Achieving Global End-to-End Maxmin in Multihop Wireless Networks

Liang Zhang Shigang Chen Ying Jian
Department of Computer & Information Science & Engineering
University of Florida, Gainesville, FL 32611, USA *

Abstract

Following the huge commercial success of WLAN, multihop wireless networks are expected to lead in the next wave of deployment. Fundamental methods for traffic engineering must be developed to support diverse application requirements in these networks. This paper studies the problem of how to support weighted bandwidth allocation among all end-to-end flows in a multihop wireless network. Our goal is to enable the network to adapt the flow rates such that global maxmin can be achieved. Our approach is to transform the global maxmin objective into four local conditions and design a distributed rate adaptation protocol based on those local conditions. Comparing with the prior art, our protocol has a number of advantages. It is designed for the popular IEEE 802.11 DCF. It replaces per-flow queueing with per-destination queueing. It achieves far better fairness (or weighted fairness) among end-to-end flows.

1 Introduction

In recent years, research on bandwidth management in multihop wireless networks has been greatly intensified in order to support new applications. While much research concentrates on the MAC layer, the user's perception on these networks is however determined mainly based on the networks' end-to-end effectiveness. For example, for new users to participate in a wireless mesh network, they want to be sure that their end-to-end traffic is treated fairly as everyone else. Moreover, if a user contributes more to the network, she may demand that her traffic is given more weight than others' traffic. In order to meet diverse user requirements, it is important for us to develop flexible tools for traffic engineering in multihop wireless networks. This paper studies a fundamental problem, how to support weighted bandwidth allocation among all end-to-end flows in a multihop wireless network based on IEEE 802.11 DCF. A more precise but less intuitive definition of the problem is how to adapt the flow rates to achieve the *global maxmin objective* [1]: the rate of any flow in the network cannot be increased without decreasing the rate of another flow which has an equal or smaller normalized rate, where the *normalized rate* is defined as the flow rate divided by the flow weight.

The maxmin solution on wired networks [6] requires not only per-flow queueing but also a fixed bandwidth capacity for each link, which makes it not applicable in wireless networks. Utility-based solutions on wired networks [10] also require each link to have a fixed capacity. Efforts have been made to adapt utility-based solutions in wireless networks by assuming each wireless node has a fixed bandwidth capacity [15], considering only single-hop flows [7], eliminating contention among neighboring nodes by using separate CDMA/FDMA channels for wireless links [19], modeling resources as maximal contention cliques instead of wireless links [18], or relying on cross-layer design to integrate end-to-end rate adaptation with MAC-layer packet scheduling [2]. The maximal clique approach [18] requires that each clique's *effective* capacity is known, but it is not clear how to accurately measure such capacity, which is a complex function of nearby contention and environmental noise. The cross-layer approach [2] requires the nodes to dynamically establish globally coordinated (or locally approximated [12]) time-slotted transmission schedules at the MAC layer, which does not fit well with IEEE 802.11's random access model. More importantly, the utility function that approximates maxmin fairness contains an exponent approaching to infinity [14], which makes the system hard to stabilize. In summary, existing utility-based approaches do not provide a maxmin solution for IEEE 802.11 DCF.

There are other works that are not utility-based. Most of them are designed to achieve MAC-layer fairness [13] or maxmin fairness [8, 17] among one-hop flows. While some study multihop flows, each has its limitation. Basic end-to-end fairness in wireless ad-hoc networks is achieved in [11]. However, the basic fair share guaranteed for each flow is highly conservative; it can be far below the maxmin rate. End-to-end maxmin is investigated in [16], which assumes a separate CDMA/FDMA channel for each wireless link. A distributed algorithm that achieves aggregate fairness in sensor networks is proposed in [4], assuming that all flows are destined to the same base station. To the best of our knowledge, no distributed algorithm has been proposed to provide weighted maxmin bandwidth allocation in a multihop wireless network based on IEEE 802.11 DCF.

This paper studies the problem of adapting the end-to-end flow rates to achieve global maxmin. In order to design a fully distributed solution that is compatible with IEEE 802.11 DCF, we transform the global maxmin objective to four local conditions. We have proved that, if the four local conditions are satisfied in the whole network, then the global maxmin objective must be achieved. We then design a distributed rate adaptation protocol based on the four conditions. Whenever a local condition is tested false at a node, the node informs the sources of certain selected flows to adapt their rates such that the condition can

*This work was supported in part by the US National Science Foundation under grant CNS-0644033 and grant CNS-0721731.

1063-6927/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDCS.2008.66

be satisfied. Comparing with [11], which we believe is the most related work, our protocol has a number of advantages. First, it does not modify the backoff scheme of IEEE 802.11. Second, it replaces per-flow queueing with per-destination queueing. Packets from all flows to the same destination is queued together. Third and most important, our protocol achieves far better fairness (or weighted fairness) among end-to-end flows than the basic fair scheme in [11].

2 Preliminaries

2.1 Network Model and Problem Statement

We consider a *static* multihop wireless network (such as wireless mesh network with external power supply for each node) based on IEEE 802.11 DCF with congestion avoidance enhancement [3]. Mobile ad-hoc networks are beyond the scope of this paper. Two nodes are neighbors of each other if they are able to perform RTS/CTS/DATA/ACK exchange. Two nodes that are not neighbors communicate via a multihop wireless path. Time is not slotted. Radio interference is resolved by random backoff. Two wireless links contend if they cannot transmit simultaneously. Based on the most popular MAC protocol, this model excludes the majority of related works [15, 19, 2, 12, 16].

