A Novel Bandwidth Estimation Algorithm for IEEE 802.11 TCP Data Transmissions

Zhenhui Yuan, Hrishikesh Venkataraman and Gabriel-Miro Muntean
Performance Engineering Lab, School of Electronic Engineering, Dublin City University, Ireland
Email: {yuanzh, hrishikesh, munteang}@eeng.dcu.ie

***Abstract*—Efficient bandwidth estimation is significant for Quality of Service (QoS) of multimedia services in IEEE 802.11 WLANs. Many bandwidth estimation solutions have been developed such as probing-based technique and cross-layer scheme. However, these solutions either introduce additional traffic or require modification to the standard protocols. This paper develops a model based bandwidth estimation algorithm at application layer when TCP traffic is delivered in the IEEE 802.11 networks. The proposed model firstly considers both TCP congestion control mechanism and IEEE 802.11 contention-based channel access mechanism. The bandwidth estimation process at the server estimates the achievable bandwidth using two types of parameters: data size to be sent and the feedback information from receivers, i.e. packet loss rate and the number of clients. The two tailed T-test analysis demonstrates that there is a 95% confidence level that there is no statistical difference between the results from the proposed bandwidth estimation algorithm and the results from the real test. Additionally, the proposed algorithm achieves higher accuracy and lower standard deviation of bandwidth, in comparison with other state-of-the art bandwidth estimation schemes.**

***Index Terms*—bandwidth estimation, TCP, IEEE802.11**

I. Introduction

Bandwidth estimation techniques have proved to be significant to improve the Quality of Service (QoS) for many multimedia applications [1] [2] [3] [4]. Most of these techniques are proposed to provide accurate bandwidth estimations for wired networks, however, bandwidth estimation for wireless networks is more challenging due to the wireless channel characteristics such as Bit Error Rate (BER), fading, interference, contention, retry limit, etc. Current wireless bandwidth estimation solutions can be grouped into two categories: 1) **Probing-based Technique**. *DietTOPP* [5] estimates the available bandwidth by comparing the adapted probing rate and the corresponding throughput, in order to find out the turning point. WBest [6] uses the packet-pair dispersion technique to estimate the effective capacity of the wireless networks and a packet train technique to infer mean and standard deviations of available bandwidth; 2) **Cross-layer Technique**. IdleGap [7] introduces an idle module located between the link layer and network layer. The idle module obtains the wireless link idle rate from the Network Allocation Vector (NAV) and sends it to the application layer. The bandwidth is then computed using link idle rate and the known capacity. Shah et al. [8] proposes another cross-layer solution to infer the available bandwidth based on the measured busy time of wireless channel. The probing-based techniques negatively impact the data traffic due to the additional probing traffic introduced. The cross-layer techniques have lower impact on existing traffic in comparison with the probing based techniques. However, they are difficult to be deployed due to the requirement of modifications on both devices and standard protocols.

This paper introduces a novel bandwidth estimation solution to predict the maximum achievable bandwidth for TCP-based data transmissions over IEEE 802.11 WLANs. Our solution extends Jitendra's TCP throughput model [9] to include the IEEE 802.11 characteristics (transmission error, contention, and retry attempts) based on Chatzimisios's 802.11 DCF model [10]. These two models are briefly described next.

The structure of the paper is as follows. Section II introduces the existing models for TCP throughput and IEEE 802.11 DCF mechanism. Section III presents the details of the proposed bandwidth estimation algorithm. Experimental setup and results analysis are presented in section IV. Finally, section V concludes the paper.

II. Existing Models

This section introduces the TCP throughput model and the IEEE 802.11 model that are utilized by our proposed algorithm.

1) *TCP Reno Throughput Model*

The fast retransmission and timeout mechanisms of TCP have been modeled by Jitendra's et al. [9], as shown 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 represents the timeout value to trigger the retransmission.

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

2) *IEEE 802.11 DCF Model*

Chatzimisios et al. [10] extend Bianchi's Markov Chain model on IEEE 802.11 DCF [11] by taking into account packet retry limits, collisions and propagation errors (fading, interference). The key assumption of Chatzimisios's model is that the transmission loss probability (P_{DCF}) of a transmitted

This work was supported by the China Scholarship Council and Enterprise Ireland.

