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

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

***Abstract*— Wireless bandwidth estimation is a critical issue for Quality of Service (QoS) provisioning in IEEE 802.11 WLANs. Current bandwidth estimation solutions focus on either probing techniques or cross-layer techniques and either require 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 the IEEE 802.11 networks. The *MBE* module for available bandwidth estimation is developed based on novel TCP/UDP throughput models for wireless data communications. The novel aspects in comparison with other works include the fact that no probing traffic is required and no modification of 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 noise. Additionally, *MBE* has lower overhead and lower error rate than other state-of-the-art bandwidth estimation techniques.**

***Index Terms*—model, bandwidth estimation, IEEE 802.11**

I. INTRODUCTION

Recently, an increasing number of rich media applications exchange data over IEEE 802.11WLANs. Bandwidth estimation schemes have been widely 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] [5] [6]. Research in [7] shows 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.

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 which has a negative influence on the 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, the cross layer solutions require modifications of standard protocols which make it complex and not desirable.

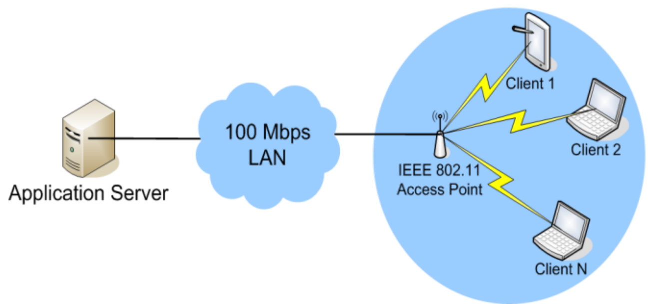


Fig. 1 Network architecture of wireless bandwidth estimation

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 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 UDP throughput model based on UDP packet transmission probability and IEEE 802.11 channel delay. Third, the paper derives a formula estimating the bandwidth when TCP and UDP traffic co-exists in IEEE 802.11 networks and proposes *MBE*. Note, unlike most existing estimation techniques, *MBE* neither require modification of current transmission protocols nor use the probing traffic.

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

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

This work was supported by the China Scholarship Council and Enterprise Ireland's Innovations Partnership Programme (IP/2010/0072).

Z.Yuan, H.Venkataraman and G.-M.Muntean are with PEL, NIC, School of Electronic Engineering (http://www.eeng.dcu.ie/~pel), Dublin City University, Ireland (e-mail: {yuanzh, hrishikesh, munteang}@eeng.dcu.ie).

II. Related Works

This section presents the related works regarding *MBE*. To begin with, the existing bandwidth estimation solutions are introduced and then subsequently, current models for TCP throughput and IEEE 802.11 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 the 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 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 layer and network layer. The idle module obtains the link idle rate from the Network Allocation Vector (NAV) 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 MAC layer by measuring the channel busy time, and use it to infer the available bandwidth. The probing-based techniques rely on the probing traffic which impact the wireless communication services due to the additional data introduced. Significantly, the 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 802.11MAC

Current models for analyzing the traffic throughput basically focus on TCP . To the best of our knowledge, Mahdavi et al. [20] propose 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] develope 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 considers UDP traffic and wireless network conditions.

IEEE 802.11 MAC protocol has been modeled in many research works. Bianchi [25] proposes a two dimensional Markov chain model to describe the 802.11 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 [26] improves Bianchi's model by introducing finite retry and also assumed saturated traffic. However, Wu's model failed to consider wireless errors. Recently, Chatzimisios [27] extends 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: 11Mbps, 5.5Mbps, 2Mbps, and 1Mbps. Link rate adaptation algorithms have been developed to dynamically adjust the data rate. *Auto Rate Fallback (ARF)* based solutions [28] [29] is one of the earliest rate adaptation algorithm. *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 achieves. *Adaptive Auto Rate Fallback (AARF)* [30] is developed based on *ARF* to resolve the bit-rate selection problem. *AARF* increases the threshold exponentially whenever the transmission attempt with the higher rate fails. *AARF* resets the threshold to the initial value when the rate is decreased and thereby provides support to both short-term and long-term adaptation. However, both *ARF* and *AARF* do not consider packet loss due to collision, and therefore, cannot apply for multi-stations scenario. *Receiver Based Auto Rate (RBAR)* based solutions [31] [32] use 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, Jaehyuk [33] 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 in order to estimate the data rates. Further, *SoftRate* [34] uses confidence information to estimate the prevailing channel BER which is calculated at 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 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 (*MBE*)

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.

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

TCP throughput model is described in equation (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 equation (2), where *N* indicates the number of contending stations, *BER* is the bit error rate, *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 equation (3), where *W* represents the initial contention window size and *m* means retry limit.

