Queue-based priority MAC for WLANs

Technical Report, September 2015

Juan J. Galvez
Coordinated Science Lab
University of Illinois at Urbana-Champaign
Email: jjgalvez@illinois.edu

Nitin Vaidya
Department of ECE and Coordinated Science Lab
University of Illinois at Urbana-Champaign
Email: nhv@illinois.edu

Abstract—**Recent studies have shown that it is possible to approximate throughput optimal scheduling using distributed Carrier Sense Multiple Access (CSMA). These studies build on the idea of maximal-weight scheduling, where the set of links scheduled in each slot are the non-conflicting links with maximum sum of their queue lengths. However, conditions imposed by real environments including imperfect carrier sensing, collisions and the distributed nature of the protocol make it challenging to develop throughput optimal schemes for CSMA. In this paper we propose an implementation of queue-based scheduling that addresses these challenges and is suitable for implementation in 802.11 networks with minimal modifications. We evaluate the proposed protocol and have shown significant performance improvements of up to 98% in TCP flows in a typical WLAN setting, without negatively impacting fairness.**

I. INTRODUCTION AND RELATED WORK

Wireless Local Area Networks (WLANs) are widely deployed. Most of them are based on the popular IEEE 802.11 standard, which uses the Distributed Coordination Function (DCF) for medium access. DCF itself is based on Carrier Sense Multiple Access (CSMA) with Collision Avoidance (CA). DCF has the advantage of being a fully distributed random-access mechanism, but one of its main drawbacks is that it does not achieve optimal performance.

Achieving throughput-optimality in wireless networks has been studied in the past using comprehensive theoretical frameworks leveraging cross-layer optimization. Examples are the work by Tassiulas and Ephremedes [1] and more recently [2]. These studies showed that maximal-weight scheduling (MWS) is throughput-optimal. In MWS, the set of links scheduled in each slot are the non-conflicting links with maximum sum of their queue lengths.

Recent works have built on this framework in an attempt to design a throughput-optimal CSMA protocol. One major result, by Jiang and Walrand [3], has shown that it is possible to achieve throughput optimality with a distributed CSMA protocol under theoretical conditions. In their work, they present a protocol where nodes use only local information, i.e. no exchange of control messages is needed. Although the protocol is distributed, this work makes several simplifying assumptions which don't occur in practice, like absence of collisions and perfect carrier sensing (i.e. nodes which might interfere can instantaneously sense each other and nodes which don't interfere will not sense each other).

In real environments, collisions are a major limiting factor for performance in CSMA. Imperfect carrier sensing or the fact that nodes can choose to transmit in the same slot are common occurrences in CSMA leading to collisions. The challenge is approximating maximal-weight scheduling in these conditions. Recent proposals attempt to to do so, using ideas extracted from the above mentioned theoretical works.

DiffQ [4] implements a differential backlog MAC scheduling suitable for off-the-shelf hardware. Packets are prioritized at the MAC level in order to give higher probability of transmission to packets from queues with large queue differential. When used with TCP, DiffQ is shown to improve fairness in multihop scenarios but does not improve throughput. Optimal DCF (O-DCF) [5] is a protocol which dynamically adapts CSMA parameters to approximate optimal scheduling. In O-DCF, each node adapts its back-off time and transmission length based on the backlog differential of its queues. A-DCF [6] introduces improvements to O-DCF to perform better with TCP, but is otherwise based on the same principles.

These works present some important limitations: DiffQ does not prevent multiple packets from being transmitted at the same time with high priority from different senders, which can notably increase collisions, as demonstrated in [5]. Also, while they improve performance in asymmetric contention scenarios, they do not in the common case where all nodes are in the same collision domain, typical of WLANs with one access point (AP). Finally, nodes make decisions based only on local information, i.e. nodes don't exchange information and don't know the queue information of other nodes, which can lead to suboptimal decisions.

In this work we propose a practical distributed protocol called QBP that performs queue-based scheduling by exploiting different MAC priority classes[1]. In this respect it is similar to the practical CSMA protocols described above. The main difference is that priority of a transmission is selected not only based on the queue length of the node, but on the known queue lengths of other nodes in the network, thus enabling better distributed scheduling decisions. QBP is built on top of the EDCA (Enhanced Distributed Channel Access) of 802.11e.

