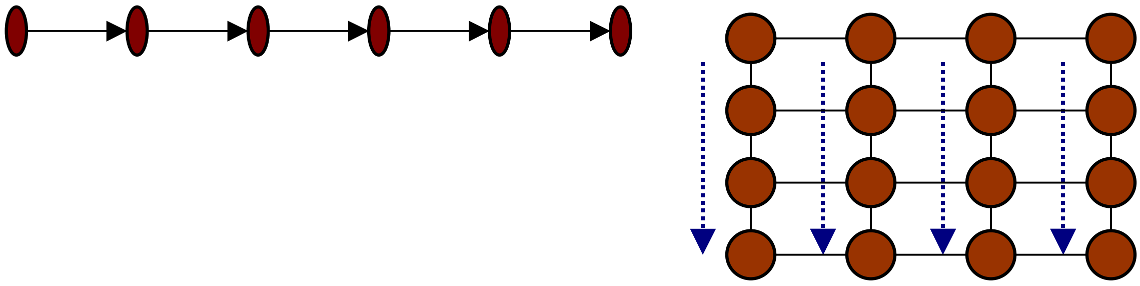


Figure 7: Different Simulation Topologies (chain and grid)

Figure 8 shows the performance results of a 7-hop chain topology. While the throughput improvements for DCMA are just around 20%, the latency improvements are quite significant, ranging from 100% (256byte packets) to 63% (1536bytes). Note that the latency also includes the buffering delay at the source node.

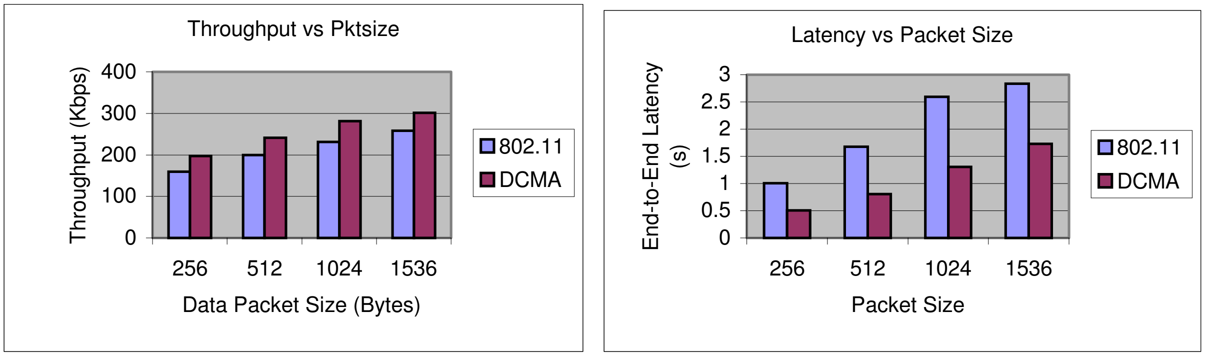


Figure 8: Comparative Performance for Different Packet Sizes on a 7-Hop Chain

We also studied the behavior of DCMA as a function of the number of hops in the CHAIN topology. Figure 9 plots the performance results of DCMA vs. the number of hops for a packet size of 1536 Bytes. Consistent with the earlier graphs, it can be seen that the latency benefits are significant (reduction of almost 50%), while throughput improvements are marginal. Note also that it is well-known [capacity], [multihop] that throughput performance over multihop networks degrades with number of hops due to contention between neighboring hops, and this is reflected in the graph below for both 802.11 and DCMA. While the throughput of a chain using 802.11b saturates at around 0.25 Mbps as the chain grows longer, the throughput using DCMA is around 0.325 Mbps.

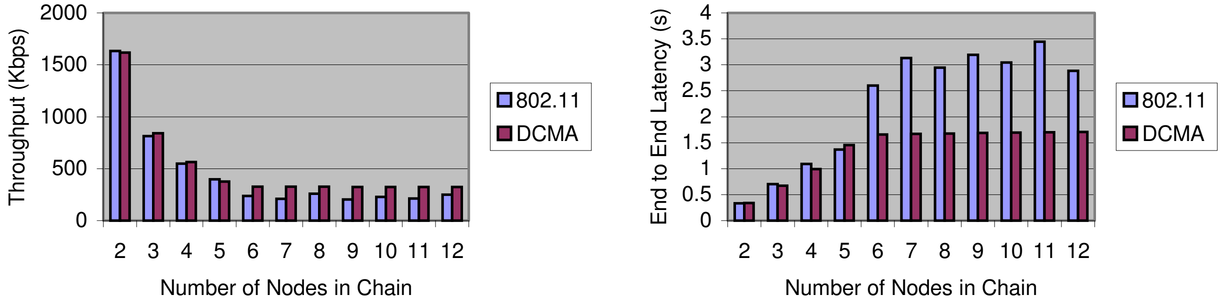


Figure 9: Comparative Performance for Different Chain Lengths. DCMA provides much lesser latency and higher throughput over 802.11

Figure 10 shows the throughput of a 12 node chain when the send rates at the source is varied. 802.11b performance is degraded when the send-rate at the source is increased beyond .375 Mbps. At a rate higher than 0.375Mbps, the throughput saturates at around 0.24 Mbps when 802.11 is used. For DCMA, the peak throughput is obtained at a send-rate of 0.425Mbps and it saturates at 0.375 Mbps for higher rates. The degradation of performance at higher rates in 802.11 is attributed to poor scheduling at the MAC level caused due to contention among the packets of the same flow (also observed in [capacity]). DCMA has an advantage over 802.11 as it does not experience contention among packets of the same flow.