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

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

Chatzimisios has described a unique solution for (2) and (3) and has derived relation for the probability that at least one transmission occurs in a random time slot (P_{tr},). This could be written as shown in (4).

$$P_{tr} = 1 - (1-\tau)^{N} \tag{4}$$

When the retransmission reaches a retry limit *m*, the packet is dropped immediately. Consequently, we derived the drop probability P_{drop}, as presented in equation (5).

$$P_{drop} = P_{DCF}{}^{m+1} \tag{5}$$

However, the TCP throughput model does not offer accurate results in the 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 in order to consider wireless delivery conditions: 1) *packet loss probability update* (P_{tcp}); 2) *Round-trip Time* (*RTT*) *update*; 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: *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 RFC2309 [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 equation (6), where w_q is the weight factor.

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

$$P_{cong} = \begin{cases} 0 & \textit{if} \quad \bar{q}_{k+1} \le q_{\min} \\ 1 & \textit{if} \quad \bar{q}_{k+1} \ge q_{\max} \\ \dfrac{\bar{q}_{k+1} - q_{\min}}{q_{\max} - q_{\min}} & \textit{otherwise} \end{cases} \tag{7}$$

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 threshold of the queue. P_{cong} is collected in the sender's queue. Note that, *DropTail* can be considered a special case of *RED*.

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_{retr}^{TCP}) based on the 802.11 standard is derived as shown in equation (8), where:

$$\begin{aligned} P_{retr}^{TCP} &= P_{cong} + P_{DCF} + P(drop \mid DCF) \\ P(drop \mid DCF) &= \frac{P(DCF \mid drop) \times P_{drop}}{P_{DCF}} \end{aligned} \tag{8}$$

P(drop|DCF) refers to the packet drop probability of 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 attempts 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(DCF|Drop)* equals 1, as this dependency always exist.

Consequently, the probability of successful transmission, P_{succ}^{TCP} , is written as shown in equation (9).

$$P_{succ}^{TCP} = 1 - P_{retr}^{TCP} \tag{9}$$

RTT Update

As shown in Fig. 2 and Fig. 3, the overall delay for transmitting the data can be decomposed into seven components based on the OSI layers:

1) *App_Delay*: delay of application layer process such as video encoding/decoding, etc.

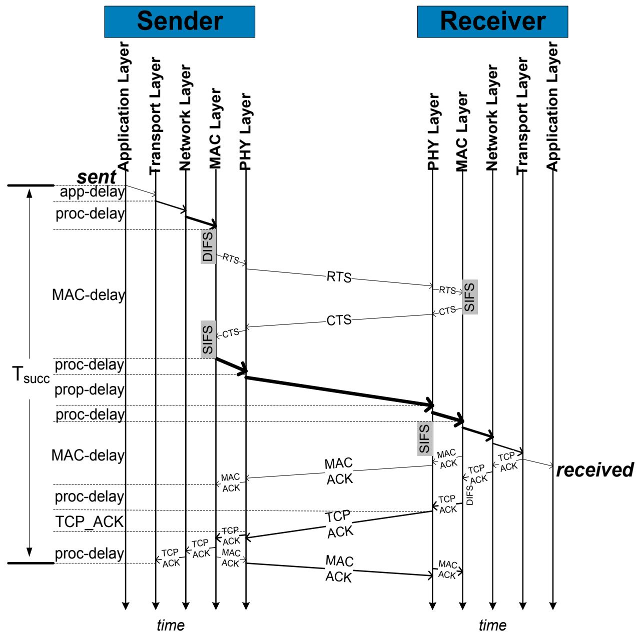


Fig. 2 Successful transmission when TCP runs over 802.11 networks

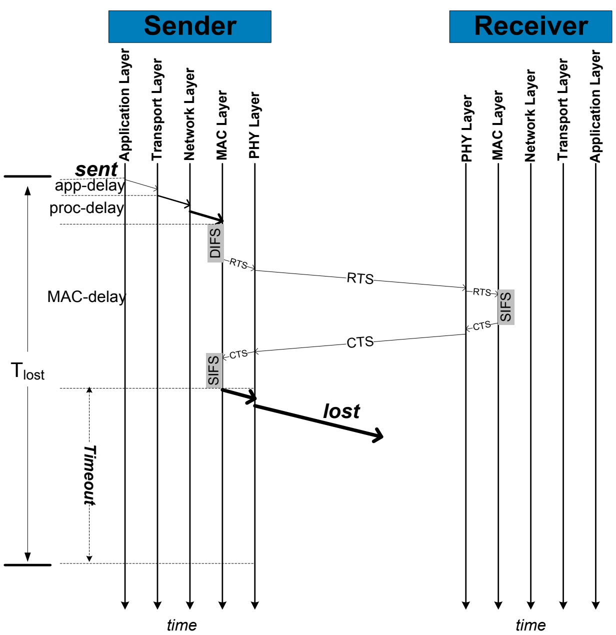


Fig. 3 Packet loss when TCP runs over 802.11 networks

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*** and ***retransmission***. The delay for successful transmission is denoted as T_{succ}. We derived equation (10) and equation (11) to present the 802.11 MAC layer delay for basic access mode (MAC_Delay_{basic}) and *RTS/CTS* mode (MAC_Delay_{RTS}), where *DIFS(Distributed Inter-Frame Space)* and *SIFS(Short Inter-Frame Space)* are contention control parameters defined in 802.11 MAC specifications. *MAC_ACK* represents the acknowledgment packet sent by the MAC receiver.