The rest of the paper is organized as follows. In Section II we explain the challenge of throughput-optimal scheduling in a realistic environment. In section III we describe the queue-based scheduling protocol proposed in this paper. Section IV presents simulation results in *ns-3*. Finally, section V concludes the paper.

---

[1] Priority classes were introduced to DCF with 802.11e to support QoS.

II. PRACTICAL THROUGHPUT-OPTIMAL SCHEDULING PROBLEM

In previous works [4], [5], [6], nodes with similar queue length will transmit with the same priority (or similar backoff time), which is specially problematic when that priority is high. Higher priorities have smaller contention windows and the probability of collision is higher. Collisions reduce the efficiency and performance of the MAC protocol. Therefore, the benefits of "throughput-optimal" scheduling in this case will be limited by increased collisions.

A solution to this problem is to give nodes knowledge of the current queue state of any contending nodes at the time of transmission, thus allowing nodes to adapt their priority based on their queue length and that of their neighbors. This can be achieved by allowing nodes to exchange queue information. However, in a real environment there are practical challenges that must be overcome:

- The information cannot be expected to be completely accurate at the time of transmission, i.e. nodes cannot know the *real* queue length of other nodes at the time of transmission, only their last advertised queue lengths.
- A node cannot be guaranteed to know the queue state of every node in its collision domain for various reasons, e.g. a neighbor has not yet advertised its queue information, or the message was lost due to collisions, interference or low Signal-to-Noise ratio (SNR).
- To achieve benefits in performance, a solution must add minimal overhead and avoid frequent exchange of control messages.

In summary, a solution must minimize overhead and assume a degree of uncertainty in the knowledge of queue length information. As we will see, the QBP protocol effectively addresses these challenges in a simple manner.

III. QUEUE-BASED PRIORITY (QBP) MAC PROTOCOL

In this section we develop the QBP protocol.

A. QBP overview

In QBP, every node in the WLAN (stations and APs) advertises its current queue length by piggybacking this information when it transmits a packet, thereby avoiding the need of explicit control messages. Nodes overhear packets transmitted by other nodes, which allows them to have a view of the state of queues in the network. Given this information, each node will refrain from transmitting if its queue length is less than the maximum advertised queue length α that the node is aware of. Note that at any particular instant α may be inaccurate and there may be several nodes with queue length greater than α, i.e. they have not had the chance to transmit and advertise their (possibly updated) queue length. These nodes should therefore be the ones to compete for the channel in the next transmission epoch. Moreover, the one with the largest queue should win. To achieve this, QBP uses the different MAC priority levels of 802.11e.

TABLE I. 802.11E DEFAULT PARAMETERS FOR EACH CLASS

	AIFS	CWMin	CWMax
BK	7	15	1023
BE	3	15	1023
VI	2	7	15
VO	2	3	7

TABLE II. SYMBOLS USED IN DESCRIPTION OF PROTOCOL

q_n	Queue length of node n.
Q	Queue length table in each node: $Q[n]$ stores q_n.
α	Maximum advertised queue length (known by a node).
$holPrio$	Priority chosen for packet at the head of transmit queue.
Δ	Multiplier to determine what priority to use when a node's queue length is $\geq \alpha$. Range is $1 \leq \Delta \leq \Delta^{max}$. A lower multiplier increases the chances of using a higher priority.
Δ^{max}	Maximum possible value of Δ.
σ	Δ is modified by σ every time a transmission is succesful.
δ	Δ is increased by δ every time a transmission fails.

In EDCA, higher priority traffic has more chances of being sent than lower priority traffic. This is achieved by using different values for parameters like contention window (CWMin, CWMax) and interframe spacing (AIFSN) in each class. EDCA supports four priority levels, from lowest to highest: Background -BK-, Best Effort -BE-, Video -VI- and Voice -VO-. Table I shows the default EDCA parameters of each class.

Based on this, stations with queue length greater or equal than the maximum advertised queue length α will transmit using a higher MAC priority. The chosen priority will depend on the distance to α, i.e. nodes with larger queue sizes will transmit with higher priority.

Table II lists the symbols used to describe the protocol. In the following subsections we explain the protocol in detail.

B. QBP protocol

QBP manages its own queue of packets and only the head-of-line packet is sent to the MAC for transmission. After that packet is transmitted, the next head-of-line packet is sent to the MAC. This facilitates inserting the latest queue size information at the moment a packet is transmitted and managing the priority of packets.