Let F be a set of end-to-end flows in the network. Each flow f has a desirable rate $d(f)$ and a weight $w(f)$. But the flow source will generate new packets at a smaller rate if the network cannot deliver its desirable rate. The actual rate of flow f is denoted as $r(f)$ ($\leq d(f)$). The *normalized rate* of flow f is defined as

$$\mu(f) = r(f)/w(f) \quad (1)$$

In this paper, when we refer to "flow rate" or "normalized rate of a flow", we mean "end-to-end rate". The *global maxmin objective* is defined as follows: The normalized rate $\mu(f)$ of any flow f cannot be increased without decreasing the normalized rate $\mu(f')$ of another flow f', for which $\mu(f') \leq \mu(f)$.

In a more intuitive but less precise description, our goal is to equalize the normalized rates of all flows as much as possible, particularly, raising the smallest ones. Directly competing flows tend to receive bandwidth in proportional to their weights. Achieving global maxmin is a fundamental function of end-to-end traffic engineering in multihop wireless networks. It adds a new entry in the existing tool box (which includes price-based and other solutions) for traffic differentiation among applications. For example, we may establish several service classes in the network and assign larger weights to applications belonging to higher classes. How to enforce a certain weight assignment scheme through service contract or other means is beyond the scope of this paper.

We assume there exists a routing protocol that establishes a routing table at each node. The routing table may be implicit under geographic routing [9], or explicitly established by a distance-vector or link-state routing protocol. Consider a specific destination. A node may receive packets from multiple *upstream neighbors* and forward them to a *downstream neighbor* towards the destination. The links from the upstream neighbors to a node are called *upstream links* of the node, and the link from a node to its downstream neighbor is called the *downstream link*.

2.2 Congestion Avoidance and Buffer-Based Backpressure

Suppose packets to different destinations are queued separately. This assumption is necessary to achieve global maxmin, as we will explain in Section 5.1. Note that other works [11, 16] require per-flow fair queueing, which is a more stringent requirement. Now consider the packets to a single arbitrary destination. A node buffers packets received from upstream links before forwarding them to the downstream link. The buffer space for the queue is limited. To avoid packet drops due to buffer overflow, we adopt the congestion avoidance scheme in [3], which allows a node i to send its downstream neighbor j a packet only when j has enough free buffer space to hold the packet. Suppose the buffer space is slotted with each slot storing one packet. To keep the neighbors updated with j's buffer state, whenever j transmits a packet (RTS/CTS/DATA/ACK), it piggybacks its current buffer state, for example, using one bit to indicate whether there is at least one free buffer slot. When an upstream neighbor i overhears a packet from j, it caches the buffer state of j. If j's buffer is not full, i transmits its packet. If j's buffer is full, i will hold its packet and wait until overhearing new buffer state from j. Note that the residual buffer at node j changes only when j receives or sends a data packet. Whenever this happens, j will send either CTS/ACK or RTS/DATA, immediately informing the neighbors of its new buffer state through piggybacking. No cyclic waiting is possible if routing is acyclic. To handle failed overhearing, i will stop waiting and attempt transmitting if it does not overhear j's buffer state for certain time. Readers are referred to [3] for discussion on other issues.

When there is a bottleneck in the routing path of a flow, the buffer at the bottleneck node will become full, forcing the upstream node to slow down its forwarding rate, which in turn makes the buffer of that node full. Such buffer-based backpressure will propagate all the way to the source of the flow. When the buffer at the source is full, the source has to slow down the flow rate (at which new packets are generated), in order to match the rate it sends our packets. Ultimately, the flow rate is determined by the forwarding rate at the bottleneck. There is no explicit signaling for the above buffer-based backpressure. The only overhead is the buffer-state bit piggybacked in each packet.

3 Link Classification

We classify the wireless links into different types based on the buffer state. In the discussion, we consider packets to a single arbitrary destination.

3.1 Saturated Buffer

When the combined rate from the upstream links of node j exceeds the rate on the downstream link, if no action is taken, the excess packets will be dropped due to buffer overflow, reducing the effective capacity of the network. With the congestion avoidance scheme [3], when the buffer at j becomes full, it forces the upstream neighbors to slow down to a combined rate that matches the rate on the downstream link.[1] *Whenever*

[1] Slowing down the rate from upstream can even help raising the rate on the downstream link due to less contention.

j *sends out a packet*, it frees some buffer space such that the upstream neighbors can compete for transmission. *Whenever* j *receives a packet*, its buffer may become full again and the upstream neighbors may have to wait for the next release of buffer at j. A buffer is *saturated* if it continuously switches between full and unfull, which slows down the rates of upstream links as the upstream neighbors have to spent time waiting for buffer release. A buffer is *unsaturated* if it stays unfull (for most of the time).

3.2 Three Link Types

We classify wireless links into three types: bandwidth-saturated links, buffer-saturated links, and unsaturated links.