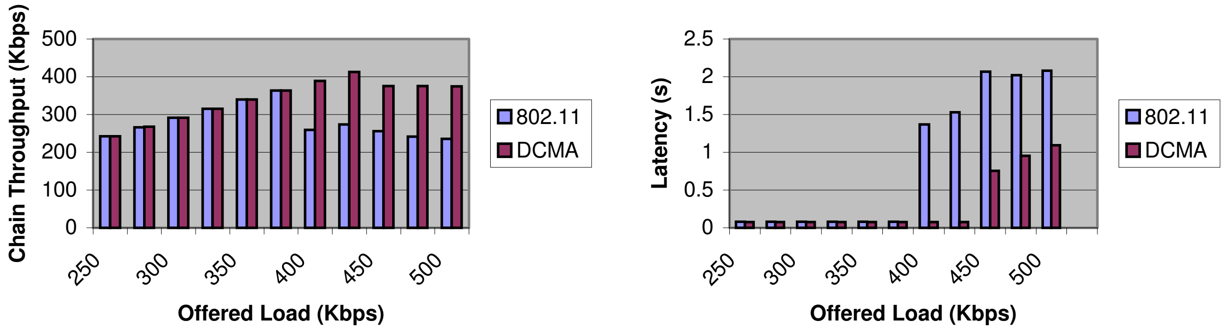


Figure 10 : Comparative Throughput and Latency of DCMA vs 802.11

Another interesting point to note is that the latency values jump up if the rate is increased even marginally above the maximum rate the MAC layer can support. The jump in latency is due to the queuing delay at the nodes. While 802.11 experiences queue build-ups at intermediate nodes, DCMA packets experience queue build up only at the source node in the one-flow chain scenario causing much lesser delays. This is confirmed by Figure 11 which shows that while 802.11 suffers buffer-buildups at nodes 2,3 and 4 (due to high arrival and low departure rates), DCMA shows uniform send rate at each hop causing no buffer build-ups due to no intra-flow packet contention.

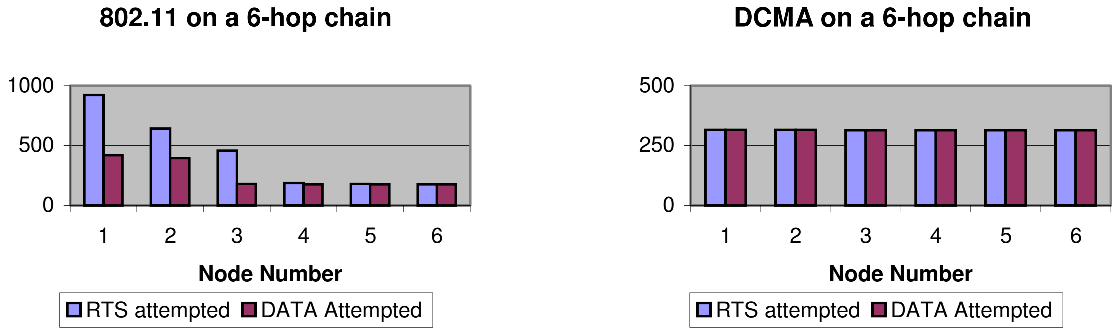


Figure 11 : Number of RTS/DATA packets attempted by each node in a 7 node chain

Figure 12 shows the throughput and average end-to-delay metrics incurred by DCMA vis-à-vis 802.11 in a 4X4 grid, for greedy sources in the first column sending horizontal streams of 1024-byte sized UDP packets to the nodes in the last column. DCMA scores over 802.11b as far as latency is concerned. When we consider the throughput, we find that DCMA allocates the bandwidth between competing flows much more fairly. This is illustrated in Figure 11 where we show the throughput of individual columns in a grid. Since the middle nodes in a grid compete for the channel with 12 other nodes, the throughput of the middle columns is considerably lesser than the outer columns. Since 802.11 requires a packet to contend for the channel at every hop, the middle columns are starved. On the other hand, DCMA allows the middle columns a much higher throughput. Hence DCMA proves to be very useful in scenarios where the forwarding nodes are in high interference regions.

