J. Parallel Distrib. Comput. 72 (2012) 462–474

Contents lists available at SciVerse ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

End-to-end maxmin fairness in multihop wireless networks: Theory and protocol

Liang Zhang [a], Wen Luo [b], Shigang Chen [b,*], Ying Jian [c]

[a] *Juniper Networks, Sunnyvale, CA 94089, USA*
[b] *Department of Computer Science, University of Florida, Gainesville, FL 32611, USA*
[c] *Google, Mountain View, CA 94043, USA*

ARTICLE INFO

Article history:
Received 21 June 2011
Received in revised form
27 October 2011
Accepted 29 November 2011
Available online 8 December 2011

Keywords:
Distributed rate adjustment protocol
Multihop wireless networks
Max–min fairness

ABSTRACT

To promote commercial deployment of multihop wireless networks, the research/industry communities must develop new theories and protocols for flexible traffic engineering in these networks in order to support diverse user applications. This paper studies an important traffic engineering problem – how to support fair bandwidth allocation among all end-to-end flows in a multihop wireless network – which, in a more precise term, is to achieve the *global maxmin fairness objective* in bandwidth allocation. There exists no distributed algorithm for this problem in multihop wireless networks using IEEE 802.11 DCF. We have two major contributions. The first contribution is to develop a novel theory that maps the global maxmin objective to four local conditions and prove their equivalence. The second contribution is to design a distributed rate adjustment protocol based on those local conditions to achieve the global maxmin objective through fully distributed operations. 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

Following the huge commercial success of WLAN, multihop wireless networks are expected to lead in the next wave of deployment. In recent years, the advent of wireless mesh networks has greatly accelerated the research on resource management in such networks to support new applications [10,40]. 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 peer-to-peer 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 an important problem—how to support weighted bandwidth allocation among all end-to-end flows in a multihop wireless network based on the random access model of IEEE 802.11 DCF (CSMA/CA). A more precise but less intuitive definition of the problem is how to adapt the flow rates to achieve the *global maxmin objective* [3]: 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. Most prior solutions are not designed for end-to-end flows in 802.11 DCF-based multihop networks. They rely on non-CSMA models [38,31,30,26], assume non-interfering WLANs [20,21], consider only single-hop flows [33,34,14,32], require coordinated time-slotted transmission schedules at the MAC layer [5], or have other restrictions unsuitable for DCF. More related work will be discussed in greater details in the next section.

We develop a novel theory and its distributed protocol to achieve the global maxmin objective among end-to-end flows in a multihop wireless network. We have two major contributions. First, 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 and prove that, if the four local conditions are satisfied in the whole network, then the global maxmin objective must be achieved. Second, we design a distributed rate adjustment protocol based on the four conditions. Whenever a local condition is found to be false at a node, the node informs the sources of certain selected flows to adapt their rates such that the condition can be satisfied.

Comparing with the existing work on bandwidth management in wireless networks, 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 are queued together.

* Corresponding author.
E-mail address: sgchen@cise.ufl.edu (S. Chen).

0743-7315/$ – see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jpdc.2011.11.012

Third and most important, our protocol achieves far better fairness (or weighted fairness) among end-to-end flows, which will be demonstrated by simulations.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 defines the network model. Section 4 classifies wireless links into three categories. Section 5 presents the local conditions for global maxmin in wireless networks with a single destination. Section 6 presents the local conditions for networks with multiple destinations. Section 7 designs a distributed global maxmin protocol based the local conditions. Section 8 evaluates the protocol by simulations. Section 9 draws the conclusion.

2. Related work

The maxmin solutions on wired networks [12,19] not only require per-flow queueing but also assume a fixed bandwidth capacity for each link, which makes them not applicable in random-access wireless networks.

It is well known that TCP does not perform well in wireless networks [2,39]. Much research has been done to improve TCP's performance [37], and a recent survey can be found in [7]. Most existing solutions employ heuristic mechanisms for better congestion signaling. However, they are not designed for solving the problem of *provable weighted bandwidth allocation* as this work does. As pointed out in [7], none of the existing solutions can work well in all scenarios or meet all the challenges, including lossy links, hidden terminals, unsynchronized congestion signaling in contention neighborhood, and throughput degradation.

Many congestion-control and rate-adaptation solutions in wireless networks move away from CSMA to multi-channel CDMA/FDMA model [38,31], node-exclusive interference model [5,25,30] (where links can transmit simultaneously as long as they do not share a common node), or the centralized scheduling model [11,26]. They cannot be applied to a multihop network using 802.11 DCF (CSMA/CA). Other work requires per-flow queueing at all nodes [16], assumes all wireless links share the same local channel [1], or assumes a fixed bandwidth capacity for each wireless node [28], which is too restrictive or inefficient for a large multihop wireless network based on DCF. It has been shown recently [21] that the rate region of a large class of 802.11 mesh networks is log convex, assuming that the WLANs are non-interfering, i.e. that they either transmit on orthogonal channels or are physically separated so that transmissions on the same channel do not interfere. Similar assumption is made in [20].

Also related are solutions that are designed to achieve MAC-layer fairness [24,23,18] or maxmin fairness [14,32,34] among one-hop flows. Those are different problems than the global maxmin fairness for end-to-end multihop flows considered in this paper. While some work studies multihop flows, each has its limitation. Basic end-to-end fairness in wireless ad-hoc networks is achieved in [22]. However, the basic fair share guaranteed for each flow is highly conservative; it can be far below the maxmin rate. The temporal fairness in multi-rate wireless networks is studied in [13], which however does not provide an algorithm that computes the temporally-fair rates. Rate fairness in sensor networks is studied in [29,8], assuming that all flows are destined to the same base station. They do not handle flows with different destinations or implement maxmin fairness. 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.

IEEE 802.11e [15] has been under development to support QoS, primarily for WLAN. Its EDCA provides prioritized channel access only for four access categories (background, best effort, video, and voice). It does not provide fine-level control for weighted bandwidth allocation among end-to-end flows.

3. Problem and assumptions

3.1. 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 [6] (see Section 3.2 for a brief description). 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.

Let F be a set of end-to-end flows in the network. Each flow f has a desired 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 desired rate. The actual rate of flow f is denoted as $r(f)$ ($\leq d(f)$). The *normalized rate* of flow f is defined as

$$\mu(f) = r(f)/w(f). \tag{1}$$

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 an important traffic engineering function 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 [4,17], or explicitly established by a distance-vector [27] or link-state routing protocol. Consider a specific destination.[1] A node may receive packets from multiple *upstream neighbors* and forward them to a *downstream neighbor* toward 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*.

