

A Cache Based Traffic Regulator for Improving Performance in IEEE 802.11s based Mesh Networks

Nagesh S. Nandiraju, Deepti S. Nandiraju, Lakshmi Santhanam, Dharma P. Agrawal

Center for Distributed and Mobile Computing, Dept. of ECECS, University of Cincinnati - Cincinnati, OH 45221
(nandirms, nandirds, santhal, dpa)@ececs.uc.edu

Abstract – Wireless Mesh Networks (WMNs) are evolving to be the key technology of the future. They aim to provide broadband wireless internet service to a large community of users by exploiting multi-hop wireless communications between Access Points thus replacing wired infrastructure. However, in multi-hop networks, flows spanning multiple hops suffer from extremely low throughputs compared to flows traversing fewer hops. In particular, some shorter hop length flows can severely degrade the performance of flows traversing more hops. This is highly undesirable in the envisioned scenarios of mesh networks. In this paper, we propose an elegant *most frequently seen* cache based traffic regulator that identifies and appropriately regulates the rate of such flows. Extensive simulations reveal that our proposed scheme effectively identifies and controls the traffic from aggressive flows and helps in substantially improving the performance of longer hop length flows.

Keywords: Multihop Networks, End to End Fairness, Queue Management, Performance Evaluation.

1. Introduction

Recently, Wireless Mesh Networks (WMNs) [2][9][10] are gaining increasing attention due to their flexibility and ease of deployment. WMNs 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) also referred to as Mesh point (MP), communicate wirelessly and forward each other's traffic in a multi-hop fashion towards the Internet gateways (IGW). This kind of cooperative behavior helps in extending the network coverage without employing any additional infrastructure. Figure 1 illustrates a simple mesh network scenario. Increasing commercial interests in WMNs has prompted the IEEE to setup a new task group (802.11s) for formalizing the PHY and MAC layer standards to support 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 2008.

Unfortunately, in multi-hop mesh networks, flows spanning multiple hops experience dismal throughput performance compared to flows traversing fewer hops, leading to a spatial bias [7]. This in turn leads to severe unfairness and starvation of such flows. As the packets from far away nodes are relayed by intermediate nodes, they undergo channel contention at each hop which is controlled by an underlying channel access mechanism like CSMA/CA based MAC. Thus, the inter-arrival times of packets belonging to longer hop length flows is usually very high. Moreover increased channel contention can result in filling up the IFQ pretty quickly even with moderate traffic loads. Hence, the flows traversing multiple

hops suffer inordinately as majority of their packets are often dropped at the intermediate nodes due to lack of buffer space. On the other hand, the shorter hop length flows enjoy higher performance as they dominate the space in the IFQ of APs near the IGW.

Popular queuing mechanisms such as Random Early Detection (RED) [4] and Fair RED (FRED) [5] are proposed for wired networks to control the domination of aggressive flows. Although these schemes effectively address the deficiencies of drop tail queuing, their effectiveness in a wireless network is debatable. Firstly, the idea of randomly dropping packets belonging to different flows does not ensure that the performance of longer hop length flows will be improved. In a multihop mesh network, it would be more appropriate to give priority to longer hop flows as they have already consumed fair amount of network bandwidth. Secondly, RED cannot effectively control the rate of all non-adaptable flows such as UDP. Recently, Gambiroza et al. [3] have addressed the 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 proposed a layer 2 congestion control mechanisms to control the rate of traffic generation at the source nodes. However, the schemes proposed in [3] and [8] do not consider the link layer buffer management. In [9] we illustrated the role of buffer management in the dismal performance of multihop flows and proposed a fair buffer sharing algorithm at the intermediate nodes to improve the performance of longer hop length flows. However if there are multiple flows originating from the same mesh point, it cannot ensure fair sharing of the buffer between these flows.

