MBE: Model-Based Available Bandwidth Estimation for IEEE 802.11 Data Communications

Zhenhui Yuan, *Student Member, IEEE*, Hrishikesh Venkataraman, *Member, IEEE*, and Gabriel-Miro Muntean, *Member, IEEE*

***Abstract*—Wireless bandwidth estimation is a critical issue for quality-of-service (QoS) provisioning in IEEE 802.11 wireless local area networks (WLANs). Current bandwidth estimation solutions focus on either probing techniques or cross-layer techniques and require either significant bandwidth resources or protocol modifications. To alleviate these problems, this paper proposes an analytical model-based bandwidth estimation algorithm (MBE) for multimedia services over IEEE 802.11 networks. The MBE module for available bandwidth estimation is developed based on novel transmission control protocol/user datagram protocol throughput models for wireless data communications. The novel aspects in comparison with other works include the fact that no probing traffic is required and that no modification of medium access control (MAC) protocol is needed. Extensive simulations and real tests were performed, demonstrating that MBE has very good bandwidth estimation results for content delivery in conditions with different packet sizes, dynamic wireless link rate, and different channel noises. Additionally, MBE has lower overhead and lower error rate than other state-of-the-art bandwidth estimation techniques.**

***Index Terms*—Bandwidth estimation, IEEE 802.11, model.**

I. Introduction

RECENTLY, an increasing number of rich media applications exchange data over IEEE 802.11 wireless local area networks (WLANs). Bandwidth estimation schemes have widely been used to improve the quality of service (QoS) of multimedia services [1]. Shah *et al.* [2] utilize a novel bandwidth estimation algorithm and propose an admission control-based resource management approach to provide fairness of existing traffic. Li *et al.* [3] develop a playout buffer and rate optimization algorithm to improve the performance of video streaming service. The basic idea is to optimize the streaming bit rate and initial buffer size based on the estimated wireless bandwidth. Efficient bandwidth estimation scheme is also significant for adapting the data transmission rate to the available bandwidth [4]–[6]. In [7], it is shown that the awareness of network resources can benefit the proposed QoS negotiation scheme that allows users to dynamically negotiate the service levels required for their traffic and to reach them through one or more wireless interfaces.

Manuscript received August 17, 2011; revised November 29, 2011 and February 3, 2012; accepted February 28, 2012. Date of publication March 13, 2012; date of current version June 12, 2012. This work was supported by the China Scholarship Council and in part by Lero - the Irish Software Engineering Research Centre (www.lero.ie) through the Science Foundation Ireland grant 10/CE/I1855. The review of this paper was coordinated by Dr. T. Taleb.

The authors are with the Performance Engineering Lab, Network Innovations Centre, School of Electronic Engineering, Dublin City University, Dublin, Ireland (e-mail: yuanzh@eeng.dcu.ie; hrishikesh@eeng.dcu.ie; munteang@eeng.dcu.ie).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2012.2190760

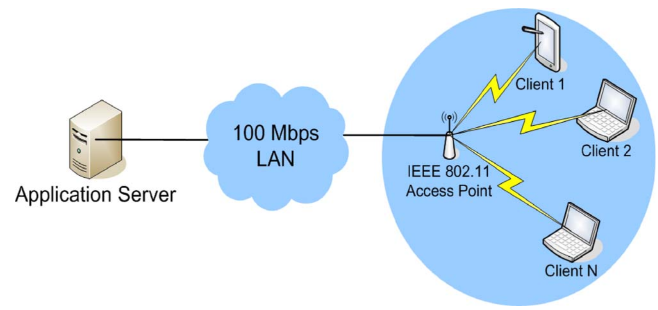


Fig. 1. Network architecture of wireless bandwidth estimation.

Many bandwidth estimation techniques have been proposed to provide estimations in wired networks, such as Spruce [8], Pathload [9], pathRate [10], pathChirp [11], IGI/PTR [12], SProbe [13], etc. However, bandwidth estimation in wireless networks is a more challenging issue due to flexible wireless conditions, such as increased and variable packet error rate (PER), wireless link rate adaptation, signal fading, contention, transmission retries, etc. Most of the existing wireless bandwidth estimation solutions such as WBest [14] and DietTOPP [15] use probing-based techniques. Probing techniques introduce extra traffic that has a negative influence on multimedia applications. Recently, mechanisms like iBE [16] and IdleGap [17] that employ cross-layer-based techniques have been proposed to estimate the wireless channel bandwidth. Unfortunately, cross-layer solutions require modifications of standard protocols that make it complex and not desirable.

This paper proposes a model-based bandwidth estimation algorithm (MBE) to estimate the available bandwidth for data transmissions in IEEE 802.11 WLANs, as shown in Fig. 1. There are three major contributions. First, MBE relies on a novel transmission control protocol (TCP) model for wireless data communications, which extends an existing TCP throughput model by considering the IEEE 802.11 WLAN characteristics (transmission error, contention, and retry attempts). Second, MBE utilizes a new user datagram protocol (UDP) throughput model based on UDP packet transmission probability and IEEE 802.11 channel delay. Third, this paper derives a formula for estimating the bandwidth when TCP and UDP traffic coexists in IEEE 802.11 networks and proposes MBE. Note, unlike most existing estimation techniques, MBE neither requires modification of current transmission protocols nor uses probing traffic.

In this paper, standalone and comparison-based experiments have been carried out using both simulations and real tests. The MBE model is studied in terms of feedback frequency, variant packet size, dynamic wireless link rate, and different wireless PERs. Furthermore, MBE is compared with existing wireless bandwidth estimation techniques using two performance metrics: 1) error rate and 2) overhead.

This paper is structured as follows. Section II introduces related works on wireless bandwidth estimation. Sections III and IV describe the MBE algorithm. Section V introduces the experimental setup. Conclusions are given in Section VI.

II. Related Works

This section presents related works regarding MBE. To begin with, existing bandwidth estimation solutions are introduced, and then subsequently, current models for TCP throughput and IEEE 802.11 medium access control (MAC) are described. Finally, different wireless link rate adaptation algorithms are presented. MBE uses these related techniques for both model development and experimental design.

A. Wireless Bandwidth Estimation Techniques

Current bandwidth estimation solutions for wireless channel can be grouped into two categories.

Probing-Based Techniques: WBest [14] uses a probing packet-pair dispersion solution to estimate the effective capacity of wireless networks. It uses a packet-train technique to infer mean and standard deviations of available bandwidth. However, WBest has not been compared with other wireless bandwidth estimation techniques. DietTOPP [15] dynamically changes the bit rate of probing traffic. The available bandwidth is obtained when the probing traffic throughput experiences the turning point. The weakness of DietTOPP is the enormous amount of overhead introduced. AdhocProbe [18] sends fixed size and back-to-back probing packet pairs from sender to receiver. The transmission time is stamped on every packet by the sender. The path capacity is then calculated at the receiver. However, the main limitation of AdhocProbe is that it is only suitable for measuring the path capacity of fixed rate wireless networks. ProbeGap [19] probes for "gaps" in the busy periods and then multiplies by the capacity to obtain an estimate of the available bandwidth. The main disadvantage of ProbeGap is the dependency on other capacity estimation schemes.

Cross Layer-Based Estimation Techniques: iBE [16] estimates the wireless network bandwidth using the packet dispersion technique, which records the packet payload size and one-way delay at the MAC layer. The estimation results are then sent to the application layer for intelligent adaptation. iBE uses the application data packets themselves instead of probing traffic, reducing the estimation overhead. However, iBE requires modification of the 802.11 MAC protocol. IdleGap [17] develops an idle module between link and network layers. The idle module obtains the link idle rate from the network allocation vector and sends it to the application layer. The bandwidth is calculated using link idle rate and known capacity. Shah *et al.* [2] propose an estimation solution to capture the wireless channel conditions at the MAC layer by measuring the channel busy time and use it to infer the available bandwidth. Probing-based techniques rely on probing traffic that impacts the wireless communication services due to the additional data introduced. Significantly, cross-layer techniques have lower overhead than packet dispersion solutions. However, they are difficult to be deployed widely due to the modifications required in the devices and standard protocols.

B. State-of-the-Art Models on Throughput and 802.11 MAC

Current models for analyzing the traffic throughput basically focus on TCP. To the best of our knowledge, Mahdavi and Floyd [20] proposed the initial TCP throughput model wherein they analyze the TCP congestion avoidance mechanism. However, the model provides low accuracy when the loss is greater than 5%. Kurose *et al.* [21], [22] developed a more accurate TCP throughput model by capturing both TCP fast retransmission and time out mechanism. On similar lines, the works described in [23] and [24] propose accurate TCP transmission models for video traffic, since the TCP flows impact significantly on video delivery performance. However, none of these throughput models consider UDP traffic and wireless network conditions.

The IEEE 802.11 MAC protocol has been modeled in many works. Bianchi [25] proposes a 2-D Markov chain model to describe the 802.11 distributed coordination function (DCF) backoff mechanisms. However, the model relies on several assumptions, such as constant and independent packet collision probability, infinite retry limit, saturation traffic, and infinite buffer size. Wu *et al.* [26] improve Bianchi's model by introducing finite retry and have also assumed saturated traffic. However, Wu *et al.*'s model failed to consider wireless errors. Recently, Chatzimisios *et al.* [27] have extended Bianchi's model by including retry limit, collision, and transmission-related packet error under saturated traffic.

