End-to-End Flow Fairness over IEEE 802.11-based Wireless Mesh Networks

Ashish Raniwala Pradipta De Srikant Sharma Rupa Krishnan Tzi-cker Chiueh
Department of Computer Science, Stony Brook University
Stony Brook, NY 11794-4400, USA
Email:{raniwala, prade, srikant, krishnan, chiueh}@cs.sunysb.edu

***Abstract*—Economies of scale make IEEE 802.11 an attractive technology for building wireless mesh networks (WMNs). However, the IEEE 802.11 protocol exhibits serious link-layer unfairness when used in multi-hop networks. Existing fairness solutions either do not address this problem, or require proprietary MAC protocol to provide fairness. In this paper, we argue that an ideal transport protocol should be able to achieve fairness even on top of an unfair MAC layer such as 802.11. Towards this end, we propose a *co-ordinated congestion control* algorithm that performs global bandwidth allocation and provides end-to-end flow-level max-min fairness [1] despite weaknesses in the MAC layer. The proposed algorithm features an advanced topology discovery mechanism that detects the inhibition of wireless communication links, and a general collision domain capacity re-estimation mechanism that effectively addresses such inhibition. Through an ns-2-based simulation study we demonstrate that the proposed algorithm substantially improves the fairness across flows, eliminates starvation problem, and simultaneously maintains a high overall network throughput.**

I. Introduction

An ideal transport protocol efficiently utilizes all available network resources and allocates them fairly among competing flows even when the MAC protocol is inherently unfair. In wireless mesh networks (WMNs), there are intricate interdependencies among neighboring wireless links that greatly complicate the problem of fair bandwidth allocation. Previously proposed transport protocols for WMNs either fail to provide end-to-end fairness, or rely upon proprietary MAC protocols to schedule packet transmissions and achieve inter-flow fairness. Consequently, none of them qualifies as a satisfactory transport protocol for 802.11-based WMNs.

Economies of scale make IEEE 802.11 an attractive technology for building wireless mesh networks. The MAC protocol of IEEE 802.11, however, introduces serious unfairness among competing nodes when used in multi-hop WMNs [1]. The well-known *hidden node problem* [13] causes one wireless link's transmission to be inhibited by another link, eventually leading to unequal bandwidth allocation between the two. More specifically, while the RTS/CTS messages in 802.11's MAC protocol effectively stop a hidden node from interfering with an on-going communication transaction, they cannot prevent the hidden node from initiating its RTS/CTS sequence at inopportune times and subsequently suffering from long backoff delays. TCP exacerbates this unfairness problem because TCP senders further back off when their packets take a long time to get through the inhibited links. As a result, TCP flows traversing on an inhibited link could be completely suppressed in the worst case. Another related fairness problem in multi-hop WMNs is that when two TCP flows share the same wireless link, the flow traversing a fewer number of hops tends to acquire a higher share of bandwidth. In a multi-hop WMN, this translates into smaller bandwidth share to nodes that are farther away from wire-connected gateways. Finally, existing transport protocols at best attempt to allocate a radio channel's bandwidth fairly among flows from a single node, rather than among all flows from all nodes that share the radio channel. As a result, a flow emanating from a node with fewer flows tends to get a larger than fair share of channel bandwidth. Figure 1 qualitatively illustrates these unfairness problems with simple examples, whereas Figure 2 quantifies the extent of unfairness by presenting the throughput of the participating flows in each instance.

