

# A Distributed Solution for Fair Bandwidth Distribution among Data Flows in Wireless Networks

Ji-Yu Chen

Department of Computer Science, East China Normal  
University, Shanghai, China  
E-mail: jiyuchen1995@gmail.com

Shi-Ping Chen

School of Optical-Electrical and Computer Engineering  
Network Center  
Engineering Research Center of Optical Instrument and  
System (Ministry of Education), University of Shanghai  
for Science and Technology, Shanghai, China  
E-mail: chensp@usst.edu.cn

**Abstract**—In wireless networks, it is important to ensure fairness among data flows in their access to network bandwidth. Providing end-to-end fairness in mobile networks is particularly challenging. This paper proposes a packet tagging and prioritized scheduling scheme that achieves fairness under network dynamics, without the need of maintaining any flow-level state in the network. The new scheme is a distributed solution that can work with any routing protocol. Numerical evaluation shows that the proposed scheme achieves far better global fairness.

**Keywords**—wireless bandwidth allocation; fairness control

## I. INTRODUCTION

In wireless networks, it is an important issue to ensure fairness among data flows in their access to network bandwidth [1]. Imagine an Internet without TCP and all applications use UDP to grab as much bandwidth as possible without caring others. Some users will see a truly high speed Internet, while others, particularly those communicating via long paths, will be starved of bandwidth. The same thing is true for wireless networks. Studies have shown that multi-hop wireless networks using IEEE 802.11 DCF can be extremely unfair among end-to-end flows [2].

Much work has been done to achieve fairness in a wireless environment. Maxmin fairness in FDMA/CDMA networks with only one-hop flows was studied in [3]. Proportionally-fair congestion control in FDMA/CDMA networks was studied in [4], under the assumptions that each flow has a single routing path and that per-flow state information is maintained at intermediate nodes. A mixed zone fairness problem on wireless infrastructure was studied in [5]. Fairness in TDMA networks was studied in [6] with the goal of maximizing the minimal flow rate, assuming a tree routing structure from all data sources to a sink. Woo and Culler proposes a rate-control mechanism, which ensures that, at every congested node, the ratio between the rate of through traffic from  $n$  upstream data sources and the rate of locally generated traffic is  $1 : n$  [7]. Fairness is achieved only in a single-path routing network where all packets from the  $n$  upstream sources are routed through the congested node. Ee and Bajcsy [8] assumes a tree routing structure from all data sources to a base station. Each node learns the number of upstream data sources in the subtree rooted at itself. It

measures its downstream forwarding rate, and computes per-source fair rate, which is propagated upstream such that the data sources do not send packets beyond that rate. The assumption of a tree routing structure is restrictive because many popular routing protocols for wireless networks support dynamic multipath routing [9], [10], [11], [12]. Fairness at the MAC layer is studied in [13], [14], [15], [16], [17], [18], [19]. The TAP fairness in multi-rate wireless backhaul networks is investigated in [20], which achieves temporal fairness instead of throughput fairness.

Different from the above prior work, this paper considers end-to-end fairness among multi-hop flows in the more widely-used CSMA/CA networks, without any restriction on routing structure and without the need for the wireless nodes to keep track of per-flow state. We take a different approach to handle network dynamics and propose a packet tagging and prioritized scheduling scheme, called TFS (Tag-based Fairness Scheme), where we tag each packet with a counter value, which will assist the intermediate nodes on the routing path to make scheduling decision. Prioritized scheduling based on the packets' tags is designed to schedule packets among neighboring wireless nodes for partially ordered transmission. Each node makes localized scheduling decision, and the net outcome of the scheduling decisions made by all nodes will collectively ensure the fairness between competing flows, which may be generated from data sources that are distant from each other. We evaluate TFS by simulations. The results demonstrate that TFS achieves far better global fairness.

## II. NETWORK MODEL AND PROBLEM STATEMENT

A mobile wireless network consists of nodes that may be stationary or moving. Each node has a single radio and may either transmit or receive at a time. Assume the MAC-layer protocol is CSMA/CA. There is a wireless communication "link" from one node to another if the latter can correctly receive the former's signal. Two nodes are neighbors if they can receive data from each other, which is needed in order to perform RTS/CTS/DATA/ACK exchange. We assume there exists a neighbor discovering protocol. For example, each node periodically transmits a beacon message to identify itself, so that every node knows the set of its neighbors. Assume there exists a routing protocol. When a node

receives a packet, it knows the next hop(s) to which the packet should be forwarded.