C. Wireless Link Rate Adaptation

IEEE 802.11a/b/g standards all provide multiple link rates. For instance, 802.11b offers four transmission rates: 11, 5.5, 2, and 1 Mb/s. Link rate adaptation algorithms have been developed to dynamically adjust the data rate. Autorate fallback (ARF)-based solutions [28], [29] is one of the earliest rate adaptation algorithms. ARF increases the data rate after consecutive successful transmission and decreases the data rate when transmission error occurs. The limitation is that ARF selects a higher data rate whenever a fixed threshold of successful transmissions is achieved. Adaptive ARF (AARF) [30] is developed based on ARF to resolve the bit-rate selection problem. AARF increases the threshold exponentially whenever the transmission attempt with higher rate fails. AARF resets the threshold to the initial value when the rate is decreased and thereby provides support to both short- and long-term adaptations. However, neither ARF nor AARF consider packet loss due to collision and therefore cannot be applied to a multistation scenario. Receiver-based autorate (RBAR)-based solutions [31], [32] use request to send/clear to send (RTS/CTS) frames to deliver the negotiated maximum transmission rate to both senders and receivers. The purpose of RBAR is to optimize the application throughput. However, RBAR requires modification of 802.11 protocols and is of little practical interest. Recently, Choi *et al.* [33] have proposed a novel rate adaptation scheme that mitigates the collision effect on the operation of rate adaptation. Instead of using explicit RTS/CTS frames, the authors utilize the "retry" information in 802.11 MAC headers as feedback to reduce the collision effect. Previous rate adaptation schemes such as ARF and AARF use frame loss or frame reception to estimate the data rates. Further, SoftRate [34] uses confidence information to estimate the prevailing channel bit error rate (BER), which is calculated at the physical layer and delivered to higher layers via the SoftPHY interface. Senders then pick up an optimal data rate based on the BER. Notably, MBE does not need to know which link rate adaptation policy is used since different access points (APs) have various adaption solutions. Instead, MBE will look at the effect of the link rate adaptation and perform bandwidth estimation.

III. Model-Based Bandwidth Estimation

A. TCP Throughput and IEEE 802.11 Models

This section first introduces the TCP throughput and the 802.11 models, which are used by the TCP over WLAN throughput model. The update processes for the two models are then described. MBE estimates TCP and UDP traffic separately. The behaviors of the TCP's fast retransmission and timeout mechanisms are captured in Kurose's model, which can be used to estimate the maximum bandwidth share that a TCP connection could achieve.

$$B=\frac{MSS}{RTT\times\sqrt{\frac{2bP_{\text{tcp}}}{3}}+T_o\times\min\left(1,3\sqrt{\frac{3bP_{\text{tcp}}}{8}}\right)\times P_{\text{tcp}}\times\left(1+32{P_{\text{tcp}}}^2\right)} \quad (1)$$

The TCP throughput model is described in (1), where B is the throughput received, MSS denotes the maximum segment size, RTT is the transport layer roundtrip time between sender and receiver, b is the number of packets that are acknowledged by a received ACK, P_{tcp} is the steady-state loss probability, and T_o is the timeout value to trigger retransmission.

The IEEE 802.11 model was introduced by Chatzimisios *et al.* They extended Bianchi's IEEE 802.11 DCF Markov chain model by taking into account packet retry limits, collisions, and propagation errors (fading, interference). The key assumption of the model is that the transmission loss probability P_{DCF} of a transmitted packet is constant and independent of the number of the collisions or errors occurred in the past. The probability P_{DCF} is given by and (2), where N indicates the number of contending stations, L is the packet size, H is the packet header, and τ denotes the probability that a station transmits a packet in a randomly chosen slot time. The probability τ is given by (3), where W represents the initial contention window size, and m means retry limit.

$$P_{\text{DCF}}=1-(1-\tau)^{N-1}\times(1-\text{BER})^{L+H} \quad (2)$$

$$\tau=\frac{2\times(1-2P_{\text{DCF}})\times\left(1-{P_{\text{DCF}}}^{m+1}\right)}{W\times(1-(2P_{\text{DCF}})^{m+1})\times(1-P_{\text{DCF}})+(1-2P_{\text{DCF}})\times(1-P_{\text{DCF}}^{m+1})} \quad (3)$$

Chatzimisios *et al.* have described a unique solution for (2) and (3) and derived the relation for the probability that at least one transmission occurs in a random time slot P_{tr}. This could be written as

$$P_{\text{tr}}=1-(1-\tau)^N. \quad (4)$$

When the retransmission reaches a retry limit m, the packet is dropped immediately. Consequently, we derived the drop probability P_{drop} as

$$P_{\text{drop}}={P_{\text{DCF}}}^{m+1}. \quad (5)$$

However, the TCP throughput model does not offer accurate results in situations when TCP runs over IEEE 802.11 networks, since the wireless channel characteristics are not considered. For this reason, this paper extends the TCP throughout model by considering both TCP congestion control mechanism and 802.11 characteristics.

B. TCP Over WLAN Throughput Model

There are three steps to update the original TCP model to consider wireless delivery conditions: 1) packet loss probability update P_{tcp}; 2) roundtrip time (RTT) update; and 3) consideration of both TCP and 802.11 DCF models.

Packet Loss Update: There are two types of packet loss when transmitting TCP traffic over wireless: 1) congestion loss P_{cong} and transmission loss P_{DCF}. TCP assumes that all packet loss is caused by congestion and therefore reduces the congestion window.

The value of P_{cong} depends on the queuing protocol. MBE considers the popular random early discard (RED) queuing protocol proposed in RFC 2309 [35]. RED determines the action of packet forwarding based on current queue size (q_{k+1}) and updates the average queue size $(\bar{q}_{k+1})$ for each arriving packet. The RED specification defines the average queue size, as given in (6), where w_q is the weight factor.

The packet drop probability due to queue congestion P_{cong} is given in (7), where $q_{\min}$ and $q_{\max}$ denote the minimum and maximum thresholds of the queue. P_{cong} is collected in the sender's queue. Note that $DropTail$ can be considered a special case of RED, i.e.,

$$\bar{q}_{k+1}=(1-w_q)q_k+w_q\times q_{k+1} \quad (6)$$

$$P_{\text{cong}}=\begin{cases}0, & \text{if} \quad \bar{q}_{k+1}\le q_{\min}\\ 1, & \text{if} \quad \bar{q}_{k+1}\ge q_{\max}\\ \frac{\bar{q}_{k+1}-q_{\min}}{q_{\max}-q_{\min}}, & \text{otherwise.}\end{cases} \quad (7)$$

TCP and 802.11 MAC trigger a packet retransmission event when packet loss is detected. The packet loss can be caused by either queue congestion P_{cong}, wireless transmission error P_{DCF}, or retry-based drop P_{drop}.

The probability of retransmission $P_{\text{retr}}^{\text{TCP}}$ based on the 802.11 standard is derived as shown in (8), shown below, where

$$P_{\text{retr}}^{\text{TCP}} = P_{\text{cong}} + P_{\text{DCF}} + P(\text{drop}|\text{DCF})$$
$$P(\text{drop}|\text{DCF}) = \frac{P(\text{DCF}|\text{drop}) \times P_{\text{drop}}}{P_{\text{DCF}}}. \quad (8)$$

$P(\text{drop}|\text{DCF})$ refers to the packet drop probability of the IEEE 802.11 MAC layer. The parameter P_{drop} is dependent on P_{DCF}, since in the IEEE 802.11 MAC layer, the packet is dropped if the retransmission reaches the maximum number of attempt limit. The parameters P_{cong} and P_{DCF} are independent from each other, as they are determined by the queue status and wireless channel, respectively. Consequently, the conditional probability is used for drop probability. The probability $P(\text{DCF}|\text{Drop})$ is equal to 1, as this dependency always exist.

Consequently, the probability of successful transmission $P_{\text{succ}}^{\text{TCP}}$ is written as shown in

$$P_{\text{succ}}^{\text{TCP}} = 1 - P_{\text{retr}}^{\text{TCP}}. \quad (9)$$

RTT Update: As shown in Figs. 2 and 3, the overall delay for transmitting the data can be decomposed into the following seven components based on the Open Systems Interconnection layers:

1) App_Delay: delay of application layer process such as video encoding/decoding, etc.;
2) $Transport\_Delay$: delay caused by transport layer protocol such as TCP congestion control;
3) IP_Delay: delay of network layer process like routing;
4) MAC_Delay: delay introduced by CSMA/CA mechanism;
5) Phy_Delay: delay at physical layer;
6) $Prop\_Delay$: propagation delay during the transmission;
7) $Proc\_Delay$: determined by terminal's processing ability such as CPU, memory, power mode, etc.

During the round-trip time RTT, the receiver can be in one of the following states: idle, successful transmission, or retransmission. The delay for successful transmission is denoted as T_{succ}. We derived (10) and (11), shown below, to present the 802.11 MAC layer delay for basic access mode MAC_Delay_{basic} and RTS/CTS mode MAC_Delay_{RTS}, where distributed interframe space (DIFS) and short interframe space (SIFS) are contention control parameters defined in 802.11 MAC specifications. MAC_ACK represents the acknowledgment packet sent by the MAC receiver, i.e.,

$$MAC\_Delay_{\text{basic}} = DIFS + SIFS + MAC\_ACK \quad (10)$$
$$MAC\_Delay_{\text{RTS}} = DIFS + 3 \times SIFS + \text{RTS} + CTS + MAC\_ACK. \quad (11)$$

Combining (10) and (11), the delay for successful transmission is given by (12), shown below, where TCP_ACK represents the acknowledgment packet sent by the TCP receiver. Note that the propagation delay is the time taken to transmit

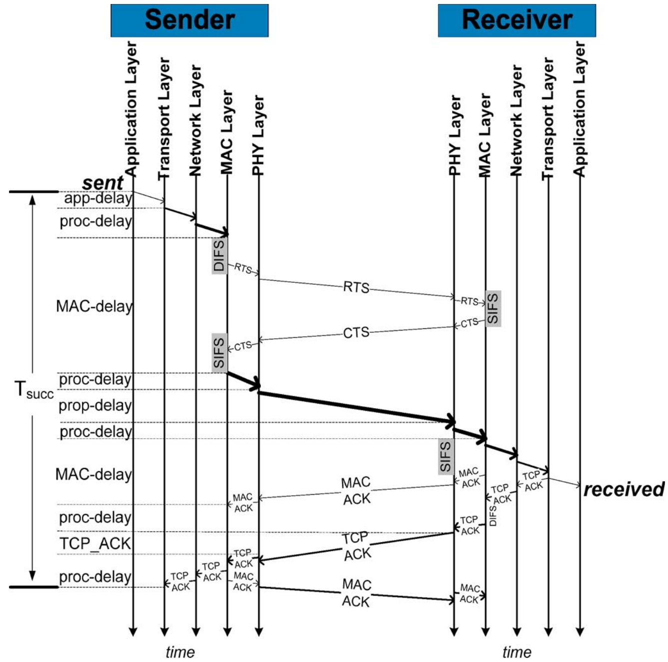


Fig. 2. Successful transmission when TCP runs over 802.11 networks.

