U.C. ECECS Technical Report 2005-04
October 2005

A Novel Queue Management Mechanism for IEEE 802.11s based Mesh Networks

by

Nagesh S. Nandiraju, Deepti S. Nandiraju, Dave Cavalcanti, Dharma P. Agrawal

† Center for Distributed and Mobile Computing,

Dept. of ECECS, University of Cincinnati - Cincinnati,OH 45221

(nandirns, nandirds, cavalcdt, dpa)@ececs.uc.edu

UNIVERSITY OF Cincinnati

Abstract

Abstract – Wireless Mesh networks exploit multi-hop wireless communications between Access Points to replace wired infrastructure. However, in multi-hop networks, effective bandwidth decreases with increasing number of hops, mainly due to increased spatial contention. Longer hop length flows suffer from extremely low throughputs which is highly undesirable in the envisioned scenarios for mesh networks. In this paper, we show that Queue/Buffer management, at intermediate relay mesh nodes, plays an important role in limiting the performance of longer hop length flows. We propose a novel queue management algorithm for IEEE 802.11s based mesh networks that improves the performance of multihop flows by fairly sharing the available buffer at each mesh point among all the active source nodes whose flows are being forwarded. Extensive simulations reveal that our proposed scheme substantially improves the performance of multihop flows. We also identify some important design issues that should be considered for the practical deployment of such mesh networks.

1. Introduction

Recently, Wireless Mesh networks are gaining increasing attention for their flexibility and ease of deployment. Wireless mesh networks aim to provide broadband Internet access to residential areas and offices by replacing the wired backbone with a wireless backhaul network. In such networks, the Access Points (APs) communicate wirelessly and forward traffic in a hop-by-hop fashion (similar to the Internet routers) in order to reach the Internet gateways (APs connected to the wired backbone to provide access to the Internet). Increasing commercial interests in wireless mesh networks has prompted the IEEE to setup a new task group (802.11s) for formalizing the PHY and MAC layer standards. Figure 1 illustrates a simple mesh network scenario.

The idea of mesh networks is similar to multi-hop ad hoc networks, where intermediate nodes act as routers by forwarding data between nodes that are not within the transmission range of each other. This kind of cooperative behavior helps in extending the network coverage without requiring any additional infrastructure. Unfortunately, in multi-hop networks, effective bandwidth decreases [9] with the increase in number of nodes and/or hops. Moreover flows spanning multiple hops experience dismal throughput performance compared to flows traversing fewer hops, thus leading to a spatial bias. Hidden and exposed terminal problems substantially contribute to this spatial bias. Another important factor that also exacerbates this spatial bias is the link layer buffer/queue management scheme at the intermediate nodes. Current queuing mechanisms do not take into account the number of hops a packet has traversed when inserting it into the link layer queue. This results in severe unfairness and starvation of flows spanning multiple hops. Whenever a packet has to be forwarded by a mesh point (APs), the routing process sends the packet down to the link layer, which inserts the packet into the interface queue (IFQ). Generally, the link layer uses a Drop tail queue management scheme in which newly arriving packets are dropped if the queue is full, regardless of the number of hops the packet has already traversed. Typically, when the mesh APs generate similar traffic loads, the packets from neighboring nodes or smaller hop length flows arrive more frequently at APs and fill up the link layer buffer. Thus, when packets from far away nodes (longer hop length flows) arrive at intermediate APs, they often find the buffer already filled and are eventually dropped. The arrival rate of forwarding traffic is usually controlled by the underlying channel access mechanism. For instance, when the CSMA/CA based MAC is used in a multi-hop scenario, the channel access delay due to spatial contention contributes to reduce the arrival rate of packets from multi-hop flows. Additionally, due to the increased channel access delays, the link layer IFQ fills up pretty quickly, even with moderate traffic loads. As a result, flows traversing multiple hops suffer inordinately as their packets are often dropped at intermediate nodes due to lack of buffer space. On the other hand, packets traversing smaller number of hops and packets originated at the node have less dropping probability and thus enjoy higher performance.

The envisioned goal of wireless mesh networks to replace the wired backbone implies an implicit requirement of unbiased treatment to all flows. Thus, there is a definite need for eliminating the aforementioned problem of spatial bias and provide an impartial service to all flows irrespective of the number of hops they traverse.

