# Practical Rate Adaptation for Very High Throughput WLANs

Arafet Ben Makhlouf, *Student Member, IEEE,* and Mounir Hamdi, *Fellow, IEEE*

***Abstract*—Wireless Local Area Networks (WLANs) have become increasingly popular due to the recent availability of affordable devices providing multiple and high rate capabilities. Optimizing the performance of WLANs for emerging new Internet applications that demand High Throughput (HT) is an important and a highly challenging issue. In this paper, we present a new practical Rate Adaptation (RA) algorithm, termed *L3S*, which extends legacy schemes with new MIMO features suitable for the forthcoming 802.11n high-speed MIMO-based WLAN products. We have implemented our rate adaptation algorithm in real hardware devices and evaluated its performance and compared it to the existing rate control mechanisms. Our experiments demonstrate that our scheme allows the current 802.11n devices to have a greater adaptability to a variety of wireless channel conditions and performs better than state-of-the-art rate adaptation algorithms.**

***Index Terms*—Rate adaptation, IEEE 802.11n/ac, MIMO, Ath9k, implementation, experimentation.**

## I. Introduction

IN the next generation WLAN standard, termed IEEE 802.11n, new Physical (PHY) and Medium Access Control (MAC) layer enhancements have been introduced. The improvements have given birth to high data rates to keep up with current and upcoming Internet applications. A fundamental problem is that an 802.11 wireless channel suffers from time-varying losses due to mobility, interference and contention from hidden stations, leading to poor and inconsistent throughput performance. In particular, the technique of Rate Adaptation (RA) is a fundamental resource management issue for IEEE 802.11 devices; its goal is to optimize the wireless link throughput in various environments. In addition, all of the IEEE 802.11 standards do not specify any algorithm for automatic rate control. The basic idea of rate control is to estimate the current wireless channel condition and dynamically select the best data rate out of multiple available transmission rates. The wide variety of data rates provided in 802.11n high speed WLANs demands an efficient control scheme for optimal rate selection and also fast rate adaptation based on time-varying channel conditions.

In the literature, there are some RA algorithms designed for this recent standard. But, to the best of our knowledge, many of them are implemented and tested only by using the network simulator NS-2. Moreover, these proposed RA algorithms have not been implemented in practice using existing device drivers so far because they require modifications to the IEEE 802.11 standard. For WLANs, the existing rate control algorithms are simple and have intuitive rules to find the best rate, but they are very slow and their performance degrades in highly variable channels. Therefore, we need a compatible 802.11 standard and a robust algorithm against various dynamics that solves the responsiveness and the stability simultaneously.

In this paper, we design a practical rate control algorithm for 802.11n WLANs, based on a probing system that guarantees that it is has Long-Term Stability and Short-Term Responsiveness (*L3S*). We then implement it in commercial devices using the actual Ath9k driver without modifications to the existing standard. The new rate adaptation classifies transient and sustained changes in the link conditions. Then, it controls both short-term and long-term channel quality variations respectively by monitoring continuously the transmission history and intelligently probing at new data rates that may outperform the current rate. Our proposed rate control algorithm adapts rapidly to these changes by adjusting the efficient transmission rate. Thus, it optimizes the throughput (or delay) performance on a wireless link.

The rest of the paper is organized as follows. In Section II, we briefly introduce the IEEE 802.11n standard and related work. The basic ideas and the practical implementation of the proposed algorithm, *L3S*, using the actual Ath9k driver are presented in the section III in detail. Section IV gives an extensive experimental evaluation of *L3S* compared with the current state-of-the-art existing MIMO rate control scheme used in 802.11n Atheros chipsets. Finally, Section V concludes the paper.

## II. Background and Related Work

### A. IEEE802.11n Standard

The IEEE 802.11n [1] incorporates new features to boost the performance of WLANs. It makes a number of changes to the PHY layer to increase the effective throughput. These improvements are as follows:

- Multiple-Input Multiple-Output (MIMO): Based on OFDM technology [2], the 802.11n devices are able to simultaneously transmit and/or receive data through multiple antennas. Data signals are split into multiple parts. Each independent signal, called a spatial stream, is sent from a transmitter antenna. The current standard allows for a maximum of four spatial streams.
- Guard Interval (GI): This is a period of time added between adjacent OFDM symbols to minimize Inter Symbol Interference (ISI) caused by the effects of multipath. The legacy 802.11a/b/g standards use only a GI of 800ns but 802.11n can operate with 800ns and 400ns.
- Channel Bandwidth: Legacy 802.11 systems use a radio channel spacing that is approximately 20MHz wide, while 802.11n can operate in the 20MHz and 40MHz

Manuscript received May 4, 2012; revised September 16 and December 7, 2012; accepted December 10, 2012. The associate editor coordinating the review of this paper and approving it for publication was C. Xiao.

This research is sponsored by HKUST Grant RPC10EG21.

The authors are with the Department of Computer Science and Engineering, HKUST (Hong Kong University of Science and Technology), Clear Water Bay, Kowloon, Hong Kong (e-mail: {ninoara, hamdi}@cse.ust.hk).

Digital Object Identifier 10.1109/TWC.2013.13.120626

modes. Over two adjacent channels bonded together, the 40MHz operation doubles the data rate.

The 802.11n provides in total 77 Modulation Coding Schemes (MCSs) or transmission rates that vary from 6.5Mbps to 600Mbps at the maximum. Using these high transmission rates leads to unwanted overhead caused by inter-frame gaps, header and acknowledgement (ACK) of each transmitted frame. For this reason, the 802.11n addresses these issues by also making changes in the MAC layer to increase the amount of transmission time. The IEEE 802.11n MAC enhancements are principally based on the reduction of overhead by including a Reduced Inter-Frame Space and aggregation mechanisms. The main idea of aggregation is to bundle multiple frames or ACKs together for a single transmission. To summarize, the 802.11n standard has introduced new PHY and MAC layer enhancements. Mainly, there are two major improvements over legacy devices, which are the employment of MIMO technology and the greater efficiencies of the channel utilization. These improvements have given birth to High Throughput.

### B. IEEE Future Standards

Late in 2008, two new Task Groups, TGac for the IEEE 802.11ac amendment and TGad for the IEEE 802.11ad amendment, were formed with the goal of significantly improving the data throughput of 802.11 standards so that the performance of a wireless network could be equivalent to a wired network. Known as Very High Throughput (VHT), both amendments are currently under development, and their finalizations are anticipated at the end of 2012.