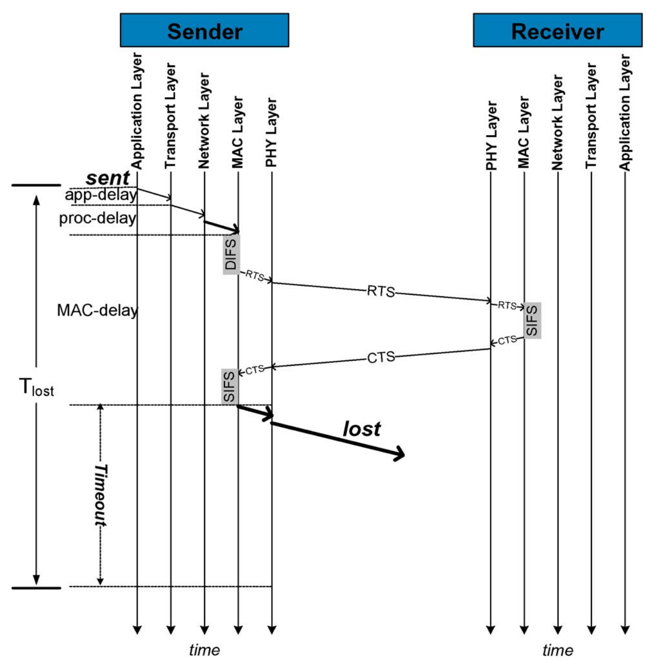


Fig. 3. Packet loss when TCP runs over 802.11 networks.

data, which include the original data packet plus the stack protocol header, i.e.,

$$T_{\text{succ}}^{\text{TCP}} = APP\_Delay + \Pr oc\_Delay + \{MAC\_Delay_{\text{basic}} \; MAC\_Delay_{\text{RTS}}\} + \Pr op\_Delay + TCP\_ACK. \quad (12)$$

The TCP-Reno congestion control starts retransmission if any of the following two conditions occur:

1) Three duplicate ACKs are received at the sender as described in RFC 2581 [36].

2) TCP sender does not receive ACK after waiting a period equal to the timeout (T_o^{TCP}). RFC 2581 gives suggestions on how to calculate timeout, as shown in (13)–(15). In (13), shown below, the parameter β is a smoothing factor determining the weight given to the previous value of RTT, namely, RTT'. The parameter M denotes the time taken for ACK to arrive. D_{RTT} is the estimation of the standard deviation of RTT. D'_{RTT} is the previous value of D_{RTT}. Whenever an ACK is received, the difference between expected and measured values |RTT-M| is computed, and D_{RTT} is updated as in (14), shown below. Subsequently, T_o^{TCP} is given by (15), shown below, based on dynamic timeout adjustment. A typical TCP implementation uses $\alpha = 0.875$ and $\beta = 0.75$, i.e.,

$$RTT = \beta \times RTT' + (1-\beta) \times M \tag{13}$$

$$D_{RTT} = \alpha \times D'_{RTT} + (1-\alpha) \times |RTT - M| \tag{14}$$

$$T_o^{\mathrm{TCP}} = RTT + 4 \times D_{RTT}. \tag{15}$$

Further, the delay $T_{\mathrm{lost}}^{\mathrm{TCP}}$ caused by timeout is subsequently given by

$$T_{\mathrm{lost}}^{\mathrm{TCP}} = \Pr oc\_Delay + MAC\_Delay + T_o^{\mathrm{TCP}}. \tag{16}$$

When three duplicate ACK packets are received at the sender, TCP enters fast retransmission, and the delay caused by the three ACK (T_{3ACK}) is $T_{\mathrm{succ}}^{\mathrm{TCP}}$. The average retransmission delay $T_{\mathrm{retr}}^{\mathrm{TCP}}$ is derived in (17), shown below. The retransmission delay can be $T_{\mathrm{succ}}^{\mathrm{TCP}}$ or T_{lost}, depending on how the retransmission is triggered: three duplicate ACKs or the timeout, i.e.,

$$T_{\mathrm{retr}}^{\mathrm{TCP}} = \left\{T_{3ACK}, T_{\mathrm{lost}}^{\mathrm{TCP}}\right\} = \left\{T_{\mathrm{succ}}^{\mathrm{TCP}}, T_{\mathrm{lost}}^{\mathrm{TCP}}\right\}. \tag{17}$$

Combination of TCP Model and 802.11 DCF Model: By combining (4), (8), (9), (12), and (17), the new roundtrip time $MRTT$ is written as

$$MRTT = (1-P_{\mathrm{tr}}) \times \sigma + P_{\mathrm{retr}}^{\mathrm{TCP}} \times T_{\mathrm{retr}}^{\mathrm{TCP}} + P_{\mathrm{succ}}^{\mathrm{TCP}} \times T_{\mathrm{succ}}^{\mathrm{TCP}}. \tag{18}$$

The parameter σ is the MAC slot time. Note that P_{tr} defined in 802.11 MAC is adopted in the new model since it is independent of the protocols. It is necessary to use $MRTT$ as it considers the transmission and acknowledgement times contributed by both transport layer and MAC layer protocols. The RTT defined in Kurose's model (1) includes the time computed at transport layer only.

Based on (1), (8), and (18), the application layer throughput B^{TCP} for each TCP connection is described in (19), where b is the number of packets acknowledged by a received ACK.

$$B^{\mathrm{TCP}} = \frac{MSS}{MRTT \times \sqrt{\frac{2bP_{\mathrm{retr}}^{\mathrm{TCP}}}{3}} + T_o \times \min\left(1, 3\sqrt{\frac{3bP_{\mathrm{retr}}^{\mathrm{TCP}}}{8}}\right) \times P_{\mathrm{retr}}^{\mathrm{TCP}} \times \left(1+32 P_{\mathrm{retr}}^{\mathrm{TCP}2}\right)} \tag{19}$$

If the network, device, and application service remain the same for a user, then the MBE would need to know values of only the following two types of parameters:

1) static parameters: application delay, processing delay, 802.11 MAC configurations such as minimum contention window, $DIFS$, $SIFS$, slot time, retry limit, and capacity;
2) dynamic parameters: the number of contending stations, packet loss, and data packet size.

C. UDP Over WLAN Throughput Model

We first propose the throughput estimation model for UDP over IEEE 802.11. Unlike TCP, the UDP protocol does not support packet retransmissions, and therefore, the UDP over WLAN throughput model should consider this. Hence, the terms P_{retr} and $MRTT$ defined in (8) and (18), which consider TCP fast retransmission and timeout respectively, should be removed in MBE's UDP version. By combining (2) and (5), the probability of retransmission when UDP traffic run over 802.11 networks can be written as

$$P_{\mathrm{retr}}^{\mathrm{UDP}} = P_{\mathrm{DCF}} + P_{\mathrm{drop}}. \tag{20}$$

Similar to the TCP transmission delay described in (12), the UDP transmission delay can be derived and is shown as

$$T_{\mathrm{succ}}^{\mathrm{UDP}} = APP\_Delay + \Pr oc\_Delay + \{MAC\_Delay_{\mathrm{basic}}\ MAC\_Delay_{\mathrm{RTS}}\} + \Pr op\_Delay \tag{21}$$

$$T_o^{\mathrm{UDP}} = \Pr op\_ACK + \Pr op\_UDP + SIFS. \tag{22}$$

Furthermore, the retransmission delay is triggered by 802.11 time-out mechanism as given in

$$T_{\mathrm{retr}}^{\mathrm{UDP}} = \Pr oc\_Delay + MAC\_Delay + T_o^{\mathrm{UDP}}. \tag{23}$$

Importantly, the average delay $Delay\_UDP$ for successfully transmitting the individual UDP packet could be written as

$$Delay\_UDP = (1-P_{\mathrm{tr}}) \times \sigma + P_{\mathrm{retr}}^{\mathrm{UDP}} \times T_{\mathrm{retr}}^{\mathrm{UDP}} + P_{\mathrm{retr}}^{\mathrm{UDP}} \times T_{\mathrm{retr}}^{\mathrm{UDP}}. \tag{24}$$

The available bandwidth for UDP traffic over 802.11 WLANs is given in (25), shown below, where $Payload$ is the total information in bytes, transmitted during one time period

$$B^{\mathrm{UDP}} = \frac{\int_{T_0}^{T_1} \frac{Payload}{Delay\_UDP} dt}{T_1 - T_0}. \tag{25}$$

D. MBE for Coexisting TCP and UDP Traffic

This section introduces MBE, which considers the combined effect of TCP and UDP traffic over WLAN and makes use of TCP and UDP over WLAN throughput models introduced before.

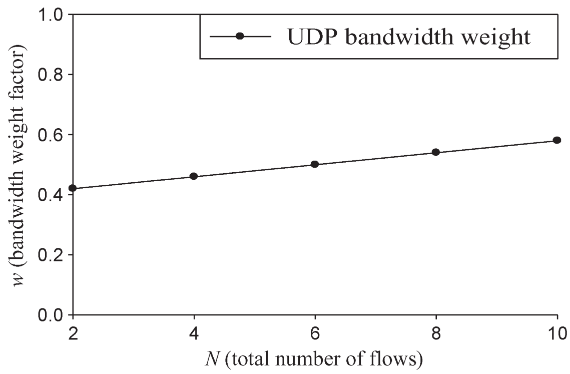


Fig. 4. Relationship between w and N.

When TCP and UDP traffic are transmitted together, their throughputs are different with those when TCP and UDP are delivered alone. TCP adopts a congestion control mechanism to adjust the transmission rate to the available bandwidth. UDP is more aggressive and always takes as much bandwidth as possible, therefore affecting the TCP traffic. The major difference between the models for TCP and UDP is with regard to consideration of lost packet retransmissions. To address this effect of UDP on TCP traffic, the weight w is introduced, as shown in Fig. 4 and (26), shown below.

By combining TCP and UDP over WLAN throughput models, the estimated aggregated throughput for coexisting TCP and UDP can be written as

$$B^{\mathrm{TCP+UDP}} = w \times \sum_{i=1}^{N} B^{\mathrm{UDP}} + (1-w) \times \sum_{j=1}^{N} B^{\mathrm{TCP}}. \quad (26)$$

The parameter w is the bandwidth weight factor, N represents the total number of TCP and UDP flows, and i and j are the numbers of TCP and UDP flows, respectively. Notably, for each value of N, the number of TCP flows and the number of UDP flows are considered equal.