In this paper, we address this problem by proposing a novel queue management scheme for multihop networks (QMMN) in an attempt to boost the performance of longer hop length flows. First, we consider the following question: *Is CSMA/CA MAC solely responsible for the dismal performance of multihop flows in a wireless multihop network?* In other words, can we improve the performance of multihop flows without changing the MAC layer? We show that, along with the CSMA/CA MAC, buffer management also plays a key role in the degradation of throughputs for multi-hop flows. Then, we propose the QMMN algorithm to provide fair treatment to all flows. Our scheme limits the maximum buffer share occupied by source node at each intermediate mesh router and dynamically manages the unused buffer space to enhance utilization. Simulation results indicate that QMMN is indeed effective in protecting the longer hop length flows and substantially improving their performance. Although our proposed scheme achieves up to 4 fold increase in the throughputs of 3 hop flows, it is limited by the inherent shortcomings of the underlying MAC layer. Further, our proposed scheme requires no modifications to the MAC layer and thus can work in conjunction with any underlying MAC. In fact, the use of QMMN in conjunction with a fair MAC (that solves the hidden and exposed terminal problems) can provide further performance gains.

The rest of this paper is organized as follows. In Section 2, we describe some background information and show, through simulation, the unfair treatment received by the multi-hop flows. We describe our proposed link layer queue management scheme in Section 3 and provide a comprehensive simulation performance evaluation in Section 4. We discuss the related work in Section 5 and finally, conclude the paper in Section 6, highlighting some open problems and future research directions.

2. The Unfairness Problem in Multi-hop Networks

In this section, we briefly discuss the wireless mesh networks [2] and illustrate the existing unfairness problem in multihop wireless mesh networks through simulations in *ns-2* [6].

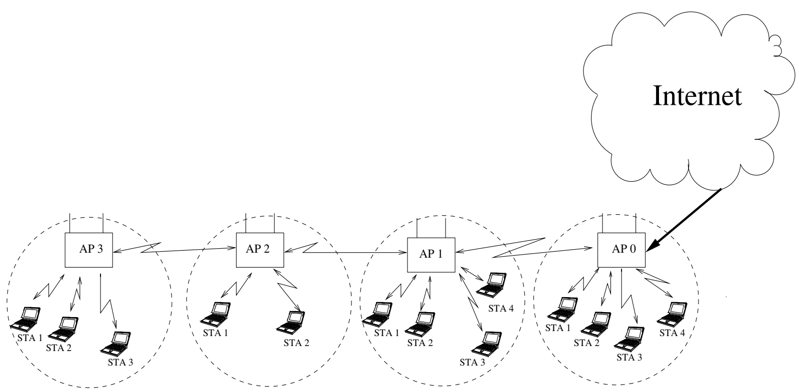

Figure 1:Simple Mesh Networking scenario

Commercial interests in mesh networks have prompted the IEEE 802.11 to setup a new task group (802.11s) to formalize PHY and MAC standards for these networks. The basic goal of this task group is to extend the existing 802.11 protocol and its architecture for supporting the mesh network paradigm. Several proposals to amend the 802.11 architecture are being considered and the 802.11s standard is expected to be completed before the end of 2006. The amended protocol should allow multiple Access Points communicate with each other wirelessly and in turn form a wireless distribution system.

We consider a simple IEEE 802.11s based mesh network (with four APs) as shown in Figure 1. These four APs (mesh points[1]) communicate with each other using the legacy IEEE 802.11 based interfaces, forming a wireless backbone. AP 0 is

[1] We use the terms AP and mesh point interchangeably

attached the gateway that provides Internet connectivity to the other APs. As assumed in [2], we also consider the mesh points communicate with their serving nodes (mobile stations) using another 802.11 interface that works in a different channel. Thus, the communications between a mesh point and its serving nodes do not interfere with the communications between two mesh points.

We assume that all mobile stations (STAs) employ IEEE 802.11 DCF operating at 2 Mbps with RTS-CTS handshake enabled. The radio propagation model used was the two-ray ground model with a transmission range of 250 m and carrier sensing range of 550m. Additionally, we employ static routing to avoid route failures. For ease of illustration, we consider that the STAs generate only UDP flows that are aggregated at the corresponding serving mesh points and forwarded towards the Internet gateway node (AP 0). This aggregate load is enough to backlog each AP and represents a worst case scenario in terms of packet dropping probability at AP. Without loss of generality, we assume a constant packet size of 1024 bytes for all the UDP flows.

Figure 2(a) shows the effective throughput (measured at AP 0) of the aggregate flows originating from each mesh point. As can be observed, there is a huge unfairness in the throughput achieved by different flows. Flows with increasing hop count (AP 2 and AP 3) suffer inordinately and their throughput almost dries up. Flows from AP 3 travel 3 hops to reach AP 0 and obtain a meager 9 kbps of total throughput while the flows from AP 1 enjoy a relatively high throughput of 566 Kbps.