1) IEEE 802.11ad Enhancements: The 802.11ad [3] is an amendment to the 802.11 standard that enables multi-gigabit wireless communication channels by using very wide bandwidth of 2.16GHz in the 60GHz frequency band. It will offer even higher data rates than 802.11ac but only at relatively short distances and where obstructions don't appear in the path. The 802.11ad will therefore be more suitable for cable replacement applications. This new standard will offer data rates up to 7Gbps.
2) IEEE 802.11ac Enhancements: The Initial Technical Specification Draft 1.1 [4] was published by IEEE 802.11 TGac on 20 January 2011. The 802.11ac will use frequencies of 5GHz to provide backwards compatibility with the existing 802.11a and 802.11n devices, which are operating with the same band. The IEEE 802.11ac devices are required to support 20, 40 and 80MHz channels and 1 spatial stream. Several optional features are defined as follows:
   - Channel bandwidths (80+80MHz and 160MHz),
   - Higher modulation support (256-QAM) with advanced coding schemes,
   - Up to 8 spatial streams using SDMA (Space Division Multiple Access),
   - MU-MIMO: for transmitting and receiving of independent data simultaneously,
   - 400ns short guard interval,
   - STBC (Space Time Block Coding).

   However, an 802.11ac device making use of only the mandatory parameters [5] (i.e., 80MHz bandwidth, 1 spatial stream and 64-QAM 5/6) will be capable of a data rate of ~293Mbps, while a device that uses all optional parameters (i.e., 8 spatial streams, 160MHz bandwidth and 256-QAM 5/6 with a short guard interval) will be able to achieve almost 7Gbps. As in 802.11n, the 802.11ac PHY is based on OFDM and will maintain the same modulations, interleaving and coding architecture. But, 802.11ac will add one more modulation option, which is 256-QAM. Also, there is a difference in the number of MCS indices; only 10 MCSs are defined in 802.11ac. Note that for 802.11n, there are in total 77 MCS indices because it supports unequal modulations (i.e., a sender might get BPSK on one stream and 16-QAM on another). In the 802.11ac standard, the decision to only allow equal modulations makes sense for the reason that in practice no IEEE 802.11n devices support unequal modulations, and with the given additional options in 802.11ac (i.e., 256-QAM, 160MHz bandwidth), the number of possibilities would be impractical.

TABLE I
STATISTIC-BASED VS. SIGNAL-BASED APPROACHES

| Statistic-based Approaches | Signal-based Approaches |
|---|---|
| ARF, AARF, ONOE, Sample Rate | CHARM, RBAR, OAR, Goodput Analysis |
| collect transmission statistics | measure the signal strength |
| not require RTS/CTS | may require RTS/CTS |
| no changes to standard | need changes to the standard |
| performance degradation in many cases | good performance (if neglect the overhead) |

### C. Rate Adaptation Algorithms

This section gives a brief description of the various RA algorithms that have been proposed in the recent literature. Based on the Channel State Information (CSI) used for channel quality estimation, the RA algorithms can be grouped into two classes, as follows: The first class contains statistic-based RA algorithms (e.g., ARF [6], AARF [7], SampleRate [8], and ONOE[9]) and the second class comprises signal-based algorithms (e.g., CHARM [10], RBAR [11], OAR [12], and Goodput Analysis [13]). Table I illustrates the classification of some related work. In [14] a hybrid approach was proposed, which uses the measured RSSI (Received Signal Strength Information) and the collected transmission statistics to improve the performance of network multimedia applications. Performance evaluations of some rate control algorithms are available in [, , ]. The legacy 802.11 standards support only open-loop rate adaptation, while the 802.11n standard supports both open-loop and closed-loop rate adaptation schemes to optimize the performance on a point-to-point link. The open-loop schemes are based on implicit feedbacks by monitoring the ACK packets to estimate the link quality between the sender and a specific receiver. On the other hand, the closed-loop schemes are based on explicit feedbacks that are carried in control packets. In this case, the receiver provides instantaneously its estimated CSI or suitable MCS to help the sender to choose the optimal MCS that satisfies both sides.

### D. MIMO Atheros Chipsets and Ath9k Driver

A driver is a piece of software used to communicate with a hardware device. In Linux, the device drivers are added as kernel modules. Ath9k [18] is a completely free and open-source Linux device driver for Atheros chipsets. It is designed to operate with 802.11n WLAN cards. The current Ath9k driver implements most, but not all, functionalities of the 802.11n standard. It can operate on both 2.4GHz and 5GHz bands with both 20MHz and 40MHz modes. Also, it supports up to two spatial streams. Then, the data rates go up to 135Mbps with 20MHz, and they go to 300Mbps with 40MHz. The IEEE 802.11n standard introduces an MCS feedback mechanism, but the current driver does not implement this feature. The Ath9k driver maintains a number of FIFO (First-In First-Out) queues of transmission descriptors with a great deal of detailed control information. Whenever data is received from the network layer, the driver adds the appropriate IEEE 802.11 MAC header with the necessary information and inserts the packet into one of the FIFO queues. According to the transmission descriptor parameters, the PHY layer is responsible for sending the packet over the wireless medium. In Ath9k, Atheros Communications specifies 4 retry series (*r0*/*c0*, *r1*/*c1*, *r2*/*c2*, *r3*/*c3*) for every frame ready to be sent, each pair with a specific rate ri and its maximum transmission attempts ci. Therefore, the lost packet is transmitted at four different decreased rates until a success or an exceeding of the retry limit. Every frame is discarded after *c0*+*c1*+*c2*+*c3* unsuccessful transmission attempts. In addition, the Ath9k driver uses the ONOE rate adaptation algorithm, as mentioned in [19], which sends a constant small fraction (10%) of the data packets at the adjacent rates of the current rate. The throughput calculation is based on the measured PER (Packet Error Rate) that is periodically updated. Finally, based on the status field, the descriptor is updated after the frame has been transmitted successfully or discarded.