$$MAC\_Delay_{basic} = DIFS + SIFS + MAC\_ACK \tag{10}$$

$$MAC\_Delay_{RTS} = DIFS + 3\times SIFS + RTS + CTS + MAC\_ACK \tag{11}$$

Combining equation (10) and (11), the delay for successful transmission is given by (12), where *TCP_ACK* represents the acknowledgment packet sent by the TCP receiver. Note that, the propagation delay is the time taken to transmit data which includes the original data packet plus the stack protocol header.

$$T_{succ}^{TCP} = APP\_Delay + \Pr oc\_Delay + \{MAC\_Delay_{basic}, MAC\_Delay_{RTS}\} + \Pr op\_Delay + TCP\_ACK \tag{12}$$

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 an ACK after waiting a period equal with the timeout (T_o^{TCP}). RFC 2581 gives suggestions on how to calculate timeout, as shown in equations (13), (14) and (15). In equation (13), 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 equation (14). Subsequently, T_o^{TCP} is given by equation (15) based on dynamic timeout adjustment. A typical TCP implementation uses α=0.875 and β =0.75.

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

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

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

Further, the delay (T_{lost}^{TCP}) caused by timeout is subsequently given by equation (16),

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

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_{succ}^{TCP}. The average retransmission delay $T_{\mathrm{retr}}^{\mathrm{TCP}}$ is derived in equation (17). The retransmission delay can be T_{succ}^{TCP} or T_{lost}, depending on how the retransmission is triggered: three duplicate ACKs or the timeout.

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

Combination of TCP model and 802.11DCF model

By combining (4), (8), (9), (12), and (17), the new Round-Trip Time (*MRTT)* is written as shown in equation (18),

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

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 (equation (1)) includes the time computed at transport layer only.

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

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

If the network, device and the application service remains same for a user, then the *MBE* would need to know values of only 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 equations (8) and (18) which consider TCP fast retransmission and timeout respectively should be removed in *MBE*'s UDP version. By combining equations (2) and (5), the probability of retransmission when UDP traffic run over 802.11 networks can be written as shown in equation (20):

$$P_{retr}^{UDP} = P_{DCF} + P_{drop} \quad (20)$$

Similar to the TCP transmission delay described in equation (12), the UDP transmission delay can be derived and is shown in equation (21) and equation (22), respectively:

$$T_{succ}^{UDP} = APP\_Delay + \Pr oc\_Delay + \{MAC\_Delay_{basic}, MAC\_Delay_{RTS}\} + \Pr op\_Delay \quad (21)$$

$$T_o^{UDP} = \Pr op\_ACK + \Pr op\_UDP + SIFS \quad (22)$$

Furthermore, the retransmission delay is triggered by 802.11time out mechanism as given in equation (23):

$$T_{retr}^{UDP} = \Pr oc\_Delay + MAC\_Delay + T_o^{UDP} \quad (23)$$

Importantly, the average delay, *Delay_UDP*, for successfully transmitting the individual UDP packet could be written as in equation (24):

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

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

$$B^{UDP} = \frac{\int_{T_0}^{T_1} \frac{Payload}{Delay\_UDP} dt}{T_1 - T_0} \quad (25)$$

D. MBE for Co-Existing TCP and UDP Traffic

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

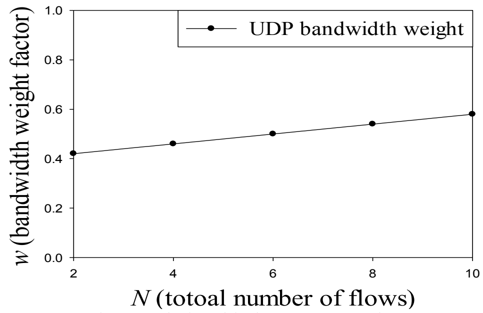


Fig. 4 Relationship between *w* and *N*.

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. In order to address this effect of UDP on the TCP traffic, the weight *w* is introduced, as shown in Fig. 4 and equation (26).

By combining the TCP and UDP over WLAN throughput models, the estimated aggregated throughput for co-existing TCP and UDP can be written as shown in equation (26):

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

The parameter *w* is the bandwidth weight factor, *N* represents the total number of TCP and UDP flows, *i* and *j* are the number 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 flows and UDP flows is equal. Fig. 4 shows the relationship between *w* and *N*. The throughput of UDP increases linearly 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. In order 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 in equation (27).

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

A similar comparison of the relationship between the TCP and UDP flows was done by Bruno in [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 equation (26) can be applied for real world TCP and UDP traffic mix scenarios.

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: NOAH[1] and Marco Fiore patch[2].

NOAH (No Ad-Hoc) 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, the LANForge acts as a server which generates traffic transmitted via a 100Mbps Ethernet and a Linksys WRV210 access point to multiple virtual wireless stations. The transmission power of AP is 20dBm through two omni-directional antennas. *MBE* configures the input parameters based on the IEEE 802.11b specifications, as shown in Table I, where MSS = 1500, b = 2, $DIFS$ = 50µs, $SIFS$ = 10µs, slot time = 20µs, TCP/IP protocol header = 40bytes and MAC protocol header = 36bytes. Each traffic connection consists of one server-wireless station pair.