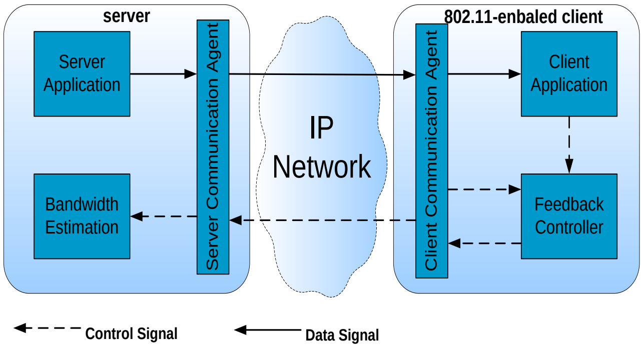

Fig 1. Block architecture of the proposed bandwidth estimation algorithm

packet is constant and independent of the number of the collisions or errors occurred in the past. The probability P_{DCF} is given by (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 (3), where W represents the initial contention window size and m means retry limit. Chatzimisios's model has proved that there is a unique solution for (2) and (3).

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

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

The probability that at least one transmission occurs in a random time slot, P_{tr} , is given in (4).

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

When the transmission loss reaches the retry attempt limit m, the packet would be dropped immediately. Consequently, the retry attempt drop probability P_{drop}, is given in (5).

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

III. Proposed Bandwidth Estimation Algorithm

This section introduces the architecture of the bandwidth estimation system and the details of the proposed bandwidth estimation algorithm.

1) *Block-based System Architecture*

Fig. 1 presents the architecture of the proposed bandwidth estimation system which consists of two main building blocks: *server* side module and *802.11-enabled client* side module. The *server* is responsible with sending multimedia traffic and estimating the achievable bandwidth of the 802.11 network. The *client* collects information of the delivered traffic, which is sent as feedback to the *server*. Multimedia traffic is delivered using TCP/IP protocol. Details of each sub-module in the proposed system are discussed next.

The communication between the server application and client application uses a control communication link which is established when the client sends a TCP connection request to the server. Subsequently, the multimedia communication link is created between the server and client allowing for multimedia data transmission.

The **Server Communication Agent (SCA)** and **Client Communication Agent (CCA)** located at both sides of the communication link are responsible with managing the transmission of multimedia traffic and control traffic. The CCA component maintains the receiving buffer and forwards feedback information to the **Feedback Controller (FC)** component. The SCA component maintains the sending buffer and forwards the feedback information received from FC to the **Bandwidth Estimation (BE)** component. The SCA also extracts the data size information and forwards to the BE component for bandwidth estimation.

The main function of FC component, situated at the client side, is to collect the feedback related parameters from client applications and CCA and send the feedback information to the CCA. There are two types of feedback parameters are gathered by the FC component: packet loss rate and the number of clients. The measurement of instant packet loss rate is done by analyzing the packets' sequence numbers as presented in [12]. Additionally, the **Client Application (CA)** component sends the client device's MAC address to FC and forward to SCA where the number of wireless clients is computed. The BE component then estimates the achievable bandwidth based on the feedback information. The details of the process in BE component are presented next.

2) *Bandwidth Estimation Algorithm*

The proposed algorithm updates the TCP throughout model by considering the 802.11 MAC-based channel contention mechanism. There are three steps to update the original TCP model: 1) *Packet Loss Update*; 2) *RTT Update*; 3) *Combination of TCP Model and 802.11DCF Model*.

A. *Packet Loss Update*

Queue overflow-related loss and *transmission loss* are the major packet loss when transmitting TCP traffic in the wireless networks. The queue-related loss depends on the queue scheduling algorithm adopted. The widely used *Random Early Discard (RED*) queuing protocol [13] is considered in this paper. *RED* determines the process of packet scheduling based on the current queue size (q_{k+1}), and updates the average queue size ($\bar{q}_{k+1}$) for each arrived packet. The average queue size is given in (6), where w_q is the weight factor.

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

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

The probability of packet loss caused by the *RED* queue (P_{queue}) is given in (7), where $q_{\min}$ and $q_{\max}$ denote the minimum and maximum threshold of the queue size. The parameter P_{queue} is captured by the SCA component where the sending buffer is maintained and sent to BE component.