3.2. 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 6.1. Note that other works [22,31] 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 [6], 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

[1] When we say "destination" in the paper, we always mean the final receiver of an end-to-end flow".

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 [6] for discussion on other issues.

When there is a bottleneck[2] in the routing path of a flow, the buffer at a 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 out 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.

4. Link classification

We classify the wireless links into different types based on the buffer state. Without losing generality, we consider packets to a single arbitrary destination. When we refer to the queue of a node, we mean the queue for that destination. When we refer to buffer, we mean the buffer space of that queue. Be aware that each wireless link may be classified differently for different destinations because the buffer state of per-destination queues may be different. This will become clear when we introduce virtual links in Section 6.

4.1. 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 [6], 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.[3] *Whenever 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).

4.2. 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. *Link (i, j) is a bottleneck if it is bandwidth-saturated.* 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.

4.3. Saturated clique

A set of mutually contending wireless links forms a *contention clique* [14,22,36]. 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 implicitly 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]

5. 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 7. For now, we assume that all flows go to the same destination. The assumption will be removed in the next section.

5.1. Basic idea

Clearly, letting IEEE 802.11 DCF decide flow rates will not achieve the global maxmin objective. Consider a network with

[2] A formal definition will be given in Section 4.2 after some necessary concepts are introduced.

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

[4] For a bandwidth-saturated link (i, j), node i is constantly backlogged by definition. It constantly attempts to send whenever the channel is idle, and therefore uses up all bandwidth available to it in the statistical sense under the random access framework of IEEE 802.11 DCF. Note that we assume IEEE 802.11 DCF, not a time-slotted MAC protocol with coordinated transmission schedules. The only reason that prevents the rate on (i, j) from being higher is the rest of the channel capacity is fully occupied by contending links. In other words, if the rate on (i, j) were to increase, when other contending links access the media at randomized times, their probability of finding media occupied by transmission on (i, j) would be proportionally higher. Consequently the rate of one or more contending links would go down.

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. However, for this simple example, the global maxmin objective requires the rates of all three flows to be the same. One approach to meet this goal is to inform the source of f_1 to lower the flow rate by self-imposing an appropriate rate limit. The problem is how to decide which flows should have rate limits in an *arbitrary* network 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. We shall prove that, if the conditions are all satisfied, then the global maxmin objective must 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 in a distributed manner.

For the above example, one may argue that, although IEEE 802.11 DCF does not provide fairness, many MAC protocols [24,23,18,14,32] have been proposed to achieve that. IEEE 802.11e can also provide coarse-level rate control. But the example is a single one with only one-hop flows. These MAC solutions cannot provide *end-to-end* fairness, let alone *provable* weighted maxmin.

5.2. 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.

The set of flows that pass (i, j) consists of all flows passing the upstream links and all flows that begin from i. By the definition of normalized rate, we have the lemma below.

Lemma 1. *The normalized rate of link (i, j) is equal to the largest value among the normalized rates of all upstream links of i and the normalized rates of all flows whose sources are i.*

5.3. 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.

Checking the above conditions does not require global state of the entire network. As we will see in Section 7 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 illustrate the purpose of the four local conditions by providing a couple of examples. First, examine the simple case in Section 5.1, where the network has only two wireless links, (i, t) and (j, t). There are three flows, one from i to t and two from j to t. Assume both i and j have saturated buffer. Satisfying the source condition ensures that the two flows on (j, t) have the same normalized rate. Satisfying the bandwidth-saturated condition ensures that they also have the same normalized rate as the flow on (i, t) does. Hence, the maxmin objective is achieved. Regardless of what the flows' weights are, equalizing normalized rate means that the flows' rates will be proportional to their weights. One may be puzzled by the contradictive fact that IEEE 802.11 DCF would assign equal bandwidth to (i, t) and (j, t), which means the flows on (j, t) would each have half of the bandwidth received by the flow on (i, t). The answer is that, to satisfy the four local conditions, rate limits must be enforced on some flow sources. In this example, a rate limit at i will reduce its flow rate such that the flows on (j, t) can receive more bandwidth. (Detailed operations will be given in Section 7.)

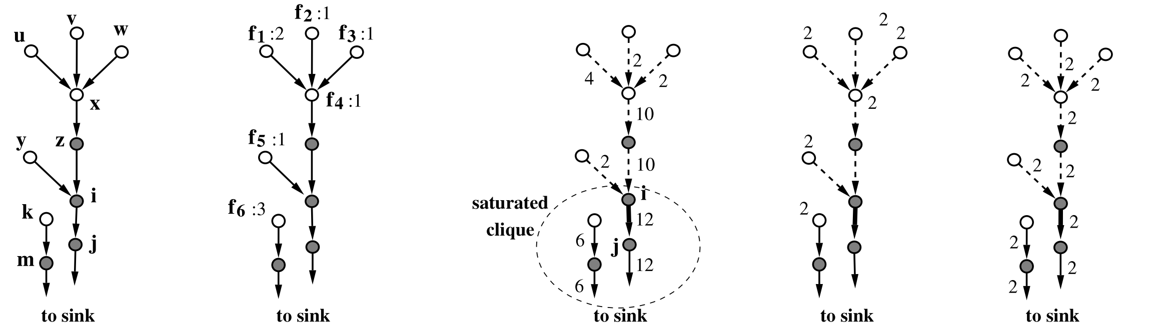
Fig. 1 gives a more sophisticated example. Satisfying the source condition ensures that the normalized rate of flow f_4 is as high as that of any other upstream flow. The buffer-saturated condition requires that flow f_1 has the same normalized rate as f_2, f_3 and f_4. Because f_1's weight is 2, its actual rate should be twice that of f_2, f_3 or f_4. To satisfy this condition, rate limits must be applied at v, w and x to give more bandwidth to u. Satisfying the bandwidth-saturated requirement ensures that the normalized rates of flows (f_1 through f_5) passing the bandwidth-saturated link (i, j) are as large as any contending flows (f_6). This may require a rate limit to be applied at k on f_6. The rate-limit condition makes sure that this rate limit is not set too low.

5.4. Correctness proof

We prove the equivalence between the global maxmin objective and the four local conditions.

The portion of a flow's routing path from the first node whose buffer is saturated to the first bandwidth-saturated link is called the *primary saturated subpath* of the flow. It is easy to see that the primary saturated subpath of a flow consists of a chain of buffer-saturated links and a bandwidth-saturated link at the end. The chain of buffer-saturated links in the primary subpath is the result of buffer-based backpressure originated from the bandwidth-saturated link, which is demonstrated in Fig. 1(c), where the bottleneck link (i, j) causes the upstream links buffer-saturated. It is possible that the primary saturated subpath of a flow does not have a buffer-saturated link. For a flow f, given the fact that the buffer at its destination is unsaturated,[5] there must be a node

[5] We assume the destination of each flow is capable of timely dealing with incoming packets and keeping its buffer unsaturated.



