Q-CMRA: Queue-Based Channel-Measurement and Rate-Allocation

Vidur Bhargava, *Student Member, IEEE,* Jubin Jose, *Member, IEEE,*
Kannan Srinivasan, *Member, IEEE,* and Sriram Vishwanath, *Senior Member, IEEE*

***Abstract*—In traditional wireless protocols, medium-access-control and physical-layer rate-allocation are performed separately. This paper makes the case for combining the two into a single cross-layer framework. It presents the design, implementation, and evaluation of queue-based channel-measurement and rate-allocation (Q-CMRA) that is based on this single cross-layer framework. Q-CMRA's distributed algorithms utilize both queue-state and channel-state to jointly control medium-access and rate-allocation. Such joint control is essential to improving spatial-reuse and total network throughput. In our experiments, Q-CMRA outperforms traditional CSMA-CA and doubles the total network throughput in some setups.**

***Index Terms*—Cross-layer design, medium-access-control, rate-allocation, ad-hoc networks, mesh networks.**

I. Introduction

WITH an ever increasing number of wireless devices operating over the same medium, a central research challenge is the development of cross-layer protocols that make optimal use of the available spectrum. Two key mechanisms affect the quality of spectrum utilization: (i) medium-access-control (MAC) that determines which node gets to use the wireless channel; and (ii) physical-layer rate-allocation that determines how fast a node can send its bits. Both mechanisms are well researched.

MAC is a well-studied research topic [1], [2], [3], [4]. MAC protocols based on carrier-sense multiple-access with collision-avoidance (CSMA-CA), such as IEEE 802.11's distributed coordination function (DCF), have become ubiquitous. In traditional CSMA-CA, a node backs off from using the wireless channel if it senses interference or noise above a **rigid** carrier-sensing threshold. This simple carrier-sensing strategy causes the node to miss several opportunities to transmit frames in parallel with other wireless nodes [5], [6], [7]. Thus, MAC protocols based on traditional CSMA-CA are limited to orthogonal-access and use the wireless spectrum very inefficiently.

Manuscript received June 7, 2012; no revision; accepted July 26, 2012. The associate editor coordinating the review of this paper and approving it for publication was X. Dong.

V. Bhargava and S. Vishwanath are with the Department of Electrical and Computer Engineering, Wireless Networking and Communications Group, The University of Texas at Austin, Austin, TX 78712 USA (e-mail: vidurbhargava@utexas.edu, sriram@austin.utexas.edu).

J. Jose is with Qualcomm Research, Bridgewater, NJ (e-mail: jjose@qualcomm.com).

K. Srinivasan is with The Ohio State University, Columbus, OH (e-mail: kannan@cse.ohio-state.edu).

Digital Object Identifier 10.1109/TWC.2012.091812.120813

There is also much research on physical-layer rate-allocation [8], [9], [10], [11] for orthogonal-access CSMA-CA MAC protocols. In these rate-allocation schemes, a node assumes that it will have exclusive access to the wireless channel when it gets a chance to transmit a frame. Thus, the node *greedily* uses the highest feasible physical-layer rate given *only* the channel-conditions of the link to its receiver. When all else is equal, a transmitter needs to transmit at a higher power to sustain a higher physical-layer rate (refer to Section III-A). Thus, a node that (successfully) transmits frames at a high physical-layer rate also causes more interference for nearby nodes. Overlooking the impact of this excessive interference on nearby nodes hurts spatial-reuse and total network throughput.

Now, consider a parallel-access MAC that is capable of facilitating multiple interfering transmissions to happen in parallel whenever such parallel transmissions are feasible. Such a parallel-access MAC can tremendously improve spatial reuse and enhance the total network throughput in some topologies. However, a node using a parallel-access MAC cannot simply use the highest feasible physical-layer rate given *only* the channel-conditions of the link to its receiver. This is because the node will need to use a higher transmit-power to sustain the higher physical-layer rate (refer to Section III-A). Thus, it will cause more interference for nearby nodes and prevent other transmissions from happening in parallel. In order to facilitate multiple transmissions to happen in parallel, the node must also take the medium-access state of nearby nodes into account when doing physical-layer rate-allocation.

Several theoretical studies [12], [13], [14], [15] have also shown that combining medium-access and rate-allocation in a cross-layer framework can provide significant throughput gains. The goal of this paper is to jointly control medium-access and rate-allocation in order to increase spatial-reuse and total network throughput. To this end, we design, implement, and evaluate a queue-based channel-measurement and rate-allocation mechanism (**Q-CMRA**). Q-CMRA uses a distributed queue-based rate-allocation algorithm that is **not** limited to orthogonal-access scenarios. The rate-allocation algorithm is designed and theoretically analyzed in [16], where it has been shown to be throughput-optimal and to achieve the convex-closure of the information-theoretic rate-region for the network. In order to function inside a real-world wireless network, the rate-allocation algorithm requires a channel-measurement algorithm that can test the feasibility of a transmission in parallel with other transmissions.

In brief, we make the following research contributions in this paper:

- We design a distributed channel-measurement algorithm that can test the feasibility of a transmission in parallel with other transmissions. The channel-measurement algorithm works with the rate-allocation algorithm from [16] to enable Q-CMRA.
- We demonstrate the benefits of Q-CMRA through a real implementation using commercially available 802.11 wireless cards. Moreover, we show that Q-CMRA outperforms traditional CSMA-CA and almost doubles the total network throughput in some topologies.

The rest of this paper is organized as follows. Section II presents other work that is relevant to Q-CMRA. Section III provides some background needed to understand the rest of this paper. Section IV presents the key design steps and the main algorithms. Section V details the implementation steps and Section VI presents experimental results. Section VII provides some additional discussion on Q-CMRA and Section VIII concludes the paper.

II. Related Work

In addition to literature presented in Section I, this section presents other work that is relevant to Q-CMRA. In order to improve the spatial-reuse of orthogonal-access CSMA-CA MAC protocols, [17], [18], [19] propose schemes to dynamically tune the carrier-sensing threshold and validate the schemes via simulations. In the proposed schemes, a node *greedily* adjusts its carrier-sensing threshold to improve its own performance and does not consider the impact on transmissions of nearby nodes. This greedy adjustment of the carrier-sensing threshold causes the proposed schemes to suffer from some hidden-node problems. Moreover, [20], [21], [22] show via simulations that the optimum carrier-sensing threshold that maximizes the throughput allows hidden-nodes to exist. Thus, tuning the carrier-sensing threshold to improve spatial-reuse makes nodes blind and creates artificial hidden-node problems. Q-CMRA improves spatial-reuse without creating artificial hidden-node problems by leaving the carrier-sensing threshold **unchanged**.