In the case that packet loss occurs, the packet retransmission process is triggered by TCP and 802.11 MAC protocol. Since packet loss can be caused by queue drop, transmission loss over wireless channel or retry attempt drop, the probability of retransmission (P_{retr}) can be derived by (8). The parameters P_{DCF} and P_{drop} are computed based on feedback parameters such as packet loss rate and data packet size, using equations (2)-(5).

$$P_{retr} = P_{queue} + P_{DCF} + P_{drop} \tag{8}$$

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

Consequently, the probability of successful transmission, P_{succ}, is given in (9)

B. RTT Update

The end-to-end delay for TCP data transmission can be decomposed into seven components based on the OSI model:

1) Application Layer Delay (*App_Delay*) - the delay at application layer.

2) Transport Layer Delay (*Transport_Delay*) - the delay cost to implement transport protocol such as TCP congestion control.

3) Network Layer Delay (*IP_Delay*) - the delay at the IP layer for routing protocol.

4) MAC Layer Delay (*MAC_Delay*) - the delay caused by the backoff due to MAC-based channel contention.

5) Physical Llayer Delay (*Phy_Delay*) - the delay at physical layer depending on raw bits transmission type.

6) Propagation Delay (*Prop_Delay*) - the delay caused by data transmission over the channel medium. Propagation Delay is the function of data size and medium type.

7) Terminal Processing Delay (*Proc_Delay*) - determined by terminal's processing ability such as CPU, memory, power mode, operation system, etc.

There three states for the receiver***: idle, successful transmission and retransmission***, in a typical round-trip time (*RTT*). The delay for successful transmission is denoted as T_{succ}. Equation (10) and (11) represent the 802.11 MAC layer delay for basic access mode (MAC_Delay_{basic}) and *RTS/CTS* mode (MAC_Delay_{RTS}), respectively, where *DIFS (Distributed Inter-Frame Space)* and *SIFS (Short Inter-Frame Space)* are contention control parameters defined in 802.11 MAC specifications. The parameter *MAC_ACK* represents the acknowledgment packet sent by the receiver of MAC layer.

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

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

The delay for successful transmission is given in (12), where the value *TCP_ACK* represents the acknowledgment packet sent by the TCP receiver. Note that the propagation delay indicates the time to transmit data including the original data packet and the packet header corresponding to different layers: TCP/IP/MAC.

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

TCP-Reno congestion control starts retransmission under two conditions: (1) three duplicate ACK packets received at the sender as defined in RFC2581; (2) the TCP sender doesn't receive ACK packet after waiting for a period equal to timeout (T_o). T_o is given in (13) based on the dynamic timeout adjustment. The variance of *RTT* (D_{RTT}) is given in (14) where *M* denotes the time taken for *ACK* to arrive and *α* is typically equals 7/8. The first condition makes TCP enter the fast retransmission and the delay caused by three duplicate ACD packets is denoted as T_{3ACK}, which is the same as T_{succ}. The delay for the second condition, T_{lost}, is given in (15),

$$T_o = RTT + 4 \times D_{RTT} \tag{13}$$

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

$$T_{lost} = \Pr oc\_Delay + MAC\_Delay + T_o \tag{15}$$

Consequently, the average retransmission delay is given in (16). The retransmission delay could be T_{succ} or T_{lost}, depending on how the retransmission is triggered according to the two conditions mentioned before, i.e., three duplicate ACKs or the timeout.

$$T_{retr} = \{T_{3ACK}, T_{lost}\} = \{T_{succ}, T_{lost}\} \tag{16}$$

C. Combination of TCP Model and 802.11 DCF Model

By combining (4), (8), (9), (12) and (15), the new Round-trip Time *MRTT*, for data transmission is described in (17), where σ is the constant MAC slot time.

$$MRTT = (1 - P_{tr}) \times \sigma + P_{retr} \times T_{retr} + P_{succ} \times T_{succ} \tag{17}$$

$$B' = \frac{MSS}{MRTT \times \sqrt{\frac{2bP_{retr}}{3}} + T_o \times \min\left(1, 3\sqrt{\frac{3bP_{retr}}{8}}) \times P_{retr} \times (1 + 32P_{retr}^{\ 2})\right)} \tag{18}$$

Note that, P_{tr} defined in 802.11MAC is directly adopted in the new model since it is independent of the stack protocols. Based on (1) (8) and (17), the application layer throughput *B'* for each TCP connection is described in (18), where *b* is the number of packets acknowledged by a received ACK.