The throughput performance of TCP and UDP is studied by sending TCP and UDP flows together without any background traffic. Note that TCP throughput consists of both TCP downward data stream and TCP ACK upward stream. The number of TCP and UDP flows is equal. Fig. 4 shows the relationship between w and N. The throughput of UDP linearly increases as the total amount of TCP and UDP traffic increases. When TCP and UDP traffic are transmitted together, their throughputs are different with those when TCP and UDP are delivered alone. This is mainly due to the fact that TCP adopts a fast congestion control mechanism to adjust the transmission rate based on packet loss. To address the influence of UDP over TCP, the weight w is introduced. By analyzing Fig. 4, a suggested value for w could be written as

$$w = 0.02 \times N + 0.38. \quad (27)$$

A similar comparison of the relationship between TCP and UDP flows was done by Bruno *et al.* [37]. Further, Bruno's work also demonstrated that the direction of TCP streams (upstream or downstream) does not affect the throughput performance. Hence, MBE, as described in (26), can be applied for real-world TCP and UDP traffic mix scenarios.

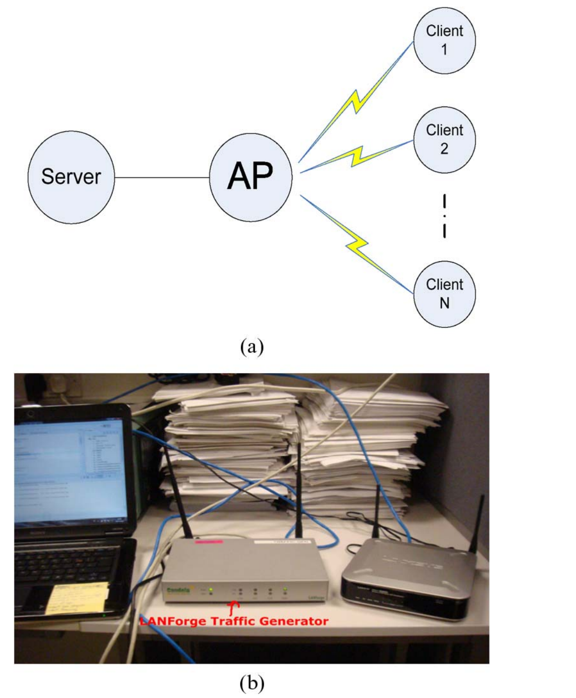


Fig. 5. (a) Test bed topology. (b) Real test bed including traffic generator and 802.11AP.

The next section presents the experimental setup, scenarios, and testing results.

IV. Experimental Setup and Scenarios

This section describes the experimental setup, including the configurations for specific estimation tool, test software introduction, and evaluation metrics used. Additionally, two experiment scenarios are introduced.

A. Setup for MBE

MBE has been evaluated by using both modeling and prototyping and by employing the NS-2.33 [38] simulator and the Candela Technologies' LANForge traffic generator V4.9.9-based network test bed. Both setups used IEEE 802.11b networks, as shown in Fig. 5. Two additional wireless patches are deployed in the NS-2: 1) No Ad-Hoc (NOAH)[1] and 2) Marco Fiore patch.[2]

NOAH was used for simulating the infrastructure WLAN environment, whereas Marco Fiore's patch provides a more realistic wireless network environment. In the prototype-based test bed, LANForge acts as a server that generates traffic transmitted via a 100-Mb/s Ethernet and a Linksys WRV210 access point to multiple virtual wireless stations. The transmission power of AP is 20 dBm through two omnidirectional antennas. MBE configures the input parameters based on the IEEE 802.11b specifications, as shown in Table I, where $MSS = 1500$, $b = 2$, $DIFS = 50\ \mu$s, $SIFS = 10\ \mu$s, slot time $= 20\ \mu$s, TCP/IP protocol header = 40 B, and MAC protocol header = 36 B. Each traffic connection consists of one server–wireless station pair.

[1] NOAH NS-2 extension, http://icapeople.epfl.ch/widmer/uwb/ns-2/noah/
[2] M. Fiore patch, http://www.telematica.polito.it/fiore

TABLE I
SIMULATION SETUP PARAMETERS IN NS-2.33

Experimental Input Parameters	Values
Routing Protocol	NOAH
Queue	DropTail
Error Model	Marco Fiore patch
DIFS	0.12
SIFS	0.08
Slot time	0.04
TCP/IP header	0.11
MAC header	0.15
Maximum Segment Size	1500 bytes
Queue buffer	50 packets
TCP packet size	1380 bytes

The wireless access mode RTS/CTS was enabled to avoid the wireless hidden node problem. DropTail was adopted as the default queue algorithm, and the queue length was set to 50. The length of TCP packet size was 1380 B. Both simulation and real test used FTP/TCP as application traffic, which used the entire wireless capacity. The sending buffer was set to 8 kB. There are two assumptions considered during the tests. First of all, the application and hardware processing delays were assumed to be negligible. This is reasonable because the IP packet processing delay in terminals depends on CPU and memory specifications, and these are state of the art in our setup. This delay is very low and is in general negligible. Second, it was assumed that the last hop wireless network is the bottleneck link of the end-to-end path. This was supported by connecting the IEEE 802.11 WLAN with a 100-Mb/s wired LAN. In this condition, the bandwidth estimation can closely reflect the wireless network capacity.

B. Setup for Other Bandwidth Estimation Techniques

Three bandwidth estimation schemes that employ different types of techniques were selected for comparison. These include nonprobing technique (iBE) [16], probing-based technique (DietTOPP) [15], and cross-layer technique (IdleGap) [17].

iBE was implemented at the 802.11 MAC layer. The 802.11 WLAN was assumed to be the bottleneck link in the end-to-end path. The feedback frequency of iBE client was set to 10 ms, as indicated by the authors [16]. The RTS/CTS function was enabled to achieve the best performance of iBE in all conditions.

DietTOPP relies on probe packet size and cross traffic, with the condition that the wireless link is the bottleneck in the end-to-end path. Hence, 1500-B probing packet and 250-kb/s cross traffic were used to obtain better estimation performance, as indicated by the authors [15].

IdleGap was implemented between 802.11 link and network layers. The cross traffic for IdleGap was set to 10 kb/s, as suggested [17]. The application packet size was set to 700 B since IdleGap achieved good accuracy for packet size ranges from 512 to 896 B. RTS/CTS was also enabled.

C. Evaluation Metrics

To evaluate the MBE performance, two estimation-based evaluation metrics were introduced: 1) error rate and 2) overhead. Error rate is defined as the difference between the MBE estimation results and the ground truth result. A lower error rate indicates higher accuracy of bandwidth estimation. The error calculation is given by (28)

$$ErrorRate = \frac{|ESTIMATEDBandwidth - REALBandwidth|}{REALBandwidth}. \quad (28)$$

Overhead is depicted as the total number of bytes sent by the model to perform the estimation. A lower overhead is critical for streaming applications over wireless networks.

D. Experimental Scenarios

Two experiments were designed to study the performance of MBE. Their goals are as follows: 1) Evaluate the robustness of the MBE model, and 2) evaluate the bandwidth estimation quality. Generally, the robustness of the MBE model depends on the feedback frequency, data packet size, wireless PER, and wireless link rate adaptation scheme. The impacts of the four factors were studied in separate tests. Additionally, the bandwidth estimation quality was studied using a comparison-based methodology in terms of error rate and overhead.

V. EXPERIMENTAL RESULTS AND RESULT ANALYSIS

This section presents the details of the experimental tests performed, as well as the result analysis.

A. Robustness of the MBE Model

To study the robustness of the MBE model in a wireless network, separate tests were performed in terms of feedback frequency, packet size, PER, and wireless link adaptation. For each test scenario, the variable-controlling method was adopted. Each scenario included a specific experimental setup that was based on the test bed described in Section IV.

Scenario A-1—Impact of Feedback Frequency: The purpose of this test was to investigate the impact of feedback traffic introduced by MBE and select a good feedback frequency for future tests. Too frequent feedback causes high overhead, which reduces the performance of multimedia traffic. MBE uses RTCP receiver report [39] to deliver the feedback (8-B RTCP receiver report packet header, 8-B UDP header, 20-B IP header, and 4-B feedback payload) due to the low cost and high reliability of this approach. Since the feedback size and the number of flows are relatively static, the bandwidth taken by feedback relies on the feedback interval. RTCP traffic uses UDP as the underlying transport protocol, so the single feedback packet size can be written as

$$FeedbackSize = RTCPheader + UDPheader + IPheader + Payload. \quad (29)$$

TABLE II
MEAN ESTIMATION ERROR, OVERHEAD, AND α DEPENDENCY ON THE FEEDBACK INTERVAL. TIME DURATION = 100 s

Feedback interval (s)	Mean Error Rate	Overhead (Mb)	α (%)
0.001	0.31	64	6.3
0.005	0.24	12.8	3.2
0.01	0.17	6.4	2.1
0.1	0.12	0.64	0.5
0.5	0.08	0.128	0.04
1.0	0.04	0.064	0.007
2.0	0.11	0.032	0.001
4.0	0.15	0.016	0.0005
6.0	0.19	0.0106	0.00009
8.0	0.22	0.008	0.00002
10.0	0.23	0.0064	0.000006

The value of the feedback size is 40 B. Consequently, the feedback rate for each flow is

$$FeedbackRate = FeedbackSize/FeedbackInterval. \tag{30}$$

When the number of flows is N and the time duration is T, the overhead can be computed by

$$Overhead = FeedbackRate \times T \times N. \tag{31}$$

Experimental setup: Scenario A-1 built up the test environment in the simulation environment. The MBE system starts sending traffic with packet size set to 1000 B as part of a single 6-Mb/s constant bit rate (CBR)/UDP. PER was set to 1×10^{-5}. The mobile nodes stay close to AP at a distance smaller than 10 m, where the link data rate is 11 Mb/s. The duration of the experiment was 100 s. The feedback interval was varied from 0.001 to 10.0 s.