The procedure used by a node to schedule the head-of-line packet is detailed in Algorithm 1. When node n wants to transmit the packet at the head of the queue, it includes its current queue length (q_n) in the packet and chooses a MAC priority for the packet (lines 1-4).

The selection of priority is shown in lines 6-14. QBP is designed to use 3 MAC priorities (we have not observed

Algorithm 1 Node n needs to send packet p

```
1:  Include $q_n$ in $p$
2:  p.prio ← currentPrio()
3:  $holPrio$ ← p.prio
4:  Send $p$ to MAC layer for transmission
5:
6:  function currentPrio() do
7:      $\alpha \leftarrow \max\{q_i\} \quad \forall i \in Q$ // $\alpha$ is maximum network queue length known by node $n$
8:      if $q_n < \alpha$ then
9:          return BK
10:     else
11:         if $q_n > \Delta\alpha$ then
12:             return VO
13:         else
14:             return VI
```

benefits of using more in our experiments), but note that any number of priority levels could be used by modifying the MAC layer and QBP can be adapted to use them. It is also possible to tweak the parameters of each level.

Nodes whose current queue length is smaller than α should refrain from transmission. However, they should also be given a chance to transmit in case any nodes with queue length greater or equal than α fail to do so. For this reason, they are assigned the lowest priority BK (lines 8-9). In the experiments we have increased the value of AIFS of the BK level so that these transmissions normally occur after all other higher priority transmissions.

For nodes with queue length greater than α, a high priority is chosen based on the distance to α (lines 11-14).

Algorithm 2 Node n receives/overhears a packet p from A to B

```
1:  if $p$ is DATA then
2:      $Q[A]$ ← p.q // p.q is queue size information in p
3:  else if p is ACK and $B \neq n$ then
4:      $Q[B]$ ← p.q
5:  if $n$ has pending transmission and currentPrio() $>$ $holPrio$ then
6:      $p_{hol}$ ← pending head-of-line packet of $n$
7:      cancel pending MAC transmission of $p_{hol}$
8:      resend $p_{hol}$ to MAC (with new priority) // Following Algorithm 1
```

Algorithm 2 shows the procedure followed when a node overhears a packet. When node n receives/overhears a data packet from A, it stores q_A in its table (i.e. $Q[A] \leftarrow q_A$). Note that if the packet is an ACK the queue information contained refers to node B (this is explained in more detail in the next subsection). Nodes keep track of the queue lengths of all nodes they overhear. An expiration timer is set to remove old entries.

If a packet has been queued for transmission at a low priority (e.g. BK) but the priority of the node suddenly increases (e.g. due to an increase in its queue length), that packet is re-queued at higher priority (lines 5-8). Otherwise the node could wait a long time to transmit even though its queue size is larger than other nodes.

from=A	to=B	q_{from}

(a) DATA packet

from=B	to=A	q_{to}

(b) ACK packet

Fig. 1. This figure shows a DATA packet from A to B and corresponding ACK packet from B to A. The third field in each packet contains a queue length. In the case of the DATA packet, it is the queue length of the node sending the packet. The ACK rebroadcasts the queue length of the node that sent the DATA packet, therefore it contains the queue length of the node to which the ACK is directed at.

Advantages of QBP over CSMA include reduced channel idle time due to the use of higher priorities with smaller back-off intervals, and reduced collisions due to approximating Maximal Weight Scheduling (MWS). Moreover, the risk of increased collisions when using high priorities is greatly reduced thanks to the exchange of queue information between nodes, and the adaptive control of priorities which will be discussed later. As we will see in section IV, this translates into increased efficiency and throughput.

C. Queue length in ACKs

In an infrastructure WLAN, there is only a requirement that stations be able to communicate with the AP. In other words, there is no guarantee that any pair of stations can decode each other's messages[2]. This means that, in general, when a node advertises its queue length in a data packet, not all nodes will receive it, thus limiting the performance of the QBP protocol. This is specially aggravated in multirate networks, where stations can transmit using different bit rates based on their signal strength to the AP. In multirate networks, the capability to overhear packets varies drastically from node to node and depends on the rate used for transmission. For example, if a station with high signal strength transmits at a high bit rate, this transmission likely cannot be decoded by stations far away from the transmitter or AP. It is not even guaranteed that every station can overhear an AP's transmission (when a transmission by the AP is not specifically directed to a station it may use a higher bit rate than what the station can decode).