[23] shows via simulations that when discrete physical-layer rates are available (as shown in Table I, IEEE 802.11 standards specify several discrete physical-layer rates), tuning transmit-power offers several advantages over tuning carrier-sense threshold. In [23], however, a transmitter uses the highest possible physical-layer rate based only on the interference-level perceived at the transmitter. Moreover, the transmitter does not take the medium-access state of nearby nodes into account when doing rate-allocation. Q-CMRA improves upon [23] by: (*a*) using the **throughput-optimal** *Rate Allocation* algorithm to constrain the maximum allowed physical-layer rate based on queue-length (see Section IV-A); and (*b*) using the *Channel Measurement* algorithm to take the medium-access state of nearby nodes into account when doing rate-allocation (see Section IV-C). Next, we provide some background for Q-CMRA.

TABLE I: Receiver minimum input sensitivity as specified by the IEEE 802.11 standard [24] and the IEEE 802.11ac draft-specification [25]. When all else is equal, higher physical-layer (PHY) rates often use higher modulation-schemes and/or higher coding-rates. As a result, higher PHY-rates often require a higher **Minimum Sensitivity** (i.e. a higher average SINR) at the receiver to be correctly decoded.

Modulation Scheme	**Coding Rate**	**PHY Rate (Mbps) [800ns GI; 1x1 ANT]**	**Minimum Sensitivity (dBm) [20MHz CH]**
BPSK	1/2	6.5	-82
QPSK	1/2	13	-79
QPSK	3/4	19.5	-77
16-QAM	1/2	26	-74
16-QAM	3/4	39	-70
64-QAM	2/3	52	-66
64-QAM	3/4	58.5	-65
64-QAM	5/6	65	-64
256-QAM	3/4	78	-59
256-QAM	5/6	86.7	-57

III. Background

In this paper, we design, implement, and evaluate Q-CMRA for IEEE 802.11 networks because such wireless networks are very popular. However, the algorithms presented in Section IV could be adapted for use in other types of wireless networks. The following sub-sections provide some background material on IEEE 802.11 that is necessary to understand the rest of this paper.

A. Physical-Layer Rates

Like most wireless standards, the IEEE 802.11 standard [24] and the IEEE 802.11ac draft-specification [25] specify different modulation schemes, coding rates, spatial streams, guard intervals, and bandwidths that result in multiple effective physical-layer rates, referred to as "rates" from now on. When all else is equal, higher rates often use higher modulation-schemes and/or higher coding-rates. As a result, higher rates often require a higher average signal-to-interference-and-noise-ratio (SINR) at the receiver to be correctly decoded (refer to Table I). A transmitter could sustain a higher SINR (and hence a higher rate) at its receiver by using a higher transmit-power. However, a higher transmit-power causes more interference for nearby nodes, thereby reducing opportunities for parallel transmissions. Next, we describe inter-frame-spacing durations.

B. Inter-Frame-Spacing Durations

Random-access wireless standards, such as the IEEE 802.11 standard [24], use different inter-frame-spacing (IFS) durations to facilitate MAC protocol design. For example, a short-IFS (SIFS) duration separates a related sequence of RTS,

CTS, DATA, and ACK frames. A slightly longer polling-IFS (PIFS) duration separates previously transmitted frames from some higher priority management-frames. An even longer distributed-IFS (DIFS) duration separates previously transmitted frames from subsequent data-frames. Different IFS durations differ by a multiple of the mini-slot-time (MST). For example, the following is a commonly used setting: *(i)* PIFS = SIFS + MST; and *(ii)* DIFS = SIFS + 2*MST. Thus, by delegating different tasks to different IFS durations, wireless networks can be controlled, managed, and operated in a distributed manner. Such a distributed architecture provides us a good foundation, especially when designing MAC protocols for ad-hoc and mesh networks. Next, we describe MAC-layer frames.

C. MAC-Layer Frames

In this paper, nodes can operate either in *CSMA-CA Mode* (refer to Section III-D) or *Q-CMRA Mode* (refer to Section III-E). Regardless of mode-of-operation, the following apply:

- Beacon-frames (similar to standard 802.11 beacon-frames) are periodically broadcast by nodes that transmit data-frames and do not solicit any ACK-frames. They also carry information regarding the mode-of-operation of nodes. This enables nodes to switch from *Q-CMRA Mode* to *CSMA-CA Mode* in certain situations (refer to Section IV-B).
- Request-frames (these are non-standard frames) are used by a node to make various requests to other nearby nodes. We introduce reQuest-IFS (QIFS) and redefine standard IFS values as follows: *(i)* PIFS = SIFS + MST; *(ii)* QIFS = SIFS + 2*MST; and *(iii)* DIFS = SIFS + 3*MST. Please note that QIFS $<$ DIFS. Moreover, in some implementations, QIFS may be equal to PIFS to facilitate backwards-compatibility.
- Beacon-frames, request-frames, and ACK-frames are modulated using the lowest physical-layer rate. Hence, they can be decoded even when the SINR at the receiver is **very low** (refer to III-A). This enables these frames to be: (*a*) received widely; and (*b*) decoded often (i.e. even in the presence of significant interference from two or more neighbors).

Next, we describe *CSMA-CA Mode*.

D. CSMA-CA Mode

When nodes are operating in *CSMA-CA Mode*, they *always* use IEEE 802.11 DCF to transmit frames (i.e. all frames are transmitted using exponential randomized back-off). Moreover, nodes operating in *CSMA-CA Mode* transmit frames using the following channel-access rules:

- *Beacon-frames*: Similar to IEEE 802.11 DCF, beacon-frames are scheduled DIFS time after the previous frame **while** using exponential randomized back-off.
- *Request-frames*: Nodes operating in *CSMA-CA Mode* do not use request-frames.
- *Data-frames*: Similar to IEEE 802.11 DCF, data-frames are scheduled DIFS time after the previous frame **while** using exponential randomized back-off.

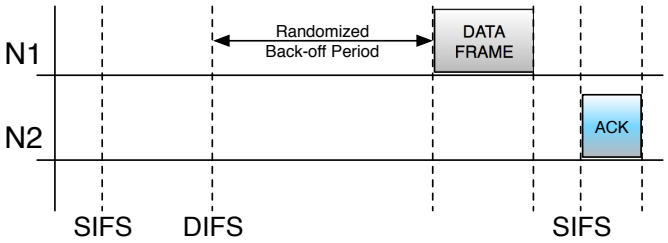


Fig. 1: Example exchange of a data-frame between nodes operating in *CSMA-CA Mode*. Data-frames are scheduled DIFS time after the previous frame **while** using exponential randomized back-off.

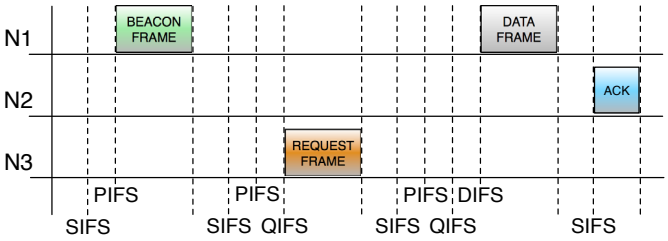


Fig. 2: Channel-access rules for *Q-CMRA Mode*. In *Q-CMRA Mode*, frames are scheduled at exact IFS times (depending on frame-type) after the previous frame **without** using exponential randomized back-off. However, a node still uses carrier-sense to defer its transmission in case the wireless-medium becomes busy before its transmission is scheduled.