Experimental result analysis: Let α represent the ratio between the feedback rate and the channel bandwidth. MBE performance-related metrics in terms of mean estimation error rate, overhead, and α are shown in Table II. The RTCP standard recommends that α should account for less than 5% of the bandwidth to optimize the quality of application. By analyzing the results, the overhead introduced by MBE increases with the decrease of feedback interval and the mean error rate changes with different feedback interval. For instance, in the case of feedback interval equal to 1 ms, the estimation overhead was 64 Mb during 100 s. This consists approximately 6.3% of the overall bandwidth, and the mean error was 31%. High packet loss reduces the MBE bandwidth estimation accuracy and increases the estimation error. Subsequently, the optimal feedback frequency is selected based on Table II. A good tradeoff between the amount of overhead and mean error recommends a feedback interval of 1.0 s.

Scenario A-2—Impact of Packet Size: Scenario A-2 investigates the impact of packet size on the MBE estimation accuracy. The feedback frequency suggested from scenario A-1 was adopted in this test.

Experiment setup: Both simulation and real test experiments were performed to study the impact of packet size.

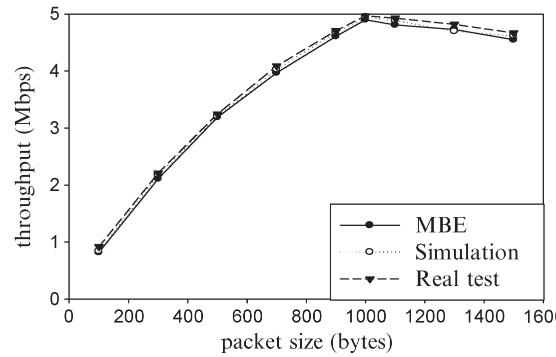


Fig. 6. Comparison of bandwidth as estimated by MBE and measured by NS-2 simulations and in the real-life tests for increasing packet size.

Single 6-Mb/s CBR/UDP traffic was sent from server to mobile station. Packet size was varied from 100 to 1500 B (Ethernet MTU) with a step of 200 B. Feedback frequency was set to 1.0 s. It was noticed that 6-Mb/s traffic was used to saturate the network so that the effect of packet size will be studied in a loaded network. The mobile node stays close to AP at a distance smaller than 10 m, where the link data rate was 11 Mb/s. Experiment time duration was set to 100 s.

Experimental result analysis: The estimation and measurement results of the packet size study are shown in Fig. 6. It is shown that the available bandwidth increases along with the increase of packet size. Since a smaller packet size leads to more frequent transmissions and higher packet overhead. Throughput is the highest when the packet size is 1000 B, as 1000 B was the fragmentation threshold. Packets with size bigger than 1000 B are fragmented into multiple packets, resulting in a decrease in throughput. According to Fig. 6, following a two-tailed T test analysis, it can be said with 95% confidence level that there is no statistical difference between the MBE results and those of the real test. It can be concluded that MBE is able to adapt variable packet size with high accuracy.

Scenario A-3—Impact of PER: In contrast with wired communications, wireless networks suffer from environmental factors, e.g., building block, or terminal generated noise, e.g., thermal noise. These affect the communications and decrease the estimation accuracy. The purpose of scenario A-3 was to study the performance of MBE in various PERs. The suggested feedback interval was used based on conclusion from scenario A-1.

Experimental setup: The impact of PER was investigated under both simulation and real test environments. Similar with the test setup in scenarios A-1 and A-2, this experiment also transmitted single CBR/UDP traffic with packet size of 1000 B. Feedback frequency was set to 1.0 s. NS-2 provides functions to increase the PER from 1×10^{-8} to 1. For each PER, there was a corresponding average packet loss ratio that was then imported to the MBE model to estimate the available bandwidth. In real test, it is difficult to inject packet error into the wireless channel. An alternative solution is to adjust the AP transmitting power to mimic the effect of PER. As shown in Fig. 5(b), we added the Pascall[3] signal manual attenuator between the AP and

[3]http://www.pascall.co.uk

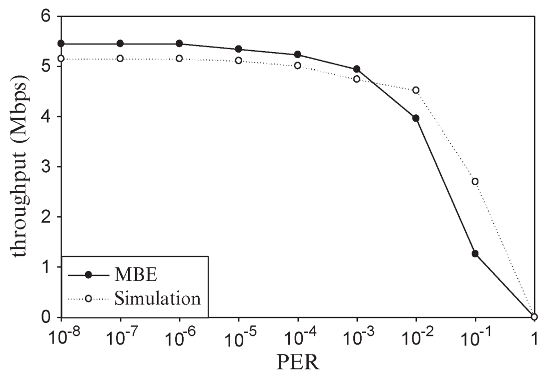


Fig. 7. PER effect on throughput.

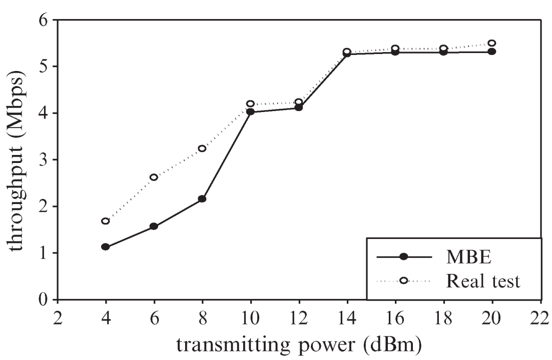


Fig. 8. Transmitting power effect on throughput.

an external N-type antenna. Since the maximum transmission power of AP is 20 dBm, the attenuator gradually reduced the transmitting power with a 2-dBm step. For both simulation and real test, the mobile nodes stay close to AP at a distance smaller than 10 m, where the link data rate was 11 Mb/s. Experiment time duration was set to 100 s, and the feedback time interval was set to 1.0 s. The bandwidth estimated by MBE is given based on the packet loss information under different simulation and real test conditions.

Experimental result analysis: Simulation and real test-based results of PER influence are shown in Figs. 7 and 8. It was noticed that the available bandwidth generally decreases along with the increase of PER. The bandwidth is equal to 1 when the PER is equal to 1. This implies that no successful transmission will be achieved even with maximum retry limit (number of retry limit $= 7$). Additionally, the available throughput decreases along with the reduction of transmission power. When the transmit power was lower than 10 dBm, the throughput start decreasing significantly. This can be explained by that the receiving signal strength might lower the receiving threshold defined at the AP. The two-tailed T-test analysis presents with 90% confidence level that there is no statistical difference between MBE results and those of the real test. Hence, it could be concluded that MBE is able to adapt the estimation to variable PER with high accuracy.

Scenario A-4—Impact of Wireless Link Adaptation: The goal of scenario A-4 was to assess the performance of MBE under variable wireless link capacity. Unlike wired networks, the capacity of wireless networks changes due to the link

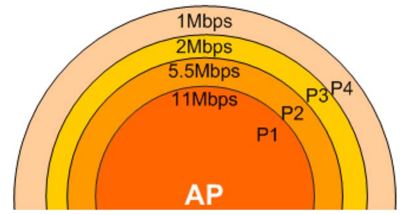


Fig. 9. Theoretical wireless link capacity for IEEE 802.11b.

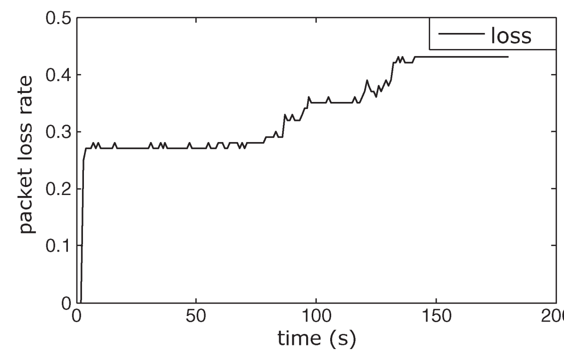


Fig. 10. Packet loss rate variation while mobile node moves away from AP.

TABLE III
IMPACT OF DISTANCE FROM AP IN TERMS OF PACKET LOSS RATE AND THROUGHPUT

	11Mbps	5.5Mbps	2Mbps	1Mbps
Loss	0.27%	0.32%	0.36%	0.43%
Throughput	4.95M	3.11M	2.62M	1.67M
MBE	5.01M	3.08M	2.58M	1.62M

rate adaptation. The signal strength of 802.11b-enabled AP is divided into four subareas according to the link rate distribution defined in 802.11b, as shown in Fig. 9. Darker colors indicate higher signal strength.

Experimental setup: Three test scenarios were implemented in the simulation environment to study the impact of wireless link adaptation. They are the following: 1) single mobile nodes located in the areas labeled P1, P2, P3, and P4 in Fig. 9, respectively; 2) four mobile nodes evenly distributed around AP; 3) multiple mobile nodes located at random locations around AP. These tests used the same test bed. The differences focused on the mobile node mobility, mobile node location, and application traffic. The transmit power of the 802.11b AP in NS2 was set to 20 dBm. According to the documentation of the Cisco Linksys WRV210, this can cover around 300 m. NS2 provided methods to calculate the distance threshold for the signal change: 70 m (P1–P2), 100 m (P2–P3), and 130 m (P3–P4), where P1, P2, P3, and P4 were four positions in each area.

1) Single Traffic to a Node Moving From P1 to P4: Single CBR/UDP traffic with an average rate of 6 Mb/s was sent from server to mobile station. The mobility was considered with the mobile station moving away from AP toward P4 at the speed of 1 m/s. Figs. 10 and 11 show the variations in throughput and packet loss during the transmission. Table III presents the comparison results between the simulation-based measured throughput and the estimated bandwidth from MBE.

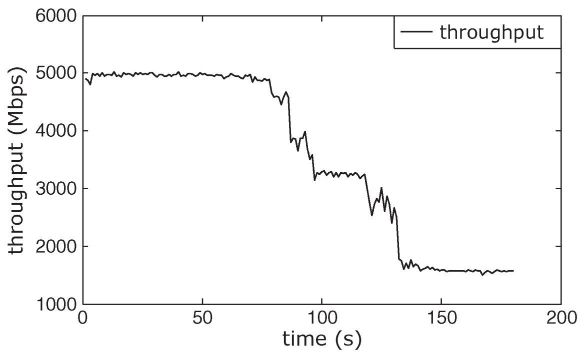


Fig. 11. Throughput variation while mobile node moves away from AP.