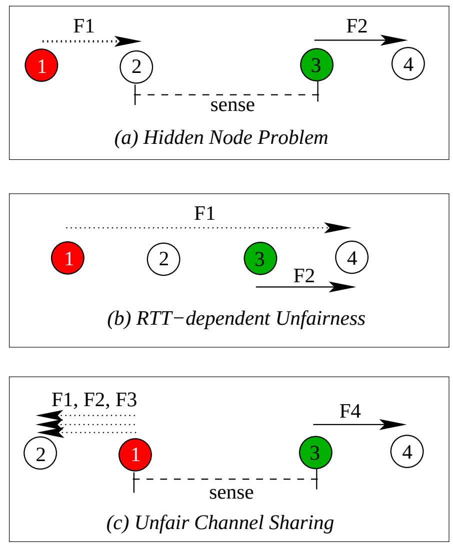

Fig. 1. Three scenarios in which significant unfairness among flows arises. The wireless node getting a lesser than fair share of bandwidth is marked as '1' (and colored in red or white), whereas the one getting a larger share is marked as '3' (and colored in green or black). (a) Node 1 lacks informations about Node 3's transmissions, attempts its communication at inopportune times, and eventually backs off unnecessarily. (b) Flow F1 traverses more hops than Flow F2. Some transport protocols, such as TCP, give more bandwidth to flow F2. (c) Flow F1, F2, F3, and F4 all share the same channel, but most transport protocols allocate more bandwidth to F4 than to others.

In this project, we aim to achieve max-min fairness over a WMN using unfair 802.11 MAC layer. Towards this end, we design and implement a *co-ordinated congestion control algorithm (C3L)* that performs global bandwidth allocation and

[1]The actual algorithm can be modified to suit other definitions of fairness[13]. For clarity of discussion, we focus on max-min fairness.

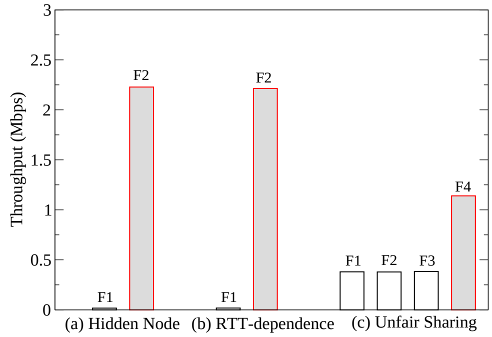

Fig. 2. TCP unfairness over 802.11 WMN for scenarios depicted in Figure 1. (a) TCP amplifies the MAC-layer unfairness due to the hidden node problem. (b) TCP gives a much higher than fair share of bandwidth to flows with smaller hop counts (especially one-hop flows [1]). (c) TCP does not allocate channel bandwidth equally among flows that share the same channel but traverse through nodes with different numbers of flows.

thus provides end-to-end flow-level max-min fairness despite weaknesses in the MAC layer. We take a centralized traffic engineering approach, which based on the latest traffic loads continuously computes the max-min fair share of individual wireless links. Unlike previous solutions to this problem, our algorithm is designed to work with multi-hop flows and takes into account both inter-flow and intra-flow dependency. Furthermore, it incorporates a general collision domain capacity re-estimation algorithm that can effectively resolve the unfairness problem due to hidden nodes. Once each wireless link is assigned a bandwidth share, flows sharing each link are in turn assigned their allocated shares in a fair way.

The traffic engineering approach is justified for WMNs. In a WMN, individual nodes do not move and route changes occur very rarely. Additionally, the traffic pattern, being aggregated from multiple end-users, does not change on a second-by-second basis. Finally, most of the traffic is directed to/from the gateway nodes that are connected to the wired Internet, resulting in a more *predictable traffic pattern*. A gateway node can also act as the central coordinator that performs the traffic engineering.

The rest of the paper is organized as follows. In Section II, we review the existing congestion control mechanisms and reason why they fail to achieve flow fairness in 802.11 WMNs. In Section III, we define the problem and discuss the coordinated congestion control algorithm. Section IV analyzes the performance of the proposed algorithm based on ns-2 simulations. We conclude the paper with a summary of contributions in Section V.

II. Existing Flow Fairness Mechanisms

Initial efforts to provide max-min fairness only considered single-hop flows [2], [3], [6], and treated each multi-hop flow as multiple independent single-hop flows. As shown in [5], single-hop fairness does not translate into multi-hop fairness because of the traffic dependency among different hops of a multi-hop flow. The algorithm presented in this paper provides end-to-end max-min fairness to multi-hop flows.

Another set of researchers proposed algorithms to achieve max-min fairness across multi-hop flows [4], [5], [15]. Most of these algorithms work with a specific interference model that does not necessarily match the real-world interference. Furthermore, all these algorithms requires MAC layer modifications, in contention resolution or packet scheduling, and thus cannot be directly applied to IEEE 802.11-based WMNs. In contrast, the algorithm proposed in this paper is specifically designed to work with unmodified 802.11 MAC.