Figure 1 shows an example exchange of a data-frame between nodes operating in *CSMA-CA Mode*. Data-frames are scheduled DIFS time after the previous frame **while** using exponential randomized back-off. Next, we describe *Q-CMRA Mode*.

E. Q-CMRA Mode

When nodes are operating in *Q-CMRA Mode*, they **never** use exponential randomized back-off to transmit frames. However, a node still uses carrier-sense to defer its transmission in case the wireless-medium becomes busy before its transmission is scheduled. Moreover, nodes operating in *Q-CMRA Mode* transmit frames using the following channel-access rules:

- *Beacon-frames*: As shown in Figure 2, beacon-frames are scheduled exactly PIFS time after the previous frame **without** using exponential randomized back-off. Moreover, nodes could use collision-avoidance mechanisms (e.g. mesh beacon collision avoidance (MBCA) from the 802.11 standard [24]) to mitigate collisions between beacon-frames in dense settings.
- *Request-frames*: As shown in Figure 2, request-frames are scheduled exactly QIFS time after the previous frame **without** using exponential randomized back-off. They are broadcast and do not solicit any ACK-frames. Request-frames are used by a node to make various requests to other nearby nodes without causing collisions with data-frames (because QIFS $<$ DIFS). Nodes interpret predefined requests based on the contents of received request-frames.

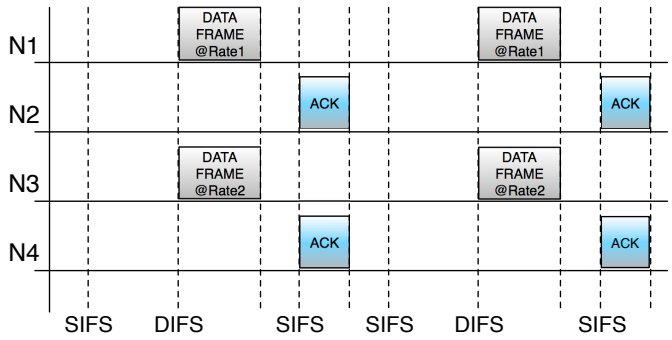


Fig. 3: Example exchange of data-frames between nodes operating in *Q-CMRA Mode*. Data-frames are scheduled (in parallel) **exactly** DIFS time after the previous frame **without** using exponential randomized back-off. However, a node still uses carrier-sense to defer its transmission in case the wireless-medium becomes busy before its transmission is scheduled.

- *Data-frames*: As shown in Figure 2, data-frames are scheduled exactly DIFS time after the previous frame **without** using exponential randomized back-off. In this paper, we use this simple scheduling strategy to show the fundamental benefits of *Q-CMRA Mode* over *CSMA-CA Mode* in the simple setups of Figure 5 (see Section VII for discussion on scheduling).

Figure 3 shows an example exchange of data-frames between nodes operating in *Q-CMRA Mode*. Data-frames are scheduled (in parallel) **exactly** DIFS time after the previous frame **without** using exponential randomized back-off. However, a node still uses carrier-sense to defer its transmission in case the wireless-medium becomes busy before its transmission is scheduled. Moreover, the size (in bytes) of each data-frame is adjusted (depending on the physical-layer rate being used to transmit the data-frame) so that **all** data-frames take similar amounts of air-time (or TXOP) to finish transmitting. When nodes are operating in *Q-CMRA Mode*, this ensures maximum overlap between data-frames that are transmitted in parallel. Next, we present the design of Q-CMRA.

IV. Design

This section describes the three core Q-CMRA modules: (i) *Rate Allocation*; (ii) *Rate Feasibility*; and (iii) *Channel Measurement*. *Rate Allocation* repeatedly chooses the maximum allowed physical-layer rate based on queue-length. *Rate Feasibility* finds the highest rate (less than or equal to the maximum allowed rate) that is feasible in the current wireless environment by using *Channel Measurement*. A summary of these modules is presented in Figure 4.[1] Obviously, Q-CMRA modules can only be used by nodes operating in *Q-CMRA Mode*. Although we implement the Q-CMRA mechanism using 802.11 hardware in Section V, the concepts presented in this section also apply to other decentralized wireless networks. Next, we describe the *Rate Allocation* module.

[1]We omit details on handling of rare cases and use standard C/C++ like syntax in our algorithms.

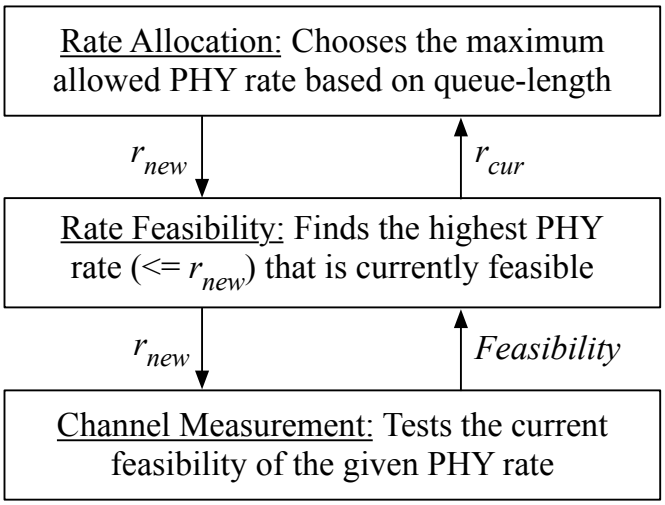


Fig. 4: Summary of Q-CMRA Modules

Algorithm 1 Rate Allocation Algorithm

Repeatedly chooses the maximum allowed physical-layer rate based on queue-length(s).

Initialize:

Let $R = [r_0, r_1, r_2, ..., r_k]$ be the set of rates. $r_0 = 0$ corresponds to no transmission.
Let $f(x)$ be an increasing function, such as $\log(x)$.
$endLoop$ is set to 0 when connection is opened.
$endLoop$ is set to 1 when connection is closed.

Rate_Allocation()

$r_{cur} = r_0$; $p_{cur} = 0$;
while $endLoop \neq 1$ **do**
  $qLen =$ Get_Q_Length();
  $S = R - \{r_{cur}\}$;
  $t_{sum} = 1/(\sum_{s \in S} e^{s*f(qLen)})$;
  sleep(t_{sum});
  $P[r_{new} = s] = e^{s*f(qLen)} * t_{sum}$;
  Rate_Feasibility(r_{new}, &r_{cur}, &p_{cur});
  Update_Max_Allowed_Rate(r_{cur}, p_{cur});
end while

A. Rate Allocation

This section describes a *Rate Allocation* algorithm (outlined in Algorithm 1) that enables the Q-CMRA framework. *Rate Allocation* uses the theoretical framework of [16] to probabilistically choose the maximum allowed physical-layer rate based on queue-length(s). It has been shown to be throughput-optimal under certain conditions in [16]. In other words, if any algorithm can stably operate a network, then *Rate Allocation* can also stabilize the network. Please note that Q-CMRA could use other rate-allocation algorithms in the future to further improve performance.