To work correctly, QBP requires stations to overhear queue length advertisements of other nodes. A simple solution is to rebroadcast this information in ACKs. Because ACKs are much smaller than regular DATA packets and are sent at lower base rates, they are much easier to overhear by other stations in the network.

The mechanism works as follows (shown in Fig. 1). When a node sends a DATA packet it includes its queue length in the packet. The queue length field therefore refers to the station in the *from* field. When a station sends an ACK it includes the queue length of the station that sent the DATA packet: in this case the queue length field refers to the station in the *to* field.

D. Adaptive priorities

Allowing nodes to transmit at high MAC priorities helps to approximate MWS, but there is a risk of increased collisions. Higher priorities have smaller contention windows, meaning that fewer nodes can use them simultaneously without incurring high collisions. To avoid this problem, the priorities used

---

[2] Note that being able to sense a transmission is not the same as being able to decode it.

by nodes to transmit can be controlled adaptively based on the state of collisions in the network. That is, if there are no collisions or very low number of collisions, transmissions from nodes with $q_n \geq \alpha$ can be done at maximum priority. If there are too many collisions, we can separate traffic from nodes with $q_n \geq \alpha$ into more priorities, i.e. nodes whose queue length is closer to α will have lower priority, while nodes whose queue length is farther from α will have higher priority.

When sending a packet to the MAC layer (Algorithm 1), the value of Δ determines the chance of transmitting at the highest priority VO: higher Δ makes it more difficult, while a smaller Δ makes it easier. By adaptively changing Δ it is possible to adapt the transmission priority to collisions in the network.

Algorithm 3 Transmission successful

```
$\sigma \leftarrow$ CalculateSigma(retries)
$\Delta \leftarrow \Delta + \sigma$ // $\sigma$ can be negative
Truncate $\Delta$ s.t. $1 \leq \Delta \leq \Delta^{max}$
```

Algorithm 4 Transmission failed

```
$\Delta \leftarrow \Delta + \delta$ // $\delta > 0$
Truncate $\Delta$ s.t. $1 \leq \Delta \leq \Delta^{max}$
```

This is done in the following manner: when a transmission is successful (Algorithm 3), Δ is modified based on the number of retransmissions that were necessary to send the packet. The change in Δ is given by σ and calculated as a function of the number of retries (CalculateSigma). In section IV we will specify the function used in our experiments. The idea is that if a transmission is successful with a low number of retries, Δ will be decreased. If it is successful but required many attempts, Δ will be increased. Similarly, if a transmission fails (Algorithm 4) Δ will be increased. Note that each node has its own value of Δ and that it is modified locally based on the performance observed by the node. The value of Δ is not exchanged between nodes.

IV. PERFORMANCE EVALUATION

In this section we evaluate the QBP protocol in 802.11 WLANs through simulation in *ns-3* [7].

A. *Simulation environment*

The experiments are conducted in WLANs with one AP. We have generated ten random topologies. In each topology, there are 20 stations located randomly around the AP, within a radius of 200 m of their AP[3]. For the purpose of establishing Internet traffic flows there is a server on the Internet connected to the AP by a point-to-point 1 Gbps link (in this way the performance observed is only limited by the WLAN). Traffic can be both uplink and downlink, and transmissions occur in the same channel. The parameters used to configure wireless interfaces and propagation in *ns-3* are shown in Table III.

The data rate of each link depends on its SNR. That is, not all links will transmit at the same rate and this will depend on factors like the distance from the station to the AP.

TABLE III. SIMULATION PHYSICAL PARAMETERS

Wifi standard	802.11a
Frequency band	5 GHz
Set of (link) data rates	$\{6, 9, 12, 18, 24, 36, 48, 54\}$ Mbps
Path loss model	Log-distance, exponent = 2.7
Tx power	18.0 dBm
RTS/CTS	Disabled
Energy detection threshold (EDth) (W)	RxPower(200m)

TABLE IV. QBP PARAMETERS USED IN SIMULATIONS

Δ^{max}	0.15
δ	0.1
σ	$+0.05$ if tries ≥ 4 0 if tries $= 3$ -0.05 if tries $= 2$ -0.1 if tries $= 1$

B. *Traffic patterns*

Traffic in the experiments consists of random TCP flows. We evaluate three categories of traffic patterns: UL, DL and MIXED. In the DL category every flow is download traffic; an Internet server is the source of all flows and the destination of a flow is chosen randomly among the stations. Note that because flows are TCP there is still a small amount of control traffic (e.g. TCP ACKs) in the uplink direction. In the UL category every flow is uplink: the destination is the Internet server and the source is chosen randomly among the stations. In category MIXED flows can be uplink and downlink and are chosen randomly. Each pattern is evaluated with 5, 10, 15 and 20 flows, and ten different scenarios are generated for each individual configuration (e.g. ten random scenarios with 5 flows for each topology). Each scenario is run ten times.