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 bytes. Both the simulation and real test used FTP/TCP as application traffic which used the entire wireless capacity. The sending buffer was set to 8K bytes. 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. Secondly, 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 100Mbps 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 which employ different types of techniques were selected for comparison. These include, non-probing technique-*iBE* [16], probing based technique-*DietTOPP* [15], and the 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 10ms as indicated by the authors [16]. *RTS/CTS* function was enabled to achieve 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 bytes probing packet and 250Kbps cross-traffic were used to obtain better estimation performance as indicated by the authors [15].

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

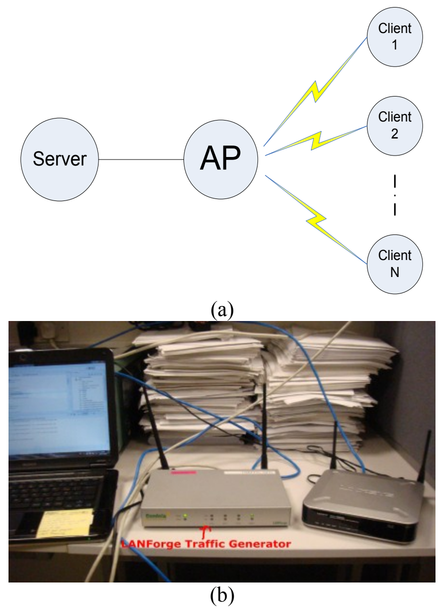


Fig. 5 (a) Test bed topology
(b) Real test bed including traffic generator and 802.11AP

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

IdleGap was implemented between 802.11 link layer and network layer. The cross-traffic for *IdleGap* was set to 10Kbps as suggested [17]. Application packet size was set to 700 bytes since *IdleGap* achieved good accuracy for packet size ranges from 512 bytes to 896 bytes. *RTS/CTS* was also enabled.

C. Evaluation Metrics

In order to evaluate the *MBE* performance, two estimation-based evaluation metrics were introduced: error rate and overhead. Error rate is defined as the difference between the *MBE* estimation results and the ground truth result. Lower error rate indicates higher accuracy of bandwidth estimation. The error calculation is given in (28).

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

Overhead is depicted as the total number of bytes sent by the model in order to perform the estimation. 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 *MBE* model; 2). evaluate bandwidth estimation quality. Generally, the robustness of *MBE* model depends on the feedback frequency, data packet size, wireless Packet Error Rate (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 *MBE* model in wireless network, separate tests were performed in terms of feedback frequency, packet size, packet error rate, and wireless link adaptation. For each test scenario, the variable-controlling method was adopted. Each scenario included a specific experimental setup which 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 the multimedia traffic. *MBE* uses RTCP Receiver Report [39] to deliver the feedback (8 bytes-RTCP receiver report packet header, 8 bytes bytes-UDP header, 20 bytes-IP header, and 4 bytes- feedback payload) due to the low cost and high reliability of this approach. Since the feedback size and the number of flows are relative 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 shown in equation (29).

$$FeedbackSize = RTCPheader + UDPheader + IPheader + Payload \tag{29}$$

The value of feedback size is 40bytes. Consequently, the feedback rate for each flow is given in equation (30).

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

When the number of flows is N and the time duration is T, the overhead can be computed by equation (31).

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

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 bytes as part of a single 6Mbps CBR/UDP. PER was set to $1\text{x}10^{-5}$. The mobile nodes stay close to AP at a distance smaller than 10m where the link data rate is 11Mbps. The duration of the experiment was 100s. The feedback interval was varied from 0.001s to 10.0s.

Experimental Result Analysis: Let α represents 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 in order to optimize the quality of application. By analyzing the results, the overhead introduced by *MBE* increases with the decrease of the feedback interval and the mean error rate changes with different feedback interval. For instance, in the case of feedback interval equals 1ms, the estimation overhead was 64Mb during 100s. This consists approximate 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 trade-off between the amount of overhead and mean error recommends a feedback interval of 1.0s.

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

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

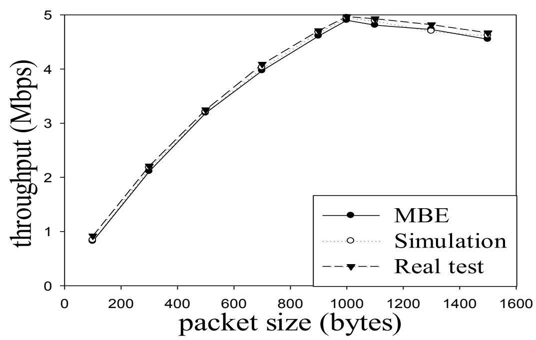


Fig. 6 Comparison of bandwidth as estimated by MBE and measured by NS-2 simulations and in the real-life tests for increasing packet size

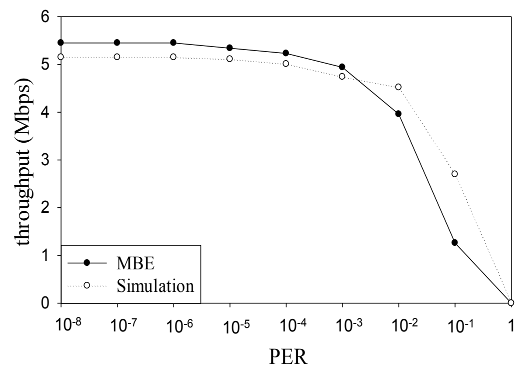


Fig. 7 PER effect on throughput

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.