At the start of each iteration, *Rate Allocation* finds $qLen$,

i.e. the length of the transmitter's queue.[2] Before computing the rate, it determines a waiting time parameter $t_{sum} = 1/(\sum_{s \in S} e^{s*f(qLen)})$, where S is the set of all physical-layer rates excluding the current rate r_{cur}. $f(x)$ is a monotone increasing function; for implementation, a slowly increasing function (such as $\log(x)$) is found to keep the frequency of updates within reasonable limits. Note that t_{sum} depends on $qLen$ and controls the frequency at which the algorithm tries to update the rate. When $qLen$ is large, these attempts are more frequent.

Once the waiting time passes, the algorithm chooses $r_{new} = s$, $s \in S$, with probability $e^{s*f(qLen)} * t_{sum}$. Thus, r_{new} is one of the rates in S chosen according to the above mentioned probability distribution (refer to [16] for the theory behind this probability distribution). This distribution chooses a higher rate with higher probability, and the probability of choosing a higher rate increases as $qLen$ increases.

Rate Allocation is only capable of choosing r_{new}, i.e. the maximum allowed physical-layer rate based on queue-length. It has no idea about what rates are feasible in the current wireless environment. Thus, it invokes *Rate Feasibility* to find r_{cur} (i.e. the highest rate [less than or equal to r_{new}] that is **currently** feasible) and p_{cur} (i.e. the required transmit-power-level). Next, *Rate Allocation* updates the maximum allowed physical-layer rate for transmission of data-frames to the r_{cur} obtained from *Rate Feasibility*. *Rate Allocation* then moves on to its next iteration.

Rate Allocation does not perform congestion-control and deal with throughput-fairness. However, it could be used with a congestion-control algorithm that uses queue-status. In the context of CSMA, congestion-control using queues has been studied in [3]. We believe that the approach in [3] is applicable to Q-CMRA as well. Next, we describe the *Rate Feasibility* module.

B. Rate Feasibility

This section describes a *Rate Feasibility* algorithm (outlined in Algorithm 2) that enables the Q-CMRA framework. As explained in Section III-A, a higher physical-layer rate requires a higher average SINR at the receiver to be decoded correctly. Thus, certain physical-layer rates may not be feasible in the current wireless environment due to excessive interference or noise. *Rate Feasibility* uses *Channel Measurement* to find the highest rate (less than or equal to r_{new}) that is currently feasible. It also determines the required transmit-power-level. Please note that Q-CMRA could use other rate-feasibility algorithms in the future to further improve performance.

Rate Feasibility invokes *Channel Measurement* to test whether the rate r_{new} is feasible in the current wireless environment. If *Channel Measurement* indicates that the rate r_{new} is not feasible, *Rate Feasibility* repeats the same process with the next lower rate to check whether it is feasible. However, if *Channel Measurement* indicates that the rate r_{new} is feasible, then *Rate Feasibility* obtains the required transmit-power-level p_{new} from *Channel Measurement*. Finally, *Rate Feasibility* returns r_{cur} (i.e. the highest rate [less than or equal to the originally provided r_{new}] that is **currently** feasible) and p_{cur} (i.e. the required transmit-power-level) to *Rate Allocation*.

[2]For multi-hop networks, this must equal the "back-pressure", i.e. the difference in the queue-lengths of the transmitter and the next receiver on the multi-hop path to the final destination.

Algorithm 2 Rate Feasibility Algorithm

Uses channel-measurements to find the highest rate (less than or equal to the originally provided r_{new}) that is currently feasible.

Rate_Feasibility(r_{new},*r_{cur},*p_{cur})

```
updateRate = 0;
if r_new < *r_cur then
  p_new = Find_Lowest_Feasible_TX_Power(*p_cur);
  updateRate = 1;
else
  while r_new > *r_cur && updateRate == 0 do
    p_new = *p_cur;
    if Channel_Measurement(r_new, &p_new) == 1 then
      updateRate = 1;
    else
      Get_Next_Lower_Rate(&r_new);
    end if
  end while
end if
if updateRate == 1 then
  *r_cur = r_new; *p_cur = p_new;
end if
```

Channel Measurement incurs some overhead when testing the feasibility of a rate. Using past indications from *Channel Measurement*, *Rate Feasibility* could maintain a cache of feasible rates on a node. Instead of invoking *Channel Measurement* every time, *Rate Feasibility* could first search the node's cache to approximate the feasibility of a given rate in the current wireless environment. Such an approach may improve overall performance by reducing *Channel Measurement* overhead.

In some topologies, nodes can only communicate via orthogonal-access (e.g. Figure 5c). In such situations, *Rate Feasibility* (running on a node) may **repeatedly** find that **none** of the non-zero physical-layer rates are feasible. Subsequently, *Rate Feasibility* will cause the node to: *(i)* update the mode-of-operation in its beacon-frame to *CSMA-CA Mode* and transmit its beacon-frame; and *(ii)* switch from *Q-CMRA Mode* to *CSMA-CA Mode*. When other nearby nodes receive the node's updated beacon-frame, they will: *(i)* update the mode-of-operation in their beacon-frames to *CSMA-CA Mode* and transmit their beacon-frames; and *(ii)* switch from *Q-CMRA Mode* to *CSMA-CA Mode*. Next, we describe the *Channel Measurement* module.

C. Channel Measurement

This section describes a simple *Channel Measurement* algorithm (outlined in Algorithm 3) that enables the Q-CMRA framework. *Rate Feasibility* invokes *Channel Measurement* to **actively** test whether a parallel transmission at a given physical-layer rate is feasible in the current wireless environment. Although *Channel Measurement* incurs some

Algorithm 3 Channel Measurement Algorithm

Uses request-frames to test whether rate r_{new} is feasible in the current wireless environment.

Channel_Measurement(r_{new},*****p_{new})

```
ourFlow = TX_Frame(r_new, *p_new);
while ourFlow != ACK do
  if ourFlow == NO_ACK then
    if *p_new == MAX_POWER then
      return 0;
    end if
    Get_Next_Higher_TX_Power(p_new);
  else if ourFlow == REQUEST then
    return 0;
  end if
  ourFlow = TX_Frame(r_new, *p_new);
end while
wait = REQUEST_WAIT;
while wait > 0 do
  if TX_Frame(r_new, *p_new) == REQUEST then
    return 0;
  end if
  wait = wait - 1;
end while
return 1;
```

overhead, it is simple to implement and enables Q-CMRA to significantly improve the total network throughput in some topologies (see Section VI). Please note that Q-CMRA could use other channel-measurement algorithms (one such example is described in Appendix A) in the future to further improve performance.

Channel Measurement works as follows. Consider a Device_A trying to transmit a frame at rate r_{new} and transmit-power-level p_{new}. Suppose that there is another Device_B operating in *Q-CMRA Mode* and transmitting data-frames without using exponential randomized back-off. When testing feasibility, Device_A tries to transmit frames in parallel with Device_B. If Device_A receives no ACK-frame from its intended receiver, it increases its transmit-power-level by a single step and retransmits the frame. Power-level is increased as required to a maximum of MAX_POWER.