Flows consist of TCP flows where the size is 2 MB and each flow starts in a random instant in 30 ± 2.0 seconds. A flow ends when it has transmitted all of its data. Flows start at slightly different times but will overlap for most of their duration, and the throughput of each flow (and hence its duration) will depend on the dynamics of the TCP and MAC protocols. Simulations last until all flows finish.

Table IV shows the values used to configure the O-CS protocol. These are not necessarily optimal, but rather values which have given good experimental results.

C. *Performance metrics*

We use the following metrics to study protocol performance: (i) *Flow throughput* - the number of bytes successfully delivered divided by the duration of the flow; (ii) Fairness Index - rates the fairness of the throughput achieved by flows in one scenario using Jain's fairness index [8]. The index varies from 0 to 1, and when equal to 1 all flows have the same throughput; (iii) *Aggregate flow utility* - measure of the proportional fair throughput of the network, given by:

$$\sum_{f \in \mathcal{F}} \ln\left(\text{throughput}(f)\right) \tag{1}$$

---

[3] There are a total of 40 links in each network, counting uplinks and downlinks.

where $\mathcal{F}$ is the set of flows in the network and the throughput of a flow is measured in bytes per second. Proportional fairness [9] is a useful metric because it allows to compare in one metric the overall network performance in terms of minimum flow rate, average rate and aggregate flow rate. It takes into account the total flow throughput, fairness and starvation.

Regarding flow throughput, in each scenario we measure the minimum, median, mean and maximum throughput of flows. Both the aggregate flow utility and minimum throughput are important metrics to detect unfairness and flow starvation.

Points shown in graphs represent the average (mean) result of the scenarios that fall in that class.

D. Performance Analysis

In this section we evaluate the performance of QBP and the CSMA protocol of 802.11 under the traffic patterns described above. First we analyze the performance with a fixed number of active flows (ten) in all traffic categories. The results are shown in Fig. 2.

There are a number of interesting conclusions that we can extract from the results. First, the median and mean flow rates are notably higher when using QBP. The increase in mean flow rate is 60%, 88% and 129% when using MIXED, UL and DL traffic, respectively. An important reason for this increase in average rates is the much higher maximum rate of flows with QBP (Fig. 2 (d)). Note that QBP effectively allows flows with longer queues to transmit in longer bursts, which also prevents packet drops and allows higher TCP throughput. In these tests, the maximum rate is much higher than the average rate. A more representative measure of the typical flow rate is *median* flow throughput, as shown in Fig. 2 (b). With this metric, we still observe a notable increase in performance with QBP: median throughput is 47%, 31% and 86% higher under patterns MIXED, UL and DL. This shows that QBP in fact improves efficiency of medium access.

As we can see, the performance of QBP is much higher when using the DL traffic pattern. Note that in this case there is still some amount of uplink traffic present (TCP ACKs traveling back to the source), but the vast majority of the traffic is transmitted by only one node (the AP). It is clear that using QBP allows the AP to transmit in longer bursts (using the highest MAC priority and being seldom interrupted by other nodes), which allows TCP to achieve a much higher throughput. It is important to note that DL is one of the most common traffic patterns in WLANs and we can therefore expect QBP to be highly useful in this scenario.

Another important observation is that the increase in throughput with QBP does not come at the expense of starving some flows. As we can see, the minimum flow rate with QBP is comparable to CSMA, and much higher in the case of UL scenarios. In the UL scenarios some flows suffer from starvation and unfairness with CSMA: minimum rate and aggregate utility are negatively affected as seen in Fig. 2 (a) and (f). QBP does not suffer from this limitation.

Finally, we can see that the throughput across flows is more similar when using CSMA, i.e. Jain's Fairness Index is closer to 1 with CSMA. This comes at the expense of aggregate throughput. With QBP, even though there is more difference in flow throughput, the aggregate utility is higher, and because the utility measured is proportional fairness this means that QBP effectively increases aggregate throughput without severely affecting fairness (e.g. no flow starvation).