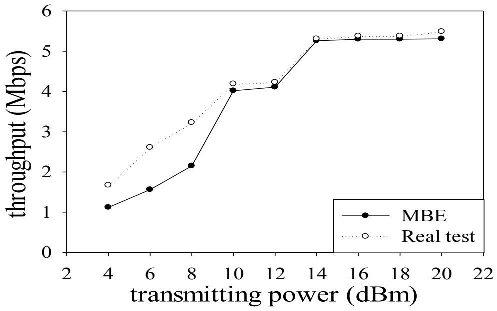


Fig. 8 Transmitting power effect on throughput

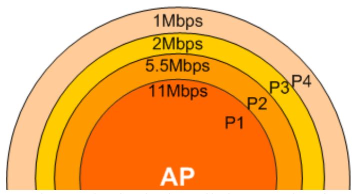


Fig. 9 Theoretical wireless link capacity for IEEE 802.11b

Experiment Setup: Both simulation and real test experiments were performed to study the impact of packet size. Single 6Mbps CBR/UDP traffic was sent from server to mobile station. Packet size was varied from 100 bytes to 1500 bytes (Ethernet MTU) with a step of 200 bytes. Feedback frequency was set to 1.0s. It was noticed that 6Mbps 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 10m where the link data rate was 11Mbps. Experiment time duration was set to 100s.

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 smaller packet size leads to more frequent transmissions and higher packet overhead. Throughput is the highest when packet size is 1000 bytes, as 1000 bytes was the fragmentation threshold. Any packets with size bigger than 1000 bytes are fragmented into multiple packets, resulting 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 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 Packet Error Rate (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 PER. 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 environment. Similar with tests setup in scenario A-1 and A-2, this experiment also transmitted single CBR/UDP traffic with packet size of 1000 bytes. Feedback frequency was set to 1.0s. NS-2 provides functions to increase the *PER* from $1\text{x}10^{-8}$ to 1. For each PER, there was a corresponding average packet loss ratio which 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 an external N-type antenna. Since the maximum transmission power of AP is 20dBm, the attenuator gradually reduced the transmitting power with a 2dBm step. For both simulation and real test, the mobile nodes stay close to AP at a distance smaller than 10m where the link data rate was 11Mbps. Experiment time duration was set to 100s and the feedback time interval was set to 1.0s. The bandwidth estimated by MBE is given based on the packet loss information under different simulation and real test conditions.

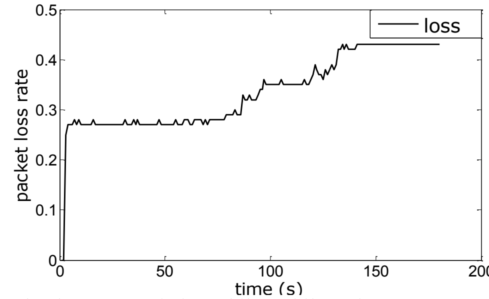


Fig. 10 Packet loss rate variation while mobile node moves away from AP

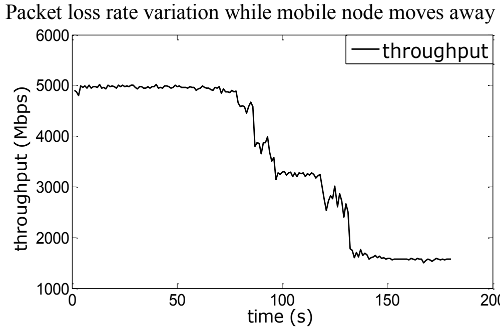


Fig.11 Throughput variation while mobile node moves away from AP

Experimental Result Analysis: Simulation and real test based results of PER influence are shown in Fig. 7 and Fig. 8. It was noticed that the available bandwidth generally decreases along with the increase of PER. The bandwidth equals zero when PER equals one. This implies that not 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 10dBm, the throughput start decreasing significantly. This can be explained by that the receiving signal strength might lower than 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 the wired networks, the capacity of wireless networks changes due to the link rate adaptation. The signal strength of 802.11b-enabled AP is divided into four sub-areas according to the link rate distribution defined in 802.11b, as shown in Fig. 9, Darker colors indicate higher signal strength.