• A link (i, j) is *bandwidth-saturated* if i's buffer is saturated but j's buffer is unsaturated. The fact that j's buffer is unsaturated means the downstream path from j to the destination is able to deliver all packets that i forwards to j. The fact that i's buffer is saturated means that (i, j) does not have sufficient bandwidth to timely deliver the packets received by i. Therefore, link (i, j) is the bottleneck. The only reason that prevents i from sending more packets to j is that the channel capacity has been fully utilized by (i, j) and its contending links. Hence, the rate on a bandwidth-saturated link cannot be increased without decreasing the rate of a contending link.

• A link (i, j) is *buffer-saturated* if both i's buffer and j's buffers are saturated. The fact that j's buffer is saturated means the downstream path has a bottleneck that cannot timely deliver the packets received by j. The backpressure from that bottleneck causes j's buffer to be saturated, which in turn causes i's buffer to be saturated. The rate on link (i, j) is limited not because the local channel capacity is fully used, but because the downstream path is bottlenecked and i has to spend a fraction of its time waiting for j to release buffer.

• A link (i, j) is *unsaturated* if i's buffer is unsaturated. Both link (i, j) and the downstream path from j to the destination are able to timely deliver all packets received by i.

3.3 Saturated Clique

A set of mutually contending wireless links forms a *contention clique* [8, 11, 18]. A proper clique is a clique that is not contained by a larger clique. In the following, when we refer to a contention clique, we already mean a proper clique. A link may belong to multiple cliques, consisting of nearby contending links. Packet transmissions on the links of a clique must be made serially. Therefore, the combined rate on all links of a clique is bounded by the channel capacity. A clique is *saturated* if the links have utilized all available bandwidth such that increasing the rate on one link will always lead to decreasing the rate on another link in the clique. Because a bandwidth-saturated link uses up all available bandwidth that it can acquire, it must belong to one or multiple saturated cliques.

4 Local Conditions for Global Maxmin: Single-Destination Case

We transform the global maxmin objective to several local conditions to be satisfied. Essentially our goal is to transform a global non-linear optimization problem into a fully distributed optimization problem (represented by the local conditions), which lays down the theoretical foundation for designing a distributed solution in Section 6. For now, we assume that all flows go to the same destination. The assumption will be removed in the next section.

4.1 Basic Idea

Clearly, letting IEEE 802.11 DCF decide flow rates will not achieve the global maxmin objective. Consider a network with two contending links, one carrying a single flow, f_1, and the other carrying two flows, f_2 and f_3. Suppose the weights of all flows are one. IEEE 802.11 DCF allocates channel capacity equally between the two links. Hence, f_1 can send at twice the rate of other flows. For this simple example, the global maxmin objective requires the rates of all three flows to be the same. One approach to meet this objective is to inform the source of f_1 to lower the flow rate by self-imposing an appropriate rate limit. The problem however becomes much harder for *large, arbitrary* networks where we have to decide which *end-to-end* flows should have rate limits and, after applying rate limits, whether the resulting flow rates achieve *global maxmin*. Our solution is to establish a set of conditions that are testable based on the current network state. When the conditions are all satisfied everywhere in the network, the global maxmin objective will be met. Moreover, if a condition is tested to be false, it should tell us which flows should increase their rates and which should decrease, so that we can inform the sources of those flows to adjust their rate limits. Finally, in order to make the solution fully distributed, the conditions have to be localized. Namely, they can be tested distributedly.

4.2 Normalized Rate

The data rate on link (i, j) is denoted as $r(i, j)$. The *normalized rate* on (i, j) is defined as the largest normalized rate of any flow that passes (i, j).

$$\mu(i,j) = \max_{f \in F,\ (i,j) \in p(f)} \{\mu(f)\} \tag{2}$$

where $p(f)$ is the routing path of flow f. There is an easy way for each link to know its normalized rate. When the source of a flow produces new packets, it lets the packets carry the flow's normalized rate. The nodes of a link inspect the passing packets and take the largest normalized rate carried in the packets as the link's normalized rate.

4.3 Local Conditions for Global Maxmin

We transform the global maxmin objective into four localized conditions below.

- *Source Condition:* For every node i with a saturated buffer, if i is the source of a flow, then the normalized rate of the flow is no less than that of any upstream link of i and no less than that of any other flow whose source is i.
- *Buffer-Saturated Condition:* For every buffer-saturated link (i, j), the normalized rate of (i, j) is no less than that of any

other upstream link of j and no less than that of any flow whose source is j.

- *Bandwidth-Saturated Condition:* Each bandwidth-saturated link has the largest normalized rate in at least one saturated clique that it belongs to.
- *Rate-Limit Condition:* The rate limit at a flow source should be set the highest without violating the previous three conditions.

Intuitively speaking, satisfying the first two conditions equalizes the normalized rates of all flows whose rates are constrained by the same bottleneck downstream on the path to the sink. Satisfying the third condition equalizes the normalized rates of the flows whose rates are constrained by the same bottleneck channel (saturated clique). The fourth condition ensures that all available bandwidth is utilized. Checking the above conditions does not require global state of the entire network. As we will see in Section 6 where we design the protocol, the first two conditions can be tested by each node individually and the third condition only requires information exchange among nearby nodes, which can be efficiently done. The fourth condition requires the rate limit at a flow source to be additively increased until a source, buffer-saturated or bandwidth-saturated condition is violated in the network. When this happens, the source will be signaled to tighten its rate limit. For example, if the bandwidth-saturated condition is violated, a link l that has the highest normalized rate in the saturated clique will be asked to reduce its rate in order to give up some bandwidth for the bandwidth-saturated link. Link l will identify the packets carrying the largest normalized rate and inform the sources of those packets to reduce their rates. In response, the sources will self-impose tighter rate limits.

