Improving the Throughput of TCP over TDMA-based Random Access Links Using the delayed ACK

Nak Woon Sung* and Kyungsoo

Electronics and Telecommunications Research Institute(ETRI),

161 Kajong-dong, Yusong-Gu, Daejeon, Korea 305-350

{nwsung, kksu}@etri.re.kr}

Abstract—**In this paper, we focus on the effect of random access delay of TDMA MAC on the TCP performance over IEEE 802.16 d/e. The random access protocol of IEEE 802.16 d/e is based on orthogonal frequency-division-multiple-access (OFDMA)-code-division-multiple-access (CDMA) with time-division-duplexing (TDD) mode and increases round-trip-time (RTT) of TCP flows over uplink transmission. And to speed TCP throughput up without revising existing TCP implementations, RTT of wireless portion in the wired-cum-wireless environment among many factors needs to be reduced by the MAC layer. In this paper, we analyze the delay factor of the random access protocol of IEEE 802.16 d/e considering the implementation complexity and show that it is closely related to TCP performance. And we propose the algorithm to minimize that delay factor using the characteristic of TCP delayed ACK which is generally adopted by most of the TCP versions.**

Keywords-TCP; Random Access; TDMA; 802.16; MAC

I. INTRODUCTION

IEEE 802.16d wireless metropolitan area network (MAN) which has been standardized for band from 10 to 66 GHz has been emerged as a promising solution to provide broadband data to business and homes and as the alternative of the expensive wired infrastructures such as T1, DSL (Digital Subscriber Line) or Cable-modem-based connection BWA (Broadband Wireless Access) [1]. In particular, IEEE 802.16e including WiBro (Wireless Broadband Internet) which has been launched in Korea since 2006 can provide subscribers with mobility over 60 km/hour [2]. In this paper, we assume that the physical layer features orthogonal frequency-division-multiplexing (OFDM) with TDD mode and the MAC (Medium Access Control) layer features TDMA (Time Division Multiple Access) mode among various modes of IEEE 802.16a/b/c/d/e. OFDM/TDMA (Time-Division Multiple Access) TDD systems has proved to provide a good solution for high-speed packet transmission which has the specific features such as the discontinuous data transmission and the traffic asymmetry between uplink and downlink [3].

In this paper, we focus on the effect of random access delay of TDMA MAC on the TCP performance. We developed one test environment, which is WiBro testbed sponsored by Samsung and developed by our lab and supporting about total 4 Mbps data rates, and measured three types of real Internet throughput in the SS. For this measurement, a general lab top computer and above the mentioned SS are used and connected with PCMCIA interface and multimedia movie file is streaming-serviced into a web browser, Microsoft Internet Explorer, using a media player, Microsoft Media Player, with each streaming protocol option as UDP or TCP with and without random access process from wired Internet Server. In these tests, we can see the definite throughput difference between UDP and TCP which arises from the protocol characteristic as UDP does not need congestion control mechanism. And the throughput gap between with the same TCP options but with and without random access process is also significant and certainly related to the uplink random access delay specified by IEEE 802.16 d/e TDMA MAC. Therefore, in this paper, we analyze the effect for the random access protocol to have on the TCP flow and suggest the algorithm to minimize that effect using the characteristic of TCP delayed ACK which is generally adopted by most of the TCP versions.

This paper is organized as followings. In section II, the random access protocol of 802.16 d/e and TCP delayed ACK are described. The structure and operation of IEEE 802.16 d/e SS is discussed in III. The new random access algorithm using TCP delayed ACK is given in IV. Result and concluding remarks are given in section V and IV respectively.

II. TCP THROUGHPUT OVER IEEE 802.16 D/E TDMA MAC

A. 802.16 Uplink Random Access Protocol

As shown in Figure 1, a frame which can be 2.5, 4, 5, 8, 10, and 12 ms consists of downlink (DL)- and uplink (UL)-subframe with Tx/Rx transmission gap (TTG) and Rx/Tx transition gap (RTG) (In our tests, 5 ms are assumed).