(a) Topology. (b) Flow weights. (c) Data rates in links. (d) Normalized rates of flows. (e) Normalized rates on links.

Fig. 1. White circles represent flow sources. Gray circles represent other nodes. Thick arrows represent bandwidth-saturated links. Thin arrows represent unsaturated links. Thin dashed arrows represent buffer-saturated links. (a) A portion of the network is shown with each arrow pointing from an upstream node to its downstream neighbor. (b) There are six flows, f_1 through f_6, whose weights are shown beside their sources. (c) The actual data rates of the links are shown. (i, j) is a bandwidth-saturated link, which sends buffer-based backpressure upstream, creating buffer-saturated links all the way to the flow sources and slowing the flow rates. (d) The normalized rates of the flows are shown beside the sources. (e) The normalized rates on the links are shown.

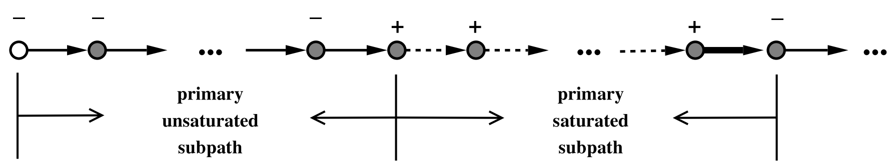


Fig. 2. White circle represents the flow source. Gray circles represent other nodes. Thick arrows represent bandwidth-saturated links. Thin arrows represent unsaturated links. Thin dashed arrows represent buffer-saturated links. "−" on top of a node indicates an unsaturated buffer at that node. "+" indicates a saturated buffer.

whose buffer is unsaturated on its routing path. If the buffer at the source of f is saturated, f must have a primary saturated subpath.

For a flow f with $r(f) < d(f)$, the first bandwidth-saturated link whose normalized rate is equal to $\mu(f)$ on the routing path of f is called the *primary bandwidth-saturated link* of f.

Lemma 2. *For any flow f with $r(f) < d(f)$ and a saturated buffer at the source, the primary bandwidth-saturated link is the first bandwidth-saturated link on the routing path if the source condition and the buffer-saturated condition are satisfied.*

Proof. First we consider the case where the first link on the routing path of f is a bandwidth-saturated link, which forms the entire primary saturated subpath of f. By the source condition and Lemma 1, the normalized rate of this link must be equal to $\mu(f)$. Then this link is the primary bandwidth-saturated link.

Next we consider the case where the first link on the routing path is not a bandwidth-saturated link. Let the primary saturated subpath be $i_1 \to i_2 \to \cdots \to i_k \to i_{k+1}$, where i_1 is the source of the flow, (i_l, i_{l+1}), $1 \le l < k$, are all buffer-saturated and (i_k, i_{k+1}) is bandwidth-saturated. The buffers at nodes i_l, $1 \le l \le k$, are all saturated.

We prove by induction that the normalized rates of links (i_l, i_{l+1}), $1 \le l \le k$, are all equal to $\mu(f)$. By the source condition and Lemma 1, $\mu(i_1, i_2) = \mu(f)$. Suppose $\mu(i_{l-1}, i_l) = \mu(f)$, $1 < l \le k$. Because (i_{l-1}, i_l) is buffer-saturated, by the buffer-saturated condition and Lemma 1, we must have $\mu(i_l, i_{l+1}) = \mu(i_{l-1}, i_l) = \mu(f)$, which completes the induction proof. Therefore, (i_k, i_{k+1}) is the primary bandwidth-saturated link of flow f. □

For a flow with an unsaturated buffer at the source, the portion of its routing path from the source to the first node whose buffer is saturated, or to the destination if there is no such node, is called the *primary unsaturated subpath* of the flow. Its routing path begins with the primary unsaturated subpath followed by the primary saturated subpath, as shown in Fig. 2. It is possible that a flow does not have a primary saturated subpath. In that case, the primary unsaturated subpath of the flow forms the entire routing path.

Lemma 3. *When the four localized conditions are satisfied in the network, for any flow f with a rate limit and an unsaturated buffer at the source, if $r(f)$ is increased by a small amount, the violation incurred by f must occur at a link on its routing path and the normalized rate of the link must be equal to $\mu(f)$ before the rate increment on f.*

Proof. By the rate-limit condition, when $r(f)$ is increased, one or more local conditions are violated and the source of f is required to reduce the rate of f. Suppose the amount of f's rate increment is very small and the buffer at the source of f is still unsaturated. The violation will not appear at the source because the source condition and the buffer-saturated condition are not applicable at a node with an unsaturated buffer. Therefore, the violation must appear on at least one link on the routing path of f. Because any small amount of rate increment on f can introduce a violation and the rate reduction request will always be sent to the source of f, the normalized rate of the link where the violation appears must be equal to $\mu(f)$ before the rate increment on f. □

Lemma 4. *For any flow f with $r(f) < d(f)$ and an unsaturated buffer at the source, if the four localized conditions are satisfied, there must be a link on the routing path of f such that* (1) *the link has the largest normalized rate in at least one saturated clique it belongs to, and* (2) *the link's normalized rate is equal to $\mu(f)$.*

Proof. There must be a rate limit at the source of f because $r(f) < d(f)$ and the buffer at the source of f is unsaturated. Suppose the rate of f is increased by a small amount. By rate-limit condition, the rate increment on f will violate at least one of the first three local conditions.

The violation can occur on the primary unsaturated subpath. In this case, the bandwidth-saturated condition may be violated by f. Let (i, j) be the first link on the primary unsaturated subpath on which the bandwidth-saturated condition is violated: namely, there exists a bandwidth-saturated link, denoted as (i', j'), which no longer has the largest normalized rate in any saturated clique. Because any small amount of rate increment on f can introduce the violation, both (i, j) and (i', j') must have the largest normalized rate in a saturated clique before f's rate is increased. By Lemma 3, the normalized rate of (i, j) is equal to $\mu(f)$.

For a flow f with $r(f) < d(f)$, the first unsaturated link on the routing path that has the largest normalized rate equal to $\mu(f)$ in at least one saturated clique it belongs to is called the *primary unsaturated link* of f. In the above case, (i, j) is the primary unsaturated link of f.