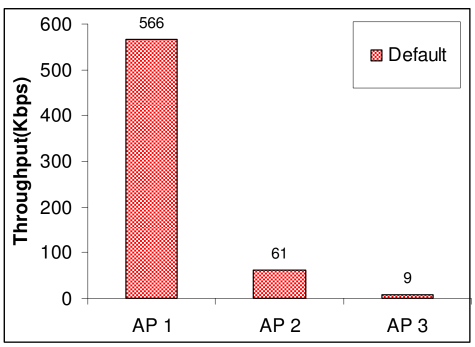

Figure 2(a): Aggregate Throughputs of flows from each AP

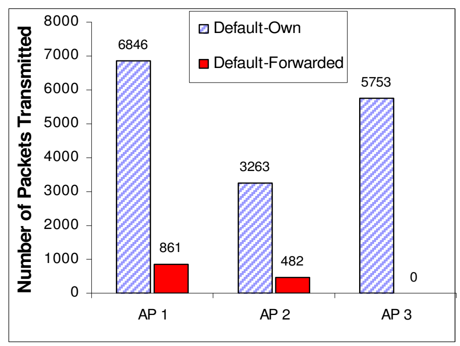

Figure 2(b): UDP Packets transmitted by each AP's MAC layer

Gambiroza et al. [2] report similar results and identify the hidden and exposed terminal problems as the reasons for such unfairness. However, the link layer buffer management scheme also plays an important role in this unfairness. In order to illustrate this effect, we show the number of packets transmitted by each mesh point in Figure 2 (b). As can be observed, AP 3 transmits 5753 UDP packets, out of which only 482 are forwarded by AP 2, while the remaining packets are dropped at the link layer due to lack of buffer space. The local traffic at AP 2 (from its corresponding STAs) arrives more frequently and occupies a major share of the buffer. Packets from AP 3 are often dropped either at AP 2 or AP 1 due to buffer overflow. As it turns out, only 118 packets (equivalent to 9 Kbps) of the flows from AP 3 finally make to AP 0. Clearly, this unfair sharing of the buffer at each intermediate node aggravates the already existing unfairness problem introduced by the hidden and exposed terminal problems.

Another interesting and important observation to make is the relatively small number of packets transmitted by AP 2, which affects the throughput of AP 2 and AP 3. There are two reasons for this unfair performance. First, AP 0 does not have any traffic to transmit and thus whenever AP 1 transmits an RTS, AP 0 immediately responds with a CTS. In contrast, AP 2 has two contenders (AP 1 and AP 3) and thus it often looses the contention. The second and most important cause is the EIFS deferring period imposed by the IEEE 802.11 DCF MAC [1]. Whenever a STA hears a transmission which it cannot decode, the STA should defer for an EIFS time period. But, it cancels the EIFS deference whenever it receives another packet that can be properly decoded. This protection was effective in WLANs operating in infrastructure mode, where no two STAs are more than 2 hops away of each other. However, as our results illustrate, it proves to be very inefficient and leads to unfairness in multi-hop networks. In our example, AP 0 is not in the transmitting range, but within the sensing range of AP 2. Then, whenever AP 1 sends a DATA packet, the corresponding ACK from AP 0 cannot be decoded by AP 2. This forces AP 2 to defer for an EIFS period. Before AP 2 can again start contending for the channel, either AP 1 or AP 3 captures the channel with a higher probability (since EIFS includes the transmission time of an ACK at basic rate, SIFS and DIFS). Therefore, AP 2 gets fewer transmit opportunities compared to AP 3 or AP 1, and even though AP 3 transmits considerable number of packets they are eventually dropped at AP 2 due to buffer overflow.

3. Queue Management in Multihop Networks (QMMN)

Experiments in the previous section indicate that unfair sharing of the buffer at intermediate nodes leads to significant throughput degradation of longer hop length flows. In order to solve this problem, we propose a novel link layer queue management scheme in multihop networks (QMMN). Our main goal is to guarantee a fair buffer share at each intermediate mesh point, for all flows traversing the mesh point, irrespective of their hop length.

In the QMMN algorithm, an arriving packet at any mesh point is either admitted into the queue or dropped depending on its source node's granted buffer quota. If the source has fewer packets buffered than its maximum allowable, the packet will be admitted, otherwise the packet will be dropped if there is no residual buffer space available. Thus, aggressive nodes can not overwhelm other peer nodes by merely increasing their traffic generation rate, because QMMN limits the allocated buffer shares, protecting the flows from nodes multiple hops away.