Broadly speaking, the fairness problem is to ensure that each flow has an equal access to the end-to-end network bandwidth. More specifically, two flows sharing the same bottleneck should have the same rate. For flows that pass different bottlenecks, their rates do not have to be equal because they have different constraints.

### III. TAG-BASED FAIRNESS SCHEME (TFS)

This section proposes a packet tagging and prioritized scheduling scheme for fair bandwidth allocation in a mobile wireless network.

#### A. Design Goals

The proposed tag-based fairness scheme, called TFS, has the following design goals.

##### 1) Routing independence

It must be able to work with any routing protocol. A flow may follow a single or multiple routing paths.

##### 2) Freedom from per-flow state at nodes

Due to performance and scalability concerns, it is undesirable to deposit flow status information on intermediate nodes.

##### 3) Flow-level fairness

Competing flows at the same bottleneck are entitled to the same amount of network bandwidth. The fairness problem becomes complex when there are a large number of flows and each may follow multiple paths.

We design TFS based on a simple rule: during each period of time, the  $i$ th packet from a flow is given higher scheduling priority in the network than the  $j$ th packet from another flow if  $i$  is smaller than  $j$ . Not only can this rule be efficiently implemented, but it is very effective in ensuring the flow-level fairness, as will be demonstrated by our simulations. The details of TFS is explained below.

#### B. Packet Tagging

Each node  $x$  maintains a tag counter  $C_x$ . The counter is initialized to zero. Whenever the node generates a new data packet, the packet is tagged with the current counter value, and then the counter is increased by one. The tag may be implemented as an optional field in the packet header. The counter is reset to zero after each preset time interval, whose value is a system-wide parameter that are preconfigured at all nodes.

Our basic idea is to tag packets at the data sources and perform prioritized scheduling in the network based on the tags carried by the packets. A node places all received packets in a queue according to the ascending order of the tags. The packets with a tag value of zero have the highest priority for transmission; the packets with a tag value of one have the second highest priority; and so on. Packets with the smallest tags are transmitted first.

Before going to the design details, we use two examples to illustrate how the tag-based prioritized scheduling can achieve fairness. The examples do not involve mobile nodes, but our solution is suitable for mobile ad-hoc networks because each node makes scheduling decision locally based

on the packets' tags, without performing any end-to-end signaling to maintenance flow-level state.

The first example is a single-path routing case shown in Figure 1. Flow sources are presented by white nodes, and intermediate nodes are represented by black nodes. Consider an intermediate node  $x$ , which is able to forward eight packets downstream per time interval without causing congestion. In the absence of prioritized scheduling, this bandwidth is fairly divided between the two upstream neighbors of  $x$  by CSMA/CA. Such MAC-fair division is recursively performed upstream. Figure 1 (a) shows the average number of packets from each flow source that will be successfully delivered through  $x$  per time interval. MAC fairness does not mean flow-level fairness. The packet rate per source ranges from one to four packets per time interval. Figure 1 (b) shows that prioritized scheduling based on packet tags can equalize the data rates from different sources.

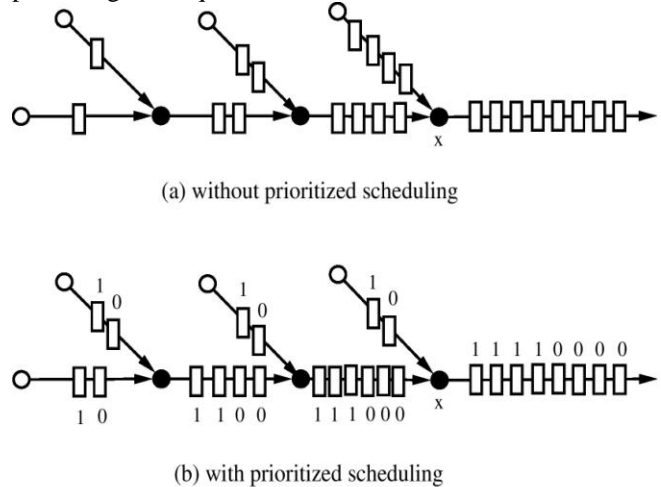


Figure 1. Achieving fairness with packet tagging — single-path routing case.