If f has a primary saturated subpath, the source condition or the buffer-saturated condition may also be violated at the last link of the primary unsaturated subpath of f (denoted by (i_0, i_1)). Let the primary saturated subpath be $i_1 \to i_2 \to \cdots \to i_k \to i_{k+1}$, where (i_l, i_{l+1}), $1 \le l < k$, are all buffer-saturated and (i_k, i_{k+1}) is bandwidth-saturated. We will prove below that (i_k, i_{k+1}) is the primary bandwidth-saturated link of f.

By Lemma 3, $\mu(i_0, i_1) = \mu(f)$. By the source condition and the buffer-saturated condition, among all upstream links and local flows of i_1, the buffer-saturated upstream links and the local flows of i_1 have the largest normalized rates. Because any small amount of rate increment on f will violate the source condition or the buffer-saturated condition, (i_0, i_1) must also have the largest normalized rate before f's rate is increased. By Lemma 1, we have $\mu(i_1, i_2) = \mu(f)$. By the buffer-saturated condition and Lemma 1, all other links on the primary saturated subpath have the same normalized rate as (i_1, i_2). Then $\mu(i_k, i_{k+1}) = \mu(f)$. Therefore, (i_k, i_{k+1}) is the primary bandwidth-saturated link of f.

If the violation happens on a link after (i_0, i_1) on the routing path, the normalized rate of (i_1, i_2) must be equal to $\mu(f)$ before $r(f)$ is increased. This can be proved by contradiction. Assume $\mu(i_1, i_2) > \mu(f)$ before $r(f)$ is increased. By Lemma 1, all links on the routing path after (i_1, i_2) also have normalized rates larger than $\mu(f)$. Among all links on the routing path from (i_1, i_2), there is a link on which the violation of flow f occurs. The normalized rate of that link is larger than $\mu(f)$ before $r(f)$ is increased, which contradicts with Lemma 3. By the buffer-saturated condition and Lemma 1, all other links on the primary saturated subpath have the same normalized rate as (i_1, i_2). Then $\mu(i_k, i_{k+1}) = \mu(f)$. Therefore, (i_k, i_{k+1}) is the primary bandwidth-saturated link of f.

By the bandwidth-saturated condition, (i_k, i_{k+1}) has the largest normalized rate in at least one saturated clique. □

For a flow f, its primary bandwidth-saturated link or primary unsaturated link, whichever appears first, is called the *primary link* of f. By summarizing Lemmas 2 and 4, we can get the lemma below.

Lemma 5. *For any flow f with $r(f) < d(f)$, if the four localized conditions are satisfied, f must have a primary link. The primary link of f has the largest normalized rate which is equal to $\mu(f)$ in at least one saturated clique it belongs to.*

Theorem 1. *When all flows have a common destination, the global maxmin objective is achieved if the four local conditions are satisfied.*

Proof. Suppose the local requirements are achieved. For an arbitrary flow f with $r(f) < d(f)$, we need to prove that, in order to increase the normalized rate $\mu(f)$, we have to decrease the normalized rate $\mu(f')$ of another flow f', for which $\mu(f') \le \mu(f)$.

We prove it by contradiction. Assume to the contrary that there exists such a flow f that $\mu(f)$ can be increased without decreasing $\mu(f')$ for all flows f' with $\mu(f') \le \mu(f)$.

By Lemma 5, flow f has a primary link (i, j) and $\mu(i, j) = \mu(f)$. It means that the normalized rates of all other flows passing (i, j) are not greater than $\mu(f)$. When we increase $\mu(f)$ by increasing the rate of f, based on the assumption, the normalized rates of all other flows passing (i, j) will not be decreased, which means that (i, j)'s rate will go up. By Lemma 5, (i, j) has the largest normalized rate in a saturated clique. When (i, j)'s rate goes up, the rate of another link (i', j') in the saturated clique will have to go down. Among all flows passing (i', j'), at least one flow f' has to decrease its rate (and thus $\mu(f')$). Since (i, j) has the largest normalized rate in the clique, we have $\mu(f') \le \mu(i', j') \le \mu(i, j) = \mu(f)$, which contradicts with the previous assumption. □

6. Local conditions for global maxmin: multiple-destination 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.

6.1. 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 Fig. 3(a). First, we show that one queue per node will unnecessarily reduce the rate of f_2 in Fig. 3(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,[6] f_2 will have the same rate as f_1, even though there is 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 be destined for the same destination, i.e., the gateway to the Internet. In Fig. 3(c), when i and j keep separate queues for destinations t and v, f_2 will be able to send at its desired rate of 5.

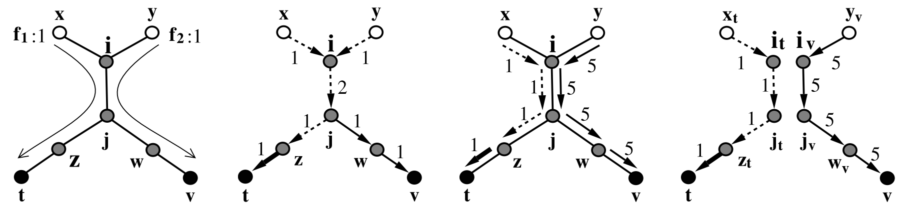
Separate queues achieve "isolation" between packets for different destinations, which allows us to model the physical wireless network as a set of overlapping virtual networks, each for one destination. Fig. 3(d) shows that f_1 and f_2 are delivered in two virtual networks with separate packet queues but sharing the same channel.

6.2. 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 toward 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 Fig. 3(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 3.2) 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 4.2. Other concepts can also be trivially extended to virtual networks.

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



(a) Topology. (b) Link rates (single queue). (c) Link rates (multiple queues). (d) Two virtual networks.

Fig. 3. 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 desired 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.

6.3. Localized requirements for global maxmin

Below we modify the local conditions in Section 5.3 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.

Lemmas 1–5 can be easily extended to virtual networks. The proofs are omitted to avoid excessive repetition.

Lemma 6. *The normalized rate of virtual link (i_t, j_t) is equal to the largest value among the normalized rates of all upstream virtual links of i_t and the normalized rates of all flows whose sources are i_t.*

Lemma 7. *For any flow f with $r(f) < d(f)$ and a saturated buffer at the source, the primary bandwidth-saturated virtual link is the first bandwidth-saturated virtual link on the routing path if the source condition and the buffer-saturated condition are satisfied.*

Lemma 8. *When the four localized conditions are satisfied in the grand virtual network, for any flow f with a rate limit and an unsaturated buffer at the source, if $r(f)$ is increased by a small amount, the violation incurred by f must happen at a virtual link on its routing path and the normalized rate of the virtual link must be equal to $\mu(f)$ before f's rate is increased.*

Lemma 9. *For any flow f with $r(f) < d(f)$ and an unsaturated buffer at the source, if the four localized conditions are satisfied, there must be a virtual link on the routing path of f such that (1) the link has the largest normalized rate in at least one saturated clique it belongs to, and (2) the link's normalized rate is equal to $\mu(f)$.*

Lemma 10. *For any flow f with $r(f) < d(f)$, if the four localized conditions are satisfied, f must have a primary virtual link. The primary virtual link of f has the largest normalized rate which is equal to $\mu(f)$ in at least one saturated clique it belongs to.*

Theorem 2. *The global maxmin objective is achieved if the four local conditions are satisfied in the grand virtual network.*

Proof. Suppose the local requirements are achieved. For an arbitrary flow f with $r(f) < d(f)$, we need to prove that, in order to increase the normalized rate $\mu(f)$, we have to decrease the normalized rate $\mu(f')$ of another flow f', for which $\mu(f') \le \mu(f)$.