The behavior of TCP and its variants over multi-hop ad hoc networks has been extensively studied [9], [12], [11]. From congestion control standpoint, TCP has several undesirable features when operating on wireless multi-hop networks [11]. Firstly, TCP's ACK clocking does not work well due to ACK bunching, which occurs because of bursty media access at different hops of a flow. It leads to further increase in traffic burstiness by skewing the RTT estimation upwards and thereby defeats TCP's self clocking strategy [11], [14]. Secondly, TCP confuses wireless channel errors, a frequent occurrence in wireless networks, with congestion related losses and triggers its congestion control mechanism unnecessarily. Finally, TCP's fairness is not RTT-independent: Flows traversing a smaller number of hops, or more generally flows with smaller RTT, get a larger share of channel bandwidth than those with larger RTT.

WTCP [10] measures the ratio of inter-packet spacing on the receiver and that on the sender, to determine whether to increase or decrease the sending rate. The instantaneous service rate of the bottleneck link along the path is reflected in the inter-packet delay at the receiver. If the sending rate is lower than the available bandwidth, the received packets would maintain their inter-packet spacing. Otherwise, the probe packets would queue up behind each other and their spacing would increase [10]. This approach assumes that all flows in the network are serviced in a strict round-robin fashion at the bottleneck links. This assumption does not hold in general on 802.11-based WMNs, because packet transmissions on wireless links tend to be bursty, and traffic bursts arriving at a bottleneck link may be serviced without any interleaving.

ATP [11] attempts to achieve fairness by maintaining exactly one packet in each link's queue from any flow traversing it. Every intermediate node measures the queuing and transmission delays for each packet going out on a wireless link. The sum of exponentially averaged queuing and transmission delays yields the *average packet service time*, which reflects the ideal dispatch interval for all the flows that share the wireless link. A multi-hop flow is given an average packet service time at each of the intermediate hops, and the sender adjusts its packet dispatch interval to the maximum of these service time estimates. One problem with ATP is that it couples the queue-size management with rate estimation, which leads to substantial rate fluctuation and eventually non-optimal estimation of channel bandwidth [8].

Finally, EXACT [7] is another explicit rate-based flow control scheme. EXACT routers measure the available bandwidth as the inverse of per-packet MAC contention and transmission time. Each router then runs a proportional max-min fair bandwidth sharing algorithm to divide this measured bandwidth among the flows passing through it. The bandwidth stamping mechanism is similar to ATP.

None of the above algorithms directly addresses the unfairness problems that occur in IEEE 802.11-based WMNs. Our experiments show that these protocols indeed produce unfair bandwidth allocation across flows.

III. CO-ORDINATED CONGESTION CONTROL

The objective of this work is to devise a congestion control mechanism that can achieve end-to-end max-min flow fairness over 802.11-based WMNs. Following are some definitions we use in our description:

- A bandwidth allocation is *max-min fair* if one can not increase the bandwidth allocation of any flow without reducing the bandwidth allocation of another flow with an already smaller share [4].
- A *collision domain* is defined as a maximal set of directed wireless links all of which interfere with each other pairwise. There is one sender associated with each directed wireless link.
- A *symmetric collision domain* is one where transmission on each link is visible to all the other senders, and therefore each contending sender gets an equal opportunity of accessing the common channel.
- An *asymmetric collision domain* is one where transmissions on some link (inhibitor link) are not visible to other senders (inhibited links), and consequently senders associated with inhibited links are at a disadvantaged position as compared with senders of inhibiting links.

A. Problem Definition

Formally, we are given the following as inputs:

1) Network graph $G = (V, E)$, where each vertex v_i corresponds to a wireless mesh network node, and each edge e_{ij} represents a direct communication link from v_i to v_j.
2) Interference matrix $I = [i_{m,n}]$, where $i_{m,n}$ is 1 if v_m and v_n interfere with each other, and 0 otherwise.
3) Flow vector $F = [f_i]$, where each flow f_i is characterized by a node pair (v_m, v_n), which represents the flow's source v_m and destination v_n.
4) Routing matrix $R = [r_{m,n}]$, where $r_{m,n}$ is the ordered set of nodes that a packet from v_m to v_n passes through.
5) Maximum channel capacity C_{max}.