Packets with tag zero are scheduled first, and those with tag one are scheduled next. Packets with tag two or higher are not drawn in the figure. However, in a steady state, those packets will have little chance to pass through  $x$  because there are sufficient packets with tag zero or one to consume all downstream bandwidth at  $x$ .

The second example is a multi-path routing case in Figure 2. Suppose  $x$  is able to send six packets downstream per time interval, and  $y$  is able to send four packets downstream per time interval. Suppose flow source  $u$  sends packets on two paths through  $x$  and  $y$ , but source  $v$  sends packets only through  $x$ . In Figure 2 (a), without prioritized scheduling,  $u$  is able to send seven packets on average per time interval, but  $v$  can only send three on average, based on CSMA/CA. In Figure 2 (b), with prioritized scheduling, they can both send five because they advance their tag counters at the same rate, one per packet. As we will see from the detailed design in Section IIID,  $u$  prefers  $y$  to forward its packets due to less competition. It also acquires additional bandwidth from  $x$ . Given that  $x$ 's bandwidth is six, which is shared by two upstream neighbors, a packet with tag greater than four will not be able to pass through  $x$ .

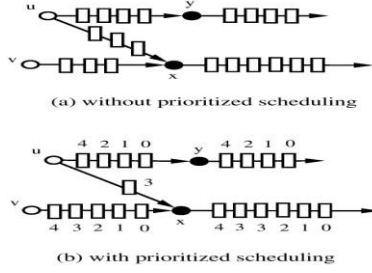


Figure 2. Achieving fairness with packet tagging — multi-path routing case.

### C. Tag Swapping

When packets from different flows are mixed at an intermediate node, as shown in Figure 3 (a), they are reordered based on packet tags before being sent out. Reordering is important to achieving fairness in prioritized scheduling. It essentially places packets from different flows in a sequence of tag-based prioritized groups. Competing flow sources produce members in a particular group (for a particular tag value) at the same rate. The only way for a flow source to generate additional packets is to use larger tags. These additional packets will be placed in lower-priority groups located at the back of the packet queue, and consequently they will not interfere with packets from other sources that produce data at smaller rates (thus using smaller tags). The problem is that reordering also happens among packets from the same flow source, as shown in Figure 3 (b). The tag space is limited to a certain number  $l$  of bits. Packets generated in different time intervals have overlapped tags. At an intermediate node, when packets generated in a later interval catch up with packets generated in an earlier interval from the same source, they are reordered, which may be undesirable for some flows (such as TCP flows) that require the arrival order at a flow's destination remains the same as the sending order at the flow's source; these flows should each be routed along a single path. Instead of reordering packets from the same source, our solution is to swap their tags. Consider two packets,  $p1$  and  $p2$ , from the same source. Even though  $p1$  is generated before  $p2$ , it may carry a larger tag due to reset of the tag counter. When they both appear in the queue of an intermediate node, the node will keep their order in the packet queue but swap their tags.

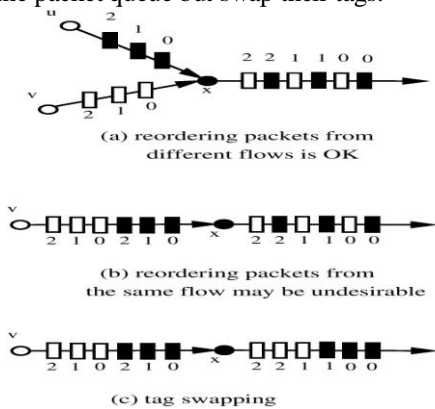


Figure 3. Packet reordering and tag swapping.

Tag swapping, as illustrated in Figure 3 (c), does not enhance or reduce the scheduling priority of packets from a flow source as a whole, but it ensures that the order of those packets in transit remains the same as the order in which they are created.

### D. Detailed Design of Prioritized Scheduling

Prioritized scheduling of data packets at one node is simple. The node simply forwards packets in the ascending order of the tags. Prioritized scheduling between neighboring nodes is more complicated. In Figure 4 (a), both  $u$  and  $y$  have a packet with tag one for transmission. They compete fairly for media access via CSMA/CA. In (b), however,  $u$  has a packet with tag one, but  $y$  only has a packet with tag two. In order for the packet with a smaller tag to be transmitted first,  $u$  should be given the priority to access media, while  $y$  has to wait. We generalize this observation and require that the transmission of a packet from  $y$  to  $x$  be withheld when one of the following two conditions is satisfied.