Before we describe our scheme in detail, we first describe the data structures and variables used at each mesh point for maintaining the per-active-node state.

3.1. Data Structures

Each mesh point maintains a fairness table that contains information about all the active nodes that are routing their packets through it. The fields in the table are explained below:

Source Address: A source node which is having one or more flows routed through this mesh point.

Max_Share: This specifies the maximum allotted buffer share, in number of packets, for the corresponding *source*. The *Max_Share* is equal to the total buffer space divided by the number of sources in the fairness table.

Fair_Share: This specifies the fair share required by the *source* node at this mesh point. When an entry for the *source* is created *Fair_Share* is set to *Max_Share*. It is later adjusted according to the estimated arrival rate of the traffic from the *source* node.

Occupied_share: This gives the number of packets currently buffered for the corresponding *source* node.

Arrival_Rate: The estimated average arrival rate of packets from the corresponding *source* node.

In addition to the above per-active-node information, QMMN also maintains a global variable *residual_share*. This variable is the aggregate count of unused share (*max_share – fair_share*) for all such *sources* whose *fair_share* is less than *max_share*. Further description of this variable is deferred to the next subsection. A sample snapshot of the fairness table is shown in Table I.

Table I: Sample snapshot of Fairness table at AP 1

Source Address	Max Share	Fair Share	Occupied_ Share	Arrival rate
1	25	25	24	0.015
3	25	17	14	0.06

3.2. The QMMN Algorithm

The QMMN algorithm is summarized in Figure 3. Whenever a packet arrives at the link layer, the mesh point checks whether there is an entry for the corresponding *source* in the fairness table. If there is no entry for that *source*, a new entry is created. At this point the *max_share* for this source is computed as the total buffer space divided by the number of sources in the fairness table. We then initialize the *fair_share* to its *max_share* and recalculate the *fair_share*'s for other *sources* in the table.

Although we initialize the *fair_share* for a *source* to its *max_share*, it is important to note that not all sources may occupy their *max_share* of the buffer. This is because the inter-arrival times of packets belonging to far away sources may be low and/or the nodes may generate traffic at different rates. Therefore, simply reserving equal buffer share for all active source nodes may underutilize the available resources, consequently, decreasing the overall throughput of the network. To address this problem, we estimate the average arrival rate of the traffic from each *source* and use this information in updating the fair buffer share (*fair_share*) needed by each *source* at each mesh point. The average arrival rate and service time are estimated using moving averages, as follows:

$$arrv\_rate = \alpha * arrv\_rate + (1 - \alpha) * curr\_arrv \ , \tag{1}$$

$$serv\_time = \alpha * serv\_time + (1 - \alpha) * serv\_time \ . \tag{2}$$

We calculate the *new_fair_share* for a given source as:

$$new\_fair\_share = \alpha * old\_fair\_share + (1 - \alpha) * \left(\frac{serv\_time}{arrv\_rate} \right) . \tag{3}$$

And then, we computed the *fair_share* as:

$$fair\_share = \mathbf{min}(max\_share, new\_fair\_share) \tag{4}$$

After extensive simulations, we found that $\alpha = 0.3$ provided good estimates for the actual arrival rates and service times.

Once we update the fair shares, the excess or unused buffer space (*max_share – fair_share*) by each *source* is accumulated to form the residual buffer share, which is calculated as:

$$residual\_share = \sum (max\_share_i - fair\_share_i) \ \forall i \in sources \cdot \tag{5}$$

An incoming packet will be deterministically enqueued if the *source* has fewer packets buffered than its *fair_share*. When the *occupied_share* exceeds the *source*'s *fair_share,* the packet is accepted, if the *source* has used less than *residual_share/num_sources* slots of the residual share, otherwise it is dropped. As can be noticed, this condition limits usage of the residual buffer space by a given node, and ensures that the residual share is not monopolized by any single source. Finally, whenever the MAC layer dequeues a packet from the buffer, the *occupied_share* of the corresponding *source* is updated.

At any time, a mesh point can buffer more packets than its estimated *fair_share*, only if there is residual space available. In this way, the *residual_share* can be temporarily utilized by the *sources* that generate higher traffic loads and thus preventing any underutilization of the buffer. In other words, the *residual_share* is introduced to account for fast variations in the traffic loads that cannot be instantaneously captured by the estimated fair share.