As Device_A tries to test the feasibility of rate r_{new}, it could cause collisions with Device_B's transmissions. When Device_B senses enough collisions (i.e. REQUEST_START number of collisions) to ascertain that the collisions are not due to occasional deep-fades or noise-bursts, it starts sending request-frames (refer to Section III-C) to indicate infeasibility to Device_A. REQUEST_START $= 3$ worked well in our experiments. If Device_A receives request-frames, it concludes that rate r_{new} is not feasible and returns a 0 to *Rate Feasibility*.

If Device_A receives an ACK-frame from its intended receiver, it tries to send REQUEST_WAIT number of queued frames at rate r_{new} and required transmit-power-level p_{new} while waiting for any request-frames from Device_B. This additional step is required because Device_B must detect REQUEST_START collisions before transmitting request-frames to indicate infeasibility to Device_A. REQUEST_WAIT should be greater than REQUEST_START to give Device_B enough time to react. REQUEST_WAIT $= 5$ worked well in our experiments. If Device_A receives request-frames during this step, it concludes that rate r_{new} is not feasible and returns a 0 to *Rate Feasibility*. Otherwise, it concludes that rate r_{new} is feasible at the required transmit-power-level p_{new} and returns a 1 to *Rate Feasibility*.

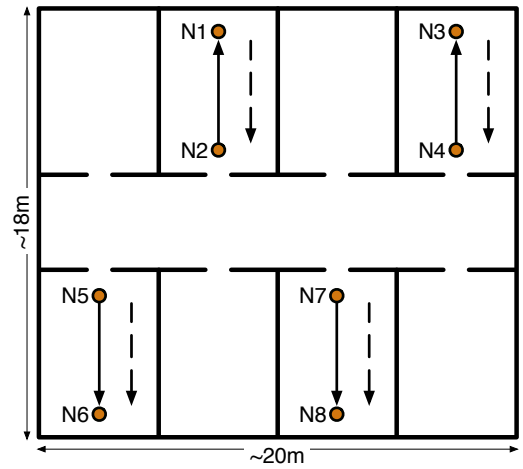
As is explained in Section III-C, request-frames are modulated using the lowest physical-layer rate. Hence, they can be decoded even when the SINR at the receiver is **very low**. Thus, *Channel Measurement* works reliably in most cases by employing request-frames. However, in some rare cases, it may not work as intended. One such rare case arises when a node using *Channel Measurement* breaks two (or more) established flows, wherein the transmitters of the established flows cause equivalent interference at the node. Thus, two (or more) request-frames may collide at the node and may not be detected by the node. In such situations, transmitters of the previously established flows can initiate a switch to *CSMA-CA Mode* using the procedure described in IV-B. Next, we describe the implementation of Q-CMRA.

V. Implementation

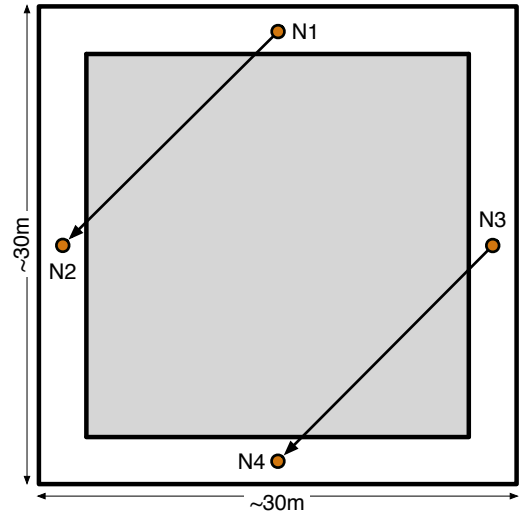
This section briefly discusses the implementation of Q-CMRA. Our experiments run on the Proteus platform [26]. Each Proteus-node uses an Atheros 802.11 wireless-card (ath5k wireless-driver) for communication and an x86-processor to run the Linux operating-system. We test and evaluate both *CSMA-CA Mode* (refer to Section III-D) and *Q-CMRA Mode* (refer to Section III-E) in an office-like environment using 4 different experimental setups (Figure 5). Please note that in all setups, arrows point from transmitters of data-frames to their corresponding receivers. Moreover, for a given mode-of-operation, transmitters are offered the same load. We leave the evaluation of Q-CMRA when transmitters are offered different loads as future work.

When nodes are using *CSMA-CA Mode* in the experimental setups of Figures 5a and 5c, they can utilize the highest physical-layer rate with less than 5% packet-error-rate because of excellent signal-to-noise conditions. Thus, rate-adaptation **is not** beneficial in these setups. When measuring the throughput of nodes using *CSMA-CA Mode* in these setups, we let the nodes use the highest physical-layer rate for each transmission. On the other hand, when nodes are using *CSMA-CA Mode* in the experimental setup of Figure 5b, rate-adaptation **is** beneficial. When measuring the throughput of nodes using *CSMA-CA Mode* in this setup, we let the nodes use the Linux-kernel's default rate-adaptation algorithm (i.e. the Minstrel algorithm).

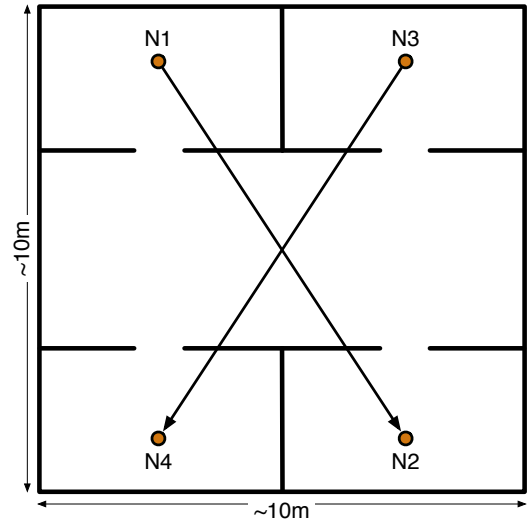
Please note that traditional CSMA-CA only allows orthogonal-access to the wireless-channel because of its rigid carrier-sensing threshold. However, nodes using *Q-CMRA Mode* often need to transmit in parallel. In order to implement *Q-CMRA Mode*, we need a way to opportunistically disregard traditional CSMA-CA's rigid carrier-sensing threshold. In *Q-CMRA Mode*, we allow interfering transmissions to happen



(a) Setup1 (solid arrows) and Setup2 (dashed arrows)



(b) Setup3. *CSMA-CA Mode* **uses** rate-adaptation in this setup.



(c) Setup4

Fig. 5: Setup1 has the most opportunity for parallel transmissions and Setup4 has the least.

in parallel by disabling exponential randomized back-off for the hardware transmit queues of the wireless-chip. Next, we present the results and evaluation of Q-CMRA.