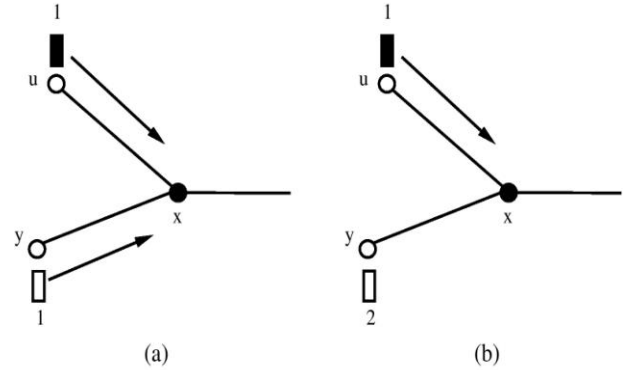


Figure 4. Prioritized scheduling between competing upstream neighbors.

Condition 1. A node  $u$  in  $x$ 's neighborhood has a packet with a smaller tag to be transmitted. The transmission by  $u$  will interfere with the transmission from  $y$  to  $x$ . If the former has a smaller tag, the latter should be withheld.

Condition 2. An upstream neighbor of  $y$  has a packet with a smaller tag to be transmitted to  $y$ . Node  $y$  cannot transmit and receive at the same time. If the packet to be received by  $y$  has a smaller tag, the transmission by  $y$  should be withheld.

The challenge is how to gather the necessary information in order for  $y$  to test the above conditions.

For any node  $x$ , we define the following variables.

- $Lout\ x$ : the tag of the next packet to be transmitted by  $x$ . If  $x$  does not have any more packet,  $Lout\ x = 2l - 1$ .
- $Lin\ x$ : the smallest tag of the packets to be transmitted by the upstream neighbors of  $x$ , i.e.,  $Lin\ x = \min\{Lout\ u \mid u \in U_x\}$ , where  $U_x$  is the set of upstream neighbors of  $x$ , from which  $x$  receives data packets. We call the nodes to which  $x$  forward data packets as downstream neighbors.
- $Ln\ x$ : the smallest tag of the packets to be transmitted by the neighbors of  $x$ , i.e.,  $Ln\ x = \min\{Lout\ u \mid u \in N_x\}$ , where  $N_x$  is the set of neighbors of  $x$ .

Let  $Ly \rightarrow x$  be the tag of the packet to be transmitted from  $y$  to  $x$ . The two conditions can be rewritten as follows.

Condition 1.  $Ly \rightarrow x > Ln x$  or  $Ly \rightarrow x > Lout x$ .

Condition 2.  $Ly \rightarrow x > Lin y$ .

Recall that a RTS/CTS/DATA/ACK exchange is performed when a data packet is transmitted for one hop. We design the following protocol.

- When a node  $x$  transmits a DATA or an ACK, it piggybacks the tag of its next packet to be transmitted,  $Lout x$ . The value of  $Lout x$  may change only when  $x$  sends or receives a packet. When that happens, the new value of  $Lout x$  will be advertised via DATA/ACK. Node  $x$  learns  $Lout u, \forall u \in Nx$ , by overhearing the packets transmitted in the neighborhood. Based on the overheard values, it updates  $Lin x$  and  $Ln x$ . Similarly  $y$  knows  $Lin y$  by overhearing DATA/ACK transmissions in its neighborhood. Node  $y$  also overhears  $Lout x$  because  $x$  is a downstream neighbor of  $y$ .
- When  $x$  transmits a RTS or a CTS, it piggybacks the value of  $Ln x$ . That is, whenever  $x$  attempts to transmit or receive a data packet, it updates its neighbors with the current value of  $Ln x$ . An upstream neighbor  $y$  learns  $Ln x$  by overhearing RTS/CTS from  $x$ .

The above protocol allows  $y$  to keep track of the values of  $Lin y$  and  $Ln x$  so that the two conditions can be tested. If one of the conditions is satisfied,  $y$  withholds from transmitting to  $x$  until it receives a new data packet with a smaller tag or overhears new information from RTS/CTS/DATA/ACK (sent by  $x$ ) such that the conditions are no longer satisfied. When both conditions fail,  $y$  attempts to transmit a packet to  $x$  based on the protocol of CSMA/CA.