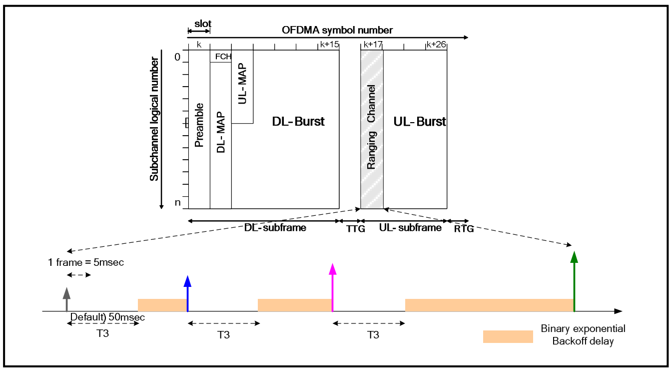


Figure 1. The frame structure and backoff algorithm of IEEE 802.16d/e

The horizontal axis shows a series of slots of DL- and UL-subframes while the vertical axis depicts the logical subchannels which consist of 48 data subcarriers respectively. In DL-subframe, the preamble representing each BS and frame control header (FCH) indicating current frame information and its DL-MAP are broadcasted to every SS in the cell. UL-and DL-data are transmitted through UL- and DL-burst regions to each SS. The information about data burst to each SS is specified by UL- and DL-MAP. For each SS to transmit UL-data, it does contention-based uplink random access procedure in the ranging channel.

Each SS in the same cell can transmit a bandwidth request (BW) code on a randomly selected ranging subchannel slot whenever it gets data sent to BS. This bandwidth code is a subsequence of a PN code and is selected from a group of PN code managed by the BS. If the BS receives the code successfully from the SS, it sends CDMA allocation message which is included in UL-MAP and assigns the bandwidth for the SS to send a bandwidth request message. The SS compares the code value to be sent to BS with the code value indicated by CDMA allocation message continuously. If the CDMA allocation message with the same code value is not received in the specified time which is indicated by T3 timer, the SS launches a binary exponential backoff algorithm to retransmit a BW code as shown in the lower part of Figure 1. On the reception of the CDMA allocation message with the same value, SS sends the bandwidth request message to include a number of bytes requested for UL-data transmission in the UL-burst region. Then, BS allocates the requested bandwidth to the SS.

IEEE 802.16 d/e MAC protocol is a class of demand-assigned multiple access (DAMA) which means contention-based random access process. By this characteristic, significant additional delay for uplink transmission is added in contrast to WCDMA system in wireless portion of the wired-cum-wireless environment, in particular, for TCP (Transmission Control Protocol) connections which require the ACK packets to be transmitted to the server in the Internet in case of web services. In this paper we define this as random access delay. In [4], the performance of IEEE 802.16d random access protocol is analytically studied according to the various parameters. T3 timer which expires without the CDMA allocation message plays the role of constant delay in the binary exponential backoff algorithm. If T3 and the size of maximum contention window increase, the difference in access delay is high while the variation of the access throughput (packets/frame) is insignificant (Assuming the population size in a cell is under 100). In addition, the effects of the number of PN codes used for bandwidth request ranging and the number of slot-subchannels in a UL-frame increase according that the population size increases with the number of population of under 100. And T16, scheduling time, by BS and piggyback probability also have the significant effect on the access delay and throughput.

B. *The impact of TCP delayed ACK and RTT*

The shortcoming of acknowledging all segments is to consume twice the power and bandwidth for transmission of ACKs compared to that of delayed TCP ACK. It leads to the increase of data packet drop rate and to frequent network congestions. So most TCP implementations are using delayed ACK scheme [5,6,7]. However, delayed TCP ACK can cause the amount of TCP transfer time to be doubled and RTO (Retransmission Timeout) for the first segment due to delayed ACK latency which forces ssThresh to shrink to 2 and leads to very slow increase of window size afterwards because the initial timeout is larger than the majority of RTTs one encounters in the Internet today. This situation has negative effect especially on the short-lived flows that compete within the slow start phase to obtain the short response time for good performance [8]. Statistical studies of Internet traffic show that most TCP are short-lived flows which can be finished in about 50 packets and are about 99% of the total TCP connections [9,10].

[11] describes transfer time t for short-lived TCP flows. The transfer time t is described by a function of RTT (round trip time), MSS (Maximum Segment Size), w (segment window size), k (the number of simultaneous TCP connections) and t_{delack} (delayed acknowledgement). With larger w, MSS and k, transfer time can become shorter. Among these factors, bigger MSS may cause larger number of retransmission for the small bit errors especially in wireless environments. And w, r and delayed ACK penalty can be minimized if the existing TCP implementations is revised to include the new algorithms such as new delayed ACK algorithm (e.g., 'DAASS' and 'BADA with SNDA') to solve the presented problem in transport layer.