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

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

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

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

Experimental Setup: Three test scenarios were implemented in the simulation environment to study the impact of the wireless link adaptation. They are: 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 20dBm. According to the documentation of the Cisco Linksys WRV210 this can cover around 300 meters. NS2 provided methods to calculate the distance threshold for the signal change: 70m (P1-P2), 100m (P2-P3), and 130m (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 6Mbps was sent from server to mobile station. The mobility was considered with the mobile station moving away from AP towards P4 at the speed of 1m/s. Fig. 10 and Fig. 11 shows the variations in throughput and packet loss during the transmission. Table III presents the comparison results between the simulation-based measured throughput and estimated bandwidth from *MBE*.

Experimental Result Analysis: It is clear from Fig. 10 and Fig. 11 that there is significant packet loss increase and throughput decrease as the mobile node moving 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 6Mbps CBR/UDP traffic were delivered in this scenario. Three test cases were considered in order to study the *MBE* performance in multiple stations 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 statically located at P1, P2, P3, P4.

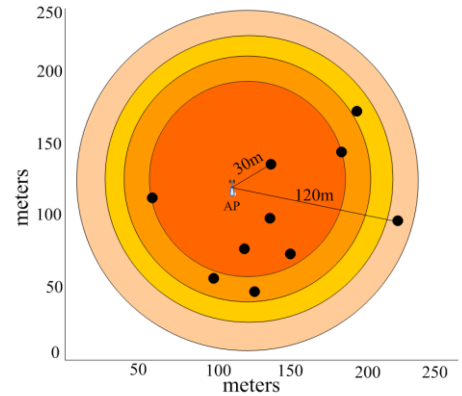


Fig. 12 Random topology in simulation.

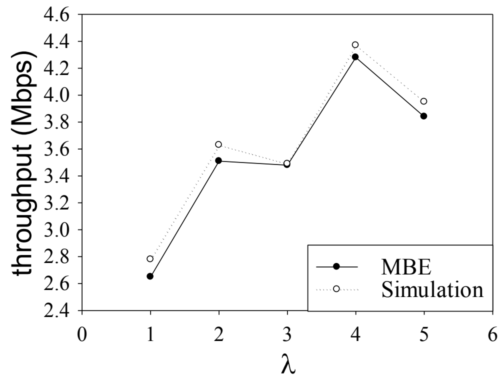


Fig. 13 Bandwidth comparison between MBE and simulation when λ increases from 1 to 5

- 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 results of case 1 and case 2 in Table IV, UDP traffic achieves higher throughput than TCP, since TCP can adapt the sending rate using congestion control.Additionally, by comparing results of case 1 and case 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 6Mbps CBR/UDP are sent. A 250m x 250m 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 30m to 120m. 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 the mobile nodes with random location. Table V and Fig. 13 give the comparison results between the simulation-based measured throughput and estimated bandwidth from *MBE*. For λ smaller than 4, both estimated bandwidth and measured bandwidth increase with increasing number of flows, and the bandwidth starts decreasing when λ equals 5. The overall throughput of the application traffic close to the wireless capacity for λ equals 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 Table III, Table IV, and Table 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 6Mbps CBR/UDP traffic load. Test case 1 to test case 5 transmitted TCP traffic only, test case 6 to test case 10 transmitted UDP traffic only while test case 11 to test case 15 sent TCP and UDP traffic simultaneously. In order 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 tests scenarios A-1, A-2, and A-3, the feedback interval was set to 1.0s, packet size was 1000 Bytes and PER was set to 10^{-5}. The overall sending rate was greater than 6Mbps and less than 7Mbps. The mobile nodes are located close to AP at a distance smaller than 10m where the link data rate is 11Mbps. Testing time duration was 100s.

Scenario B-1: Error Rate Analysis

Scenario B-1 studies the error rate which 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: 15 test cases were implemented to study the error rate of *MBE* under variable traffic load. In single flow situation, such as case 6 and case 11, *IdleGap* provides better accuracy than *MBE* in comparison with results from real test. From test case 1 to 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 5 TCP flows, the estimated bandwidth by *MBE* is 3.12 Mbps. Similarly, the impacts of UDP traffic were studied, as shown from test case 6 to 10. The number of UDP flows increased from 1to 9 with steps equals 2. Real test results show a significant different 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 case 11 to 15 study the scenario when TCP and UDP sharing the wireless network. Due to the aggressive characteristic of UDP traffic, the total throughput achieved by TCP and UDP was higher comparing 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. Also, *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.

Notably, throughput measured by simulation and real test was slightly higher than that of *MBE* in most cases. There are two reasons. First, *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 the 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 setup in Scenario B-1, Scenario B-2 also used 15 cases with variable TCP and UDP traffic load. 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 the 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 is 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 Fig. 14 and Fig. 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 the error rate. Furthermore, *MBE* obtained up to 70% lower standard deviation than *IdleGap* and 83% lower mean value than *DietTOPP* in terms of overhead.