The main idea behind the QMMN algorithm is to drop packets from aggressive *source*s in order to protect the flows that have traversed multiple hops. Consider the case in which each mesh point generates the same traffic load. Hence, packets traversing multiple hops have relatively low arrival rates at the relaying mesh points due to contention delays at multiple nodes, and seldom exceed their granted share. However, packets from local or nearby sources may arrive more frequently and quickly consume their source's share. Allowing such packets into the buffer may require dropping of packets that have already traversed several hops and consumed considerable network resources. Therefore, it is reasonable to drop the packets from near by sources. In this way, QMMN ensures that no aggressive source monopolizes the buffer at any node.

```
When a packet p arrives:

If(p->source is not in fairness_table)
    Create an entry for source
    Recalculate the Max_share and Fair_Share for
    each node
    Update the residual_share accordingly
    If a free space is found in the buffer,
    enqueue the packet p and update the
    occupied_share

Else If(p->source is in the fairness_table)
    Update the average inter-arrival rate for the
    source
    Recalculate the fair_share and update the
    residual_share.

    If(occupied_share < fair_share)
      Add the packet p to the queue
      Update the occupied_share
    Else
      used_residue = occupied_share - max_share;
      If(residual_share>0 &&
        used_residue<residual_share/num_sources))
              Add the packet p to the queue
              occupied_share++;
              residual_share--;
      Else
              Drop the packet p
      End If
    End If
End If
```

Figure 3. QMMN Algorithm

4. Performance Analysis

In this section, we compare the QMMN algorithm with the default link layer drop tail queue through simulations performed in *ns-2* (version 2.26) [6]. We have used the scenario shown in Figure 1 and the same configuration described in Section 2, in which each AP generates traffic towards the Internet gateway (AP 0). Other simulation parameters are summarized in Table II.

Table II: Simulation Parameters.

Parameters	**Value**
Packet Size	1024 bytes
Simulation Time	100 seconds
Transmission Range	250m
Carrier sensing range	550m
Bandwidth	2 Mbps
IFQ Size	50
Radio Propagation Model	Two Ray Ground
MAC Protocol	IEEE 802.11
Transport/Application Protocol	UDP(CBR), TCP(FTP)

4.1. Performance with UDP flows

We first consider each mesh point has backlogged UDP traffic and measure the performance of the aggregate flows from each mesh point.

Figure 4(a) shows the aggregate throughput of the three APs using the default queue management and using our proposed QMMN algorithm. As we can see from the Figure, with the default queue management, AP 1 achieves relatively high throughput, while AP 2 and AP 3 starve. As explained earlier, in Section 2, these unacceptably low throughputs are partly due to the unfair buffer sharing. Under high load, local flows from AP 1 quickly fill up the link layer buffer at this mesh point. When the packets from AP 2 and AP 3 arrive at AP 1, they find a full buffer and are dropped. For similar reasons, AP 2 drops the packets from AP 3. Thus, flows from AP 3 experiences a drastic decrease in their throughput.

On the other hand, with the QMMN algorithm, it can be observed that the buffer reservation at intermediate mesh points indeed improves the performance of multihop flows. AP 2 and AP 3 now achieve substantially higher aggregate throughputs. It is important to note that QMMN achieves this improvement by throttling the flows from AP 1. With QMMN, all nodes have their guaranteed buffer share at each intermediate mesh point, and the packets traversing a longer multi-hop route are not often dropped due to buffer overflow. Thus, AP 2 forwards fair number of AP 3's packets (see Figure 4(b)) out of its total MAC layer transmissions. As a result, both AP 2 and AP 3 achieve similar throughputs.

In spite of the reserved buffer for AP 2 and AP 3 at AP 1, flows from AP 1 continue to enjoy relatively higher throughput than other flows (from AP 2 and AP 3). As explained in Section 2, AP 2 often fails to win the channel contention due to the EIFS deferring, and gets fewer chances to transmit. Therefore, AP 1 receives relatively fewer packets to be forwarded from both AP 2 and AP 3 and thus, a smaller share is provided for them at AP 1. Even if we divide the buffer at AP 1 equally among all three mesh points, flows from AP 2 and AP 3 cannot utilize their share. As a consequence they receive relatively lower throughputs compared to AP 1.