## III. Our Algorithm Design

In this paper, we focus on designing a practical transmitter-based algorithm without link knowledge at the sender, which is self-tuning and has a fast response over the time-varying of wireless channel conditions. In particular, our mechanism is an open-loop RA algorithm for MIMO WLANs that does not require any explicit feedback from the receiver; instead it monitors only the binary ACK (implicit feedback). Our algorithm, termed *L3S*, is based on the following observations:

- The wireless interfaces provide the signal strength measurements to the upper layer. In practice, they are uncalibrated and only for a very limited area. Therefore, it is difficult to obtain an instantaneous and a reliable estimate of the SNR of a link. Moreover, there is no explicit method to exchange the RSSI between the stations. For these reasons, the RSSI measurements cannot be helpful for the stability and responsiveness of the algorithm.
- The static nature of some previously proposed RA algorithms with static thresholds and timers (e.g., ARF, ONOE, etc.) are application dependent and are generally fixed at the design time. For this reason, they are less versatile for very dynamic link conditions.
- In practice, it is very difficult to accurately differentiate between collision and wireless losses. The current MIMO RA algorithm, embedded in the Ath9k driver for 802.11n products, probes with only one attempt, which makes it sensitive to the single failure of a probe packet. Hence, the RA algorithms must be robust against collision losses.
- The current drivers for WLAN devices are quite slow in adapting to the time-varying channel conditions. The RA algorithm should be quickly adaptive and accurate in rate adjustment to utilize the channel efficiently, which means it is able to immediately jump to the rate that may yield the best performance.

In 802.11n WLANs, the choice of the MCS has a direct impact on the throughput performance. A static transmission (i.e., with a fixed number of antennas) cannot maximize the link's capacity due to the rapid changes in the wireless channel state. It has been shown in [19] that a dynamic transmission strategy is needed to improve the network's performance by taking advantage of spatial diversity. Then, the goal of link adaptation is to choose the suitable number of spatial streams that maximizes the throughput or minimizes PER, which are directly related to the SNR in the channel and gives precise link state information between each two peers. The signal-based algorithms use signal strength measure-ments, which are provided immediately by the radio interface, to select the optimal transmission rate. Then, the sender does not need to collect feedback statistics and use probe packets for channel quality estimation. In spite of these advantages, they have not been applied in practice so far. The main reason is due to the difficulty to obtain reliable and accurate estimates of the channel quality. Most wireless network interface cards provide un-calibrated RSSI (i.e., not instantaneous). In addition, there is no available method within the 802.11 standards to provide the CSI of the receiver to the sender. Finally, the signal quality measured at the sender may not be exactly the same at the receiver. However, the sender can use a higher level modulation for transmission but, on the other side, the received signal cannot be well decoded. Thus, we have chosen that our rate adaptation algorithm maintains present channel transmission statistics (achieved throughput, acknowledged packets, and PER) for each used transmission rate. These statistics provide an elegant way to obtain the necessary CSI feedback for link state estimation. In our proposed algorithm, the collected statistics at the sender are divided into two categories for the best adaptation to the time-varying channel characteristics. The Long-term and the Short-term Statistics are continuously updated by monitoring the transmission results. They are reset whenever the principal rate is changed. The scheme also decides how long the sender should stay at the current transmission rate, when and how to probe at candidate rates, and whether or not to switch to a new rate.

### A. Short-term Statistics for System Responsiveness

The Short-term Statistics are used to control the transmission rate against transient variations that improve the system responsiveness. A wireless link usually behaves diversely and does not follow a fixed success or failure pattern. In this case, consecutively received and lost ACK frames may respectively reflect fast improving and deteriorating channel conditions in a short interval. However, the sender should, before a probe, justify whether there are short-term or long-term variations in the channel quality. Thus, the short-term changes are handled by a multi-rate retry mechanism. To incorporating these, we use the counter *consecutive_retries* for each retry rate initialized in the transmission descriptor, which denotes the number of consecutive packets that fail at the attempts. Also, the ACK packets provide binary information to the sender as to whether its choice of the MCS was supported by the channel. Similar to the ARF-like algorithms in the SISO case, our sender maintains *consecutive_successes* and *consecutive_failures* counters, which are used to dynamically update the beginning of the next probing period. They represent the total number of consecutive packets that succeed and fail at all of the attempts. These two counters tackle the long-term variations by calling the probe function to change the long-term rate immediately or after a period of time. Their thresholds are respectively 10 and 2 similar to other ARF algorithms. However, if a sender receives 10 consecutive ACKs, then the channel condition is stable and it delays the probing. When a sender misses 2 consecutive ACKs, it falls back immediately to the previous rate. Finally, if there are 4 missed consecutive ACKs, then the channel quality is degrading and the sender accelerates the probing.

### B. Long-term Statistics for System Stability

The Long-term Statistics are maintained to adapt the transmission rate, which provides the best throughput, against sustained changes in the link. The potential best rate is determined by estimating the observed throughput performance, whether it has been used in the past transmissions or by the adjustment that enhances the system stability. Long-term Statistics (packet error rate, achieved throughput) within a window of a certain number of packets can decide whether a probing rate can be selected. These statistics are related to the effective throughput, and then they guarantee that this throughput is maximized in the long-term. Thus, our proposed algorithm probes intelligently by sending data packets at the candidate rates that may improve the performance. Based on PER, the expected throughput is calculated to gauge the performance on the currently used rate. In order to have an advanced and efficient rate selection that instantaneously estimates the channel quality, the long-term transmission rate is not changed during the probing, and its throughput performance is evaluated with that of the probing rates. With the throughput optimization goal of rate adaptation, a probe rate that outperforms the current long-term rate must replace it. As long as the sender observes that the selected rate does not perform as well as expected, it falls back immediately to the pervious MCS. Specifically, our scheme continually examines the long-term data rate during the transmission periods and can tune to the channel variations as soon as possible.

In the SISO (Single-Input Single-Output) case, the decision is simply to increase or decrease the scaling by one, which may lead to premature rate switch decisions. However, in the MIMO technique case, the additional transmitter antenna(s) can be used for diversity gain. Specifically, transmitting a single data stream across multiple antennas can greatly improve the transmission reliability. The *probe( )* function, which intelligently probes at the possible probing directions, is periodically invoked. The probing rates are adjusted according to the multi-rate retry series mechanism. Figure 1 shows the six proposed probing directions, which correspond to these different following functions: *Down_Probe( )*, *Up_Probe( )*, *Right_Probe( )*, *Left_Probe( )*, *Right_Down_Probe( )* and *Left_Up_Probe( )*. These different functions are used to increment or decrement the current rate. In particular, all main functions used in our algorithm are described in Table II.