We prove it by contradiction. Assume to the contrary that there exists such a flow f that $\mu(f)$ can be increased without decreasing $\mu(f')$ for all flows f' with $\mu(f') \le \mu(f)$.

By Lemma 10, flow f has a primary virtual link (i_t, j_t) and $\mu(i_t, j_t) = \mu(f)$. It means that the normalized rates of all other flows passing (i_t, j_t) are not greater than $\mu(f)$. When we increase $\mu(f)$ by increasing the rate of f, based on the assumption, the normalized rates of all other flows passing (i_t, j_t) will not be decreased, which means that (i_t, j_t)'s rate will go up. By Lemma 10, (i_t, j_t) has the largest normalized rate in a saturated clique. When (i_t, j_t)'s rate goes up, the rate of another virtual link (i'_v, j'_v) in the saturated clique will have to go down. Among all flows passing (i'_v, j'_v), at least one flow f' has to decrease its rate (and thus $\mu(f')$). Since (i_t, j_t) has the largest normalized rate in the clique, we have $\mu(f') \le \mu(i'_v, j'_v) \le \mu(i_t, j_t) = \mu(f)$, which contradicts with the previous assumption. □

7. 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 6.3, which is equivalent to meeting the global maxmin objective in wireless networks with multiple destinations.

7.1. 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.

Even after the conditions are satisfied, the network/traffic dynamics may cause them to be violated again. The protocol will continuously change the flow rates to restore the conditions and achieve global maxmin in the current network/traffic environment.

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.

Below we explain the operations performed in the measurement and adjustment periods. Note that the operations by a virtual node i_t are *actually* performed by the physical node i.

7.2. 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[7] 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.

7 Or just DATA/ACK if RTS/CTS are turned off.

*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 6.3, 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). However, due to the disparity between transmission range and interference range. Such a task is not always possible because a node with a contending link may not be reachable when it is outside of transmission range. As a practical compromise, our protocol sends the link state to all 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 re-broadcasts the information to its neighbors.

The state of a link is very small. Instead of making a separate transmission, such information can be disseminated by piggybacking in RTS/CTS/DATA/ACK packets, which are overheard by all nodes in one-hop neighborhood. In this design, i piggybacks the state of (i, j) in its normal transmission, and after overhearing the information, a node in i's dominating set does the same thing. When i has multiple adjacent links, if each packet piggybacks the state of one link, it takes multiple packets for i to disseminate the state of all links. In addition, to overcome failed overhearing, the same information (about the state of a link) should be piggybacked in a number of transmissions. We stress that the piggyback design can be applied to disseminate other information in the rest of the protocol as well.

7.3. 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. (The motivation for the above rate reduction scheme is straightforward. While reducing by β percentage is the norm, an optimization is added—when the gap between L_1 and S_1 is too big, reducing by half helps to close the gap quickly. The number 3 is artificially set.)

(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 adopt a two-hop interference model when computing contention cliques. In this model, a wireless link (i, j) contends with other links that are within two hops from i or j. For example, consider a topology $x-y-i-j-w-v$. Link (x, y) is two hops from i. Because the transmission of ACK by y will interfere the reception of ACK by i from j, (x, y) and (i, j) are considered to be contending links.[8]

After deployment, each node i needs to discover the wireless topology in its three-hop neighborhood in order to pre-computes the set of contention cliques it belongs to. Consider an arbitrary adjacent link (i, j). We first identify the set of contending links, which are within two hops from i or j, and then identify the contention relationship among those links (based on the similar two-hop rule). Finally, we can compute the cliques based on such contention relationship.

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. Since a node pre-computes the set of cliques it belongs to, it knows whether or not its identifier is the smallest in any of those cliques, and if so, it will choose a sequence number that it has not used before in order to ensure the uniqueness of the clique identifier.

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, the control packet keeps the rate increase request that has the smallest increase. If there is a rate reduction request, it discards all rate increase requests and only keeps the reduction request. If there are multiple rate reduction requests for the flow, the packet keeps the one that has the largest 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.

8. Simulation

We perform simulations to evaluate the proposed distributed global maxmin protocol (GMP). We created a packet-level simulation testbed for multihop wireless networks. It implements IEEE 802.11 DCF, congestion avoidance [8], GMP, and the most related work [22] for comparison. IEEE 802.11 DCF is implemented using the parameters from the standard [35]. RTS/CTS/DATA/ACK exchange is implemented for virtual carrier sensing. Radio collision happens when the receiver of a data/control packet is within the interference range of another concurrent transmission. The simulation setup is described as follows. The channel capacity is 11 Mbps. Each node has a transmission range of 250 m. Each data packet is 1024 bytes long. The desired rate of any flow is 800 packets per second. The overall buffer space at a node can hold 300 packets. Each packet queue can hold 15 packets except when IEEE 802.11 DCF is used alone, in which case there is a single queue taking all the buffer space. The length of each simulation session is 400 s. Flow rates are measured in packets per second. Each measurement or adjustment period is 2 s long. β is set to be 10%.

8.1. Effectiveness of GMP

The network topology used in the first simulation is shown in Fig. 4. 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),

8 For 802.11's virtual sensing mechanism, before i exchanges DATA/ACK with j, i transmits RTS to make sure that its neighbors do not transmit or receive (which effectively silences links within two hops from i), and then j transmits CTS to make sure that its neighbors do not transmit or receive (which effectively silences links within two hops from j).

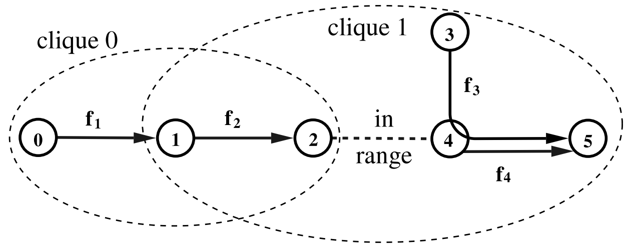


Fig. 4. Network topology of a simple scenario.