We also evaluate the adaptive nature of QMMN by considering different loads at the three APs. In particular, we assume UDP traffic loads of 1200, 800 and 100 kbps at AP 1, AP 2, and AP 3, respectively. Figure 4(c) shows the aggregate throughputs of the three APs measured at AP 0. As we can see, once again, with the default queue management, AP 3 gets only a meager 9 Kbps of its 100 Kbps traffic while AP 2 gets around 138 kbps. In contrast, QMMN boosts the throughput of flows from AP 3 by almost 10 times. This improvement is achieved by avoiding the packet drops of flows from AP 3 at AP 2 and AP 1. However, it is worthwhile to note that AP 2 doesn't gain any improvement for its flows due to the unfairness caused by MAC layer, as explained above.

In order to illustrate the efficacy of our estimation of the fair share we show the allocated fair share and the buffer occupancy of AP 3's traffic at AP 1 in Figure 4(d). As can be seen, our estimated fair share for AP 3 closely encompasses the buffered packets. In this way, QMMN prevents an underutilization of the buffer.

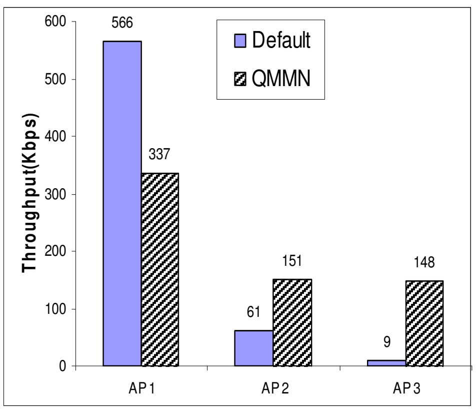

Figure 4(a): Aggregate Throughput of UDP flows with Uniform Load

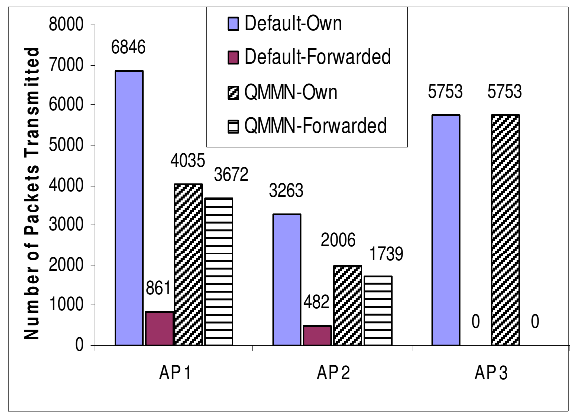

Figure 4(b): Packets transmitted by each AP (backlogged traffic)

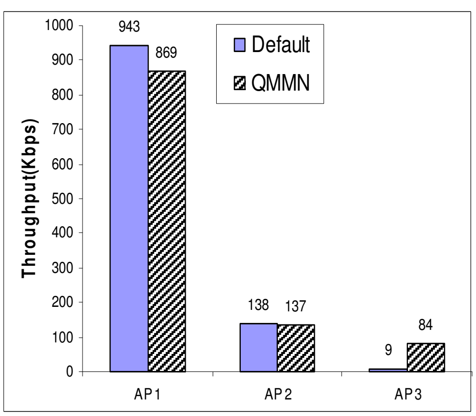

Figure 4(c): Aggregate Throughput of UDP flows with Non-Uniform Load

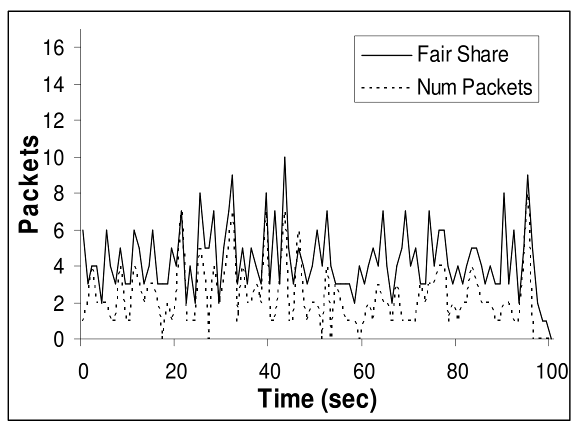

Figure 4(d): AP 3's Estimated Fair Share and Packets buffered at AP 1.

4.2. Performance with one TCP flow with interfering UDP traffic

In this section, we examine the impact of the QMMN algorithm on the TCP performance. We consider a single TCP flow at AP 3 and an interfering UDP traffic from AP 1. Both flows are routed towards the Internet gateway node (AP 0).