The above description is presented for a single pair of upstream/downstream neighbors,  $y$  and  $x$ . When  $y$  has multiple downstream neighbors,  $y$  will attempt to transmit a packet as long as both conditions fail for one of its downstream neighbors. Consider the following example where  $y$  has two downstream neighbors,  $x$  and  $x'$ . If  $y$  finds that one of the two conditions is satisfied for  $x$  but both fail for  $x'$ , then  $y$  must withhold from transmitting to  $x$ , but it can transmit a packet to  $x'$  immediately.

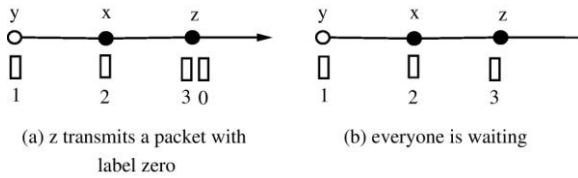


Figure 5. Deadlock due to  $y$ 's stale knowledge of  $Ln x$ .

Next, we describe a “deadlock” problem and design a heuristic solution. The value of  $Ln x$  may change when any node in  $Nx$  sends or receives a packet. Although  $x$  makes its best effort to disseminate the value of  $Ln x$  whenever it has the chance, it's not guaranteed that  $y$  will already have the correct value of  $Ln x$ . The stale knowledge of  $Ln x$  can cause a “deadlock” situation that is illustrated by the following example: In Figure 5 (a),  $Ln x = 0$  and we assume  $y$  knows

the correct value of  $Ln x$  at this moment. Node  $y$  withholds the transmission of its next packet to  $x$  because Condition 1 is satisfied. Node  $x$  also withholds because its Condition 1 is also satisfied. Node  $z$  will transmit the packet with tag zero to its downstream neighbor. Overhearing the DATA packet that piggybacks  $Lout z = 3$ ,  $x$  updates  $Ln x = 1$  in Figure 5 (b). However,  $x$  cannot advertise this new information. It has to withhold the transmission of its next packet because its upstream neighbor  $y$  has a packet with a smaller tag, which satisfies Condition 2. Node  $z$  will also withhold for the same reason. With the stale information  $Ln x = 0$ ,  $y$  will continue withholding its transmission. Everyone is waiting, which is a deadlock.

We design a heuristic approach to resolve the above deadlock. The basic idea is to let the information about  $Ln x$  decay. If  $y$  does not hear an update of  $Ln x$  from  $x$  for a period of  $(|Ny| + 1)\Delta$ , it artificially increases  $Ln x$  by one, where  $\Delta$  is the time for transmitting a data packet.  $(|Nx| + 1)\Delta$  is the time period for everyone in  $x$ 's neighborhood to transmit a packet, ignoring the media contention time. Node  $y$  does not know  $|Nx|$ ; our heuristic approach uses  $|Ny|$  to substitute  $|Nx|$ .

Eventually  $Ln x$  will be increased large enough such that Condition 1 is no longer falsely satisfied due to the stale value of  $Ln x$ , and consequently the deadlock is broken. We have performed extensive simulations, which confirm that the above heuristic approach never allows persistent deadlock and its performance is comparable to the idealized setup where the correct knowledge about  $Ln x$  is always assumed.

#### IV. SIMULATION

We have performed extensive simulations to evaluate the proposed tag-based fairness scheme (TFS). Besides TFS, the following two schemes are implemented.

- No Congestion/Fairness Control (NoCC): Neither do the data sources adjust their packet generation rates, nor the intermediate nodes adapt their forwarding rates.

TABLE I. EFFECTIVE NETWORK THROUGHPUTS OF THE THREE SCHEMES

	low mobility (Fig. 6)	high mobility (Fig. 7)
NoCC	8128.0	8847.9
STCP	10491.0	10158.3
TFS	12320.3	10315.1

- Simplified TCP Congestion Control (STCP): A simplified congestion control mechanism of TCP Reno at data sources adjusts their flow rates based on the network congestion perceived.

##### A. Simulation Environment

The simulation parameters are described as follows. 500 wireless nodes are randomly placed in a  $1000 \times 1000$  area. Four gateways are evenly spaced along one edge of the deployment area to connect the wireless network with the external network (such as the Internet). The transmission range of the nodes is 100. The transmission rate is 11 Mbps.