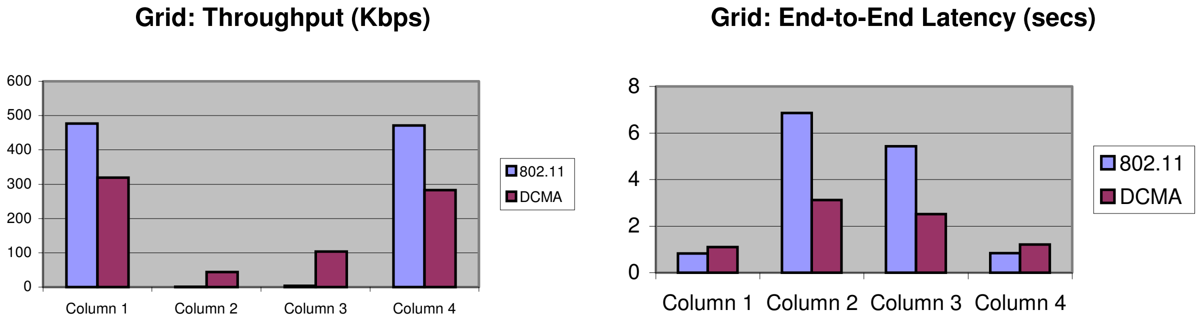


Figure 12 : Throughput and Latency of DCMA vs 802.11 for each column in a 4x4 grid vertical flows

7 Conclusions

In this paper, we presented an architecture for a “wireless router”, i.e. a forwarding node with a single wireless NIC in a multi-hop wireless network, that allows a packet to be forwarded entirely within the network interface card of the forwarding node without requiring per-packet intervention by the node’s CPU. This was made possible by enhancing the 802.11 DCF channel access scheme and by carrying a label in the RTS/ACK packet, which allowed the NIC to determine the packet’s next hop. The NIC was augmented with a label-switching table mapping incoming labels and MAC addresses to outgoing labels and MAC addresses. As part of future work, we aim to actually emulate DCMA operation within a NIC and measure the power savings possible. The simulation results presented in this paper were based on a single flow in a chain and parallel flows in a grid : we plan to look at more complex traffic patterns and topologies in the future. We believe that in addition to increases in raw link rates in 802.11a/b cards, there is a need for an integrated approach combining MAC, routing and TCP enhancements for end-to-end performance in multi-hop wireless networks to become comparable to early wireline Ethernet rates, and thereby realize the vision of multi-hop wireless networks in practice, instead of simply using wireless LAN for last-hop access to a wireline network.

8 References

[ieee] IEEE Computer Society. *802.11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, June 1997.

[wlan] B. Crow, I. Widjaja, J. G. Kim and P. T. Sakai. *IEEE 802.11 Wireless Local Area Networks.* IEEE Communications Magazine, Sept '99.

[dsr] D. Johnson and D. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks*. In Mobile Computing, chapter 5, Kluwer Academic Publishers, 1996.

[aodv] C. Perkins, E. belding-Royer and S. Das. *Ad-Hoc On-Demand Distance Vector (AODV) Routing*, draft-ietf-manet-aodv-09.txt, IETF, Work in Progress, November 2001.

[maca] P. Karn. *MACA – a new channel acces method for packet radio*. ARRL/CRRL Amateur Radio 9th Computer Networking Conference, 1990.

[macaw] V. Bhargavan, A. Demers, S. Shenker and L. Zhang. *MACAW : A Media Access Protocol for Wireless LAN's*. Proceedings of ACM SIGCOMM '94.

[dfwmac] J. Weinmuller, H. Woesner, J. Ebert, A. Wolisz. *Analyzing the RTS/CTS Mechanism in the DFWMAC Media Access Protocol for Wireless LANs*. IFIP TC6 Workshop on Personal Wireless Communications, April '95.

[seedex] R. Rozovsky and P. R. Kumar. *SEEDEX: A MAC protocol for ad hoc networks*. Proceedings of ACM Mobihoc 2001.

[jabbari] B. Jabbari, R. Papneja and E. Dinan. *Label Switched Packet Transfer for Wireless Cellular Networks*, Proceedings of IEEE WCNC, August 2000.

[fama] C. L. Fullmer and J. J. Garcia-Luna-Aceves. *Solutions to hidden terminal problems in wireless networks*. Proceedings of ACM Sigcomm '97, Sept '97.

[mpls] MPLS label switching architecture, IETF RFC 3031

[ldp] LDP Specification , IETF RFC 3036

[ns] Kevin Fall and Kannan Varadhan, *ns* Notes and Documentation. Technical Report, UC Berkeley, LBL, USC/ISI, and Xerox PARC, November 1997.

[monarch] CMU Monarch Group. CMU Monarch extensions to *ns*. http://www.monarch.cs.cmu.edu/.

[multihop] Xu SG and Saadawi T. "Does the IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Ad Hoc Networks?", IEEE Communications Magazine. 39(6): 130-137 JUN 2001

[capacity] J. Li, C. Blake, D.S.J. De Couto, H.I. Lee, R. Morris. Capacity of Ad hoc Wireless Networks. Proceedings of ACM International Conference on Mobile Computing and Networking, August 2001