Table 1
Simulation results on the topology in Fig. 4. Flow rates are measured in packets per second.

Flow	⟨0, 1⟩	⟨1, 2⟩	⟨3, 5⟩	⟨4, 5⟩
Rate	544	207	213	219

Table 2
Simulation results for weighted maxmin in Fig. 4. Flow rates are measured in packets per second.

Flow	⟨0, 1⟩	⟨1, 2⟩	⟨3, 5⟩	⟨4, 5⟩
Weight	1	2	1	3
Rate	519	231	122	372

(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. It takes only ten measurement/adjustment periods for all flow rates to converge. Fig. 5a shows the rates of the flows in the first 20 periods. 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. It takes fourteen measurement/adjustment periods for all flow rates to converge in this case. Fig. 5b shows the rates of the flows in the first 20 periods. 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.

8.2. 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 [22]. These three protocols use different buffer management strategies to accommodate their packet queueing 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 15 packets. In GMP, all flows to the same destination share a common queue that can hold 15 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*.

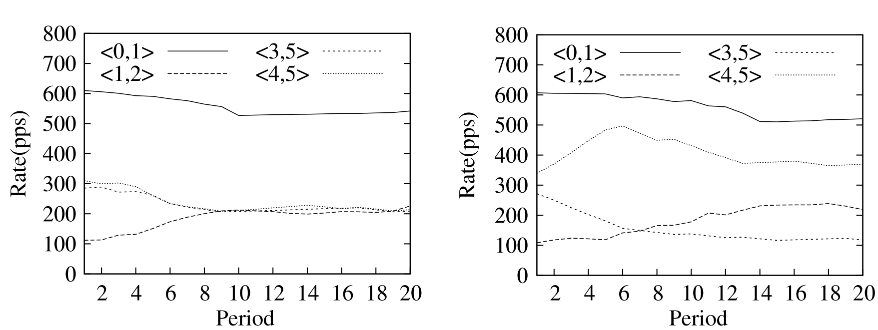
To evaluate the end-to-end fairness, we adopt the maxmin fairness index [3] (denoted by I_{mm}) and Jain's fairness index [9] (denoted by I_{jn}).

$$I_{mm} = \frac{\min\limits_{f\in F}\{r(f)\}}{\max\limits_{f\in F}\{r(f)\}}, \qquad I_{jn} = \frac{\left(\sum\limits_{f\in F} r(f)\right)^2}{|F| \sum\limits_{f\in F}(r(f))^2}$$

I_{mm} measures the ratio of the smallest flow rate to the largest flow rate. I_{jn} 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 toward 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.

One may question why U is defined as $\sum_{f\in F} r(f) \times l_f$ instead of $\sum_{f\in F} r(f)$. We explain the reason by using a simple example in Fig. 6, where three flows are carried by three links that mutually contend. To maximize $\sum_{f\in F} r(f)$, we would have to assign all channel capacity to the shortest flow ⟨2, 3⟩, which is extremely unfair. On the other hand, to be maxmin-fair, all flows should have equal rates in this example. We should assign $\frac{3}{6}$ of the channel capacity to flow ⟨0, 3⟩. This bandwidth must be divided among the three links that the flow passes, and consequently the flow rate corresponds to $\frac{1}{6}$ of channel capacity. We should assign $\frac{2}{6}$ of the channel capacity to flow ⟨1, 3⟩ that passes two links, and assign $\frac{1}{6}$ of channel capacity to flow ⟨2, 3⟩, such that each end-to-end flow has the same rate that corresponds to $\frac{1}{6}$ channel capacity. This bandwidth assignment achieves the maxmin objective. It does not



(a) Same weight. (b) Different weights.

Fig. 5. Rates of the flows in the first 20 periods in Fig. 4.

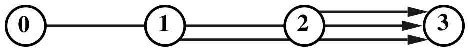


Fig. 6. A three-link topology.

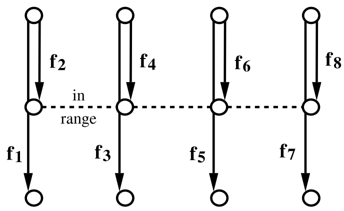


Fig. 7. Network topology.

Table 3
Simulation results on the topology in Fig. 6. Flow rates are measured in packets per second.

flow	802.11	2PP	GMP
⟨0, 3⟩	81	132	166
⟨1, 3⟩	220	189	175
⟨2, 3⟩	174	241	178
U	856	1014	1026
I_{mm}	0.366	0.547	0.935
I_{jn}	0.882	0.946	0.999

Table 4
Simulation results on the topology in Fig. 7. Flow rates are measured in packets per second.

flow	802.11	2PP	GMP
f_1	222	43	138
f_2	222	347	143
f_3	107	43	135
f_4	107	87	137
f_5	106	43	134
f_6	106	87	132
f_7	223	43	142
f_8	223	347	142
U	1977	1213	1653
I_{mm}	0.476	0.124	0.923
I_{jn}	0.890	0.514	0.999

maximize $\sum_{f\in F} r(f)$, but it maximizes the channel utilization in terms of $\sum_{f\in F} r(f) \times l_f$. In fact, all bandwidth assignment schemes that fully utilize the channel capacity while not wasting much bandwidth due to packet drops will have high effective network throughput U, although they may not achieve the same value of U because their link contention overhead can be different. A fair bandwidth assignment scheme should have high values not only for I_{mm} and I_{jn} but also for U.

We perform simulation for the scenario in Fig. 6. 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) [14]. 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.

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. 7, and the results are shown

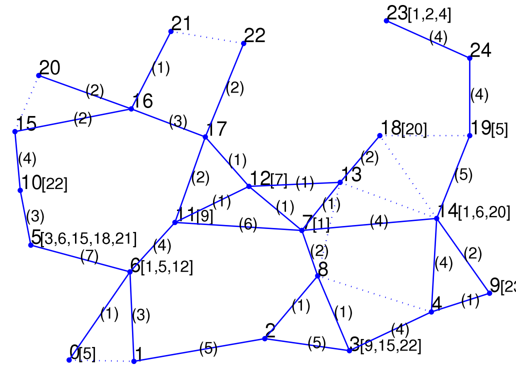


Fig. 8. Network topology.

Table 5
Simulation results on the topology in Fig. 8.