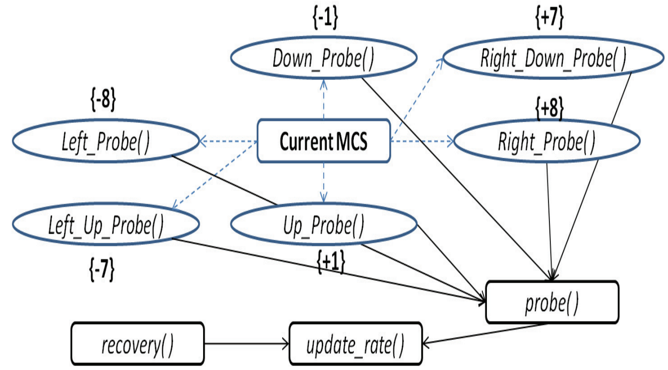



Fig. 1. Different probing directions.

TABLE II
DESCRIPTION OF DIFFERENT USED FUNCTIONS

| Function | Signal-based Approaches |
|---|---|
| rate_statistics( ) | Update the different transmission or probe statistics |
| get_rate( ) | Calculate the expected throughput and get a new rate |
| probe( ) | Fix and probe at the possible probing directions |
| recovery( ) | Fallback immediately to the previous transmission rate |
| update_rate( ) | Update the transmission rate and reset statistic counters |

In our algorithm, the short term changes in the link quality are controlled by multi-rate retry series, while the long term variations are manipulated by changing the values of *ri*. However, it keeps the mechanism of multi-rate retry with four retry series but modifies the values of *ci*'s. Thus, this mechanism is used to solve transient channel variations by retransmitting the lost packet at different data rates, which are always in decreasing order, until a there is a success or exceeding of the retry limit. The successive retry values of the four multi-rate retry series are important to responsiveness performance. A high value makes the second and later attempts using the same rate as the first, which may cause additional retransmissions where the channel condition is really degrading. On the other hand, a small value may cause a transmission at a reduced rate fairly quickly, and it is very probable that the fail transmission is caused by a collision. The multi-rate retry series mechanism is used during the transmitting (*Tx* state) and the probing (*Probe* state) periods. For both states and with more than one attempt, we do not rush to a reduced rate once the first attempt fails. Therefore, we set *c1* to two, which means that the first and second attempts use the *long-term-rate* in the case of the transmission period or the *probe-rate* in the gauging period. However, rate decrement is possible when consecutive failures

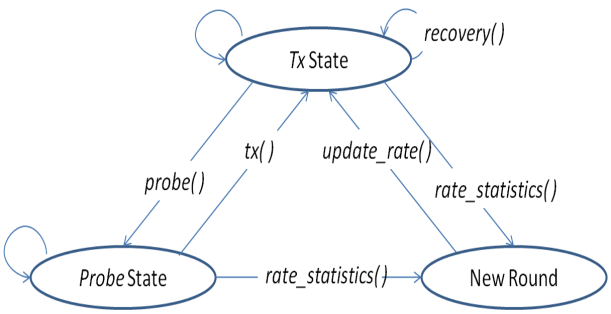



Fig. 2. State transitions at the sender.

TABLE III
DESCRIPTION OF DIFFERENT USED FUNCTIONS

| Modulation Type | Existing RA Algo. | | Proposed RA Algo. | |
|---|---|---|---|---|
| | 1 stream | 2 streams | 1 stream | 2 streams |
| BPSK | 13.50 | - | 13.50 | 27.00 |
| QPSK | 27.00 | - | 27.00 | 54.00 |
| QPSK | 40.50 | - | 40.50 | 81.00 |
| 16-QAM | 54.00 | 108.00 | 54.00 | 108.00 |
| 16-QAM | 81.00 | 162.00 | 81.00 | 162.00 |
| 64-QAM | - | 216.00 | 108.00 | 216.00 |
| 64-QAM | - | 243.00 | 121.50 | 243.00 |
| 64-QAM | - | 270.00 | 135.00 | 270.00 |

occur, which is, with a high probability, caused by imperfect channel quality rather than collisions. Similarly, *c2* and *c3* retry limits are set to two to ensure a fast adaptation to our short-term channel variations. Whereas, the first rate series *r0* with four as the successive retry limit, is always used to send the different control packets (i.e., RTS, CTS, and ACK) at the lower MCS with a single antenna before data transmissions. The driver, Ath9k, uses a basic MIMO mode where the same modulation and coding schemes are employed for all spatial streams of an MCS. Table III shows the basic MCS set, with a maximum of two spatial streams for 40 MHz and GI of 800ns, which can be supported by the current driver. For this existing MIMO rate control, the used transmission rates are sorted, which is similar to the rate control algorithms in the SISO case. This strategy makes the adaptation very slow in the presence of random wireless errors. Eventually there are too many packet losses before it finds a proper rate, for the reason that it can only step down to the next lower rate, for each interval. Whereas, our *L3S* algorithm takes advantage of all the available MCSs in the IEEE 802.11n standard. Thus, it will use all the data rates listed in Table III.

In our scheme, the long-term transmission rate *tx_rate* for each station is adapted according to the sender's state. The state transitions and the used functions are illustrated in Figure 2. Specifically, a sender has two states, the *Tx* state and the *Probe* state. In each round, the transmitter moves periodically between these two states and updates continuously the associated statistics. When the sender adjusts the long-term MCS, the statistics of both periods are reset, which means the commencement of a new round. The rate selection is controlled by the function, *get_rate( )*, which decides the rates, *ri*, and their corresponding retry numbers, *ci*, for a new outgoing packet. At the beginning of each round, the sender is in the *Tx* state and transmits data at *r1 = tx_rate*. In the same way as in ONOE, the remaining rates, *r2* and *r3*, of multi-rate retry mechanism are filled with the next successive lower rates. Meanwhile, the sender observes the transmission results of the current rate. Specifically, it calculates the throughput by monitoring the PER. In practice, the average collision rate for TCP is around 11%, so the PER values less than this threshold are ignored. After a period of time, the sender sets up two successive probe series and enters into the Probe state, but the *tx_rate* is not altered. Similarly, it keeps on observing the probing results at the candidate rate. The first series is employed to probe with the current used spatial stream(s), where a constant fraction (10%) of the data is sent at the two adjacent rates of the long-term rate. Therefore with the multi-rate retry scheme *r1 = Up_Probe( )*, *r2 = tx_rate* and *r3 = Down_Probe( )*. On the other hand, the probing in the second series is based on the number of streams currently used. Using the same method, another window of ten percent of the packets is sent at other candidate rates that are sorted, as always, in decreasing order. Thus, if the sender transmits with only one spatial stream, we call the two functions *Right_Probe( )* and *Right_Down_Probe( )*, which means an additional transmitter antenna is used for the following queued packets. Otherwise, the called functions are *Left_Up_Probe( )* and *Left_Probe( )*. For the cases, the long-term transmission rate *tx_rate* is kept in the multi-rate retry series during the probing periods. After these two probe series, the sender goes back to *Tx* state. In the *update_rate( )* procedure, the performances of all five rates are evaluated and the best rate is selected as a new long-term MCS, which is critical to the achieved throughput. In particular, the *recovery( )* function describes when the sender is needed to fall back immediately to the previous long-term MCS. After each long-term transmission rate adjustment, the next probing time, *probe_interval*, is set to the default period of 60ms.