We have proved that, when all flows have a common destination, the global maxmin objective is achieved if the four local conditions are satisfied. The proof is omitted due to space limitation.

5 Local Conditions for Global Maxmin: Multiple-Destinations Case

Removing the assumption of a single destination, we establish local conditions that are equivalent to the global maxmin objective in a general multihop wireless network.

5.1 Per-Destination Packet Queueing

We argue that, when the flows passing a node are destined for different destinations, the node should allocate a separate queue for packets to each destination. Consider the network with two flows in Figure 1 (a). First, we show that one queue per node will unnecessarily reduce the rate of f_2 in Figure 1 (b), where (z, t) is a bandwidth-saturated link, causing buffer-based backpressure to saturate the buffers at j, i, x and y. Suppose the rate of f_1 is 1 due to the bottleneck (z, t). Because the source nodes, x and y, compete fairly for transmission to i whenever i's buffer is not full,[2] f_2 will have the same rate as f_1, even though there is

[2]They also spend the same amount of time waiting for i's buffer becoming unfull, which wastes channel capacity.

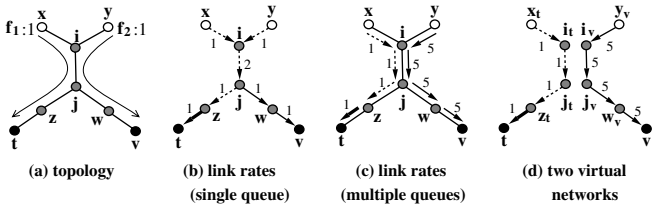

Figure 1. White circles represent flow sources. Black circles represent destinations. Thick arrows represent bandwidth-saturated links. Thin arrows represent unsaturated links. Thin dashed arrows represent buffer-saturated links. (a) A portion of the network with two flows whose weights are both one and desirable rates are both 5. (b) Each node has one queue for all destinations. (c) Each node has one queue per served destination. (d) The wireless network is modeled as two virtual networks.

no bottleneck on its routing path. With one queue at each intermediate node, f_2 is penalized because packets from f_1 saturate the shared queues along the path. To solve this problem, a node must be allowed to use multiple queues.

A node is said to *serve* a destination if it is on the routing path of a flow with that destination. A node should maintain a separate queue for each served destination, not for each passing flow. It should be noted that, in a mesh network, many flows may destine for the same destination, i.e., the gateway to the Internet. In Figure 1 (c), when i and j keep separate queues for destinations t and v, f_2 will be able to send at its desirable rate of 5.

Separate queues achieve "isolation" between packets for different destinations, which allows us to model the physical wireless network as a set of overlapping virtual networks, each for one destination. Figure 1 (d) shows that f_1 and f_2 are delivered in two virtual networks with separate packet queues but sharing the same channel.

5.2 Virtual Nodes, Virtual Links, and Virtual Networks

We model each physical node i as a set of *virtual nodes* i_t, one for each served destination t. A virtual node i_t carries one queue, storing all packets received by i for destination t. All virtual nodes for the same destination t form a *virtual network*; there exists a *virtual link* (i_t, j_t) if j is i's next hop towards t. (i_t, j_t) is called the *downstream link* of i_t and an *upstream link* of j_t. All virtual networks together are called the *grand virtual network*, which can be viewed as a "decomposed" model of the original wireless network. A wireless link (i, j) is modeled as the aggregate of virtual links (i_t, j_t) from all virtual networks. These virtual links (i_t, j_t) mutually contend because the physical node i can only transmit a packet from one of its queues at each time. An example is given in Figure 1 (d), where the wireless network is modeled as two virtual networks, and (i, j) as two virtual links.

Each virtual network carries a subset of flows, which is disjoint from the subsets carried by other virtual networks. Buffer-based backpressure (Section 2.2) is performed independently within each virtual network. The normalized rate of a virtual link is defined as the largest normalized rate of any flow passing the link. Within a virtual network, we classify virtual links

as bandwidth-saturated, buffer-saturated, or unsaturated in the same way as we did in Section 3.2. Other concepts can also be trivially extended to virtual networks.

5.3 Localized Requirements for Global Maxmin

Below we modify the local conditions in Section 4.3 to suit for a wireless network whose flows have different destinations.

- *Source Condition:* In the virtual network for destination t, for every node i_t with a saturated buffer, if i_t is the source of a flow, then the normalized rate of the flow is no less than that of any upstream link of i_t and no less than that of any other flow whose source is i_t.
- *Buffer-Saturated Condition:* In the virtual network for destination t, for every buffer-saturated virtual link (i_t, j_t), the normalized rate of (i_t, j_t) is no less than that of any other upstream link of j_t and no less than that of any flow whose source is j_t.
- *Bandwidth-Saturated Condition:* Each bandwidth-saturated virtual link has the largest normalized rate in at least one saturated clique that it belongs to.
- *Rate-Limit Condition:* The rate limit at a flow source should be set the highest without violating the previous three conditions.

We have proved that the global maxmin objective is achieved if the four local conditions are satisfied in the grand virtual network. The proof is omitted due to space limitation.