We measure the TCP throughput for varying interfering UDP traffic loads at AP 1. Figure 5(a) and Figure 5(b) show the TCP and the UDP throughputs, respectively. As can be observed, with the default queue management, the TCP throughput degrades linearly with the increasing load at AP 1 (Figure 5(a)). In contrast, the UDP flows (from AP 1) enjoy almost linear throughput increase (Figure 5(b)). With the default scheme, the TCP throughput is as low as 3 Kbps at high loads at AP 1. On the other hand, QMMN enables the TCP flow to achieve as high as 255 Kbps, even at higher loads.

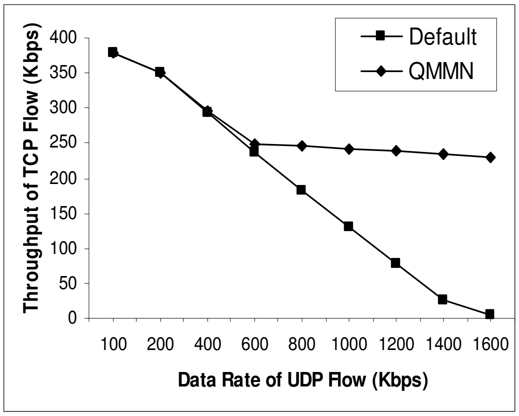

Figure 5(a): Throughput of TCP flow from AP 3

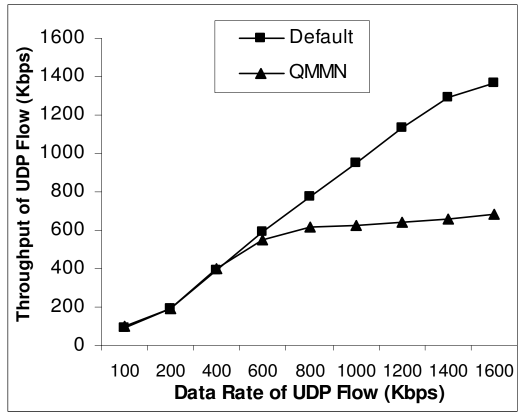

Figure 5(b): Aggregate Throughput of UDP traffic from AP 1

The reason behind the poor performance with the default scheme can once again be attributed to the unfair sharing of the buffer at AP 1. In this case, since AP 2 doesn't have any packets to transmit other than the TCP flow's packets; it forwards all the packets to and from AP 3. However, at AP 1, the UDP packets and TCP packets contend for the buffer. As explained earlier, since the TCP packets have to traverse 2 hops to reach AP 1, their arrival rate becomes relatively low when compared to the UDP packets arrival rate, the TCP packets are often dropped due to buffer overflow.

On the other hand, QMMN effectively protects the TCP flow and improves its performance. Even with the increase in the traffic load, the TCP flow is not affected and maintains almost a constant throughput. QMMN achieves this improvement by throttling the UDP flow at AP 1 and strictly making the reserved buffer at AP 1 available to the TCP flow.

QMMN allows the TCP congestion window to grow, which increases the throughput. Figure 6 depicts the congestion window growth of the TCP flow with different traffic loads at AP 1. With the default queue management, when the AP1's traffic load is 400Kbps (denoted by Default-400), the congestion window grows smoothly without any cuts. However, when the incoming UDP traffic at AP 1 is increased to 800 Kbps, more and more TCP packets are dropped. As a result, regular timeouts occur at AP 3, causing frequent congestion cuts (see Default-800 in Figure 6), which results in very low throughput.

In contrast, the QMMN algorithm does not allow the UDP flow at AP 1 to affect the TCP flow. The TCP flow is now stable even at higher loads at AP 1. The unaffected growth of the congestion window with QMMN can be seen in Figure 6. It is important to note that, no noticeable difference can be observed in the congestion window growth for different traffic loads at AP 1.