Experimental result analysis: It is clear from Figs. 10 and 11 that there is significant packet loss increase and throughput decrease as the mobile node moves away from AP. This is caused by the reduced transmission signal of AP. The two-tailed T-test analysis is applied on the results from Table III. It is shown that there is no statistical difference between MBE estimation results and the measured results under simulation with 95% confidence level.

2) Static Mobile Nodes Within the Coverage of AP: FTP/TCP and 6-Mb/s CBR/UDP traffic were delivered in this scenario. Three test cases were considered to study the MBE performance in multiple station conditions.

- Case 1: Four TCP flows were sent to four mobile stations, and each mobile station was statically located at P1, P2, P3, and P4, respectively.
- Case 2: Four UDP flows were sent to four mobile stations, and each mobile station is statically located at P1, P2, P3, and P4.
- Case 3: Two TCP flows were sent from mobile stations located at P1 and P3, and two UDP flows were transmitted from mobile stations located at P2 and P4.

Experimental result analysis: Table IV presents the comparison results between MBE estimated bandwidth and that measured in the simulation tests for all these three cases. Column "MBE" presents the overall bandwidth estimated by MBE when three test cases are considered. Column "Simulations" provides the overall bandwidth measured in NS-2 for the three test cases, respectively. According to the results of cases 1 and 2 in Table IV, UDP traffic achieves higher throughput than TCP, since TCP can adapt the sending rate using congestion control. Additionally, by comparing the results of cases 1 and 2, the throughput of UDP traffic increases 47.9% compared with that of TCP traffic. In case 3, two TCP flows and two UDP flows are transmitted together; the overall throughput is lower than that of four UDP flows (case 2) and higher than that of four TCP flows (case 1). UDP traffic affects TCP traffic due to the aggressive nature on bandwidth cost. The two-tailed T-test analysis presents with 90% confidence level that there is no statistical difference between MBE results and simulation results.

3) Mobile Nodes at Random Positions: In this scenario, FTP/TCP and 6-Mb/s CBR/UDP are sent. A 250 m $\times$ 250 m test topology was created in the simulation, as shown in Fig. 12. The position of AP is constant, and wireless stations are located around AP with a random distance ranging from 30 to 120 m. The number of TCP and UDP flows both equal λ, which increases from 1 to 5. Hence, the total number of contending stations ranges from 2 to 10, in steps of 2.

Experimental result analysis: The mean aggregate throughput was measured through simulation for mobile nodes with random location. Table V and Fig. 13 give the comparison results between the simulation-based measured throughput and the estimated bandwidth from MBE. For λ smaller than 4, both estimated and measured bandwidths increase with increasing number of flows, and the bandwidth starts decreasing when

TABLE IV
IMPACT OF DISTANCE FOR MULTIPLE TCP AND UDP TRAFFIC

	P1	P2	P3	P4	MBE	Simulation
Case 1	1TCP	1TCP	1TCP	1TCP	1.86Mbps	1.99Mbps
Case 2	1UDP	1UDP	1UDP	1UDP	3.57Mbps	3.65Mbps
Case 3	1TCP	1UDP	1TCP	1UDP	2.58Mbps	2.69Mbps

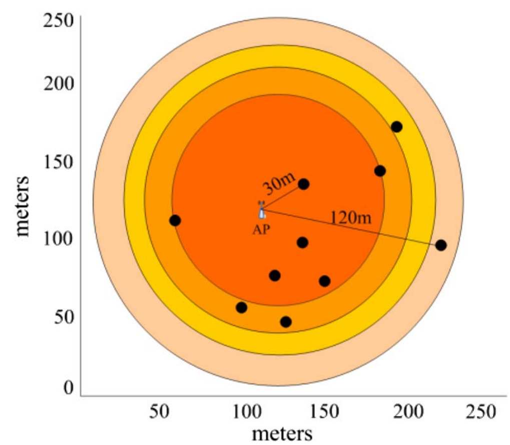


Fig. 12. Random topology in simulation.

TABLE V
BANDWIDTH COMPARISON BETWEEN MBE AND SIMULATION

λ	MBE (Mbps)	Simulation (Mbps)
1	2.65	2.78
2	3.51	3.63
3	3.48	3.49
4	4.28	4.37
5	3.84	3.95

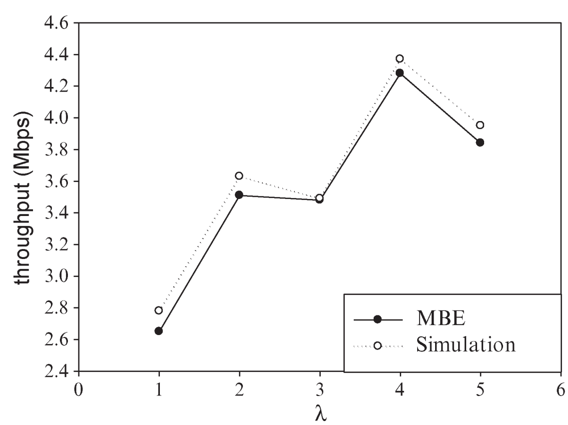


Fig. 13. Bandwidth comparison between MBE and simulation when λ increases from 1 to 5.

TABLE VI
COMPARISON OF BANDWIDTH ESTIMATED AND ESTIMATION OVERHEAD AMONG iBE, DIETTOPP, IDLEGAP, AND MBE

Case	N (Number of flows)		Comparison of Estimated Bandwidth						Comparison of Estimation Overhead			
	TCP	UDP	iBE (Mbps)	DietTOPP (Mbps)	IdleGap (Mbps)	MBE (Mbps)	Simulation (Mbps)	Real Test (Mbps)	iBE (Mbps)	DietTOPP (Mbps)	IdleGap (Mbps)	MBE (Mbps)
1	1	0	5.08	5.01	4.85	5.57	4.89	4.97	0.049	0.95	0.061	0.058
2	3	0	3.65	4.23	3.83	3.61	3.98	3.66	0.16	1.11	0.27	0.17
3	5	0	3.01	3.02	3.24	3.12	3.47	3.17	0.24	1.14	0.48	0.28
4	7	0	2.43	2.24	2.50	2.52	2.94	2.56	0.32	1.21	0.69	0.36
5	9	0	1.65	1.33	1.72	1.92	2.25	1.95	0.47	1.35	0.81	0.49
6	0	1	6.21	5.39	5.61	6.09	5.1	5.8	0.058	1.02	0.06	0.064
7	0	3	5.53	4.96	5.15	5.32	5.3	5.3	0.18	1.29	0.31	0.21
8	0	5	5.01	4.82	5.02	5.11	5.19	5.21	0.26	1.31	0.62	0.33
9	0	7	4.54	4.53	4.89	4.99	5.07	5.03	0.38	1.36	0.85	0.42
10	0	9	4.12	4.17	4.68	4.8	4.94	4.91	0.52	1.39	0.91	0.59
11	1	1	5.98	5.78	5.01	5.83	4.975	5.28	0.062	1.21	0.08	0.071
12	2	2	4.56	4.34	4.32	4.74	4.86	4.61	0.21	1.31	0.33	0.19
13	3	3	3.82	3.72	4.21	4.46	4.59	4.51	0.29	1.36	0.65	0.31
14	4	4	3.51	3.38	4.13	4.3	4.46	4.45	0.43	1.37	0.85	0.46
15	5	5	3.19	2.12	4.08	4.12	4.35	4.31	0.55	1.42	0.99	0.51

λ is equal to 5. The overall throughput of the application traffic close to the wireless capacity for λ is equal to 4, where the number of TCP and UDP flows was 8. The two-tailed T-test analysis is used and shows a 95% confidence level, i.e., there is no statistical difference between MBE results and the simulation results. Based on the test results from Tables III –V, it is concluded that MBE can adapt to the variable wireless link capacity. This can be explained that the packet loss caused by the wireless link adaptation is used by MBE to infer the available bandwidth.

B. Evaluation of Bandwidth Estimation

Three scenarios were designed to assess the MBE performance in terms of error rate, overhead, and loss. MBE analytical model results are compared with simulation and real test results. Additionally, the results of other bandwidth estimation techniques such as iBE, DietTOPP, and IdleGap were also considered

Experimental setup: Each scenario included 15 cases with variable FTP/TCP and 6-Mb/s CBR/UDP traffic load. Test cases 1–5 transmitted TCP traffic only, test cases 6–10 transmitted UDP traffic only, while test cases 11–15 sent TCP and UDP traffic simultaneously. To estimate the maximum bandwidth a network can support, it is necessary to use high traffic load to saturate the 802.11 channel. In a saturated network, any new incoming traffic will decrease the overall throughput since the available throughput is higher than the network capacity. Based on test scenarios A-1, A-2, and A-3, the feedback interval was set to 1.0 s, the packet size was 1000 B, and PER was set to 10^{-5}. The overall sending rate was greater than 6 Mb/s and less than 7 Mb/s. The mobile nodes are located close to AP at a distance smaller than 10 m, where the link data rate is 11 Mb/s. The testing time duration was 100 s.

Scenario B-1—Error Rate Analysis: Scenario B-1 studies the error rate that reflects the accuracy of MBE. Table VI shows the comparison results between bandwidth estimated and measured. Real test and simulation results were obtained according to the setup in Section IV.

Experimental result analysis: Fifteen test cases were implemented to study the error rate of MBE under variable traffic load. In single flow situation, such as cases 6 and 11, IdleGap provides better accuracy than MBE in comparison with results from real test. From test cases 1–5, the number of TCP flows increased from 1 to 9, with steps of 2. It is shown that the bandwidth estimated by the four algorithms and the bandwidth measured in simulation and real test all decrease as the overall traffic load increases. For test case 3, which transmits five TCP flows, the estimated bandwidth by MBE is 3.12 Mb/s. Similarly, the impacts of UDP traffic were studied, as shown from test cases 6–10. The number of UDP flows increased from 1 to 9 with steps equal to 2. Real test results show a significant difference in throughput achieved between TCP and UDP traffic. When the number of TCP and UDP flows increased from 1 to 9, respectively, the throughput of TCP traffic decreased by 60.8%, and the throughput of UDP traffic reduced by 15.3%. The reason is that TCP flow can adjust the sending rate using congestion control. Consequently, UDP traffic obtains more bandwidth than TCP traffic, which leads to unfair channel access. Test cases 11–15 study the scenario when TCP and UDP share the wireless network. Due to the aggressive characteristic of UDP traffic, the total throughput achieved by TCP and UDP was higher compared to TCP traffic only.