6 Distributed Global Maxmin Protocol (GMP)

In this section, we design a distributed protocol that adapts the flow rates to satisfy the four local conditions in Section 5.3, which is equivalent to meeting the global maxmin objective in wireless networks with multiple destinations.

6.1 Overview

Our basic means is to set appropriate rate limits at flow sources such that the local conditions can be satisfied in the network. Assume the system clocks at the nodes are loosely synchronized. The time is divided into alternating measurement/adjustment periods. In each measurement period, all nodes measure the state of its adjacent (virtual) links and exchange information with close-by nodes. In each adjustment period, based on the information measured by itself and close-by nodes, each node checks the first three local conditions. If one or more conditions are false, the node issues rate adjustment requests for selected flow sources, which adjust their rates accordingly. If a flow source does not receive a rate adjustment request, it will increase its rate limit to meet the fourth condition. After a series of measurement and adjustment periods, the rate limits of all flows are gradually modified to meet the four conditions.

In the protocol description, we refer to a physical node simply as "node", denoted as "i", in contrast to a "virtual node", denoted as "i_t" for destination t. We refer to a link between two physical nodes as "wireless link", denoted as "(i, j)", which may contain multiple "virtual links", denoted as "(i_t, j_t)". We refer to the original network as "wireless network", in contrast to "virtual network" consisting of virtual links. The protocol could have been designed to work entirely on virtual nodes/links, but we optimize it by working on physical nodes and wireless links whenever possible and on virtual nodes/links only when we have to. The reason is that there are a lot more virtual nodes/links than physical ones.

Flow f is a *local flow* at node i if i is the source of f. Flow f is a local flow of virtual node i_t if f is a local flow of i and its destination is t. The *primary flow* of a (virtual) link is the flow that has the largest normalized rate among all flows passing that (virtual) link. When multiple flows have the largest normalized rate, they are all primary flows.

6.2 Measurement Period

In this period, nodes measure the state of their links. At the end of the period, they exchange the link state.

Step 1: Measurement

All virtual nodes measure their buffer state, based on which they determine the types of their adjacent virtual links. The virtual nodes also measure the normalized rates of their adjacent virtual links. The physical nodes measure the channel occupancies of their adjacent wireless links; we will discuss how to determine saturated cliques based on this information. Details of measurement are given below.

Buffer State: Each virtual node i_t carries one queue for all packets received by i destined for t. A certain amount of buffer is designated for the queue. Over each measurement period, i_t measures the fraction Ω of time in which the buffer stays full. If Ω is above a threshold, i_t sets the buffer state as saturated. We find in our simulations that, if the upstream neighbors supply more packets than i_t can forward, Ω will always stay above 50%, and if the upstream neighbors supply fewer packets than i_t can forward, Ω will be almost zero. Therefore, we set the threshold to 25%.

At the end of a measurement period, for each virtual link (i_t, j_t), the end nodes exchange their buffer state, which can be piggybacked in RTS/CTS/DATA/ACK packets with one extra bit (saturated or not). Based on their buffer state, both i_t and j_t can determine the type of (i_t, j_t), which is buffer-saturated, bandwidth-saturated, or unsaturated.

Link Rate: For each virtual link (i_t, j_t), i_t measures the average data rate $r(i_t, j_t)$ on the link over each measurement period.

Normalized Rate: In the first half of each measurement period, the flows' normalized rates are measured at their sources. In the second half of the period, each flow source selects a number of data packets to piggyback the flow's current normalized rate. From the packets forwarded on a virtual link (i_t, j_t), both i_t and j_t learn the virtual link's normalized rate, which is the largest normalized rate carried in the packets. They also learn the sources of the virtual link's primary flows, which are the sources of the packets that carry the largest normalized rate. Clearly, the normalized rate of a wireless link (i, j) is equal to the largest normalized rate of its virtual links (i_t, j_t).

Channel Occupancy: The channel occupancy of a wireless link (i, j) is defined as the fraction of time in which the channel is occupied by packets forwarded by i to j, including RTS/CTS/DATA/ACK transmissions. Nodes i and j measure their transmissions over each measurement period, and exchange their measurements at the end of the period.

Step 2: Information Dissemination

Every node i has the following information at the end of a measurement period: a) the type of each adjacent virtual link, b) the data rate on each downstream virtual link, c) the normalized rate of each adjacent virtual link, d) the sources of the primary flows, e) the normalized rate of each adjacent wireless link, and f) the channel occupancy of each adjacent wireless link. In order to design an protocol that checks the bandwidth-saturated condition in Section 5.3, a node must also know the normalized rates and the channel occupancies of all wireless links that contend with any of its adjacent links.

We refer to the normalized rate and the channel occupancy of a wireless link as the *state of the link*. We must disseminate the state of each wireless link (i, j) to all nodes that have a link contending with (i, j), including all those nodes within two hops away from either i or j. The dissemination protocol is described as follows. Recall that we only consider static wireless networks. After deployment, we assume each node i discovers the wireless topology in its two-hop neighborhood, and identifies a minimum subset of one-hop neighbors, called i's *dominating set*, whose adjacent links reach all two-hop neighbors. Node i informs the nodes in its dominating set of their membership in the set. At the end of each measurement period, if the state of (i, j) changes from the previous period, both i and j broadcast the new state to their one-hop neighbors. When a node in their dominating sets overhears this information, the node rebroadcasts the information to its neighbors.