Finally, the total bandwidth (B_{total}) is given by the summing of each application layer throughput as shown in (19), where *n* is the total number of contending stations and *k* indicates the k^{th}

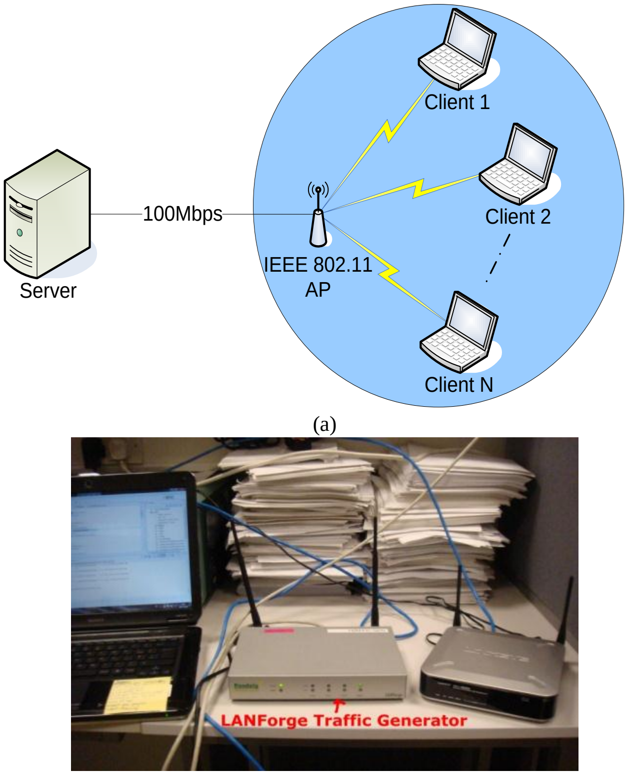

Fig. 2 (a) Simulation test bed topology (b) real test bed including traffic generator and 802.11AP

$$B_{total} = \sum_{k=1}^{n} B'_{k} \qquad (19)$$

station among the *n* stations. Suppose the network, device and the application service remains same for a user, then the proposed bandwidth estimation algorithm would need to know values of only two types of parameters:

1) **Static parameters**: application delay, processing delay, 802.11 MAC-related parameters such as minimum contention window, *DIFS*, *SIFS*, slot time and retry attempt limit. The static parameters are constant and predefined by the protocols.

2) **Dynamic parameters:** the number of contending clients, packet loss rate and data packet size to be sent. They are required by equation (2) to obtain the wireless transmission loss probability.

D. *Implementation*

The purpose of bandwidth estimation is to serve the higher level adaptive scheme such as [3] [4]. In practical, the proposed bandwidth estimation algorithm is implemented into a separate agent server which runs on Linux system with IEEE 802.11 wireless interface. The agent acts as the *server* module as shown in Fig. 1. Feedback information is collected at each 802.11-enabled device and sent back to the agent. The agent then performs the bandwidth estimation process. Since Linux system provides two types of timer, millisecond (waiting period) and microsecond (busy period), additional timer is implemented to present a unified microsecond resolution timer.

IV. Experimental Setup and Results Analysis

This section describes the experimental setup under both simulation and real test bed and the experimental results analysis.

1) *Experimental Setup*

The proposed algorithm has been evaluated using the NS-2.23 [14] simulator and Candela Technologies' LANForge traffic generator V4.9.9-based network test bed. Two additional wireless update patches are deployed in NS-2 set-up: NOAH [15] and Marco Fiore patch [16]. NOAH (No Ad-Hoc) was used for simulating the infrastructure WLAN environment and Marco Fiore's patch provides a more realistic wireless network environment. Both setups used 802.11b networks as shown in Fig. 2. 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 input parameters were configured based on IEEE 802.11b specifications, where *MSS*=1000, *DIFS*=50 μs, *SIFS*=10 μs, slot time=20 μs, TCP/IP protocol header=40bytes and MAC protocol header=36bytes. Each traffic connection consists of one server-wireless client 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 traffic which used the entire wireless capacity. The sending buffer was set to 8K bytes, which is the most common configuration in the industry.