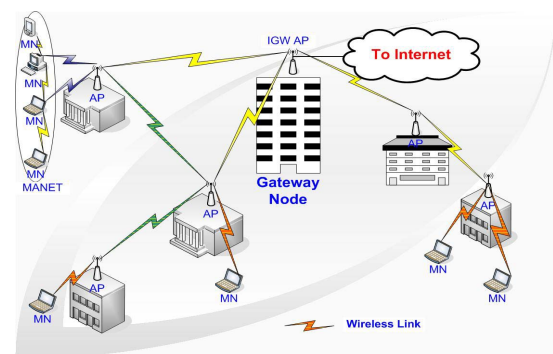


Figure 1: Simple Mesh Networking scenario

In this paper, we address this problem by proposing a novel mechanism to identify aggressive flows based on most

frequently seen cache discipline and then propose an effective rate control algorithm that regulates the traffic from aggressive flows. This ensures that the buffer at intermediate nodes is fairly shared by all the flows passing through that node and hence prevents the performance degradation of longer hop flows. We implement our proposed scheme just above the MAC layer. Hence it does not require any modifications in the firmware part of MAC and thus very flexible to deploy.

The rest of this paper is organized as follows. In Section 2, we illustrate the unfair treatment received by the multi-hop flows. We describe our proposed cache based traffic regulating mechanism in Section 3 and provide a comprehensive performance evaluation in Section 4. We finally, conclude the paper in Section 5, highlighting some open problems and future research directions.

2. The Unfairness Problem in WMNs

In this section, we conduct some experiments using *ns-2* [6] and illustrate the unfairness problem in WMNs. We setup an IEEE 802.11s based mesh network with four APs placed in a line topology and serving 16 end users/clients. These APs (a.k.a mesh points¹) communicate with each other using the IEEE 802.11a [1] based interfaces, forming a wireless backbone. AP 0 functions as the IGW for the other APs. The APs are also equipped with an 802.11b interface to communicate with the end users (mobile stations). We assume that all mobile stations (STAs) employ IEEE 802.11 DCF operating at 2 Mbps with RTS-CTS handshake enabled.

Three clients (Client 1 and client 2 under AP 3 and client 3 from AP 1) generate UDP traffic at different rates and are started at different times. Client 1 starts traffic (*flow-1*) at a rate of 150 Kbps at the start of simulation time. Figure 2 shows the measured instantaneous throughput of all the flows. As we can see, it enjoys full throughput until *flow-2* is started. At time 20 seconds when client 2 starts *flow-2* with a rate of 500 Kbps, we find that the throughput of *flow-1* is marginally affected. However, as we start the aggressive flow (*flow-3*) from client 3 at time 50 seconds, we can notice the dismal performance of both clients 1 and 2. It is clear that when *flow-3* enters the network, it causes serious starvation of both the longer hop length flows (*flow-1* and *flow-2*).

This severe performance degradation can be attributed to the underlying queuing mechanism. Usually nodes employ drop tail queuing mechanism. Whenever a packet arrives at an intermediate queue, it will be admitted into the queue if there is space. If the queue is full, the packet is simply dropped. Since the drop tail mechanism does not consider the number of hops a packet has traversed while inserting it into the IFQ, the packets belonging to aggressive flows will fill up the buffer. On the other hand, packets arriving from far away nodes have higher inter-arrival times and will often find the buffer at intermediate nodes already filled up. As a result these flows like those from clients under AP 3 obtain very low throughput.