	802.11	2PP	GMP
U	1666	1673	2652
I_{mm}	0.002	0.017	0.218
I_{jn}	0.327	0.298	0.842

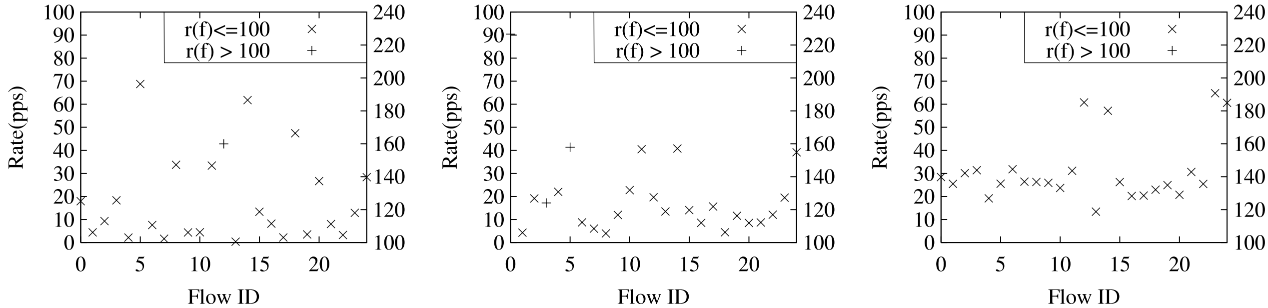
in Table 4. With this topology, the basic fair share calculated based on the formula in [22] is small, and the remaining bandwidth is distributed heavily biased toward f_2 and f_8 based on the linear programming approach in the same paper. Under 802.11, the flows in the middle (f_3, f_4, f_5 and f_6) 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. The flows in the middle have slightly lower rates for two possible reasons. First, under GMP, two flow rates are considered to be "equal" if their difference is below β, which is 10% in our simulations. Second, the maximum combined rate of the four links in the middle (which form a contention clique) is slightly lower than those of other cliques due to more contention in the middle clique.

Finally, we perform simulations on a more complex network topology shown in Fig. 8. The network consists of 25 nodes that are deployed in a 900 × 900 m^2 region. We create 25 multihop flows in the network, where the source and the destination of each flow are randomly chosen. The destinations of the flows starting from a node are listed in square brackets after the source node ID. The wireless links are shown as edges in the graph. A solid line means that the link is on the routing path of at least one flow. The total number of flows passing a links is shown in parentheses beside that link. A dotted line represents an unused link. The simulation results are shown in Fig. 9 and Table 5. In Fig. 9, the flow rates that are under 100 pps (packets per second) use the numbers on the left vertical axis; the flow rates above 100 pps use the numbers on the right vertical axis.

Under 802.11, half of all flows have rates under 10 pps. Several flows (e.g. f_7 and f_{13}) are almost starved. Under 2PP, three one-hop flows, f_0 (from node 6 to node 5 in Fig. 8), f_3 (from node 6 to node 1), and f_5 (from node 12 to node 7), have very high rates and contribute more than 50% of the total end-to-end throughput. The three flows whose rates are around 40 pps (f_{11}, f_{14} and f_{24}) are also short flows that are only one-hop or two-hops long. GMP achieves far better fairness as shown in Fig. 9c.

9. Conclusion

In this paper, we developed a novel theory and proposed a distributed protocol to achieve the global maxmin objective based on four local conditions. We introduced several new concepts,



(a) 802.11. (b) 2PP. (c) GMP.

Fig. 9. Rates of the flows on the topology in Fig. 8.

including link classification based on buffer state, virtual links, and virtual networks, which are essential for the development of the local conditions. We performed extensive simulations to evaluate the effectiveness of the proposed protocol and demonstrate that it works far better than existing protocols.

Acknowledgment

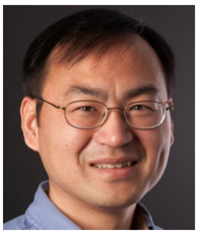
This work was supported in part by the US National Science Foundation under grant CNS-0644033 and grant CPS-0931969.