The trigger of the next probing period is dynamically changed depending on present channel characteristics. The feature of dynamic probing time can further improve the TCP throughput performance by decreasing the wastage of time on retransmissions. The sender collects transmission statistics in the function *rate_statistics( )*. After a frame transmission, the returned hardware status in the descriptor is either successes or retries, which is used to decide when the next probe state will be triggered. Note that the transmission successes status does not necessarily mean that only the first attempt is successful, but the frame may be successfully transmitted after several retries. The retries status means that the attempt has failed. *Our_Algo* below provides the pseudo code description and the detailed probing process of the proposed rate control scheme.

## IV. PERFORMANCE EVALUATION

### A. Experiment Settings

For the reasons that 802.11ac/ad devices are not yet available on the market, we test our rate selection mechanism using 802.11n devices. However, the implementation is done in the Ath9k [15] driver designed for Atheros chipsets. We consider the network topology shown in Figure 3. The different

**Our_Algo: Proposed Rate Adaptation Algorithm**

```
update_rate( ) function:
{commencement of a new round, reset the statistic counters}
tx_rate = new_rate;
consecutive_successes = 0;
consecutive_failures = 0;
consecutive_retries[ ] = 0;
if rate increase then
   probe_interval = 20;
end if

rate_statistics( ) function:
update try consecutive_retries[ ] counter for each rate
update PER counter of each rate PER[ ];
if receiving an ACK then
   consecutive_successes + +;
   consecutive_failures = 0;
   consecutive_retries[cur_rate] = 0;
   if (consecutive_successes = = 10) then
      probe_interval = 90;
end if
else if missing an ACK then
   consecutive_successes = 0;
   consecutive_failures + +;
   consecutive_retries[cur_rate] + +;
   if (consecutive_failures == 4) then
      probe_interval = 10;
end if
end if
if (missing two consecutive ACKs) then
{fallback immediately to the previous tx_rate}
recovery( );
update_rate( );
probe_interval = 30;
end if

get_rate( ) function:
calculate the best expected throughput from PER statistics;
update_rate( );
if (probe( ) == true) then
{calculate all possible probing links by calling the different
probe functions}
{fill the first probe series; return the different MCSs for the
first probe series in the multi-rate retry mechanism; possible
values are probe = +1 and -1}
   if (first_probe == false) then
      rix1 = Up_Probe( ); try = 2;
      rix1 = Up_Probe( ); try = 2;
      rix3 = Down_Probe( ); try = 2;
      first_probe = true;
      probe_interval = 10;
   else
      first_probe = false;
      probe_interval = 60;
      {fill the second probe series; return the different
       MCSs for the second probe series; possible
       values are probe = +7 and +8 or -7 and -8}
      if (tx_rate >= 0 && tx_rate <= 7) then
         rix1 = Right_Probe( ); try = 2;
         rix2 = Right_Down_Probe( ); try = 2;
         rix3 = tx_rate; try = 2;
      else if (tx_rate >= 8 && tx_rate <= 15) then
         rix1 = tx_rate; try = 2;
         rix2 = Leftt_Up_Probe( ); try = 2;
         rix3 = Left_Probe( ); try = 2;
      end if
   end if
else if (tx( ) == true) then
   {send control packets using the 1st rate series}
   rix0 = Lower_Rate( ); try =4;
   {return the different MCSs for the transmission
   Tx state; Three consecutive rates in decreasing
   order are used}
   rix1 = tx_rate; try = 2;
   rix2 = Down_Rate( ); try = 2;
   rix3 = Down_Rate( ); try = 2;
end if
```

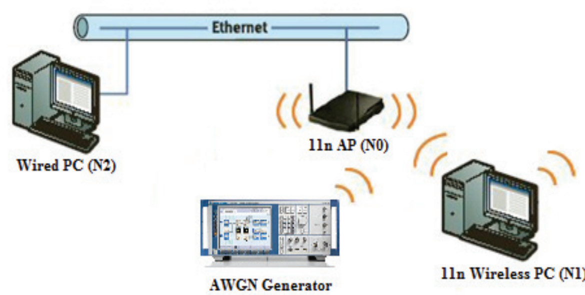



Fig. 3. Network topology in the experiments.

hardware and software configuration parameters used in our experiments are listed in Table IV. The Access Point (*N0*) and the PC (*N1*) wireless card adopt "2x2" MIMO technology for data rates of up to 300Mbps. In order to make the scenario more realistic, we use a tunable waveform generator that allows injecting of Adaptive White Gaussian Noise (AWGN) at different power levels and a desired frequency band. AWGN is commonly used to simulate background noise of the channel under study, in addition to multipath, terrain blocking, interference, ground clutter and self interference that modern radio systems encounter in terrestrial operation. However, in our experiments, the distance between two devices and their mobility variations are adjusted by changing the noise power created by the AWGN generator. To increase the distance between a sender and a receiver, then we should increase the noise power in the channel. TCP (Transmission Control Protocol) provides reliable, ordered delivery of streams and it is the protocol used by major Internet applications. Since, TCP uses processes to check that the packets are correctly sent to the receiver, we use Netperf [20] to generate continuous saturated TCP packets and report the achieved throughput for each network link between the client (*N1*) and the server (*N2*). All the equipment used are placed in a typical office with several concrete obstacles, and then the paths between the AP and the client PC may not be line-of-sight.

We run experiments with a static client but in different channel conditions by changing the SNR. To get RSSI variations such as in mobility cases, we modify the noise power level that is injected by the AWGN generator. Every experiment is run for about 20 minutes. The throughput results for each RSSI value are grouped together and a statistical average measure is reported over more than three experimental runs. For our performance evaluation, we have designed three different experimental scenarios. In scenarios *S1* and *S3*, the injected noise is, respectively, varied slowly in decreasing order and rapidly in increasing and decreasing order, at the sender area. For the scenario *S2*, we decrease slowly the SNR at both client and AP sides.