VI. Results and Evaluation

This section compares the performance (see Figure 6) of *Q-CMRA Mode* (refer to Section III-E) with *CSMA-CA Mode* (refer to Section III-D) in the 4 setups shown in Figure 5. *Q-CMRA Mode* is able to exploit the topology of nodes in Setup1, Setup2, and Setup3 to outperform *CSMA-CA Mode*. In Setup4, however, nodes automatically switch from *Q-CMRA Mode* to *CSMA-CA Mode* (refer to Section IV-B) because the topology favors orthogonal-access. Thus, *Q-CMRA Mode* and *CSMA-CA Mode* have similar performance in Setup4.

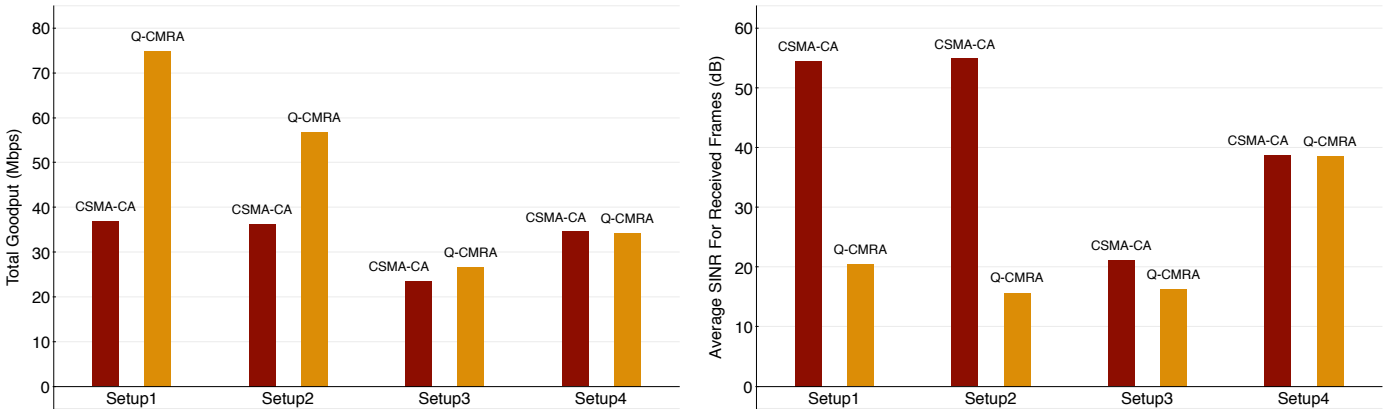
Figure 6a shows that *Q-CMRA Mode* outperforms *CSMA-CA Mode* (in terms of total goodput) by about 103.53% in Setup1, 56.47% in Setup2, and 13.19% in Setup3. The throughput gains in Setup2 are smaller because N2 and N4 experience greater average interference due to the topology. The throughput gains in Setup3 are smallest because there are fewer flows (i.e. 2-flows in Setup3 vs. 4-flows in Setup1 and Setup2) that can be scheduled in parallel.

Figure 6b shows that the average SINR for frames received by receiver nodes is considerably lower for *Q-CMRA Mode* in Setup1, Setup2, and Setup3. *Q-CMRA Mode* makes more efficient use of the wireless spectrum by intelligently allowing transmitter nodes to transmit in parallel. Parallel transmissions increase interference and hence decrease SINR for received frames; however, such parallelism improves total system throughput. *CSMA-CA Mode* is very conservative and causes transmitter nodes to miss several opportunities to transmit in parallel because of its rigid carrier-sensing threshold. On the other hand, *Q-CMRA Mode* improves total system throughput by intelligently trading off SINR for parallel transmissions.

Our simple experiments show that *Q-CMRA Mode* is able to substantially improve the spatial-reuse and total network throughput over *CSMA-CA Mode*. *Q-CMRA Mode* identifies and makes use of opportunities for parallel transmissions depending on the topology. Therefore, the throughput gains from *Q-CMRA Mode* are strictly a function of the underlying topology. Nonetheless, as shown in Setup4, *Q-CMRA Mode* never performs significantly worse than *CSMA-CA Mode*.

As described in Section II, [17], [18], [19] propose schemes to dynamically tune the carrier-sensing threshold and validate the proposed schemes **only** via simulations. Unfortunately, the proposed schemes are non-trivial to implement in a real-world network and cannot be compared to Q-CMRA. We acknowledge that the proposed schemes address the same shortcomings of traditional CSMA-CA (e.g. rigid carrier-sensing threshold) and may have comparable performance to Q-CMRA. However, as explained in Section II, the proposed schemes create artificial hidden-node problems.

On the other hand, Q-CMRA improves spatial-reuse without creating artificial hidden-node problems by leaving the carrier-sensing threshold **unchanged**. Moreover, in order to function properly, Q-CMRA requires beacon-frames and request-frames to be exchanged between far-off nodes. This works best when the carrier-sensing threshold is left at its default value. Next, we present some discussion on Q-CMRA.



(a) Total Goodput

(b) Average SINR For Received Frames

Fig. 6: Performance Comparison: *CSMA-CA Mode* vs. *Q-CMRA Mode*

VII. Discussion

A node can marginally improve spatial-reuse by adapting physical-layer-rate and transmit-power based on channel-conditions and feedback from its receiver [8], [27], [23]. In order to facilitate multiple interfering transmissions to happen in parallel and further improve spatial-reuse, the node must jointly control medium-access and rate-allocation. Our Q-CMRA mechanism does that and significantly improves the spatial-reuse and total network throughput.

One may choose different algorithms for the various Q-CMRA modules depending on external factors, such as hardware capabilities, desired complexity, choice of scheduler, system model, etc. Like [28], Q-CMRA may use QoS extensions to control channel access probabilities and further enhance performance in multi-hop settings. Beam-forming MIMO antennas, such as those used by several 802.11n wireless cards, give more directionality to a link and thus reduce the interference for other links. Given its opportunistic nature, Q-CMRA can effectively utilize beam-forming gains to further improve spatial-reuse and total network throughput.

The channel-measurement algorithms presented in this paper require transmitters to perform the majority of tasks. In addition to this, a receiver can aid its transmitter in making channel-measurements and estimating rate-feasibility. For example, if a transmitter sends an RTS to its intended receiver, the receiver can choose whether to reply back with a CTS based on its own estimate of rate-feasibility. Such an approach is expected to make channel-measurement faster and more accurate. We leave such enhancements as future work.

In order to implement Q-CMRA, we did **not** modify or disable the carrier-sensing threshold because it is essential for the decentralized operation of wireless networks (refer to Section III-B) and also facilitates other QoS enhancements [24], [28]. However, as is explained in Section V, we disable exponential randomized back-off in order to implement Q-CMRA on real hardware. Although Q-CMRA outperforms CSMA-CA with this hack, we leave more elegant implementations of Q-CMRA as future work.

In this paper, Q-CMRA does not use a scheduler; i.e. a Q-CMRA node attempts to transmit every single time (at the DIFS boundary) unless its maximum allowed physical-layer rate is zero (see Figure 3). In the simple setups of Figure 5, we are able to show the fundamental benefits of Q-CMRA over traditional CSMA-CA without using a scheduler. However, to support larger setups that mimic real-world wireless network deployments, Q-CMRA requires a complementary scheduler. We leave the design, implementation, and evaluation of such a scheduler as future work. Next, we present our conclusion for Q-CMRA.