100 data sources are randomly selected from the 500 nodes. A source generates up to 25 data packets per second. However, the actual delivery rate will be lower if there is congestion downstream. Each data packet is 1024 bytes long. The packet queue at each node can hold 30 data packets. The tag size is 5 bits.

Greedy perimeter stateless routing (GPSR) [9] is adopted to forward packets towards the gateways. In the greedy forwarding mode of GPSR, for a node  $x$ , among all its neighbors that are closer to the gateways than  $x$ ,  $x$  selects the two closest ones as downstream neighbors. When a node forwards a packet, for NoCC or STCP, it pseudo-randomly picks a downstream neighbor; for TFS, it randomly picks a downstream neighbor that does not satisfy the two conditions for withholding in prioritized scheduling.

All data sources are stationary. The motion of non-source nodes follows the random waypoint model [21]. At the beginning of a move period, a non-source node randomly chooses a destination in the simulated area region and moves to that destination at a velocity randomly chosen from a range (velocity range). Upon arriving at the chosen destination, the node pauses for a period whose length is randomly chosen from another range (pause time range). Then the node repeats move periods and pause periods alternately. In our simulations, the degree of network mobility is controlled by the width of the pause time range. We set different pause time ranges to achieve two levels of network mobility: low and high. In a network with low (or high) mobility, at an arbitrary time instant, the probability of being in a move period for a non-source node is 0.25 (or 0.5). The velocity range is between 3 and 20. The network is initialized with the method described in [22] such that simulations can start in the stationary regime of the random waypoint model directly.

### B. Performance Comparison

The first performance metric is called delivered packet rate, which is the average number of packets that are successfully delivered from a data source to the Internet gateway(s) per second. The second performance metric is effective network throughput, which is defined as the sum of all delivered packets' hop counts in the network per second. 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.

Fig. 6 shows the delivered packet rates of 100 data sources of the three schemes in a network with low mobility. The dashed line in each plot is the average rate of all flows. For NoCC, the rates are widely distributed from zero to 25 packets per second. STCP shows somewhat better fairness than NoCC as very few data sources have delivered packet rates lower than 5 packets per second. However, the rate distribution of STCP is still very wide. Figure 6 (c) shows that TFS considerably improves fairness over the other two schemes. Although NoCC has approximately the same average delivered packet rate as STCP and TFS, most nodes with high rates are close to the gateways. Fig. 7 shows that

TFS can also considerably improve fairness over the other schemes in networks with high mobility. The effective network throughputs of the schemes are listed in Table I, which demonstrates that TFS is comparable to STCP and performs better than NoCC.

## V. CONCLUSION

Fairness in bandwidth distribution is an important issue in wireless networks. This paper proposes a tag-based fairness scheme (TFS). Without any assumption of routing structure and without depositing any flow-level state in the network, we tag each packet to assist the intermediate nodes on the routing path to make scheduling decision. Prioritized scheduling based on packet tags is designed to schedule packets for partially ordered transmission. Each node makes localized scheduling decision, and the net outcome of the scheduling decisions made by all nodes will collectively ensure the fairness among competing flows.

## ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under grants 61170277 and 61472256, the Innovation Program of Shanghai Municipal Education Commission (Program Number 12zz137), and the first-class discipline construction project of Shanghai under grant S1201YLXK.

## REFERENCES

- [1] SHI Huaizhou, R Venkatesha Prasad, Ertan Onur, and IGMM Niemegeers, "Fairness in wireless networks: Issues, measures and challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 5–24, 2014.
- [2] S. Chen and Z. Zhang, "Localized Algorithm for Aggregate Fairness in Wireless Sensor Networks," *Proc. of ACM Mobicom'06*, September 2006.
- [3] L. Tassiulas and S. Sarkar, "Maxmin Fair Scheduling in Wireless Networks," *Proc. of IEEE INFOCOM'02*, June 2002.
- [4] 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.
- [5] Mario K'oppen and Masato Tsuru, "Zonal fairness in wireless infrastructure," in *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 2015, pp. 1–5.
- [6] A. Sridharan and B. Krishnamachari, "Max-Min Fair Collision-Free Scheduling for Wireless Sensor Networks," *Proc. of Workshop on Multihop Wireless Networks (MWN'04)*, April 2004.
- [7] A. Woo and D. Culler, "A Transmission Control Scheme for Media Access in Sensor Networks," *Proc. of ACM MobiCom'01*, July 2001.
- [8] C. T. Ee and R. Bajcsy, "Congestion Control and Fairness for Many-to-One Routing in Sensor Networks," *Proc. of ACM SenSys'04*, November 2004.
- [9] B. Karp and H. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proc. of ACM MobiCom'00*, August 2000.
- [10] Q. Fang, J. Gao, and L. J. Guibas, "Locating and Bypassing Routing Holes in Sensor Networks," *Proc. of IEEE INFOCOM'04*, March 2004.
- [11] C. E. Perkins and E. M. Royer, "Ad Hoc On-demand Distance Vector Routing," *Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, February 1999.
- [12] M. Marina and S. Das, "On-demand Multipath Distance Vector Routing in Ad Hoc Networks," *Proc. of IEEE International Conference on Network Protocols (ICNP)*, August 2001.