The goal is to come up with a bandwidth allocation vector $B = [b_i]$, where b_i is the bandwidth allocation to flow f_i, and B is max-min fair. That is, increasing any b_i to $b_i + \delta$ leads to reduction in allocation b_j of some flow, where $b_j < b_i$. For simplicity of description, we assume greedy flows with infinite bandwidth requirements. Our approach can be easily extended to flows with finite bandwidth requirements.

There are several difficulties in achieving max-min fairness over an 802.11-based WMN:

1) **Intra-fbw dependency:** The bandwidth allocation to a multi-hop flow across all hops should be the same as the bandwidth assignment on its bottleneck link. Allocating more bandwidth on any other hops represents a waste of resource.
2) **Shared radio channel:** Unlike a wired network, where each link can operate independently without interfering with other links, a wireless link shares the radio channel with other links in its proximity. A wireless network is composed of multiple overlapped collision domains.
3) **MAC-dependent capacity:** While the capacity of any collision domain cannot exceed C_{max}, its effective capacity is dependent on how much time the MAC layer spends in backoffs, transmission and retransmissions, and in general cannot be known beforehand.
4) **Asymmetric MAC Contention:** Channel sharing within a collision domain could be asymmetric. Here, the inhibited sender has incomplete information about the channel status (busy or idle), and attempts its communication at inopportune times. As a result, the attempts fail more frequently and the backoff delay is increased. The end result is that transmissions on inhibited link are less likely to go through successfully than on inhibiting links, commonly described as the hidden terminal problem.

B. Case 1: Single Collision Domain Network

In the simplest case, all links interfere with one another, and hence belong to a single collision domain. The bandwidth share of each link is then proportional to the the number of flows going over it. If n_i is the number of flows going over the i^{th} link, then the *link's* fair share should be $C_{max} * n_i / \sum_i n_i$.

The above allocation works if the effective channel capacity C_{eff} equals C_{max}. Even in a symmetric collision domain, C_{eff} could be less than C_{max} because the MAC-layer overheads, such as backoffs and packet errors, reduce the effective channel capacity. The situation is even more complicated for an asymmetric collision domain, where senders associated with inhibiting links may need to slow down intentionally so that senders of inhibited links can compete on a more equal footing. This slow-down leads to a reduction in the effective channel capacity.

To estimate C_{eff}, we first assume that C_{est}, the estimated overall capacity of the collision domain, equals C_{max}. If $C_{est} > C_{eff}$, then some of the links in the collision domain would not be able to support their incoming traffic load, and their queues would get built up. From the build-up we can infer that the current C_{est} is higher than the collision domain's effective capacity. The queue build up should be sufficient before we infer $C_{est} > C_{eff}$, as short queue build up may occur even because of bursty traffic. If, on the other hand, none of the links in the collision domain has its queue built up despite all nodes sending at their assigned rates, then we may have under-estimated the collision domain's capacity.

The same logic can be used to detect capacity misestimation in an asymmetric collision domain. Consider the simplest asymmetric collision domain consisting of two links with one flow each as in Figure 1(a). Unlike in a symmetric collision domain, where allocating $C_{max}/2$ to each sender would result in queue build-up at both senders, here only the inhibited sender's queue is built up. The inhibiting link's sender, on the other hand, can transmit data at a rate up to C_{max} without experiencing any queue build-up. In essence, the inhibited sender perceives an inflated picture of the inhibitor's traffic. Therefore, simply decreasing the inhibitor's rate to $C_{max}/2$ does not help the inhibited sender.

One way to resolve the starvation issue above is to adjust the capacity estimate for a collision domain. If at least one of the senders in a collision domain sees its queue built up, the channel capacity estimate is decreased to $C_{est} - \delta$. However, if none of the senders in a collision domain sees its queue built up, the capacity estimate is increased to $C_{est} + \delta$. Unlike TCP, where each flow does this probing, this algorithm probes a collision domain's capacity for all flows associated with it.