Now we will illustrate the performance under varying number of flows. Fig. 3 shows results under varying number of flows using only the MIXED traffic pattern, while Fig. 4 shows results with the DL pattern. The results show significant improvements in network capacity and flow throughput, without negatively impacting the fairness of flows.

As expected, per-flow throughput drops with increasing number of flows. QBP is the best strategy overall with higher aggregate flow utility in all cases (Fig. 3 (f)). The improvement in median flow throughput is present but tends to diminish with increasing number of flows: it is 49%, 47%, 37% and 34% with 5, 10, 15 and 20 flows, respectively. In DL scenarios (Fig. 4) QBP shows a very large increase in throughput with any number of flows: 87%, 86%, 74% and 98% in median throughput for 5, 10, 15 and 20 flows, respectively.

V. CONCLUSIONS

In this paper we have proposed a scheme that effectively uses queue-based scheduling to improve the performance of CSMA. This scheme is suitable for use in real environments and can be implemented in 802.11 wireless networks with minimal modifications.

In a distributed wireless environment, making optimal scheduling decisions based on queue length is challenging, due to the difficulty of exchanging accurate queue information, the presence of collisions and imperfect carrier sensing. The proposed QBP protocol effectively allows sharing queue length information, making distributed decisions under imprecise information and adapting to network conditions (e.g. collisions).

We have shown results in a typical WLAN setting under a variety of TCP traffic patterns, including download and upload traffic. The results have shown significant improvements in TCP throughput with respect to CSMA, particularly when most of the traffic is download traffic. Moreover, this does not come at the cost of negatively affecting flow fairness.

ACKNOWLEDGMENT

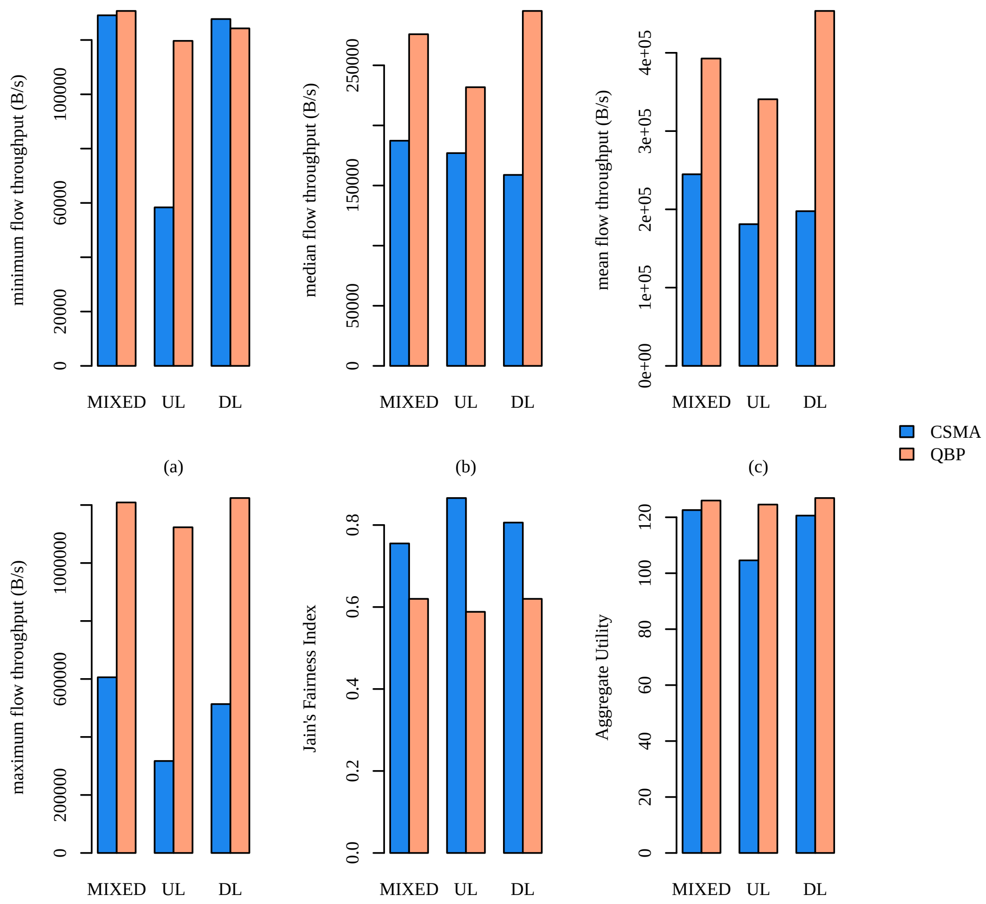
This research is supported in part by National Science Foundation awards CNS-11-17539. Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the funding agencies or the U.S. government.