- [13] George Athanasiou, Pradeep Chathuranga Weeraddana, Carlo Fischione, and Leandros Tassioulas, "Optimizing client association for load balancing and fairness in millimeter-wave wireless networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 23, no. 3, pp. 836–850, 2015.
- [14] Mohammad Shurman, Bilal Al-Shua'b, Mohammad Alsaadeen, Mamoun F Al-Mistarihi, and Khalid A Darabkh, "N-beb: New backoff algorithm for iee 802.11 mac protocol," in *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*. IEEE, 2014, pp. 540–544.
- [15] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan, "Achieving MAC Layer Fairness in Wireless Packet Networks," *Proc. of ACM MobiCom '00*, August 2000.
- [16] H. Luo, S. Lu, and V. Bharghavan, "A New Model for Packet Scheduling in Multihop Wireless Networks," *Proc. of ACM MobiCom '00*, August 2000.
- [17] B. Bensaou, Y. Wang, and C. C. Ko, "Fair medium access in 802.11 based wireless ad-hoc networks," *Proc. of ACM MobiHoc '00, Boston, Massachusetts*, July 2000.
- [18] X. L. Huang and B. Bensaou, "On Max-Min Fairness and Scheduling in Wireless Ad-hoc Networks: Analytical Framework and Implementation," *Proc. of ACM MobiHoc '01, Long Beach, California*, October 2001.
- [19] H. Luo, J. Cheng, and S. Lu, "Self-Coordinating Localized Fair Queueing in Wireless Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 1, 2004.
- [20] V. Gambiroza, B. Sadeghi, and E. W. Knightly, "End-to-End Performance and Fairness in Multihop Wireless Backhaul Networks," *Proc. of ACM Mobicom '04, Philadelphia, PA, USA*, September–October 2004.
- [21] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," *Proc. of ACM Mobicom '98*, August 1998.
- [22] J.-Y. Le Boudec, "Understanding the Simulation of Mobility Models with Palm Calculus," *EPFL Technical Report IC/2005/033*, 2005.

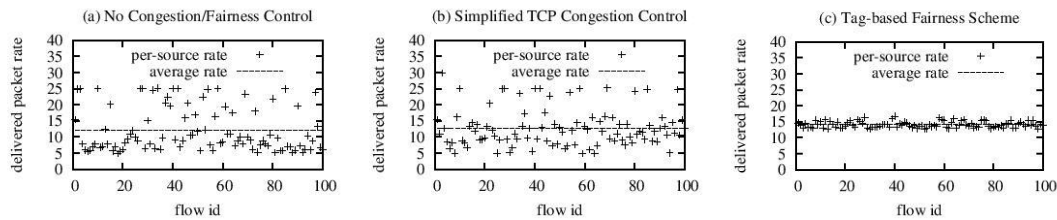


Figure 6. Low mobility network.

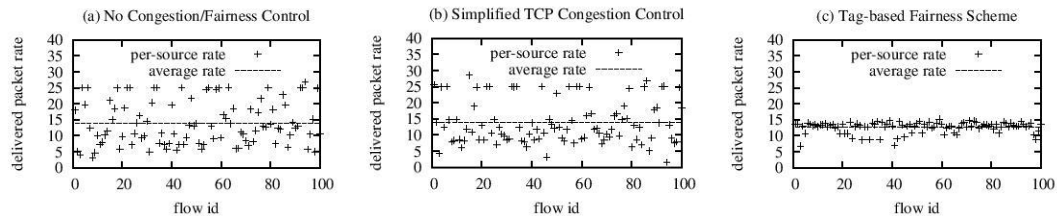


Figure 7. High mobility network.