C. Case 2: Multi-Collision Domain Network

With multiple collision domains, each link could participate in multiple overlapped collision domains, and thus receives a bandwidth allocation from each domain it participates in. The most constrained collision domain is the one that assigns the smallest bandwidth allocation and hence decides the allocation to the link.

Additionally, the flows could be multi-hop bringing the intra-flow dependency. Once a flow is allocated the bandwidth at any of its hops, that allocation needs to be propagated on all the hops. Again, the hop that participates in the most constraining collision domain is the one that ends up deciding the allocation to the flow.

D. Two-Level Allocation

While the above algorithm can already allocate share to individual flows, it is costly to run the above algorithm upon every flow joining/leaving. We therefore perform a two-level allocation: the above algorithm is used to assign bandwidth to individual wireless links. At run-time, the per-link bandwidth is allocated to all the flows going over a link by the link's sender node. The two-level allocation enables us to incorporate fluctuations in number of flows passing through a link. At run-time, if more flows pass through a link, they can be assigned bandwidth using intra-link bandwidth assignment. Similarly, extra bandwidth gets used if there are lesser number of flows than expected.

IV. Performance Evaluation

We implemented the co-ordinated congestion control algorithm (C3L), and studied its performance through both comprehensive NS-2 simulations as well as testbed experiments. Due to space constraints, this section only presents the comparison of fairness of C3L with existing congestion control mechanisms. For each protocol, we show the fairness index, the minimum-allocation flow's end-to-end bandwidth, and the average end-to-end flow bandwidth.

The fairness index indicates the degree of fairness in bandwidth allocation across flows. If X_i is the end-to-end bandwidth achieved by flow i and n is the number of flows in the network, then the fairness index is computed as [17] –

$$FairnessIndex = \frac{(\sum_i X_i)^2}{n * \sum_i (X_i)^2} \quad (1)$$

The closer the fairness index is to 1, the fairer the associated bandwidth allocation. The max-min fairness does not mean exactly equal allocation, and even the optimal fairness index may not reach 1 in many cases.

Figure 3 shows the fairness achieved by different congestion control mechanisms in a 64-node *8x8 grid network* with 20 randomly chosen flows. The fairness index for C3L is much better than other congestion control mechanisms, suggesting a more uniform allocation of bandwidth across flows by C3L. The second histogram shows the bandwidth achieved

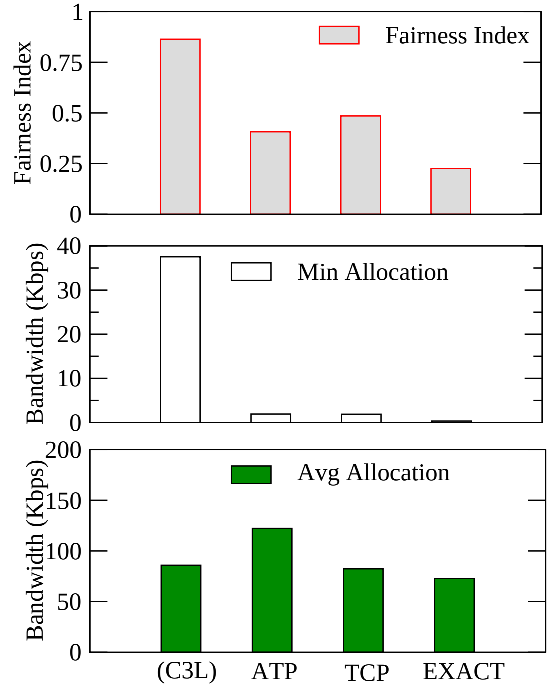

Fig. 3. Performance of C3L in a 64-node grid network with 20 end-to-end flows. The fairness index of C3L flows is closest to 1 suggesting a fair allocation of bandwidth with use of C3L. This result is also strengthened by the second histogram that shows that unlike ATP, TCP, and EXACT, no C3L flow is starved. Finally, C3L achieves this fairness without sacrificing the efficiency, as seen from its comparable performance in terms of average bandwidth allocation to each flow.

by minimum allocation flow in all the cases. C3L again does a much fairer allocation of bandwidth when compared with other protocols that end up starving some flows. Finally, the last histogram shows the average allocation across flows. The comparable average allocation of C3L suggests that C3L achieves fairness while also maximizing the utilization of the network. C3L is thus not just fair but also efficient. The average bandwidth is somewhat smaller for C3L because it gives more bandwidth to some of the flows with a larger hop count. Allocating bandwidth to a flow with a larger hop count consumes more radio resource, and therefore decreases the average flow bandwidth.