### *B. Experiment Results*

We implement of our RA algorithm on a real 802.11n wireless card based on the Atheros AR9220 chipset [21]. To

TABLE IV
CONFIGURATION PARAMETERS

| Parameters | Values |
|---|---|
| Signal Generator | Rohde&Schwarz SMU200A |
| WLAN Card | TL-WN851N (2Tx/2Rx) |
| WLAN Router | TL-WN841N (2Tx/2Rx) |
| Operating System | Linux kernel 2.6.32.12 |
| Device Driver | Ath9k |
| Wireless PHY | 802.11n |
| Channel Bandwidth | 40 MHz |
| Transmit Power | 20 dBm |
| Frequency Band | 2.427 GHz |

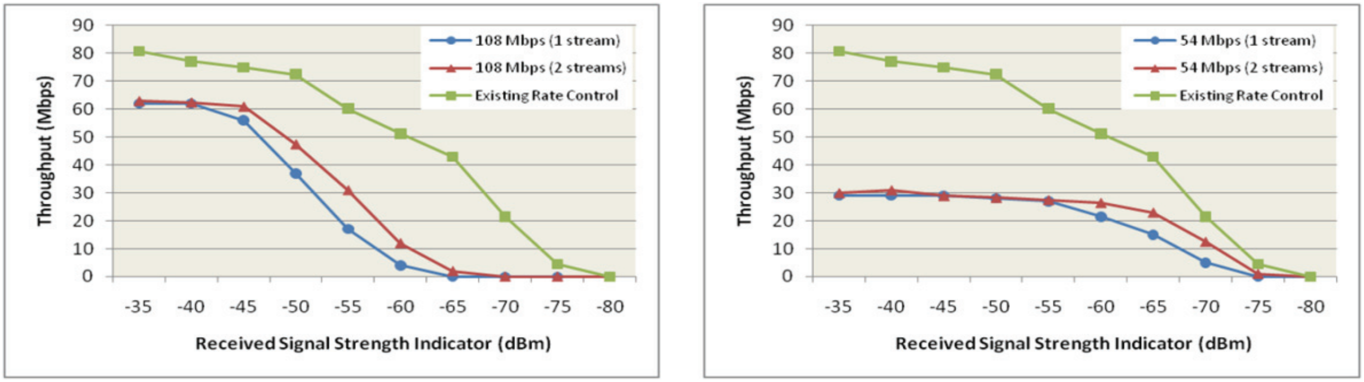
analyze its performance compared to the scheme embedded in the Ath9k driver, we carry out extensive experiments under various link conditions. So, we measure the stability (by average throughput) and the responsiveness (by response time). Stability means the ability to maintain a rate close to the optimum rate. Thus, the RA algorithm can achieve the desired performance reference and it is robust to disturbance. On the other hand, responsiveness measures how fast RA takes for the algorithm output to achieve the new desired rate in reaction to channel changes. It reflects the ability to adapt to quick channel variations in loss rate.

At first, and in order to check the effects of the improvement reached on a MIMO system by the additional transmitter antennas, we study the TCP throughput performance of the last existing Atheros MIMO RA with static rates via a different number of spatial streams (i.e., antennas). Figure 4(a) shows the throughput achieved by the transmission rate 108Mbps across one and two antenna(s) with different RSSI values. So as to verify again the effects of an extra antenna, we repeat the same experiment but with a transmission rate of 54Mbps. The throughput achieved is shown in Figure 4(b). The throughputs achieved by these two fixed rates are included for performance benchmarks. For both figures, initially, when the channel condition is good, the transmissions with a single stream and the transmissions with double streams can achieve similar optimal throughput. However, as the channel condition deteriorates, the throughput achieved by using one antenna reduces faster than the throughput achieved by simultaneous transmissions. The reason is attributable to the signal reflection and multipath that improve the received signal strength and maintain higher throughput. All these point out the fact that transmitting a single spatial stream across two antennas at the same time can greatly improve the reliability. Also, Figures 4(a) and 4(b) show that the gain in throughput achieved by the existing RA is extremely high compared to that achieved by a fixed transmission rate. This proves that the automatic rate selection capability in IEEE 802.11 devices is very important to deal with varying channel conditions.

Now, we study the throughput performance of the two algorithms under different scenarios. In all scenarios, our new RA algorithm achieves the highest throughput by managing to send more packets at higher rates, mainly due to the intelligent and dynamic probing. The comparative throughput, between the existing and our new RA algorithm, with varying RSSI is shown in Figure 5. In Figure 5(a), where the link between the client and the AP is stable and with slow RSSI variations, both algorithms can initially achieve similar high throughput when the channel condition is good (RSSI > -45). But, when the SNR increases, the throughput achieved by the existing scheme reduces faster than that of our algorithm. The main reason is attributable to the several failed transmissions and the longer time spent to reach the best data rate. The gain in throughput achieved is almost ~20%. This is also true in Figure 5(b) where the channel condition worsens but with a 15% gain. This figure demonstrates that the TCP throughput achieved by our proposed RA algorithm significantly outperforms that achieved by the existing Atheros MIMO RA scheme, which can hardly attain the threshold to scale up by one. However, our rate control can respond to the variations rapidly enough by probing intelligently at the candidate rates and jumping immediately to the new long-term data rate. Also, the feature of dynamic probing time can improve more the TCP throughput performance by decreasing the wastage of time, as shown in Figure 5(a) and Figure 6. This is due to fact that our new scheme probes quickly when it observes an improving or deteriorating channel by using the short-term statistics. On the other hand, the existing RA spends much more time before reaching the best MCS. It waits 10 seconds at each MCS before scaling to the next upper rate. Another feature of our proposed scheme is its rate stability by using long-term statistics. In our strategy, the dominant factor to decide a rate change is the expected throughput or, equivalently, the expected PER. In the probing periods, the long-term rate is not changed, since it may be suboptimal under some transient channel conditions. As long as the sender observes that the selected rate does not perform as well as expected, it falls back immediately to the previously used long-term MCS. Specifically, our scheme continually examines the long-term MCS also during the transmission periods and can tune to the channel variations as soon as possible. However, the different transient variations are resolved by using the multi-rate retry mechanism. Indeed, if it selects a wrong data rate to probe, it can perceive this error after two consecutive failed attempts and continue to transmit and test simultaneously the next lower MCS. Under favorable link conditions, the transmit queue is almost empty, resulting in less packet delay.