There were two assumptions considered in the tests: 1) the application delay and processing delay were ignored. This is reasonable because the IP packet application and 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; 2) the wireless network was the bottleneck link of the end-to-end path. This was supported by connecting with a 100Mbps wired LAN. So the bandwidth estimation can closely reflect the wireless network condition.

Two existing wireless bandwidth estimation tools were selected for the comparison: probing-based technique-*DietTOPP* [5] and cross layer-based technique-*IdleGap* [7]. They were implemented using the default configuration under NS-2.

DietTOPP relies on probe packet size and cross-traffic when the bottleneck is a wireless link. Hence, 1500 bytes probing packet and 250Kbps cross-traffic were used to obtain better estimation performance as indicated by the authors.

The *IdleGap* algorithm was implemented between the 802.11 link layer and network layer. The cross-traffic for *IdleGap* was set to 10Kbps as suggested. 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* function was also enabled.

2) *Evaluation Metric*

In order to evaluate the performance of the proposed algorithm the error rate is introduced. Error rate is defined as

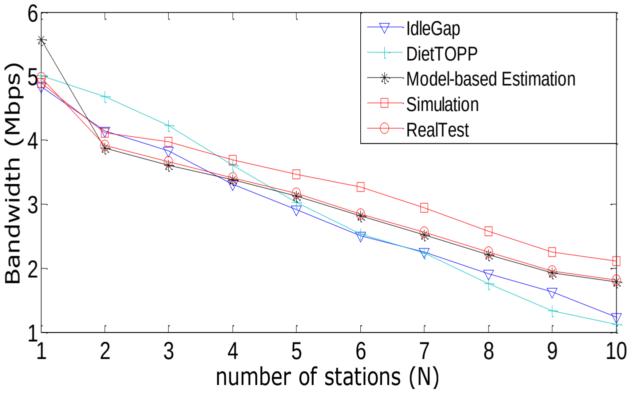

Fig. 3 Comparison results between the proposed algorithm, simulation and real test.

TABLE I

COMPARISON RESULTS AMONG ESTIMATION, SIMULATION AND REAL TEST

N	IdleGap (Mbps)	DietTOPP (Mbps)	Estimation Bandwidth (Mbps)	Simulation Bandwidth (Mbps)	Real Bandwidth (Mbps)
1	4.84	5.01	5.57	4.89	4.97
2	4.14	4.68	3.88	4.12	3.92
3	3.83	4.23	3.61	3.98	3.66
4	3.31	3.61	3.38	3.69	3.41
5	2.92	3.02	3.12	3.47	3.17
6	2.51	2.53	2.81	3.26	2.84
7	2.25	2.24	2.52	2.94	2.56
8	1.91	1.76	2.21	2.57	2.25
9	1.63	1.33	1.92	2.25	1.95
10	1.23	1.12	1.79	2.11	1.81

N: number of stations.

the difference between the estimation results and the ground truth (real test) result. The ground truth values are obtained based on the measurement test bed as shown in fig. 2 (b). Lower error rate indicates higher accuracy of bandwidth estimation. The computation of error rate is given in (20).

$$ErrorRate = \frac{\left|ESTIMATEDBandwidth - REALBandwidth\right|}{REALBandwidth} \qquad (20)$$

3) *Experimental Scenario*

The number of wireless stations was increased from 1 to 10 to test the estimation performance under different traffic load. Each wireless station receives the FTP/TCP traffic from the server to saturate the 802.11 channel. In this scenario, any incoming traffic will decrease the overall throughput since the achievable throughput is higher than the wireless network capacity. The testing duration was set to 100s.

4) *Experimental Results Analysis*

It is observed from Fig. 3 that, under all three techniques (model-based estimation, *IdleGap*, and *DietTOPP*), the achievable bandwidth decreases with the increasing number of contending stations. For instance, the bandwidth decreases by 68%, 76% and 80% for model-based estimation, *IdleGap*, and *DietTOPP*, respectively, when the number of stations increases from 1 to 10. This is because the increasing amount of traffic introduces more collision, which increases the packet loss and therefore reduces the throughput. Moreover, the TCP sender can adapt to the poor network condition by reducing its transmission rate, leading to lower throughput.