Histograms in Figure 4 show the same results for a 64-node *random mesh network* with 4 gateway nodes distributed across the network. The traffic profile comprises 15 flows originating from randomly chosen nodes, and ending at the closest gateway. Again, C3L performs a fairer and efficient allocation of bandwidth across all the flows. Figure 5 shows the actual distribution of end-to-end bandwidth achieved by different congestion control mechanisms. TCP, ATP, and EXACT give much higher bandwidth to some of the flows, while starving others. The allocation of C3L is most uniform with no obvious starvation.

V. Conclusions

The high cost of wiring, both for enterprise backbone networks and for ISP last-mile networks, makes wireless mesh network (WMN) a desirable architectural choice. At the same time, economies of scale make IEEE 802.11-based hardware

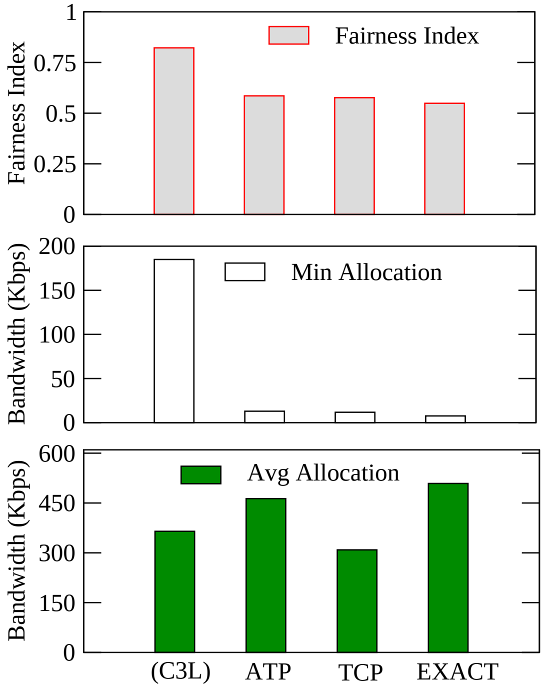

Fig. 4. Performance of C3L in a 64-node random mesh network with 4 gateways and 15 flows each destined to the closest gateway. The bandwidth distribution of C3L flows is most fair with no starvation.

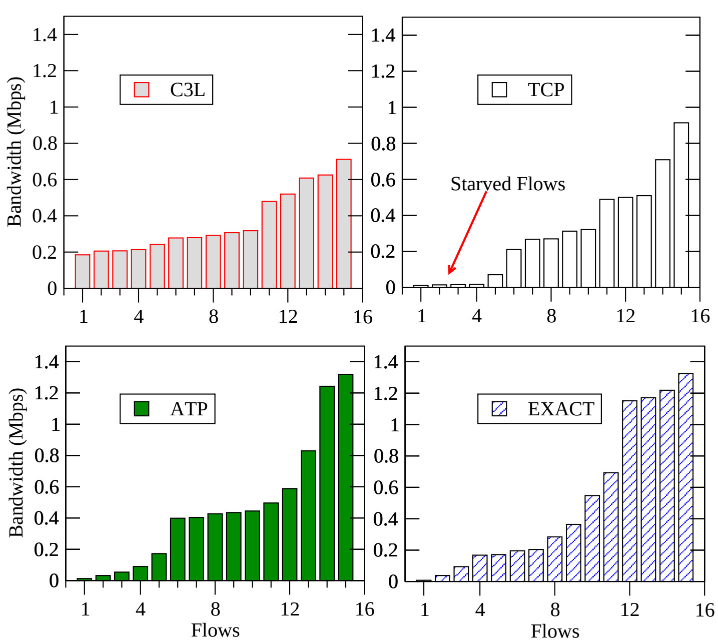

Fig. 5. Actual end-to-end bandwidth distribution across flows for results in Figure 4. C3L has the most uniform distribution of bandwidth across flows. All TCP, ATP, and EXACT lead to starvation of some of the flows.

an attractive building block for these WMNs. A major concern about using IEEE 802.11 in WMNs is its inherent unfairness at MAC layer when used in multi-hop networks [1]. Existing solutions to this problem either do not effectively resolve this unfairness, or require modifications to the MAC protocol. At the same time, the congestion control schemes in state-of-the-art transport protocols for WMNs, such as TCP, ATP and EXACT, actually exacerbate this MAC unfairness problem and end up performing a highly unfair end-to-end bandwidth allocation across flows.