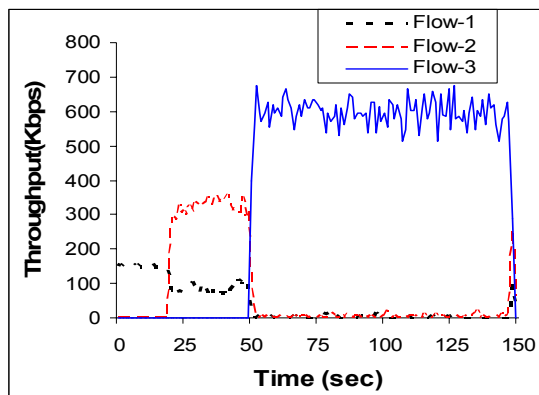


Figure 2: Instantaneous Throughputs of flows

3. Cache Based Traffic Regulator (CBTR)

Experiments in the previous section indicate that some aggressive flows traversing fewer hops can grab an unfair share of the buffer at the intermediate nodes leading to starvation of flows traversing longer hops. This means that in a mesh network scenario, proximity of the serving MP to IGW decides the throughput a mesh client would receive. We also observed that an aggressive flow spanning more number of hops does not affect the shorter hop length non-aggressive flows. The packets originating from the MP that is further away will have relatively high inter-arrival times at the relaying mesh points. This is because these packets have to traverse multiple hops and at each relaying MP, they experience the channel contention delays. Thus, even a high traffic rate flow becomes non-aggressive in nature as it traverses more number of hops. However, packets from local or nearby clients (have to traverse fewer hops) may arrive more frequently at the MPs if the nearby clients generate at high rate. Admitting such packets would fill up the buffer quickly, resulting in no space for longer hop length flows when they arrive eventually. In order to solve the unfairness problem, we propose a mechanism to identify the high rate flows that can affect the longer hop length flows and regulate the traffic of such flows. Our main goal is to provide a fair treatment or opportunity to all flows irrespective of their hop length. Our algorithm is divided into two subparts: **Most Frequently Seen Cache** module and the **Rate Regulator** module. The **most frequently seen cache** module identifies the aggressive flows at each node. Then the **rate regulator** appropriately controls the traffic from these aggressive flows.

3.1. Identifying aggressive flows

We propose to use a cache based mechanism to identify the aggressive flows. Each AP maintains a cache table that contains information about the most active flows that are currently routing their packets through it. In addition to the per-active-flow information, we also maintain a per node variable called *drop_probability* which is initially set to 0. This is used to determine whether a packet from an aggressive flow should be admitted into the queue or not.

¹ We use the terms AP, mesh router and mesh point interchangeably

Whenever a packet arrives at a node, if the flow to which this packet belongs, is already present in the cache, the frequency count of the flow is updated. Otherwise, a new entry is created in the cache for this flow. However if the cache is full, the flow that has minimal frequency count in the cache is replaced with the new flow. As the packets from an aggressive flow arrive frequently, they are more likely to be present in the cache with a large frequency count.

3.2. The Rate controller

After identifying the aggressive flows, it is the responsibility of the rate controller to regulate the traffic from the aggressive flows. If a packet belongs to non-aggressive flow, then it will be admitted provided there is space in the queue. However, if the queue is full, this packet has to be dropped; upon which we increment the *drop_probability* at this node. Thus, when a packet belonging to an aggressive flow arrives, even if there is space in the IFQ, it is dropped or admitted depending on the drop probability at this node.

As a non-aggressive flow has a larger inter-arrival time, it is less likely to be affected due to the increase in *drop_probability*. On the other hand, the shorter hop aggressive flows are more likely to be affected by this. Thus, our scheme is a safe defense against aggressive flows only and would not have a negative effect over non-aggressive flows. In fact, it makes room for the non-aggressive flows. Thus, the rate controller module prevents the aggressive flows from overwhelming the longer hop length non-aggressive flows.

Once the system stabilizes and the non-aggressive flows start getting fair treatment, CBTR dynamically adapts to the changes in the network by decrementing the *drop_probability* to favor the aggressive flows. Otherwise the high *drop_probability* could unnecessarily control the rate of aggressive flows even when they are not affecting the non-aggressive flows. CBTR employs a multiplicative increase for the *drop_probability* and therefore controls the rate of aggressive flows with a very low reaction time. However, we advocate a linear decrease for the *drop_probability* so that the aggressive flows remain under control for an appropriate time period and does not fill up the buffer quickly.

Also, when the system has stabilized and the average queue size is low, the incoming packets will often find space in the buffer. In such a scenario, the *drop_probability* may not be decremented, in which case the rate controller would continue to regulate the aggressive flows. To prevent such situations, we check the last time when there was a necessity to increment the *drop_probability*. If the time difference is more than a certain threshold, we decrement the *drop_probability*.

4. Performance Analysis

In this section, we evaluate the performance of our algorithm using the scenario and configuration described in Section 2. We also tested the effectiveness of our scheme in a random scenario with 20 APs but due to shortage of space we only

present the former scenario's results. For more comprehensive results please refer to our technical report [11].