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

TABLE VII
MEAN AND STANDARD DEVIATION OF ESTIMATION ERROR AND OVERHEAD FOR iBE, DietTOPP, IdleGap, 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

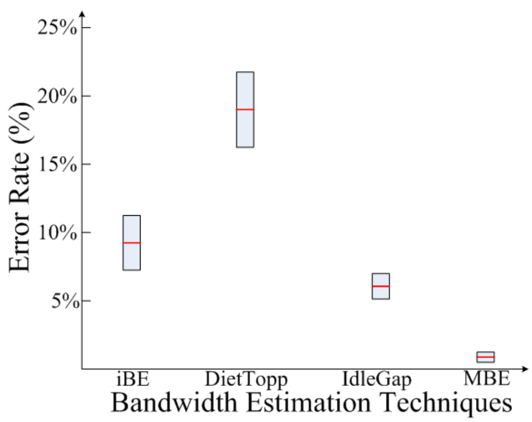


Fig. 14 Mean and standard deviation of error rate for iBE, DietTOPP, IdleGap and MBE

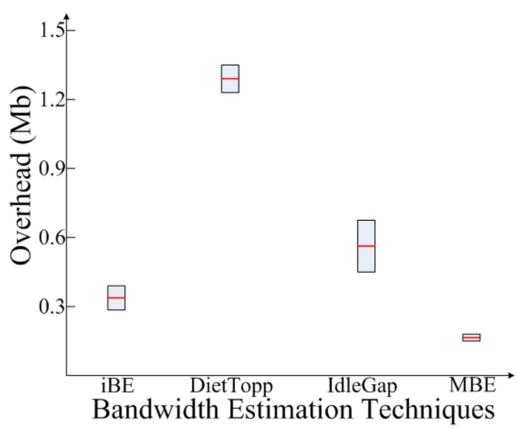


Fig. 15 Mean and standard deviation of overhead for iBE, DietTOPP, IdleGap and MBE

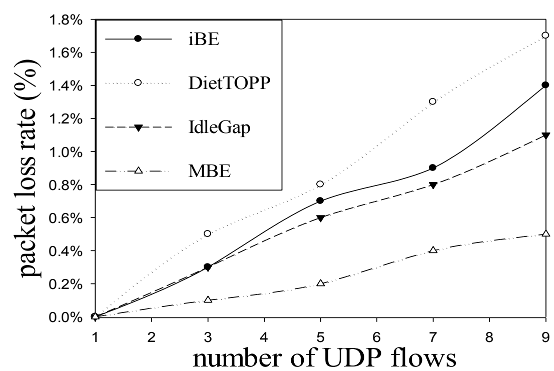


Fig. 16 Packet loss rate of UDP for iBE, DietTOPP, IdleGap and MBE

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, *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 9 UDP flows, since *DietTOPP* continuously sends probing traffic which contends with the UDP traffic. When using *MBE* to estimate the bandwidth, the packet loss rate is the lowest in comparison with all other solutions. For instance for 9 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*. Also, *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. Conclusions and Future Works

This paper proposes a novel Model-based Bandwidth Estimation algorithm (*MBE*) to estimate the available bandwidth for TCP and UDP traffic over 802.11WLANs. 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 *MBE* model is robust under different conditions: variant packet size, packet error rate 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 prioritized bandwidth allocation scheme [41] without using IEEE 802.11e [42].

In 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 Network*, vol. 17, no. 6, pp. 27-35, Dec. 2003.

[2] S. Shah, K .Chen and K. Nahrstedt, "Dynamic Bandwidth Management in Single-hop Ad Hoc Wireless Networks," *Journal. Mobile Networks and Applications*, 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 Computing, Communications and Applications*, vol. 5, no. 3, Aug. 2009.

[4] Q. Shen, X. Fang, P. Li, and Y. Fang, "Admission Control Based on Available Bandwidth Estimation for Wireless Mesh Networks," *IEEE Tran. Vehicular Technology*, vol. 58, no. 5, pp. 2519-2528, 2009.

[5] C. H. Muntean, "Improving Learner Quality of Experience by Contenet Adaptation based on Network Conditions," *Journal of Computers in Human Behavior*, vol. 24, no. 4, pp. 1452-1472, 2008.

[6] C. H. Muntean and J. McManis, "The Value of QoE-based Adaptation Approach in Educational Hypermedia: Empirical Evaluation," *4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, Dublin, Ireland, Springer-Verlag Berlin, Lecture Notes in Computer Science (LNCS) 4018, 21-23 June, 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", ACM SIGCOMM Conf. Internet Measurement, Miami Beach, FL, USA, 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. Networking*, vol. 11, no. 295-308, pp. 537-549, Aug. 2003.

[10] C.Dovrolis, P. Ramanathan, and D. Moore, "Packet-dispersion Techniques and A Capacity-Estimation Methodology," *IEEE/ACM Trans. Networking*, 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," 4th Passive and Active Measurement Workshop, La Jolla, CA, USA, Apr. 2003.

[12] N. Hu and P. Steenkiste, "Evaluation and Characterization of Available Bandwidth Probing Techniques," *IEEE Journal. Selected Areas. Communications*, 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: http://sprobe.cs.washington.edu/.

[14] M. Li, M. Claypool, and R. Kinicki, "WBest: A Bandwidth Estimation Tool for IEEE 802.11 Wireless Networks", *IEEE Int. Conf. Local Computer Networks (LCN)*, Montreal, Canada, Oct.2008, pp. 374-381.