VIII. Conclusion

In this paper, we design, implement, and evaluate a distributed cross-layer mechanism called Q-CMRA. Q-CMRA facilitates multiple transmissions to happen in parallel whenever feasible by **jointly** controlling medium-access and rate-allocation. Thus, it significantly improves spatial-reuse and doubles the total network throughput in some setups. Q-CMRA is easy to implement and enhances total network throughput. Thus, it could be useful in high-density infrastructure networks, mesh networks, and peer-to-peer networks.

Appendix A
Another Channel Measurement Algorithm

This section describes another channel-measurement algorithm (outlined in Algorithm 4). In *Channel Measurement*, the onus is on the transmitter of the established flow to indicate infeasibility. In Algorithm 4, however, the device testing feasibility of a new rate does most of the work. Algorithm 4 requires sequential and real-time guarantees to function properly. Thus, it can best be implemented in the MAC-hardware of wireless-chips. Moreover, Algorithm 4 is faster than *Channel Measurement* at testing feasibility. Thus, we believe that it should improve Q-CMRA's performance by reducing channel-measurement overhead.

There are several unused bits in the Frame Control field of the 802.11 ACK-frame [24]. Algorithm 4 uses one of these unused bits as a sequence-bit (SEQ-bit) to *quickly* test the feasibility of a rate. Please note that the SEQ-bit can either be 0 or 1. Next, we explain the functionality of the SEQ-bit using the sequence of events shown in Figure 7. A receiver uses a

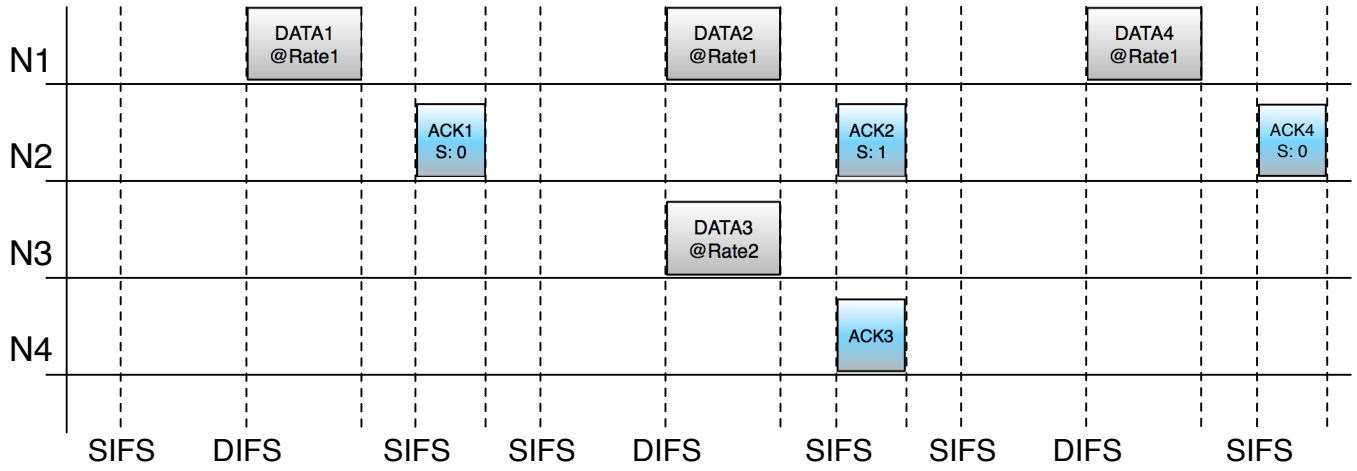


Fig. 7: N3 uses **ACK-snooping** to test the feasibility of a given rate (i.e. Rate2) while co-existing with N1.

Algorithm 4 Another Channel Measurement Algorithm

Uses the SEQ-bit of ACK-frames to test whether rate r_{new} is feasible in the current wireless environment.

Channel_Measurement(r_{new},*p_{new})

```
oldSEQBit = Get_Next_SEQ_Bit();
while 1 do
  ourFlow = TX_Frame(r_new, *p_new);
  newSEQBit = Get_Next_SEQ_Bit();
  if newSEQBit == oldSEQBit || oldSEQBit == -1
  || newSEQBit == -1 then
    if ourFlow == NO_ACK then
      if *p_new == MAX_POWER then
        return 0;
      end if
      Get_Next_Higher_TX_Power(p_new);
    else
      return 1;
    end if
  else
    return 0;
  end if
  oldSEQBit = newSEQBit
end while
```

new SEQ-bit value only when it receives a fresh frame from the same transmitter. In Figure 7, Node1 (N1) is operating in *Q-CMRA Mode* and sending data-frames to N2 without using exponential randomized back-off. N3 is using Algorithm 4 to test the feasibility of a given rate (i.e. Rate2).

In time-slot-1, N3 snoops ACK1, uses the Get_Next_SEQ_Bit function to extract the SEQ-bit from ACK1, and then assigns the extracted SEQ-bit value to the variable $oldSEQBit$. In time-slot-2, N3 transmits DATA3 to N4 in parallel with N1's transmission of DATA2 to N2. This time around, N3 will only try to receive ACK3 from N4. Assuming that N3 has no idea about ACK2, it currently does not know whether DATA2 was successfully transmitted and received while it was busy transmitting its own DATA3. If it receives ACK3, it knows that its DATA3 was successfully received by its intended receiver N4. Now, N3 must find out whether it broke the flow from N1 to N2 in time-slot-2. In time-slot-3, N3 snoops ACK4, uses the Get_Next_SEQ_Bit function to extract the SEQ-bit from ACK4, and then assigns the extracted SEQ-bit value to the variable $newSEQBit$.

Suppose $oldSEQBit$ is equal to 0. $newSEQBit$ could be either 0 or 1. If $newSEQBit$ is 0 (as shown in Figure 7), N3 did not break the flow from N1 to N2 in time-slot-2 because N1 did not retransmit any frames. If N3 did not receive ACK3 from N4 in time-slot-2, it must repeat the ACK-snooping process while using a higher transmit-power-level. Otherwise N3 concludes that rate r_{new} is feasible at the required transmit-power-level p_{new} and returns a 1 to *Rate Feasibility*. If $newSEQBit$ is 1, N3 broke the flow from N1 to N2 in time-slot-2 because DATA4 is a retransmission of DATA2. Thus, N3 concludes that rate r_{new} is not feasible and returns a 0 to *Rate Feasibility*.

As is explained in Section III-C, ACK-frames are modulated using the lowest physical-layer rate and can be decoded even when the SINR at the receiver is **very low**. Thus, vulnerable receivers are **usually** not present when no ACK-frame is snooped from nearby nodes. In such situations, the Get_Next_SEQ_Bit function makes an **approximation** and returns a -1. Please refer to Section IV-B and Section IV-C for handling of some relevant rare cases that may arise due to this approximation. Moreover, multiple SEQ-bits and more sophisticated snooping-sequences could be used to further improve the robustness of Algorithm 4 in rare cases, such as when two or more nodes use Algorithm 4 at the same time to test the feasibility of physical-layer rates.