REFERENCES

[1] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, pp. 1936–1948, 1992.

[2] L. Georgiadis, M. J, and R. Tassiulas, "Resource allocation and cross-layer control in wireless networks," in *Foundations and Trends in Networking*, 2006, pp. 1–149.

[3] L. Jiang and J. Walrand, "A distributed csma algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 960–972, Jun. 2010. [Online]. Available: http://dx.doi.org/10.1109/TNET.2009.2035046



(a) (b) (c)

(d) (e) (f)

Fig. 2. Performance comparison of regular CSMA and QBP with 10 TCP flows under different traffic patterns (MIXED, UL and DL). Shown are the average values of minimum, median, mean and maximum flow rates, Jain's Fairness Index and aggregate utility.

[4] A. Warrier, S. Janakiraman, S. Ha, and I. Rhee, "DiffQ: Practical Differential Backlog Congestion Control for Wireless Networks," in *In Proc. of INFOCOM, Rio de Janeiro*, 2009.

[5] J. Lee, H. Lee, Y. Yi, S. Chong, B. Nardelli, and M. Chiang, "Making 802.11 DCF near-optimal: Design, implementation, and evaluation," in *10th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON 2013, New Orleans, LA, USA, 24-27 June, 2013*, 2013, pp. 86–94. [Online]. Available: http://dx.doi.org/10.1109/SAHCN.2013.6644966

[6] H. Lee, S. Moon, and Y. Yi, "A-DCF: Design and Implementation of Delay and Queue Length Based Wireless MAC," in *IEEE Infocom*, 2015.

[7] "The ns-3 network simulator." [Online]. Available: http://www.nsnam.org/

[8] R. Jain, D. Chiu, and W. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems," Tech. Rep., 1984, DEC Research Report TR-301. [Online]. Available: http://www1.cse.wustl.edu/ jain/papers/ftp/fairness.pdf

[9] F. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, pp. 33–37, 1997.