6.3 Adjustment Period

When local conditions are tested false, a node proposes rate adjustments for its local flows and primary flows on the adjacent virtual links. We will discuss how to efficiently deliver rate adjustments to the sources of the primary flows at the end of this section.

In order to stabilize the flow rates quicker, we introduce a system parameter β. The data rates, normalized rates, or channel occupancies of two links (or flows, cliques when applicable) are considered to be "equal" if their difference is below β percentage (e.g., 10%). One is considered to be "smaller" than another if it is smaller by at least β percentage. The operations performed by the nodes in this period are explained below.

- *Removing Unnecessary Rate Limits*

If a local flow's actual rate is smaller than its rate limit, the node removes the rate limit because it is unnecessary.

- *Testing Source Condition and Buffer-Saturated Condition*

If a virtual node i_t has a saturated buffer, it examines the normalized rates of its upstream virtual links and local flows. Let L_1 be the largest value among them, and S_1 be the smallest among the normalized rates of the local flows and those of buffer-saturated upstream virtual links. To satisfy both source condition and buffer-saturated condition, S_1 should be equal to L_1. Otherwise we have to adapt the rates of local and/or passing flows until S_1 is equal to L_1. More specifically, i_t transmits a rate adjustment request (carrying L_1 and S_1) to all upstream neighbors. When j_t receives the request, it invokes the following procedure:

(1) If $\mu(j_t, i_t)$ is equal to L_1, then a rate reduction request is issued for the primary flows on virtual link (j_t, i_t). If $L_1 > 3S_1$, it requests the primary flows to halve their rates; otherwise, it requests the primary flows to reduce their rates by β percentage. (When the gap between L_1 and S_1 is too big, reducing by half helps to close the gap quickly.)

(2) If (j_t, i_t) is a buffer-saturated link and $\mu(j_t, i_t)$ is equal to S_1, then a rate increase request is issued for the primary flows on virtual link (j_t, i_t). If $L_1 > 3S_1$, it requests the primary flows to double their rates; otherwise, it requests the primary flows to increase their rates by β percentage.

Similarly, a rate adjustment request may be issued for a local flow f for destination t. If $\mu(f) = L_1$, i_t issues a rate reduction request for f. If $\mu(f) = S_1$ and f has a rate limit, i_t issues a rate increase request for f.

- *Testing Bandwidth-Saturated Condition*

We have assumed that, after deployment, each node i discovers the wireless topology in its two-hop neighborhood. From the topology, it pre-computes the set of cliques it belongs to. Because there is a one-to-one correspondence between cliques in the original wireless network and cliques in the grand virtual network, we are able to perform most clique-related operations based on wireless links instead of their constituent virtual links (whose number is much larger). Each clique has a system-wide unique identifier, consisting of the smallest identifier of the nodes in the clique and a sequence number. A clique's identifier is assigned by the node with the smallest identifier and disseminated to other nodes in the clique via its dominating set.

At the beginning of each adjustment period, i computes the channel occupancy of each clique, which is equal to the sum of the channel occupancies of the wireless links in the clique. For a wireless link (i, j) that has at least one bandwidth-saturated virtual link, we do the following: First, among its bandwidth-saturated virtual links, we identify the one (i_t, j_t) with the smallest normalized rate. Among all cliques that (i, j) belongs to, we treat those that have the largest channel occupancy as being saturated. Second, we check whether (i_t, j_t) satisfies the bandwidth-saturated condition. If $\mu(i_t, j_t)$ is not the largest normalized rate in any of its saturated cliques, we must increase $\mu(i_t, j_t)$ by issuing rate adjustment requests. Let L_2 be the largest normalized rate on wireless links in all saturated cliques that (i, j) belongs to. Node i disseminates L_2, $\mu(i_t, j_t)$, and the identifiers of saturated cliques via its dominating set to all nodes in two-hop neighborhood. When a node k receives this information, if a wireless link (k, m) belongs to one of those saturated cliques, k executes the following procedure: for each virtual link (k_v, m_v), (1) if $\mu(k_v, m_v)$ is equal to L_2, then a rate reduction request is issued for the primary flows on (k_v, m_v) to cut their rates by β percentage; (2) if (k_v, m_v) is a bandwidth-saturated link and $\mu(k_v, m_v)$ is equal to $\mu(i_t, j_t)$, then a rate increase request is issued for the primary flows on (k_v, m_v) to increase their rates by β percentage.

- *Rate Adjustment at Sources*

At the end of an adjustment period, the source of each flow sends a control packet that travels along the routing path to collect the rate adjustment requests for the flow. It only carries one request. If there is no rate reduction request, it keeps the rate increase request with the smallest increase. If there is a rate reduction request, it discards all rate increase requests. If there are multiple rate reduction requests for the flow, it keeps the one with the largest rate reduction. When the destination receives the control packet, it sends the packet back to the source, which will adjust its rate (by changing the rate limit) based on the request carried in the packet.

- *Meeting Rate-Limit Condition*

If a flow source does not receive any rate adjustment request and it has a rate limit, it will additively increase its rate limit by a small amount to make sure that the flow will send at the highest possible rate.