REFERENCES

[1] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," in *Proc. 2008 Allerton Conference on Commun., Control and Computing*.

[2] S. Rajagopalan, D. Shah, and J. Shin, "Network adiabatic theorem: an efficient randomized protocol for contention resolution," in *2009 ACM Sigmetrics*.

[3] L. Jiang, D. Shah, J. Shin, and J. Walrand, "Distributed random access algorithm: scheduling and congestion control," *IEEE Trans. Inf. Theory*, 2010.

[4] J. Ni, B. Tan, and R. Srikant, "Q-CSMA: queue-length based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks," in *Proc. 2010 IEEE INFOCOM*.

[5] J. Choi, J. Na, K. Park, and C. Kim, "Adaptive optimization of rate adaptation algorithms in multi-rate WLANs," in *Proc. 2007 IEEE ICNP*.

[6] T. Lin and J. Hou, "Interplay of spatial reuse and SINR-determined data rates in CSMA/CA-based, multi-hop, multi-rate wireless networks," in *Proc. 2007 IEEE INFOCOM*.

[7] H. Zhai and Y. Fang, "Physical carrier sensing and spatial reuse in multirate and multihop wireless ad hoc networks," in *Proc. 2006 IEEE INFOCOM*.

[8] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," in *Proc. 2011 ACM MobiCom*.

[9] A. Kamerman and L. Monteban, "WaveLAN-II: a high-performance wireless LAN for the unlicensed band," *Bell Labs Technical J.*, 1997.

[10] M. Lacage, M. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: a practical approach," in *Proc. 2004 ACM MSWiM*.

[11] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic media access for multirate ad hoc networks," in *Proc. 2002 ACM MobiCom*.

[12] L. Tassiulas and A. Ephremides, "Stability properties of constrained queuing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, 1992.

[13] A. Eryilmaz, R. Srikant, and J. Perkins, "Stable scheduling policies for fading wireless channels," *IEEE/ACM Trans. Networking*, 2005.

[14] X. Lin, N. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE J. Sel. Areas Commun.*, 2006.

[15] L. Georgiadis, M. Neely, M. Neely, and L. Tassiulas, *Resource Allocation and Cross-Layer Control in Wireless Networks*. NOW Publishers, 2006.

[16] J. Jose and S. Vishwanath, "Distributed rate allocation for wireless networks," *IEEE Trans. Inf. Theory*, 2011.

[17] J. Zhu, X. Guo, L. Lily Yang, W. Steven Conner, S. Roy, and M. Hazra, "Adapting physical carrier sensing to maximize spatial reuse in 802.11 mesh networks," *Wireless Communications and Mobile Computing*, 2004.

[18] A. Vasan, R. Ramjee, and T. Woo, "ECHOS - enhanced capacity 802.11 hotspots," in *Proc. 2005 IEEE INFOCOM*.

[19] H. Ma, S. Shin, and S. Roy, "Optimizing throughput with carrier sensing adaptation for IEEE 802.11 mesh networks based on loss differentiation," in *Proc. 2007 IEEE ICC*.

[20] E. Wong and R. Cruz, "A spatio-temporal model for physical carrier sensing wireless ad-hoc networks," in *Proc. 2006 IEEE SECON*.

[21] J. Deng, B. Liang, and P. Varshney, "Tuning the carrier sensing range of IEEE 802.11 MAC," in *Proc. 2004 IEEE GLOBECOM*.

[22] H. Ma, H. Alazemi, and S. Roy, "A stochastic model for optimizing physical carrier sensing and spatial reuse in wireless ad hoc networks," in *Proc. 2005 IEEE MASS*.

[23] T. Kim, H. Lim, and J. Hou, "Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks," in *Proc. 2006 ACM MobiCom*.

[24] "Specific requirements part 11: WLAN MAC and PHY specifications," IEEE Std 802.11-2012, 2012.

[25] "Specification framework for TGac," IEEE 802.11-09/0992r21, 2011.

[26] N. Paine and S. Vishwanath, "Proteus: a modular ad hoc networking testbed," 2009. Available: http://proteus.ece.utexas.edu

[27] J. Monks, V. Bharghavan, and W. Hwu, "A power controlled multiple access protocol for wireless packet networks," in *Proc. 2001 IEEE INFOCOM*.

[28] A. Warrier, S. Janakiraman, S. Ha, and I. Rhee, "Diffq: practical differential backlog congestion control for wireless networks," in *Proc. 2009 IEEE INFOCOM*.

Vidur Bhargava is a Ph.D. candidate in Electrical and Computer Engineering at the University of Texas at Austin. He received his B.S. and M.S. in Electrical and Computer Engineering from the same university in 2009 and 2010. He was a research assistant with the Wireless Networking and Communications Group at the University of Texas at Austin and has interned at Apple Inc., Cupertino, CA. His research interests include wireless communications, wireless networking, and embedded systems.

Jubin Jose received his B.Tech. in Electrical Engineering from Indian Institute of Technology, Madras in 2006. He received his M.S. and Ph.D. in Electrical and Computer Engineering from the University of Texas at Austin in 2008 and 2011. His industry experience includes positions at NEC Labs, Qualcomm, and Alcatel-Lucent Bell Labs. He is currently with Qualcomm Research, Bridgewater, NJ. His research interests include wireless communications, information theory, and network optimization.

Kannan Srinivasan is an Assistant Professor in the department of Computer Science and Engineering at The Ohio State University. He was a post-doctoral researcher at the University of Texas at Austin in 2011. He received his Ph.D in Electrical Engineering from Stanford University in 2010. He has published more than 20 papers in international conferences, workshops and journals. He has won Best Paper Award in MobiCom 2010 and Best Demo Award at MobiCom 2010.

Sriram Vishwanath received the B. Tech. degree in Electrical Engineering from the Indian Institute of Technology (IIT), Madras, India in 1998, the M.S. degree in Electrical Engineering from California Institute of Technology (Caltech), Pasadena USA in 1999, and the Ph.D. degree in Electrical Engineering from Stanford University, Stanford, CA USA in 2003. His industry experience includes positions at Lucent Bell Labs and National Semiconductor Corporation. He is currently an Associate Professor in the Department of Electrical & Computer Engineering at The University of Texas, Austin, USA. His research interests include network information theory, wireless systems, and mobile systems.

Sriram received the NSF CAREER award in 2005 and the ARO Young Investigator Award in 2008. He is the co-recipient of the 2005 IEEE Joint Information Theory Society and Communications Society best paper award. He has served as the general chair of IEEE Wireless Network Coding conference (WiNC) in 2010, the general co-chair of the IEEE Information Theory School in 2011, the local arrangements chair of ISIT 2010 and the guest editor-in-chief of IEEE TRANSACTIONS ON INFORMATION THEORY special issue on interference networks.