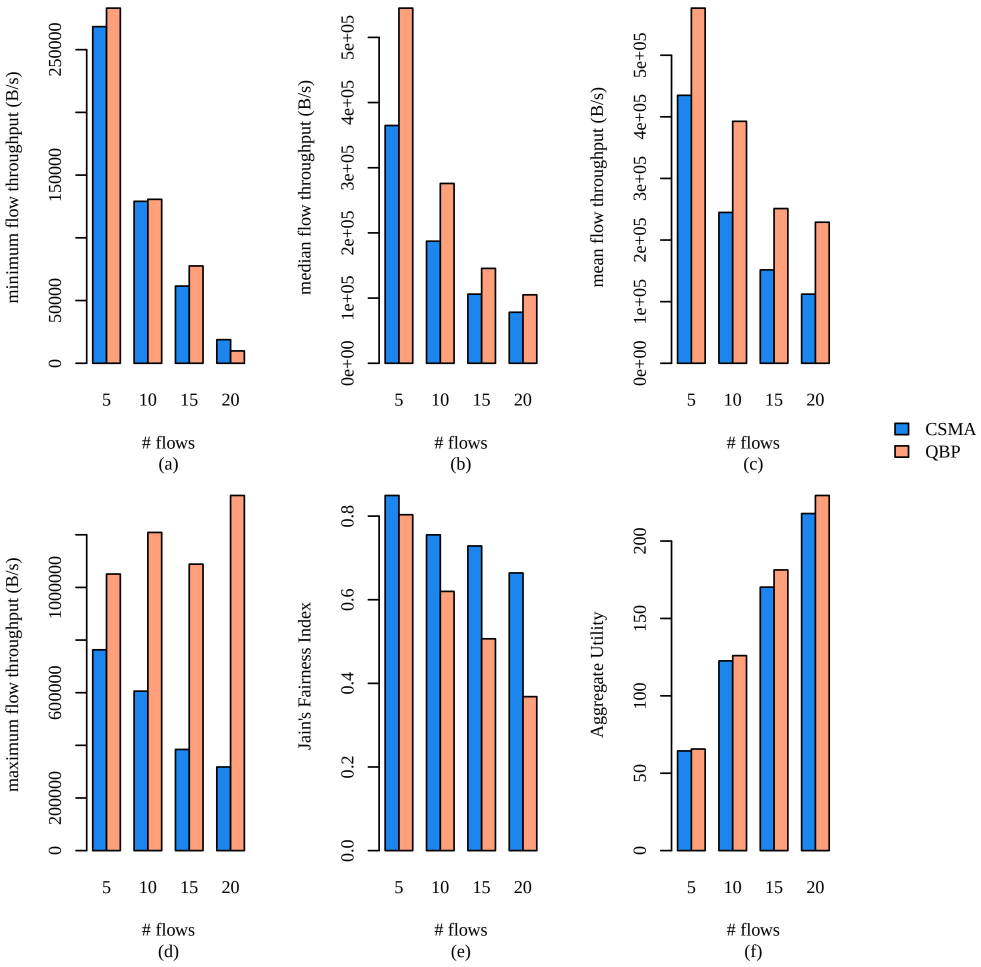


Fig. 3. Performance comparison of CSMA and QBP under the MIXED traffic pattern with varying number of flows.

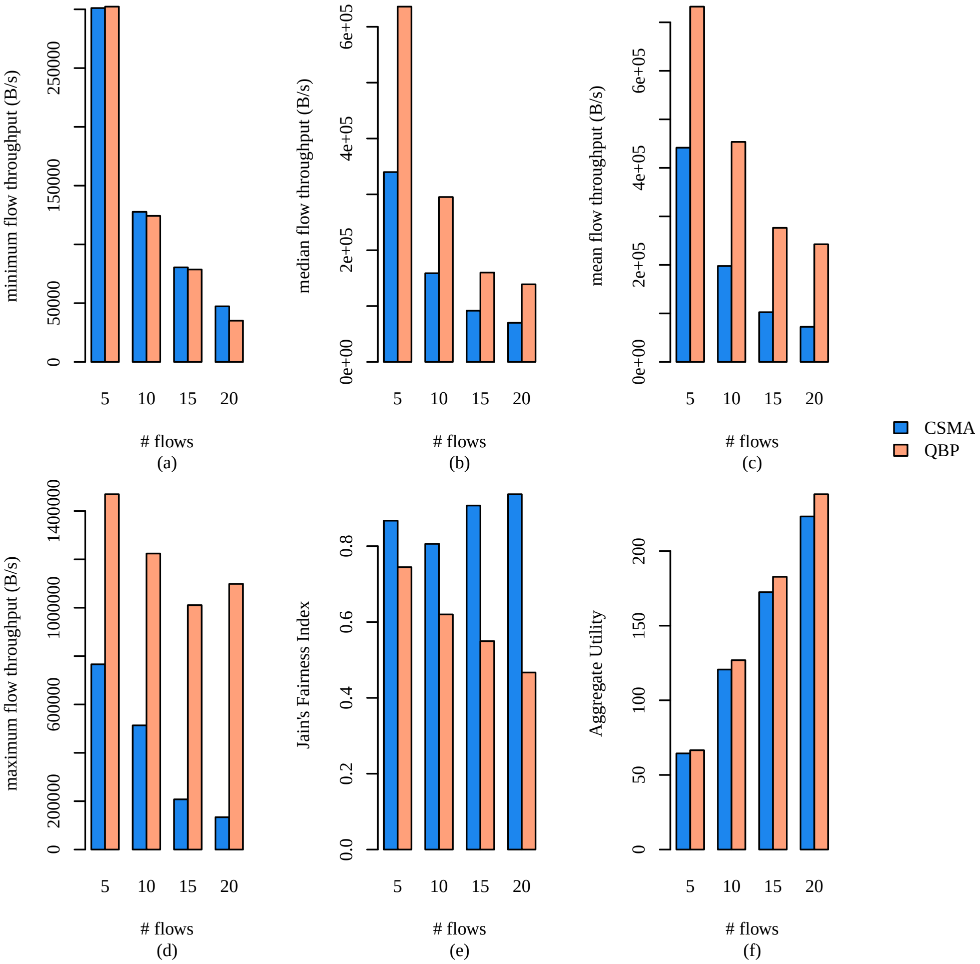


Fig. 4. Performance comparison of CSMA and QBP under the DL traffic pattern with varying number of flows.