However, it is very difficult to revise all the existing TCP implementations. In this paper we have interests on the reduction of RTT by minimizing the UL-random access delay of TDMA MAC layer in the IEEE 802.16 d/e SS.

C. *The analysis of RTTover IEEE 802.16 d/e TDMA MAC*

In this paper we focuses on RTT reduction of the wireless portion, in particular, for the MAC layer of SS side. RTT in consists of wired portion and wireless portion as following expression for IEEE 802.16 d/e wired-cum-wireless environment.

$$RTT = T_{wired} + T_{wireless} \qquad (1)$$

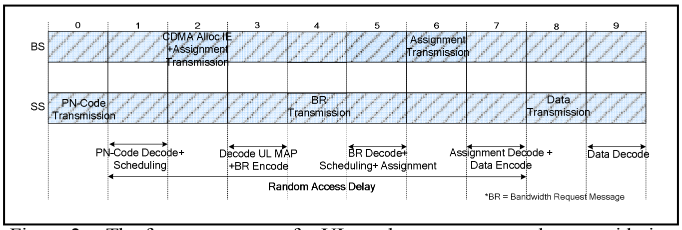


Figure 2. The frame sequences for UL random access procedure considering implemenation

in which T_{wired} component is ignored in this paper as it is not controllable throughout the Internet. Then, $T_{wireless}$ is given by

$$T_{wireless} = T_{uplink} + C_{downlink} + 2T_{wait}$$

$$= T_{random\_access\_delay} + T_{transmit} + C_{downlink} + 2T_{wait}$$

$$\left(0 \le T_{wait} < T_{frame\_duration} + T_{scheduling\_time}\right)$$

where T_{uplink} and $C_{downlink}$ are the UL- and the DL-transfer time between the BS and the SS respectively. And T_{wait} means the time during that a TCP segment has to wait until the beginnings of the DL-subframe in the BS and the UL-subframe in the SS to be transmitted after arriving and $T_{transmit}$ is the time to be transmitted to the BS from the SS. In above expression, $C_{downlink}$ portion can be assumed constant because TCP segments can be sent without delay except for scheduling time by BS when the downlink bandwidth is enough. Then the uplink random access delay can be obtained by

$$T_{random\_access\_delay} = \sum_{i=1}^{n} p_s (1-p_s)^{i-1} 2^{i-1} T_{frame} + 2T_{assignment} + T_{transmit} \qquad (2)$$

In (2), p_s is the probability that a SS transmit its PN-codes on an uplink-subframe in a cell and the BS receives it successfully when the wireless channel is assumed to follow Rician distribution. And $T_{assignment}$ is the time in which the a bandwidth for the bandwidth request message or TCP segments can be scheduled for being allocated to the SS by the BS and T_{frame} is one frame duration (e.g. 5 ms assumed in this paper). i means the number of retransmitting PN-code.

According to above (2), $T_{random\_access\_delay}$ may be 4 frames if p_s is 1(Many papers showed the algorithms to minimize this factor in dynamic wireless channel state which means dynamic value of p_s. However in this paper $T_{random\_access\_delay}$ is assumed constant.) This results in minimum 8ms to maximum 24 ms as the IEEE 802.16 d/e specification defines various frame durations which are 2, 2.5, 4, 5, 8, 10 and 12 ms. This can be ignored considering benefits from TDMA/TDD mode although this increase average RTT of TCP connections. However, considering the implementation of this specification, other delay factors can be added. Figure 2 depicts this situation in detail with the example of our WiBro project developed in 2003 with Samsung in which targeting traffic rates of a SS for uplink and downlink streams are 1 Mb/s and 3 Mb/s respectively. As shown in this figure, in addition to the above delay specified by IEEE 802.16 d/e, the additional processing time for decoding and encoding the data, and scheduling data and bandwidth resources which is equal to 4 frames and may be different in other implementation is needed. And this additional delay is related to implementation complexity. the shorter additional delay becomes, the more complex implementation becomes generally.