4.1. Performance with UDP flows

In order to investigate the effectiveness of our proposed mechanism, we consider the same setup of flows as described in Section 2. Figure 3 shows the instantaneous throughput of all the flows. Initially *flow-1* (non-aggressive) from AP 3 successfully obtains a throughput of 150 Kbps which is its traffic generation rate. When *flow-2* (that has 500 Kbps traffic generation rate) starts at time 20 sec from another client under AP 3, some packets from *flow-1* are dropped due to which the drop probability is increased. Our algorithm quickly identifies that *flow-2* is an aggressive flow and starts applying the rate regulation for *flow-2*. As a result, *flow-2* is not allowed to affect *flow-1* which now gets its fair share of the bandwidth. At time 50 seconds when another aggressive flow from client 3 under AP 1 starts, the buffer at AP 1 is predominantly occupied by *flow-3*'s packets. Thus the packets belonging to the flows from the distant AP 3 (*flow-1* and *flow-2*) often find the buffer at AP 1 already filled and are dropped due to lack of buffer space. Recall from previous section that whenever a packet from a non-aggressive flow is dropped at a node, the drop probability at that node is incremented. CBTR quickly classifies that *flow-1* and *flow-2* are non-aggressive and as a result the drop probability at AP 1 is rapidly increased. This drop probability is used when the next packet from an aggressive flow arrives. Consequently many packets belonging to *flow-3* are dropped until the system stabilizes and *flow-1* and *flow-2* get decent throughput. As we can observe from Figure 3, the longer hop length flows now get a fair share of the buffer and hence their throughput is improved substantially. *Flow-1* gets its bandwidth of 150 Kbps almost throughout its life time. Hence, we can clearly see that our scheme effectively shields the non-aggressive flows from non-adaptive aggressive flows by controlling their rate.