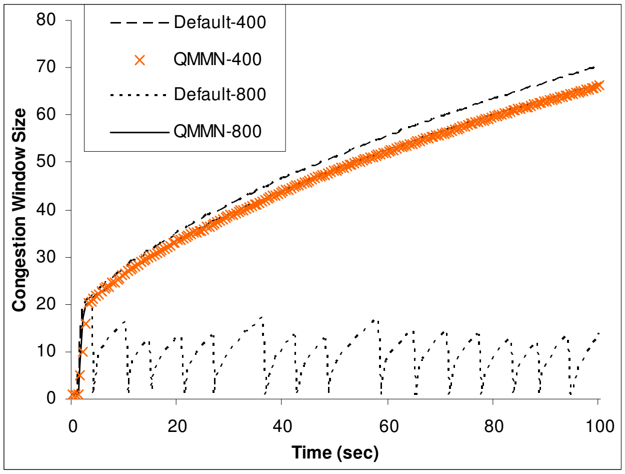

Figure 6: Congestion window growth of the TCP flow from AP 3

5. Related Work

Most existing queue management schemes have been designed for controlling congestion in the Internet routers. For instance, RED [4] was designed to control congestion and avoid queue buildups at Internet routers by measuring the average queue length. RED maintains two thresholds viz. min_{th} and max_{th}. Packets are enqueued till the queue size is below min_{th} and are dropped as the queue size exceeds max_{th}. When the queue size ranges between min_{th} and max_{th}, packets are either admitted or dropped with probability that increases with queue size. Admitted packets are marked to indicate congestion for the source nodes to take immediate action. However, there is no flow level differentiation and thus, it is not effective in controlling non responsive flows. Fair Random Early Detection (FRED) [5] is a method for differentiating (classifying) incoming flows. The authors in [5] suggest saving per-active flow information in the gateway buffer to decide about the fair buffer share for the flows. Then, packets are dropped if the flows exceed their share.

Although the above schemes address the deficiencies of drop tail queuing, they do not consider various issues inherent to multi-hop wireless networks. For example, in a multihop network, it may be better to give priority to packets arriving from farther hops when compared to near by traffic, as they would have already consumed fair amount of bandwidth. Dropping such traffic that has traversed multiple hops will result in wastage of valuable network resources.

Jun and Sichitiu [7] have investigated several queuing schemes at intermediate relay nodes for achieving fairness in multihop wireless networks and suggested maintaining a separate queue for each individual source at all relaying nodes. Maintaining a separate queue for each individual source is difficult and may render infeasible with the dynamic nature of traffic. Recently, Gambiroza et al. [2] have addressed unfairness problem in multihop mesh networks by proposing an Inter-tap fairness algorithm in which the nodes exchange channel usage information and decide their maximal channel access times. Yi and Shakkotai [8] developed analytical models for hop by hop congestion control in ad hoc networks and propose layer 2 congestion control mechanisms for controlling the traffic load generated at the source nodes. However, the schemes proposed in [2] and [8] do not consider the link layer buffer management. In contrast, the main focus of our work is to expose the role of buffer management in the dismal performance of multihop flows. QMMN alleviates the improper buffer sharing at intermediate nodes and in turn improves the performance of longer hop length flows.

6. Conclusion

In this paper, we have shown that buffer management at the intermediate nodes also plays a vital role in degrading the performance of multihop flows. We propose a novel queue management scheme for multihop networks (QMMN) to protect the longer hop length flows. Simulation results show that, QMMN substantially improves the performance of multihop flows by controlling the throughput of short hop length flows. It is important to notice that the throughput of multi-hop flows is also constrained by the inherent unfairness of the IEEE 802.11 MAC layer. Therefore, resolving channel access issues in multihop networks would enable QMMN to further improve the performance of the multihop flows. To investigate these issues is part of our future work.

Acknowledgements

This work has been partially supported by the Ohio Board of Regents, Doctoral Enhancement Funds.

References

[1] IEEE Std. 802-11. "IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification," ISO/IEC 8802-11:1999 (E), Aug., 1999.

[2] Bruno R., Conti M., and Gregori E., "Mesh Networks: Commodity Multihop Ad Hoc Networks," IEEE Communications Magazine, Vol. 43, No. 3, March 2005

[3] Gambiroza V., Sadeghi B., and Knightly E. W., "End to End Performance and Fairness in Multihop Wireless Backhaul Networks," In *Proceedings of ACM* Mobicom 2004

[4] Floyd, S., and Jacobson, V., "Random Early Detection gateways for Congestion Avoidance," IEEE/ACM Transactions on Networking, vol.1, no.4, August 1993.

[5] Lin D., and Morris R., "Dynamics of Random Early Detection," In *Proceedings of ACM* SIGCOMM 97.

[6] UCB/LBNL/VINT Network Simulator (NS-2), Available at http://www.isi.edu/nsnam/ns/index.html.

[7] Jun J. and Sichitiu M. L., "Fairness and QoS in multihop wireless networks," in *Proc. Of the IEEE Vehicular Technology Conference (VTC)*, October 2003.

[8] Yi Y. and Shakkottai S., "Hop-by-hop congestion control over a wireless multi-hop network," In *Proceedings of IEEE INFOCOM '04*, Hong Kong, March 2004.

[9] Akyildiz I.F., X. Wang, "A survey on wireless mesh networks," IEEE Communications Magazine, Sept. 2005.