According to Table I, following a two tailed t-test analysis it can be said with 95% confidence level that there is no statistical difference between the estimated results and those of the real test. The mean value and the standard deviation value of the error rate for *IdleGap* are lower than that of *DietTOPP*, with 23% and 32%, respectively. Additionally, the proposed algorithm achieves lower mean value and standard deviation value of the error rate than *IdleGap*, with 27% and 25%, respectively.

V. CONCLUSIONS AND FUTURE WORKS

This paper proposes a novel bandwidth estimation scheme which estimates the overall bandwidth for TCP traffic over 802.11WLANs. Real test and simulation results show that proposed algorithm provides higher accurate bandwidth estimations with increasing number of nodes (confidence level of 95%).

The proposed bandwidth estimation algorithm can also be extended for IEEE 802.11e [17] and IEEE 802.11p [18] protocols. 802.11e has been developed to provide QoS for multimedia services using traffic type categories and multiple frame transmission (TXOP) technique. IEEE 802.11p is an approved amendment to the IEEE 802.11 standard to incorporate wireless access in vehicular environments (WAVE). The MAC layer of 802.11p is derived from the 802.11e by excluding the TXOP function. Since the proposed algorithm is developed based on the original 802.11 DCF, and 802.11e and 802.11p are also based on the 802.11 DCF protocol, our scheme will also work in 802.11e and 802.11p. However, the high speed of mobile devices and fast network change is a critical challenge when applying the proposed algorithm in 802.11p. Future works will report the results of the bandwidth estimation in 802.11e/p networks.

REFERENCES

[1] 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, Jul.2008.

[2] C. H. Muntean and J. McManis, "Fine Grained Content-based Adaptation Mechanism for Providing High End-user Quality of Experience with adaptive hypermedia systems," *Proceedings of the 15th W3C International World Wide Web Conference (WWW2006)*, Edinburgh, UK, ACM Press, New York, U.S., 23-26 May , 2006, pp.53-62.

[3] G. M. Muntean, P. Perry and L. Murphy, "A comparison based study of quality-oriented video on demand", IEEE Trans. Broadcasting, vol. 53, no. 1, pp. 92-102, March 2007.

[4] 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.

[5] A. Johnsson, B. Melander, and M. Bjorkman, "DietTOPP: A First Implementation and evaluation of a simplified bandwidth measurement method," *Proceedings of SNCNW*, Karlstad University, Sweden, Nov. 2004.

[6] M. Li, M. Claypool and R. Kinicki, "WBest: A bandwidth estimation tool for IEEE 802.11 wireless networks", *IEEE International Conference on Local Computer Networks (LCN)*, Montreal, Canada, Oct.2008, pp.374-381.

[7] 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 International Conference on Multimedia and Expo (ICME)*, Toronto, Canada, Jul.2006, pp.1181-1184.

[8] 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.

[9] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, “Modeling TCP Reno Performance: A Simple Model and its Empirical Validation,” *IEEE/ACM Transactions on Networking*, no. 2, vol.8, pp.133-145, Apr.2000.

[10] P.Chatzimisios, A. C. Boucouvalas, and V. Vitsas, “Performance Analysis of IEEE 802.11 DCF in Presence of Transmission Errors,” in *Proceedings of IEEE International Conference on Communications (ICC)*, Paris, France, Jun.2004, pp.3854-3858.

[11] G.Bianchi, “Performance Analysis of the IEEE 802.11 Distributed Coordination Function,” *IEEE Journal of Selected Areas in Communications*, no. 9, vol. 18, pp. 1787-1800, Mar. 2000.

[12] G. Almes, S. Kalidindi, and M. Zekauskas, “A One-way Packet Loss Metrics for IP Performance Metrics (IPPM),” RFC 2680, Sept. 1999.

[13] 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.

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

[15] NS-2 extension, *http://icapeople.epfl.ch/widmer/uwb/ns-2/noah/*.

[16] M. Fiore patch, *http://www.telematica.polito.it/fiore*.

[17] “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.

[18] Draft amendment to standard for information technology telecommunications and information exchange between systems local and metropolitan area networks specific requirements Part 11: wireless LAN medium access control(MAC) and physical layer specifications Amendment 7: wireless access in vehicular environment, 2007.