7 Simulation

We perform simulations to evaluate the proposed distributed global maxmin protocol (GMP). The simulation setup is described as follows. IEEE 802.11 DCF is implemented. The channel capacity is 11Mbps. Each node has a transmission range of 250 meters. Each data packet is 1024 bytes long. The desirable rate of any flow is 800 packets per second. The buffer space at a node can hold 300 packets. The length of each simulation session is 400 seconds. Each measurement or adjustment period is 4 seconds long. β is set to be 10%.

7.1 Effectiveness of GMP

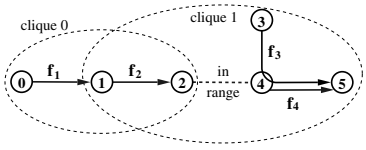

Figure 2. Network topology of a simple scenario

flow	f_1	f_2	f_3	f_4
rate	563.96	196.96	217.57	221.41

Table 1. Simulation results on the topology in Fig.2

flow	f_1	f_2	f_3	f_4
weight	1	2	1	3
rate	527.58	225.40	121.90	377.20

Table 2. Simulation results of weighted maxmin in Fig. 2

The network topology used in the first simulation is shown in Fig. 2. First, we assign all flows the same weight 1, so that a flow's normalized rate is the same as the flow rate. Wireless links $(1, 2)$, $(3, 4)$ and $(4, 5)$ mutually contend with each other and form clique 1. Links $(0, 1)$ and $(1, 2)$ form the smaller clique 0. Based on the maxmin model, flows f_2, f_3 and f_4 should have the same normalized rate, and they have equal access to the channel capacity of clique 1. Because the rate of f_2 is limited by clique 1, flow f_1 is able to send at a higher rate, fully utilizing the bandwidth not used by f_2 in clique 0. The simulation results shown in Table 1 are consistent with the above analysis. In the simulation, after the flow rates are stabilized, $(0, 1)$ and $(1, 2)$ are bandwidth-saturated links, while $(3, 4)$ and $(4, 5)$ are unsaturated links. The bandwidth-saturated condition ensures that, in the saturated clique 1, the normalized rate of f_2 is no less than those of f_3 and f_4. The rate-limit condition ensures that f_1 will send at the highest-possible rate as long as it does not drive the rate of f_2 too low that violates the bandwidth-saturated condition of f_2 in clique 1.

Next we test weighted maxmin on the same network topology by assigning different weights to flows. The simulation results are given in Table 2. The rates of the three flows in clique 1 are approximately proportional to their pre-assigned weights. Flow f_1 has a higher rate than flow f_2 even though its weight is smaller. That is because it opportunistically utilizes all remaining bandwidth in clique 0 that cannot be used by f_2.

7.2 Performance Comparison

In this subsection, we compare the performance of GMP with IEEE 802.11 DCF (abbreviated as 802.11) and the two-phase protocol (abbreviated as 2PP) proposed in [11]. These three protocols use different buffer management strategies to accommodate their packet queuing algorithms. In 802.11, all flows passing a node share the same buffer space. When a packet arrives at a node whose buffer is full, it will overwrite the packet at the tail of the queue. In 2PP, each flow is allocated a separated queue that can hold 10 packets. In GMP, all flows to the same destination share a common queue that can hold 10 packets.

Since 2PP is designed to provide fairness (instead of weighted bandwidth allocation), we compare the protocols from two aspects: *end-to-end flow fairness* (when all flows have equal weights) and *spatial reuse of spectrum*.

To evaluate the end-to-end fairness, we adopt the maxmin fairness index [1] (denoted by I_{mm}) and the equality fairness index [5] (denoted by I_{eq}).

$$I_{mm} = \frac{min_{f\in F}\{r(f)\}}{max_{f\in F}\{r(f)\}}, \qquad I_{eq} = \frac{(\sum_{f\in F} r(f))^2}{|F|\sum_{f\in F}(r(f))^2}$$

I_{mm} measures the ratio of the smallest flow rate to the largest flow rate. I_{eq} measures the overall equality among the flow rates; its value approaches to one if the rates of all flows approach toward equality.

To measure the spatial reuse of spectrum, we employ the *effective network throughput* U, which is defined as $\sum_{f\in F} r(f) \times l_f$, where l_f is the number of hops on the routing path of flow f. The packets dropped by the intermediate nodes do not count towards the effective network throughput as they do not contribute to end-to-end throughput. The effective network throughput gives us a measurement for network bandwidth utilization and the efficiency of a protocol.

Figure 3. A three-links topology

flow	802.11	2PP	GMP
$\langle 0,3\rangle$	80.63	131.86	164.75
$\langle 1,3\rangle$	220.07	188.76	176.04
$\langle 2,3\rangle$	174.09	240.85	179.21
U	856.11	1013.96	1025.54
I_{mm}	0.366	0.547	0.919
I_{eq}	0.882	0.946	0.999

Table 3. Simulation results on the topology in Fig. 3

First we simulate the scenario in Fig. 3. The simulation results are shown in Table 3. GMP is much fairer than 2PP, which is in turn much fairer than 802.11. Due to the hidden terminal problem under 802.11, a severe unfairness in media access exists between link $(0,1)$ and $(2,3)$ [8]. Node 0 has much less chance to grab the channel when it has packets to be transmitted to node 1. This explains why the flow from node 0 to node 3, which passes $(0,1)$, has the lowest rate under 802.11. The effective network throughputs of 2PP and GMP are comparable, and they are higher than that of 802.11, which drops more packets due to buffer overflow.