It was observed among iBE, DietTOPP, and IdleGap that DietTOPP produced the highest error rate and IdleGap achieved the lowest error rate. In addition, MBE achieved 47% less error rate than IdleGap. Two-tailed T-test analysis shows that there is no significant statistical difference between MBE and real test results with 95% confidence level. By looking at the mean value, it can be concluded that MBE achieves the lowest error rate.

TABLE VII
MEAN AND STANDARD DEVIATION OF ESTIMATION ERROR AND OVERHEAD FOR iBE, DIETTOPP, IDLEGAP, AND MBE

	iBE		DietTOPP		IdleGap		MBE	
	Mean	**STDEV**	**Mean**	**STDEV**	**Mean**	**STDEV**	**Mean**	**STDEV**
Estimation Error	9.7%	0.16	17.6%	0.26	6.2%	0.08	3.3%	0.03
Overhead	0.35Mb	0.15	1.25Mb	0.14	0.53Mb	0.33	0.21Mb	0.1

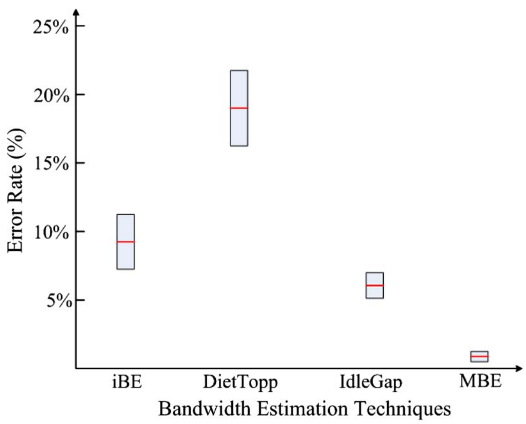


Fig. 14. Mean and standard deviation of error rate for iBE, DietTOPP, IdleGap, and MBE.

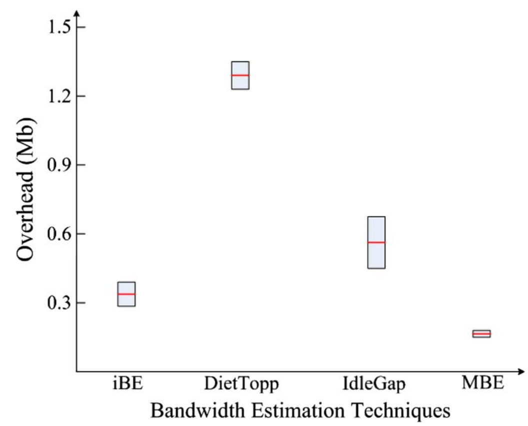


Fig. 15. Mean and standard deviation of overhead for iBE, DietTOPP, IdleGap, and MBE.

Notably, the throughput measured by simulation and real test was slightly higher than that of MBE in most cases. There are two reasons. First, the MBE model assumes that for each packet to be transmitted, the station invokes backoff mechanism and waits for a DIFS period. However, in simulation and real test, the packets might be transmitted immediately without backoff delay when the channel is sensed idle. Second, both simulation and real tests use buffers to improve the system performance.

Scenario B-2—Overhead Analysis: Similar to the setup in Scenario B-1, Scenario B-2 also used 15 cases with variable TCP and UDP traffic loads. The overhead introduced by MBE came from feedback traffic. Table VI shows the comparison results between MBE and other bandwidth estimation techniques in terms of overhead.

Experimental result analysis: For all the 15 test cases, the overhead increases with increasing number of contending flows. Among iBE, DietTOPP, and IdleGap, DietTOPP created the highest overhead since DietTOPP continually sends probing traffic. iBE introduced low overhead, but MBE has 18% lower overhead than iBE, as it relies on small feedback packets. The main difference between MBE and iBE is that the former requires packet loss information while the latter deals with packet received times. It should be noted that applications using TCP traffic caused lower overhead than those using UDP traffic. This might be explained by the fact that TCP ACK packets compete with feedback packets and therefore affect the throughput of the feedback traffic.

The mean and standard deviation of error rate and overhead for all the test cases are shown in Table VII and are further illustrated in Figs. 14 and 15, respectively. Among the existing bandwidth estimation algorithms, MBE achieved up to 89% lower standard deviation and 81% lower mean value than DietTOPP in terms of error rate. Furthermore, MBE obtained up to 70% lower standard deviation than IdleGap and 83% lower mean value than DietTOPP in terms of overhead.

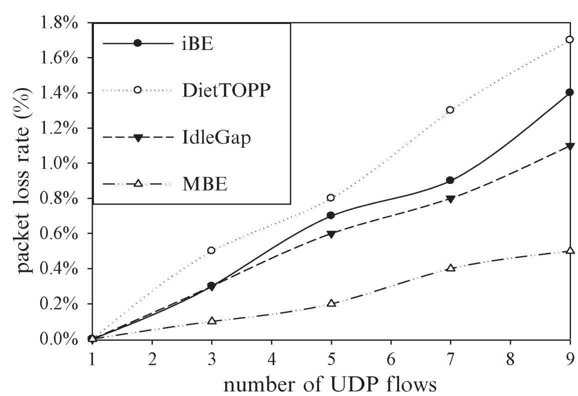


Fig. 16. Packet loss rate of UDP for iBE, DietTOPP, IdleGap, and MBE.

Scenario B-3—Loss Analysis: The purpose of scenario B-3 is to study the packet loss rate for different bandwidth estimation schemes. Fig. 16 shows the results of the packet loss rate evolution with increasing number of UDP traffic flows when iBE, DietTOPP, IdleGap, and MBE are used for bandwidth estimation, respectively.

Experimental result analysis: The number of UDP flows is increased from 1 to 9, and the bandwidth is estimated by four different bandwidth estimation schemes, i.e., iBE, DietTOPP, IdleGap, and MBE. It is shown in Fig. 16 that DietTOPP produces the highest packet loss rate of up to 1.7% for nine UDP flows, since DietTOPP continuously sends probing traffic that contends with UDP traffic. When using MBE to estimate the bandwidth, the packet loss rate is the lowest in comparison with all the other solutions. For instance, for nine UDP flows when MBE is employed, the loss rate is only 0.4%. It is worth noting that in these conditions, when using MBE, the packet loss rate decreases with up to 65% in comparison with that of DietTOPP. In addition, MBE reduces packet loss with up to 56% in comparison with that of iBE and with up to 50% in comparison with that of IdleGap.

VI. Conclusion and Future Works

This paper has proposed a novel MBE to estimate the available bandwidth for TCP and UDP traffic over 802.11 WLANs. MBE is based on novel throughput models for TCP and UDP traffic over IEEE 802.11 WLANs, which are also proposed. In contrast with current wireless bandwidth estimation techniques, MBE is fully compatible with the IEEE 802.11 standard protocol, has higher estimation accuracy, and introduces lower overhead. MBE does not use additional probing traffic, which in turn reduces the required bandwidth resources. Experiments results show that the MBE model is robust under different conditions: variant packet size, PER, and dynamic wireless link. MBE provides accurate bandwidth estimation with low overhead in comparison with existing bandwidth estimation techniques such as iBE, DietTOPP, and IdleGap. Among the three compared techniques, IdleGap gives the smallest estimation error rate, and iBE introduced the lowest overhead. MBE achieves 47% less estimation error rate than IdleGap and 18% lower overhead than iBE. Additionally, MBE produces the lowest standard deviation and mean value for both error rate and overhead.

The results of MBE are expected to benefit wireless QoS solutions. For instance, accurate estimation on available bandwidth is significant for the resource allocation scheme [40]. MBE can also be utilized for the prioritized bandwidth allocation scheme [41] without using IEEE 802.11e [42].

In the future, MBE can be extended in IEEE 802.11e and IEEE 802.11n [43] networks. 802.11e provides multimedia QoS support by introducing traffic access categories and block acknowledgement mechanism at MAC layer. 802.11n improves the multimedia transmission quality by using group-based frame at MAC layer and MIMO technique at PHY layer. Since MBE is developed based on the original 802.11 DCF, and 802.11e and 802.11n are also based on the 802.11 DCF protocol, MBE will also work in 802.11e and 802.11n. Future works will report the results of MBE in 802.11e/n networks.

References

[1] R. S. Prasad, M. Murray, C. Dovrolis, and K. C. Claffy, "Bandwidth estimation: Metrics, measurement techniques, and tools," *IEEE Netw.*, vol. 17, no. 6, pp. 27–35, Nov./Dec. 2003.

[2] S. Shah, K. Chen, and K. Nahrstedt, "Dynamic bandwidth management in single-hop ad hoc wireless networks," *J. Mobile Netw. Appl.*, vol. 10, no. 1/2, pp. 199–217, Feb. 2005.

[3] M. Li, M. Claypool, and R. Kinicki, "Playout buffer and rate optimization for streaming over IEEE 802.11 wireless networks," *ACM Trans. Multimedia Comput., Commun. Appl.*, vol. 5, no. 3, p. 26, Aug. 2009.

[4] Q. Shen, X. Fang, P. Li, and Y. Fang, "Admission control based on available bandwidth estimation for wireless mesh networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 5, pp. 2519–2528, Jun. 2009.

[5] C. H. Muntean, "Improving learner quality of experience by contenet adaptation based on network conditions," *J. Comput. Human*, vol. 24, no. 4, pp. 1452–1472, Jul. 2008.

[6] C. H. Muntean and J. McManis, "The value of QoE-based adaptation approach in educational hypermedia: Empirical evaluation," in *Proc. 4th Int. Conf. Adaptive Hypermedia Adaptive Web-Based Syst.*, vol. 4018, *Lecture Notes in Computer Science (LNCS)*, Dublin, Ireland, Jun. 21–23, 2006, pp. 121–130.