However, with link quality degrading, the number of packets in the queue grows, which is caused by the retransmissions. The goal of the rate adaptation algorithm is to select instantaneously the best MCS and transmit the packets back to back until the queue becomes empty again. In this section, we compare the responsiveness of our algorithm with the existing rate control, as shown in Figure 6. It is quite clear from this figure that our scheme responds to the variations quickly enough and especially when the link quality decreases suddenly. Indeed, our sender predicts and tunes to the channel conditions rapidly with little delay, utilizing both the long-term and the short-term statistics. Under high varying channels, the new RA algorithm still achieves better performance than the existing Ath9k scheme mostly with dynamic probing time. The reason is attributable to our proposed fast and efficient rate selection strategy.

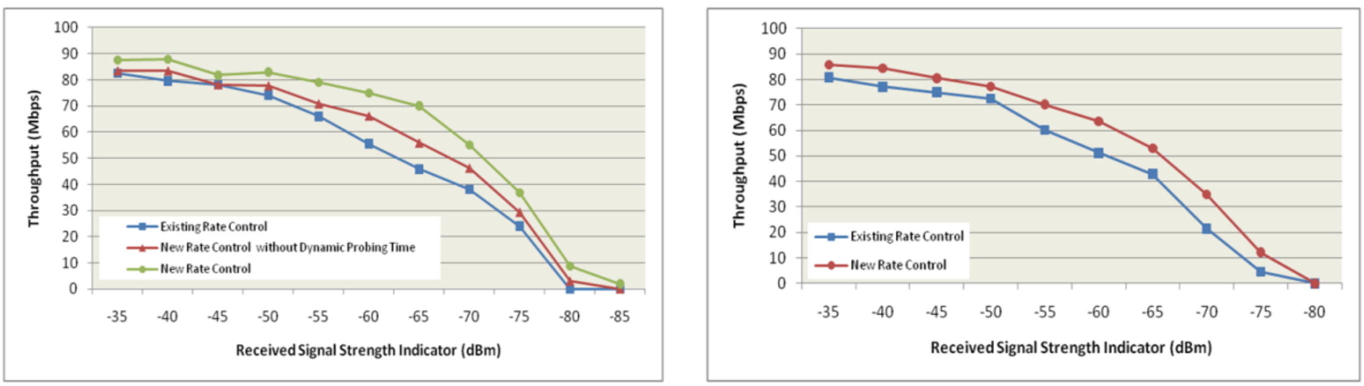
The closed-loop mechanism assumes that perfect channel knowledge is available at the transmitter, either through ex-




(a) Throughput Achieved by 108 Mbps with Different Number of Spatial Stream

(b) Throughput Achieved by 54 Mbps with Different Number of Spatial Stream

Fig. 4. Throughput comparison of fixed rates for TCP traffic under scenarios S1




(a) Comparative throughput under scenario S1

(b) Comparative throughput under scenario S2

Fig. 5. Throughput comparison for TCP traffic on two different scenarios.

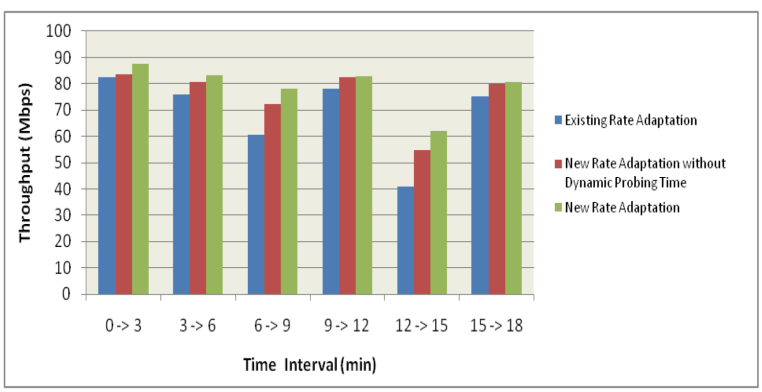



Fig. 6. TCP throughput comparison under scenario S3 (RSSI changes quickly for each period as follows: -35 =>-55 => -70 => -45 => -75 => -35).

plicit feedback from the receiver using specific control frames or through channel sounding and calculation between the transmitter and the receiver. However, this approach incurs a lot computation complexity and communication overhead. On the other hand, our mechanism is an open-loop rate adaptation algorithm that does not require any explicit feedback from the receiver; instead it monitors only the binary ACK (implicit feedback). Also, the current Ath9k algorithm and our new algorithm send a constant fraction of the data packets at the adjacent rates of the current rate using multi-rate retry series. The calculation of throughput is based respectively on the measured PER and RSSI, which are periodically updated. Then, both algorithms have the same computational complexity, but our rate control scheme is highly responsive to time-varying link conditions due to the intelligent and dynamic probing.

## V. Conclusions

IEEE 802.11 is the most popular standard used for WLANs today. IEEE 802.11n is a recent amendment for supporting many high transmission rates by introducing enhancements over the legacy 802.11 a/b/g standards. Many factors make the wireless channel very dynamic. As a result, rate adaptation is extremely important to the throughput performance of WLANs with multi-rate capabilities.

In this paper, we have proposed a new sender-based rate adaptation algorithm, which controls both short-term and long-term channel quality variations. The novelty of our scheme lies in its great adaptability to a variety of link conditions. The experiments on typical 802.11n networks show that it is capable of achieving noticeable improvements in throughput compared to the present MIMO RA implemented in Ath9k drivers. The reason is attributable to our proposed strategy of quick rate adaptation and mostly the intelligent and efficient rate selection methods we use. Finally, our new RA algorithm is still interoperable with the legacy standards and can be easily adopted by recent wireless hardware.

## References

[1] IEEE Draft Std P802.11n/D3.01, Draft Standard for Information Technology Telecommunications and Information Exchange between Systems-Local and Metropolitan Area Networks-Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment4: Enhancements for Higher Throughput, Std., Nov. 2007.

[2] B. Bjerke, J. Ketchum, R. Walton, S. Nanda, I. Medvedev, M. Wallace, and S. Howard, "Packet error probability prediction for system level simulations of MIMO-OFDM based 802.11n WLANs," in *Proc. 2005 IEEE ICC*, pp. 2538–2542.

[3] Wireless Gigabite Alliance White Paper, "Defining Future of Multi-Gigabit Wireless Communications," July 2010.

[4] IEEE 802, "IEEE P802.11ac/D1.1," Institute of Electronic Engineers, Aug. 2011.