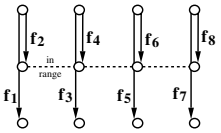

Figure 4. Network topology

flow	802.11	2PP	GMP
f_1	221.81	43.31	145.46
f_2	221.81	347.81	145.94
f_3	107.29	43.33	134.26
f_4	107.28	86.67	132.38
f_5	106.36	43.39	135.44
f_6	106.36	86.70	133.04
f_7	223.39	43.36	141.69
f_8	223.39	346.96	149.07
U	1976.54	1214.93	1674.13
I_{mm}	0.476	0.125	0.888
I_{eq}	0.890	0.514	0.998

Table 4. Simulation results on the topology in Fig.4

The design of 2PP is to ensure a basic fair share of bandwidth for all flows and then favor short flows in allocating the remaining bandwidth. The basic fair share can be very small, and there are cases in which it is outperformed by 802.11. We perform simulations on the topology in Fig. 4, and the results are shown in Table 4. With this topology, the basic fair share calculated based on the formula in [11] is small, and the remaining bandwidth is distributed heavily biased towards f_2 and f_8 based on the linear programming approach in the same paper. Under 802.11, the flows in the middle (f_3, f_4, f_5 and f_6) have lower rates than the flows on the sides (f_1, f_2, f_7 and f_8). The reason is that a flow in the middle need compete for bandwidth with more flows than a flow on the side. With GMP, all flows have approximately equal rates regardless of their locations and lengths.

8 Conclusion

In this paper, we proposed a distributed protocol to achieve the global maxmin objective based on four local conditions. We introduced several new concepts, including link classification based on buffer state, virtual links, and virtual networks, which are essential for the development of the local conditions. We performed simulations to demonstrate the effectiveness of the proposed protocol.

References

[1] D. Bertsekas and R. Gallager. *Data networks*. Prentice-Hall Inc, 2nd edition, 1992.

[2] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle. Cross-layer Congestion Control, Routing, and Scheduling Design in Ad Hoc Wireless Networks. *Proc. of IEEE INFOCOM'06*, April 2006.

[3] S. Chen and N. Yang. Congestion Avoidance based on Light-Weight Buffer Management in Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems, Special Issue on Localized Communication and Topology Protocols for Ad Hoc Networks*, 17(9), September 2006.

[4] S. Chen and Z. Zhang. Localized Algorithm for Aggregate Fairness in Wireless Sensor Networks. *Proc. of ACM Mobicom'06*, September 2006.

[5] D. Chiu and R. Jain. Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks. *Journal of Computer Networks and ISDN*, 17(1):1–14, June 1989.

[6] J. M. Faffe. Bottleneck Flow Control. *IEEE Transactions on Communications*, COM-29(7):954–962, July 1981.

[7] Z. Fang and B. Bensaou. Fair Bandwidth Sharing Algorithms based on Game Theory Frameworks in Wireless Ad-Hoc Networks. *Proc. of IEEE INFOCOM'04*, March 2004.

[8] X. L. Huang and B. Bensaou. On Max-Min Fairness and Scheduling in Wireless Ad-hoc Networks: Analytical Framework and Implementation. *Proc. of MobiHoc'01, Long Beach, California*, October 2001.

[9] B. Karp and H. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. *Proc. of ACM MobiCom'00*, August 2000.

[10] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research*, 49, 1998.

[11] B. Li. End-to-End Fair Bandwidth Allocation in Multi-hop Wireless Ad Hoc Networks. *Proc. of IEEE ICDCS'05*, June 2005.

[12] X. Lin and N. B. Shroff. The Impact of Imperfect Scheduling on Cross-Layer Congestion Control in Wireless Networks. *IEEE/ACM Trans. on Networking*, 14(2), April 2006.

[13] H. Luo, S. Lu, and V. Bharghavan. A New Model for Packet Scheduling in Multihop Wireless Networks. *Proc. of MobiCom'00*, August 2000.

[14] J. Mo and J. Walrand. Fair End-to-End Window-Based Congestion Control. *IEEE/ACM Trans. on Networking*, 8(5), October 2000.

[15] Y. Qiu and P. Marbach. Bandwidth Allocation in Wireless Ad-Hoc Networks: A Price-based Approach. *Proc. of IEEE INFOCOM'03*, March 2003.

[16] S. Sarkar and L. Tassiulas. End-to-end Bandwidth Guarantees Through Fair Local Spectrum Share in Wireless Adhoc Networks. *IEEE Transactions on Automatic Control*, 50(9), September 2005.

[17] L. Tassiulas and S. Sarkar. Maxmin Fair Scheduling in Wireless Networks. *Proc. of IEEE INFOCOM'02*, June 2002.

[18] Y. Xue, B. Li, and K. Nahrstedt. Optimal Resource Allocation in Wireless Ad Hoc Networks: A Price-based Approach. *IEEE Transactions on Mobile Computing*, 5(4), Aprile 2006.

[19] Y. Yi and S. Shakkottai. Hop-by-Hop Congestion Control over a Wireless Multi-hop Network. *Proc. of IEEE INFOCOM'04, Hong Kong, China*, March 2004.