[15]A.Johnsson, B.Melander, andM.Bjo¨rkman,"DietTOPP: A First Implementation and Evaluation of A Simplified Bandwidth Measurement Method," Swedish National Computer Networking Workshop, 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," *12th IFIP/IEEE Int. Conf. Management of Multimedia and Mobile Networks and 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," *IEEE Int. Conf. Multimedia and Expo*, Toronto, 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," *Proc. International Workshop on Wireless Traffic Measurements and Modeling*, Seattle, WA, USA, Jun. 2005, pp. 31-37.

[19] K. Lakshminarayanan, V. N. Padmanabhan, and J. Padhye, "Bandwidth Estimation in Broadband Access Networks," *Proc. ACM SIGCOMM Conf. Internet Measurement*, 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. Networking*, 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," *ACM SIGCOMM conference on Applications, technologies, architectures, and protocols for computer communication*, Vancouver, Canada, 1998.

[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 Computing, Communications and Applications*, vol. 4, no. 2, May. 2008.

[25] G.Bianchi,"Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE JSAC*, vol. 18, no. 9, pp.1787-1800, 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," *Proc. IEEE Conf. Computer Communications (INFOCOM),* New York, USA, 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," *IEEE Int. Conf. Communications (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 Communications*, 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 Comm. Letters*, 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 *IEEE Military Communications Conference*, Washington, U.S., 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," *IEE Proc. Comms*, no. 3, pp. 311- 319, 2005.

[32] M. Ashraf and A. Jayasuriya, "Improved Opportunistic Auto Rate protocols for wireless networks," *IEEE Int. Symposium on Personal, Indoor and Mobile Radio Communications*, Cannes, France, Sept.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 Journal. Selected Areas. Communications*, vol. 26, no. 8, pp. 1366-1375, Oct. 2008.

[34] M.Vutukuru, H.Balakrishnan and K.Jamieson, "Cross-Layer Wireless Bit Rate Adaptation," *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, L. Zhang, RFC 2309: Recommendations on Queue Management and Congestion Avoidance in the Internet, April 1998.

[36] W. Stevens, M. Allman, and V. Paxson, "TCP Congestion Control", RFC 2581, April 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 Computing*, vol. 7, no. 2, pp. 171-186, Feb. 2008.

[38] Network Simulator NS-2 [Online] http:// www.isi.edu/nsnam/ns/.

[39] "RTP: A Transport Protocol for Real Time Applications," RFC3550, 2003.

[40] T. Taleb and A. Nafaa, ―A Fair and Dynamic Auction-Based Resource Allocation Scheme for Wireless Mobile Networks, *IEEE International* Conference on Communications, Beijing, China, May.2008, pp. 306-310.

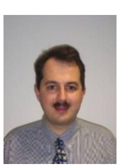
[41] Z. Yuan, H. Venkataraman, and G.-M. Muntean,"iPAS:An User Perceived Quality-based Intelligent Prioritized Adaptive Scheme for IPTV in Wireless Home Networks," IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, Shanghai, China, Mar.2010, pp.1-6.

[42] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment-Quality of Service Enhancements," IEEE 802.11e, IEEE Standard for Information Technology, 2005.

[43] Wireless LAN MAC and PHY Specifications: Enhancements for Higher Throughput, IEEE Draft P802.11n/D.2.00, Feb. 2007.

Zhenhui Yuan received his B. Eng. degree from Wuhan University, China, in 2008. He was granted the national scholarship by China Scholarship Council in 2008. He is currently working towards the Ph.D degree in the Performance Engineering Lab (PEL), Dublin City University, Ireland. His research interests include QoS and wireless multimedia delivery.

Hrishikesh Venkataraman received the Master's degree from the Indian Institute of Technology (IIT) Kanpur in 2004 and his PhD from Jacobs University, Bremen, Germany for his research on wireless cellular networks in 2007. He is currently a research fellow with Performance Engineering Laboratory in the RINCE research Institute, in Dublin, Ireland. He is an Enterprise Ireland principal Investigator and an Executive Editor of European Transactions on Telecommunications. His research interests include mobile multimedia, energy-optimal devices/networks and vehicular networks. He has published around 40 papers in journals, international conferences, book chapters and has edited 2 books till date.



Gabriel-Miro Muntean received the BEng and MSc degrees in Software Engineering from the Computer Science Department, "Politehnica" University of Timisoara, Romania and the PhD degree from Dublin City University, Ireland, for research in the area of quality-oriented adaptive multimedia delivery in 2003. He is a Lecturer with the School of Electronic Engineering, co-director of the Performance Engineering Laboratory and Director of the Network Innovations Centre at Dublin City University, Ireland. 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. 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. He is an associate editor for the IEEE Transactions on Broadcasting and a reviewer for important international journals, conferences, and funding agencies. He is a member of the IEEE, the IEEE Broadcast Technology Society, the Rince Research Institute, LERO and CloudCore Research Centres.