[5] HT PHY Specification, Enhanced Wireless Consortium v1.27. Available: http://www.hydrus.et.put.poznan.pl/~rkotrys/802_11/EWC_PHY_spec_V127.pdf

[6] A. Kamerman and L. Monteban, "WaveLAN-II: a high-performance wireless LAN for the unlicensed band," *Bell Labs Technical J.*, pp. 118–133, Summer 1997.

[7] M. Lacage, M. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: a practical approach," in *Proc. 2004 ACM MSWiM*, pp. 126–134.

[8] J. C. Bicket, "Bit-rate selection in wireless networks," Master's thesis, Feb. 2005.

[9] ONOE Rate Control. Available: http://www.madwifi.org/browser/madwifi/trunk/ath_rate/onoe

[10] G. Judd, X. Wang, and P. Steenkiste, "Low-overhead channel-aware rate adaptation," in *Proc. 2007 ACM MobiCom*, pp. 354–357.

[11] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," in *Proc. 2001 ACM MOBICOM*, pp. 236–251.

[12] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "OAR: an opportunistic auto-rate media access protocol for ad-hoc networks," *Wireless Networks*, vol. 11, pp. 39–53, Jan. 2005.

[13] D. Qiao, S. Choi, and K. Shin, "Goodput analysis and link adaptation for IEEE 802.11a wireless LANs," *IEEE Trans. Mobile Computing*, vol. 1, no. 4, pp. 278–292, Oct.-Dec. 2002.

[14] I. Haratcherev, K. Langendoen, R. Lagendijk, and H. Sips, "Hybrid rate control for IEEE 802.11," in *Proc. 2004 ACM MobiWac*, pp. 10–18.

[15] S. Pal, S. R. Kundu, K. Basu, and S. K. Das, "IEEE 802.11 rate control algorithms: experimentation and performance evaluation in infrastructure mode," in *Proc. 2006 PAM*.

[16] Y. Yang, M. Marina, and R. Bagrodia, "Experimental evaluation of application performance with 802.11 PHY rate adaptation mechanisms in diverse environments," in *Proc. 2006 IEEE WCNC*, pp. 2273–2278.

[17] K. Ramachandran, H. Kremo, M. Gruteser, I. Seskar, and P. Spasojevic, "Experimental scalability analysis of rate adaptation techniques in congested IEEE 802.11 networks," in *Proc. 2007 IEEE WoWMoM*, pp. 1–12.

[18] Ath9k 802.11n Drivers. Available: http://www.wireless.kernel.org/en/users/Drivers/ath9k/

[19] A. Roy and N. Saxena, "A new channel aware rate adaptation in high speed WLANs," *IEICE Trans. Commun. Lett.*, vol. E92.B, no. 6, pp. 2345–2348, June 2009.

[20] Netperf, The Networking Benchmark. Available: http://www.netperf.org/netperf/

[21] Atheros Communications. Available: http://www.atheros.com/news/AR9220.html/

**Arafet Ben Makhlouf** (S'08) received his B.S. degree in Computer Science from the Faculty of Sciences of Sfax in 2006 and the M.Phil. degree in Computer Science and Engineering from the Hong Kong University of Science and Technology, Hong Kong in 2010. He is currently working towards the Ph.D. degree in Computer Science and Engineering from the Hong Kong University of Science and Technology, Hong Kong. His research interests include cross layer design and optimization, link adaptation, scheduling, resource management for the IEEE802.11 WLANs, and quality of service provisioning.

**Mounir Hamdi** (S'90-M'91-SM'06-F'11) is a Chair Professor at the Hong Kong University of Science and Technology, and the Head of the Department of Computer Science and Engineering. He is an IEEE Fellow for *contributions to design and analysis of high-speed packet switching*. He received the B.S. degree in Electrical Engineering - Computer Engineering minor (with distinction) from the University of Louisiana in 1985, and the MS and the PhD degrees in Electrical Engineering from the University of Pittsburgh in 1987 and 1991, respectively. He has been a faculty member in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology since 1991, where he was a founding member of the University and the Department. He is now the Head and Chair Professor of the Department that has around 900 undergraduate and graduate students. He was the Director of the Computer Engineering Program (for 10 years) that has around 330 undergraduate students, Director of the Master of Science in Information Technology (for 3 years) which has more than 150 graduate students, and Director of the Computer Engineering research Lab and the Cyberspace Center. He is/was a member of the University Senate, University Council, and University Administration Committee. In 1999 to 2000 he held visiting professor positions at Stanford University, USA, and the Swiss Federal Institute of Technology, Lausanne, Switzerland. His general area of research is in high-speed wired/wireless networking in which he has published around 300 research publications, received numerous research grants, and graduated more 30 graduate students. In addition, he has frequently consulted for companies and governmental organizations in the USA, Europe and Asia on high-performance Internet routers and switches as well as high-speed wireless LANs. He is a frequent keynote speaker in International Conferences and Forums.

Prof. Hamdi is/was on the Editorial Board of various prestigious journals and magazines including IEEE TRANSACTIONS ON COMMUNICATIONS, *IEEE Communication Magazine*, *Computer Networks*, *Wireless Communications and Mobile Computing*, and *Parallel Computing*, as well as a guest editor of *IEEE Communications Magazine*, guest editor-in-chief of two special issues of IEEE JOURNAL ON SELECTED AREAS OF COMMUNICATIONS, and a guest editor of *Optical Networks Magazine*. He has chaired more than 20 international conferences and workshops including The IEEE International High Performance Switching and Routing Conference, the IEEE GLOBECOM/ICC Optical networking workshop, the IEEE ICC High-speed Access Workshop, and the IEEE IPPS HiNets Workshop, and has been on the program committees of more than 200 international conferences and workshops. He was the Chair of IEEE Communications Society Technical Committee on Transmissions, Access and Optical Systems, and Vice-Chair of the Optical Networking Technical Committee, as well as member of the ComSoc technical activities council. He received the best paper award at the IEEE International Conference on Communications in 2009 and the IEEE International Conference on Information and Networking in 1998. He also supervised the best PhD paper award amongst all universities in Hong Kong. In addition to his commitment to research and professional service, he is also a dedicated teacher and renowned quality-assurance educator. He received the best 10 lecturers award (through university-wide student voting for all university faculty held once a year), the distinguished engineering teaching appreciation award from the Hong Kong University of Science and Technology, and various grants targeted towards the improvement of teaching methodologies, delivery and technology.