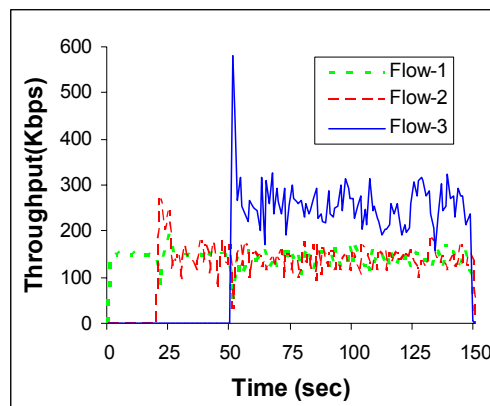


Figure 3: Instantaneous Throughputs of flows

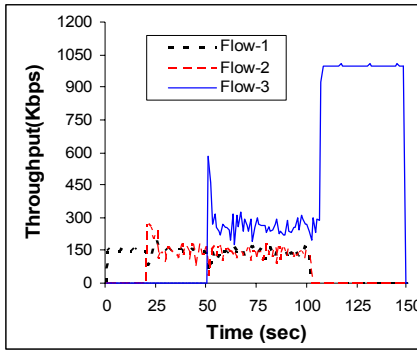


Figure 4(a): Effect of Decrement Drop Probability Step Size 0.05

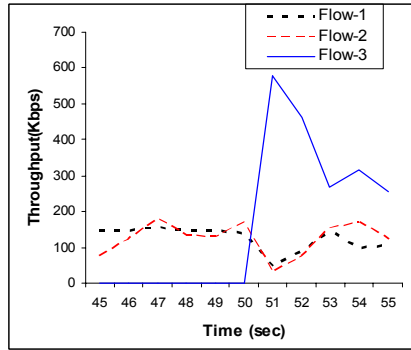


Figure 4(b): Initial value of Increment Drop Probability = 0.05

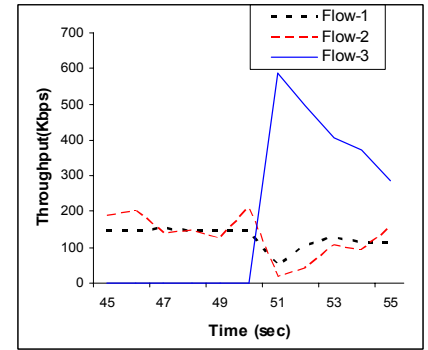


Figure 4(c): Initial value of Increment Drop Probability = 0.1

4.2. Impact of Drop Probability

We now examine how quickly CBTR adapts to the changes in the network. Figure 4(a) shows the instantaneous throughput of all the three flows. We start *flow-1* as soon as the simulation starts. *Flow-2* is started at 20 seconds and *flow-3* at 50 seconds. When we stop *flow-1* and *flow-2* at 100 seconds, the *drop_probability* is still high. To respond to such changes in the system, CBTR quickly decreases the *drop_probability*. Otherwise, it would continue regulating the rate of *flow-3* even when there are no other flows. This helps *flow-3* to improve its throughput and avoid any unnecessary packet droppings. It can be seen that *flow-3* is able to achieve a full throughput as soon as *flow-1* and *flow-2* are stopped at time 100 seconds.

We also observe the reaction time of CBTR with different initial values of the *drop_probability*. Figure 4(b) and 4(c) shows this impact of different starting values. When the *drop_probability* is set to 0.05, we observe that the reaction time (time taken to control the aggressive flow) is around 1 second (see Figure 4(b)). If we change this initial setting to 0.1, the reaction time decreases and the aggressive flow is controlled more quickly as shown in Figure 4(c).

Increasing the initial value can help decrease the reaction time, but it may lead to many false positives. After examining the overall performance of the network with various values, we conclude that an initial value of 0.05 is an optimal for achieving higher performance.

5. Conclusion & Future Work

In this paper, we have shown that some non-adaptable aggressive flows affect the performance of non-aggressive flows that may span multiple hops. We identify that buffer management at the intermediate nodes plays a vital role in this degradation and highlight the inadequacy of current queuing mechanisms. We introduce a novel cache based mechanism to identify the aggressive flows and propose a rate controller mechanism that limits the rate of aggressive flows and protects the performance of longer hop length flows. Simulation results show that, our algorithm substantially improves the performance of multihop flows by controlling the throughput of shorter hop length flows.

In the future, we plan to investigate the performance of our algorithm in the presence of TCP and UDP flows. Even an adaptive TCP flow would require a rate control algorithm as a shorter hop length flow might dominate over a longer hop length TCP flow. The longer hop length TCP flow detecting congestion in the network might cut down its congestion window resulting in very low throughput. We intend to apply our scheme for both adaptive and non-adaptive flows to ensure fair treatment of all flows.

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] R. Bruno., M. Conti, and E.Gregori, "Mesh Networks: Commodity Multihop Ad Hoc Networks," IEEE Communications Magazine, Vol. 43, No. 3, March 2005
- [3] V.Gambiroza, B.Sadeghi, and E.W.Knightly, "End to End Performance and Fairness in Multihop Wireless Backhaul Networks," In *Proceedings of ACM Mobicom 2004*
- [4] S.Floyd, and V.Jacobson, "Random Early Detection gateways for Congestion Avoidance," IEEE/ACM Transactions on Networking, vol.1, no.4, August 1993.
- [5] D.Lin and R.Morris, "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] J.Jun and M. L. Sichitiu, "Fairness and QoS in multihop wireless networks," in *Proc. Of the IEEE Vehicular Technology Conference (VTC)*, October 2003.
- [8] Y. Yi and S. Shakkottai, "Hop-by-hop congestion control over a wireless multi-hop network," In *Proceedings of IEEE INFOCOM '04*, Hong Kong, March 2004.
- [9] Nagesh S. Nandiraju, D. Nandiraju, D. Cavalcanti, D. P. Agrawal, "A Novel Queue Management Mechanism for Improving Performance of Multihop Flows in IEEE 802.11s based Mesh Networks," In *Proceedings of IPCCC-2006*.
- [10] I.F. Akyildiz, X. Wang, "A Survey on Wireless Mesh Networks," Proc of IEEE Radio Communications, September 2005.
- [11] Nagesh S. Nandiraju, D. Nandiraju, L. Santhanam, D. P. Agrawal, "A Cache Based Traffic Regulator for improving performance in IEEE 802.11s based Mesh Networks," Technical Report TR-005, University of Cincinnati.