[7] J. C. Fernandez, T. Taleb, M. Guizani, and N. Kato, "Bandwidth aggregation-aware dynamic QoS negotiation for real-time video streaming in next-generation wireless networks," *IEEE Trans. Multimedia*, vol. 11, no. 6, pp. 1082–1093, Oct. 2009.

[8] J. Strauss, D. Katabi, and F. Kaashoek, "A measurement study of available bandwidth estimation tools," in *Proc. ACM SIGCOMM Conf. Internet Measurement*, Miami Beach, FL, Oct. 2003, pp. 39–44.

[9] M. Jain and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput," *IEEE/ACM Tran. Netw.*, vol. 11, no. 4, pp. 537–549, Aug. 2003.

[10] C. Dovrolis, P. Ramanathan, and D. Moore, "Packet-dispersion techniques and a capacity-estimation methodology," *IEEE/ACM Trans. Netw.*, vol. 12, no. 6, pp. 963–977, Dec. 2004.

[11] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell, "Pathchirp: Efficient available bandwidth estimation for network paths," in *Proc. 4th Passive Active Meas. Workshop*, La Jolla, CA, Apr. 2003.

[12] N. Hu and P. Steenkiste, "Evaluation and characterization of available bandwidth probing techniques," *IEEE J. Sel. Areas. Commun.*, vol. 21, no. 6, pp. 879–894, Aug. 2003.

[13] S. Saroiu, P. K. Gummadi, and S. D. Gribble, Sprobe: A Fast Technique for Measuring Bottleneck Bandwidth in Uncooperative Environments, Aug. 2001. [Online]. Available: http://sprobe.cs.washington.edu/

[14] M. Li, M. Claypool, and R. Kinicki, "WBest: A bandwidth estimation tool for IEEE 802.11 wireless networks," in *Proc. IEEE Int. Conf. LCN*, Montreal, QC, Canada, Oct. 2008, pp. 374–381.

[15] A. Johnsson, B. Melander, and M. Björkman, "DietTOPP: A first implementation and evaluation of a simplified bandwidth measurement method," in *Proc. Swedish Nat. Comput. Netw. Workshop*, Karlstad, Sweden, 2004.

[16] Z. Yuan, H. Venkataraman, and G.-M. Muntean, "iBE: A novel bandwidth estimation algorithm for multimedia services over IEEE 802.11 wireless networks," in *Proc. 12th IFIP/IEEE Int. Conf. Manage. Multimedia Mobile Netw. Services*, Venice, Italy, Oct.2009, pp. 69–80.

[17] H. K. Lee, V. Hall, K. H. Yum, K. I. Kim, and E. J. Kim, "Bandwidth estimation in wireless LANs for multimedia streaming services," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Toronto, ON, Canada, Jul. 2006, pp. 1181–1184.

[18] T. Sun, G. Yang, L. Chen, M. Y. Sanadidi, and M. Gerla, "A measurement study of path capacity in 802.11b based wireless networks," in *Proc. Int. Workshop Wireless Traffic Meas. Model.*, Seattle, WA, Jun. 2005, pp. 31–37.

[19] K. Lakshminarayanan, V. N. Padmanabhan, and J. Padhye, "Bandwidth estimation in broadband access networks," in *Proc. ACM SIGCOMM Conf. Internet Meas.*, Sicily, Italy, Oct. 2004, pp. 314–321.

[20] J. Mahdavi and S. Floyd, "TCP-friendly unicast rate-based flow control," Technical Mote Sent to the end2end-Interest Mailing List, 1997.

[21] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP reno performance: A simple model and its empirical validation," *IEEE/ACM Trans. Netw.*, vol. 8, no. 2, pp. 133–145, Apr. 2000.

[22] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *Proc. ACM SIGCOMM Conf. Appl., Technol., Architect., Protocols Comput. Commun.*, Vancouver, BC, Canada, 1998, pp. 303–314.

[23] J. Vieron and C. Guillemot, "Real-time constrained TCP-compatible rate control for video over the Internet," *IEEE Trans. Multimedia*, vol. 6, no. 4, pp. 634–646, Aug. 2004.

[24] B. Wang, J. Kurose, P. Shenoy, and D. Towsley, "Multimedia streaming via TCP: An analytic performance study," *ACM Trans. Multimedia Comput., Commun. Appl.*, vol. 4, no. 2, p. 16, May 2008.

[25] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas. Commun.*, vol. 18, no. 9, pp. 535–547, Mar. 2000.

[26] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma, "Performance of reliable transport protocol over IEEE 802.11 wireless LAN: Analysis and enhancement," in *Proc. IEEE Conf. Comput. Commun.*, New York, Jun. 2002, pp. 599–607.

[27] P. Chatzimisios, A. C. Boucouvalas, and V. Vitsas, "Performance analysis of IEEE 802.11 DCF in presence of transmission errors," in *Proc. IEEE ICC*, Paris, France, Jun. 2004, pp. 3854–3858.

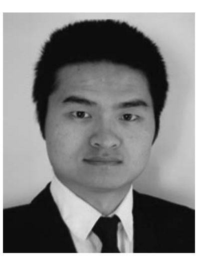
[28] C. Wang, T. Lin, and K. W. Chang, "On throughput performance of channel inequality in IEEE 802.11 WLANs," *IEEE Trans. Wireless Commun.*, vol. 7, no. 11, pp. 4425–4431, Nov. 2008.

[29] K. Hong, S. Lee, and N. Golmie, "Throughput study for admission control in IEEE 802.11 DCF with ARF," *IEEE Commun. Lett.*, vol. 13, no. 6, pp. 432–434, Jun. 2009.

[30] Y. Xi, B. S. Kim, J. Wei, and Q. Y. Huang, "Adaptive multirate auto rate fallback protocol for IEEE 802.11 WLANS," in *Proc. IEEE Military Commun. Conf.*, Washington, DC, Oct. 2006, pp. 1–7.

[31] Z. Li, A. Das, A. K. Gupta, and S. Nandi, "Full auto rate MAC protocol for wireless ad hoc networks," *Proc. Inst. Elect. Eng.—Commun.*, vol. 152, no. 3, pp. 311–319, Jun. 2005.

[32] M. Ashraf and A. Jayasuriya, "Improved opportunistic auto rate protocols for wireless networks," in *Proc. IEEE Int. Symp. Pers., Indoor Mobile Radio Commun.*, Cannes, France, Sep. 2008, pp. 1–6.
[33] J. Choi, J. Na, Y.-S. Lim, K. Park, and C.-K. Kim, "Collision-aware design of rate adaptation for multi-rate 802.11 WLANs," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 8, pp. 1366–1375, Oct. 2008.
[34] M. Vutukuru, H. Balakrishnan, and K. Jamieson, "Cross-layer wireless bit rate adaptation," in *Proc. ACM SIGCOMM*, Barcelona, Spain, Aug. 2009, pp. 3–14.
[35] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, RFC 2309: Recommendations on Queue Management and Congestion Avoidance in the Internet, Apr. 1998.
[36] W. Stevens, M. Allman, and V. Paxson, TCP Congestion Control RFC 2581, Apr. 1999
[37] R. Bruno, M. Conti, and E. Gregori, "Throughput analysis and measurements in IEEE 802.11 WLANs with TCP and UDP traffic flows," *IEEE Trans. Mobile Comput.*, vol. 7, no. 2, pp. 171–186, Feb. 2008.
[38] Network Simulator NS-2. [Online]. Available: http://www.isi.edu/nsnam/ns/
[39] RTP: A Transport Protocol for Real Time Applications, RFC 3550, 2003.
[40] T. Taleb and A. Nafaa, "A fair and dynamic auction-based resource allocation scheme for wireless mobile networks," in *Proc. IEEE Int. Conf. Commun.*, Beijing, China, May 2008, pp. 306–310.
[41] Z. Yuan, H. Venkataraman, and G.-M. Muntean, "iPAS: An user perceived quality-based intelligent prioritized adaptive scheme for IPTV in wireless home networks," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast.*, Shanghai, China, Mar. 2010, pp. 1–6.
[42] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment-Quality of Service Enhancements, IEEE Std. 802.11e, 2005.
[43] Wireless LAN MAC and PHY Specifications: Enhancements for Higher Throughput, IEEE Draft P802.11n/D.2.00, Feb. 2007.



Zhenhui Yuan (S'12) received the B.Eng. degree from Wuhan University, Wuhan, China, in 2008. He is currently working toward the Ph.D. degree with the Performance Engineering Lab, Dublin City University, Dublin, Ireland.

He was granted the national scholarship by the China Scholarship Council in 2008. His research interests include quality of service and wireless multimedia delivery.

Hrishikesh Venkataraman (M'07) received the Master's degree from the Indian Institute of Technology, Kanpur, India, in 2004 and the Ph.D. degree for his research on wireless cellular networks from Jacobs University, Bremen, Germany, in 2007.

He is currently a Research Fellow with the Performance Engineering Laboratory, Rince Research Institute, Dublin City University, Dublin, Ireland. He is an Enterprise Ireland principal Investigator and an Executive Editor of the *European Transactions on Telecommunications*. He has published around 40 papers in journals, international conferences, and book chapters and has edited two books. His research interests include mobile multimedia, energy-optimal devices/networks, and vehicular networks.

Gabriel-Miro Muntean (M'04) received the B.Eng. and M.Sc. degrees in software engineering from "Politehnica" University of Timisoara, Timisoara, Romania, and the Ph.D. degree for his research in the area of quality-oriented adaptive multimedia delivery from Dublin City University, Dublin, Ireland, in 2003.

He is currently a Lecturer with the School of Electronic Engineering, co-Director of the Performance Engineering Laboratory, and Director of the Network Innovations Centre, Dublin City University. He has published more than 130 papers in top-level international journals and conferences and has authored a book and ten book chapters and edited five books. His research interests include quality, performance, and energy saving-related issues of multimedia delivery, personalized e-learning, and other data communications over wired and wireless networks with various device types.

Dr. Muntean is a member of the IEEE Broadcast Technology Society, the Rince Research Institute, The Irish Software Engineering Research Centre, and the CloudCore Research Centre. He is an Associate Editor for the IEEE TRANSACTIONS ON BROADCASTING, IEEE Communications Survey and Tutorials, and a Reviewer for important international journals, conferences, and funding agencies.