References

[1] U. Akyol, M. Andrews, P. Gupta, J. Hobby, I. Saniee, A. Stolyar, Joint scheduling and congestion control in mobile ad-hoc networks, in: Proc. IEEE INFOCOM'08, April 2008.
[2] H. Balakrishnan, V. Padmanabhan, S. Seshan, R. Katz, A comparison of mechanisms for improving TCP performance over wireless links, in: Proc. ACM SIGCOMM, August 1996.
[3] D. Bertsekas, R. Gallager, Data Networks, second ed., Prentice-Hall Inc, 1992.
[4] P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, Proc. of 3rd Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, DialM'99, August 1999.
[5] L. Chen, S.H. Low, M. Chiang, J.C. Doyle, Cross-layer congestion control, routing, and scheduling design in Ad Hoc wireless networks, in: Proc. IEEE INFOCOM'06, April 2006.
[6] S. Chen, N. Yang, Congestion avoidance based on light-weight buffer management in sensor networks, in: Localized Communication and Topology Protocols for Ad Hoc Networks, IEEE Trans. Parallel Distrib. Syst. 17 (9) (2006) (special issue).
[7] X. Chen, H. Zhai, J. Wang, Y. Fang, TCP performance over mobile ad hoc networks, Can. J. Electr. Comput. Eng. 29 (2004) 129–134.
[8] S. Chen, Z. Zhang, Localized algorithm for aggregate fairness in wireless sensor networks, in: Proc. ACM Mobicom'06, September 2006.
[9] D. Chiu, R. Jain, Analysis of the increase/decrease algorithms for congestion avoidance in computer networks, J. Comput. Networks ISDN 17 (1) (1989) 1–14.
[10] Y. Ding, Y. Yang, L. Xiao, Multi-path routing and rate allocation for multi-source video on-demand streaming in wireless mesh networks, in: Proc. IEEE INFOCOM, April 2011.
[11] A. Eryilmaz, R. Srikant, Joint congestion control, routing and MAC for stability and fairness in wireless networks, in: Nonlinear Optimization of Communication Systems, IEEE J. Sel. Areas Commun. 24 (8) (2006) (special issue).
[12] J.M. Faffe, Bottleneck flow control, IEEE Trans. Commun. COM-29 (7) (1981) 954–962.
[13] V. Gambiroza, B. Sadeghi, E.W. Knightly, End-to-end performance and fairness in multihop wireless backhaul networks, in: Proc. Mobicom'04, Philadelphia, PA, USA, September–October 2004.
[14] X.L. Huang, B. Bensaou, On max–min fairness and scheduling in wireless ad-hoc networks; analytical framework and implementation, in: Proc. MobiHoc'01, Long Beach, California, October 2001.
[15] IEEE standard for telecommunications and information exchange between systems – LAN/MAN specific requirements – Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications: medium access control (MAC) quality of service enhancements. IEEE Std 802.11e, Nov 2005
[16] L. Jiang, J. Walrand, A distributed CSMA algorithm for throughput and utility maximization in wireless networks, in: Proc. Allerton Conference on Communication, Control, and Computing, September 2008.
[17] B. Karp, H. Kung, GPSR: greedy perimeter stateless routing for wireless networks, in: Proc. ACM MobiCom'00, August 2000.
[18] K. Kar, S. Sarkar, L. Tassiulas, Achieving proportionally fair rates using local information in aloha networks, IEEE Trans. Autom. Control 49 (10) (2004).
[19] J.M. Kleinberg, Y. Rabani, E. Tardos, Fairness in routing and load balancing, IEEE Symposium on Foundations of Computer Science, 1999.
[20] D.J. Leith, Q. Cao, V.G. Subramanian, Realising max–min fairness in 802.11e mesh networks, in: Proc. IEEE International Symposium on Wireless Pervasive Computing (ISWPC), 2010.
[21] D.J. Leith, Q. Cao, V.G. Subramanian, Max–min fairness in 802.11 mesh networks, IEEE/ACM Trans. Netw. (in press).
[22] B. Li, End-to-end fair bandwidth allocation in multi-hop wireless ad hoc networks, in: Proc. IEEE ICDCS'05, June 2005.
[23] H. Luo, J. Cheng, S. Lu, Self-coordinating localized fair queueing in wireless ad hoc networks, IEEE Trans. Mob. Comput. 3 (1) (2004).
[24] H. Luo, S. Lu, V. Bharghavan, A new model for packet scheduling in multihop wireless networks, in: Proc. MobiCom'00, August 2000.
[25] E. Modiano, D. Shah, G. Zussman, Maximizing throughput in wireless networks via gossiping, SIGMETRICS Perform. Eval. Rev. 34 (1) (2006) 27–38.
[26] M. Neely, E. Modiano, C. Li, Fairness and optimal stochastic control for heterogeneous networks, IEEE/ACM Trans. Netw. 16 (2) (2008).
[27] C.E. Perkins, E.M. Royer, Ad hoc on-demand distance vector routing, in: Proc. the 2nd IEEE Workshop on Mobile Computing Systems and Applications, February 1999.
[28] Y. Qiu, P. Marbach, Bandwidth allocation in wireless ad-hoc networks: a price-based approach, in: Proc. IEEE INFOCOM'03, March 2003.
[29] S. Rangwala, R. Gummadi, R. Govindan, K. Psounis, Interference-aware fair rate control in wireless sensor networks, in: Proc. ACM SIGCOMM, 2006.
[30] S. Sanghavi, L. Bui, R. Srikant, Distributed link scheduling with constant overhead, in: Proc. ACM SIGMETRICS, 2007.
[31] S. Sarkar, L. Tassiulas, End-to-end bandwidth guarantees through fair local spectrum share in wireless adhoc networks, IEEE Trans. Automat. Control 50 (9) (2005).
[32] L. Tassiulas, S. Sarkar, Maxmin fair scheduling in wireless networks, in: Proc. IEEE INFOCOM'02, June 2002.
[33] X. Wang, K. Kar, Distributed algorithms for max–min fair rate allocation in aloha networks, in: Proc. the 42nd Annual Allerton Conference, Urbana–Champaign, October 2004.
[34] X. Wang, K. Kar, J. Pang, Lexicographic max-min fairness in a wireless Ad-hoc network with random, in: Proc. IEEE CDC, December 2006.
[35] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Standard 802.11, 1999
[36] Y. Xue, B. Li, K. Nahrstedt, Optimal resource allocation in wireless ad hoc networks: a price-based approach, IEEE Trans. Mob. Comput. 5 (4) (2006).
[37] K. Xu, M. Gerla, L. Qi, Y. Shu, Enhancing TCP fairness in Ad Hoc wireless networks using neighborhood RED, in: Proc. ACM MOBICOM, 2003.
[38] Y. Yi, S. Shakkottai, Hop-by-hop congestion control over a wireless multi-hop network, in: Proc. IEEE INFOCOM'04, Hong Kong, China, March 2004.
[39] H. Zhai, Y. Fang, Distributed flow control and medium access in multihop ad hoc networks, IEEE Trans. Mob. Comput. 5 (11) (2006).
[40] W. Zhang, S. Bai, G. Xue, J. Tang, C. Wang, DARP: distance-aware relay placement in WiMAX mesh networks, in: Proc. IEEE INFOCOM, April 2011.

Liang Zhang received B.E. and M.E. degrees in Computer Science from Tsinghua University of China in 1999 and 2001, respectively, and received his Ph.D. degree in Computer Engineering from the University of Florida in 2009. He had worked with Oracle R & D Center in China from 2002 to 2003. Since 2008, he has been working with Juniper Networks. His research interests include sensor networks, multihop wireless networks, and network security.

Wen Luo (wluo@cise.ufl.edu) is a Ph.D. student in the Department of Computer and Information Science and Engineering at University of Florida. He obtained his B.S. degree from University of Science and Technology of China in 2008 and then he moved to University of Florida for graduate study. Currently he is a Ph.D. student under the supervision of Dr. Shigang Chen. His research interest includes radio-frequency identification systems, Internet traffic measurement, wireless resource management, etc.



Shigang Chen (sgchen@cise.ufl.edu) is an associate professor in the Department of Computer and Information Science and Engineering at University of Florida. He received his B.S. degree in computer science from University of Science and Technology of China in 1993. He received M.S. and Ph.D. degrees in computer science from University of Illinois at Urbana–Champaign in 1996 and 1999, respectively. After graduation, he had worked with Cisco Systems for three years before joining University of Florida in 2002. His research interests include network security and wireless networks. He received IEEE Communications Society Best Tutorial Paper Award in 1999 and NSF CAREER Award in 2007. He was a guest editor for ACM/Baltzer Journal of Wireless Networks (WINET) and IEEE Transactions on Vehicle Technologies. He served as a TPC co-chair for the Computer and Network Security Symposium of IEEE IWCCC 2006, a vice TPC chair for IEEE MASS 2005, a vice general chair for QShine 2005, a TPC co-chair for QShine 2004, and a TPC member for many conferences including IEEE ICNP, IEEE INFOCOM, IEEE ICC, IEEE Globecom, etc.

Ying Jian received his B.E. degree in 2001 and M.E. degree in 2004, both in computer science, from Tsinghua University of China. He received his Ph.D. degree in computer engineering from University of Florida in 2008. After graduation, he worked with Microsoft for two years and then joined Google in 2010. His research interests include QoS and security in multihop wireless networks and sensor networks.