In this paper, we show that it is indeed possible to achieve application-layer flow fairness over the unfair 802.11 MAC protocol through purely transport-layer mechanisms. Specifically, we propose a *co-ordinated congestion control* algorithm that achieves max-min fairness over the unmodified 802.11 MAC layer. Even though our approach is centralized, our performance overhead measurements show that it is indeed feasible, and does yield significantly better fairness than existing mechanisms. We take a traffic engineering approach, and continuously adapt the bandwidth allocation for individual wireless links to the latest traffic profile. The per-link bandwidth is further divided among flows passing through it by the link's sender. Unlike some of the previous proposals, our algorithm ensures end-to-end flow fairness by taking into account the intra-flow and inter-flow dependencies. To the best of our knowledge, this is the first research effort towards providing max-min fairness over 802.11 WMNs.

References

[1] S. Xu, T. Saadawi "Does IEEE 802.11 Work Well in Multi-hop Wireless Network?" IEEE Communications Magazine, Vol. 39, No. 6., 2001.

[2] X. Huang, B. Bensaou; "On Max-min Fairness and Scheduling in Wireless Ad-Hoc Networks: Analytical Framework and Implementation," ACM Mobihoc 2001.

[3] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan; "Achieving MAC Layer Fairness in Wireless Packet Networks," MobiCom 2000.

[4] L. Tassiulas and S. Sarkar; "Maxmin fair scheduling in wireless networks," In Proceedings of IEEE INFOCOM '02.

[5] B. Li; "End-to-End Fair Bandwidth Allocation in Multi-hop Wireless Ad Hoc Networks," ICDCS 2005.

[6] H. Luo, S. Lu, V. Bharghavan; "A New Model for Packet Scheduling in Multihop Wireless Networks," ACM Mobicom 2000.

[7] K. Chen, K. Nahrstedt, N. Vaidya; "The Utility of Explicit Rate-Based Flow Control in Mobile Ad Hoc Networks," IEEE WCNC 2004.

[8] Raj Jain, Shiv Kalyanaraman and Ram Viswanatha; "Rate Based Schemes: Mistakes to Avoid," ATM Forum/94-0882, September 1994.

[9] H. Balakrishnan, V. N. Padmanabhan, S. Seshan and R. H. Katz; "A Comparison of Mechanisms for Improving TCP performance over Wireless Links," Proc of ACM SIGCOMM, August, 1996

[10] P. Sinha, N. Venkitaraman, R. Sivakumar and V. Bharghavan; "WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks," Proc of ACM Mobicom, August, 1999

[11] K. Sundaresan, V. Anantharaman, H. Y. Hsieh and R Sivakumar; "ATP: A Reliable Transport Protocol for Ad Hoc Networks," Proc of ACM Mobihoc, June, 2003

[12] G. Holland, N. Vaidya.; "Analysis of TCP performance over mobile ad hoc networks," ACM Mobicom 1999.

[13] V. Gambiroza, B. Sadeghi and E. Knightly; "End-to-end performance and fairness in multihop wireless backhaul networks," Mobicom 2004.

[14] R. Chakravorty, A. Clark, and I. Pratt; "GPRSWeb: Optimizing the Web for GPRS Links," In Proc. of USENIX Mobisys, 2003.

[15] Y. Yi and S. Shakkottai; "Hop-by-hop congestion control over a wireless multi-hop network," In proc. of IEEE INFOCOM, 2004.

[16] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu; "Impact of interference on multi-hop wireless network performance," In proc. of ACM Mobicom, 2003.

[17] R. Jain, A. Durresi, G. Babic; "Throughput Fairness Index: An Explanation," In ATM Forum/99-0045, February 1999.