TABLE I. PROCESSING LATENCY COMPARSIONS(IN MICROSECONDS)

Bytes / Workload	128	1024	2048
Map Parsing	22	22	22
memcpy()	10	28	50
DES	141	162	186
CRC	26	172	347

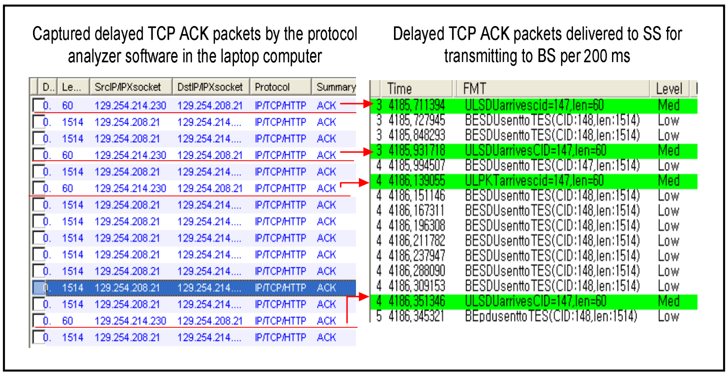
For example, such additional delay can occur in this case. According to our previous analysis [12], if UL-MAP pertains to the current frame (i.e., the frame in which UL-MAP was received), only about 1 ms can be given to MAC for the uplink because MAC can know its own opportunity and allocated bandwidth length for uplink transmission after it gets the UL-MAP. And Table I shows execution time statistics for CPU to execute presented workloads which are essential for MAC functionalities or to develop others. This test is done on the realtime embedded Linux operating system, Montavista Linux, based on the StrongARM SA-1110 processor. As shown in statistics, although overloads such as ARQ and scheduling which are difficult jobs to be analyzed quantitatively are not assumed, it takes about 600 us for one MAC PDU (Program Data Unit) to be made in MAC layer. However, in the worst case 600 us is not enough because of the characteristic of the burst of the TCP connection and processing delay variations which depends on the characteristic of the real-time operating system on the MAC board and increases in proportion to the processing latency and the traffic loads. This situation can cause droppings of packets and cause real-time applications supporting the movie or audio files over 1 Mb/s to experience the stall whenever an uplink burst such as TCP acknowledgement packets are dropped. Therefore, additional delay which is at least about 1.3 ms is required and leads to frame sequences shown in Figure 2 when the MAC layer is implemented in Montavista Realtime Linux kernel.

In the result, the $T_{random\_access\_delay}$ can increase to 8 frames and the total delay for uplink transmission leads to maximum 11 frames which are 55 ms if one frame duration is 5ms although p_s is assumed 1 that means none failure on receiving PN-code. The longer one frame duration is, the longer RTT becomes. The method to reduce the uplink random access delay in IEEE 802.16 d/e is required

III. LOOK-AHEAD RANDOM ACCESS ALGORITHM USING DELAYED ACK

In this paper, we have special concerns about the TCP performance including Web and multimedia streaming services from point of view of the SS. When we estimated the throughput of streaming service from a general Internet multimedia server using media player (e.g., Microsoft windows media player 9) over IEEE 802.16 testbed in our lab, we could easily find the difference between with UDP option and with TCP option as the streaming protocol. The one with TCP option is about one third of the other with UDP option. As known widely, this difference occurs because the TCP module at the far end needs to sends back an acknowledgement for packets which have been successfully received to handle both timeouts and retransmission for reliable and in-order delivery of data.

We changed the normal testbed in order to check how much the uplink random access delay has the effect on the TCP performance with above the same content. In this testbed, we made the BS allocate bandwidth for the SS to send its TCP packet without the uplink random access process every frame. The throughput of the third with TCP options as the streaming protocol is two third of that with UDP options. So many papers tried to propose various contention resolution algorithm to minimize $T_{random\_access\_delay}$, p_s , and n [13]. However, with a best algorithm, $T_{random\_access\_delay}$ can be minimized but not be removed. In this paper we propose the method to remove $T_{random\_access\_delay}$ delay factor from RTT of TCP connections for a IEEE 802.16 SS in the best although UL-random access processes are still executed for the UL-transmission.



D..	Le...	SrcIP/IPXsocket	DstIP/IPXsocket	Protocol	Summary
0.	60	129.254.214.230	129.254.208.21	IP/TCP/HTTP	ACK
0.	1514	129.254.208.21	129.254.214....	IP/TCP/HTTP	ACK
0.	1514	129.254.208.21	129.254.214....	IP/TCP/HTTP	ACK
0.	60	129.254.214.230	129.254.208.21	IP/TCP/HTTP	ACK
0.	1514	129.254.208.21	129.254.214....	IP/TCP/HTTP	ACK
0.	60	129.254.214.230	129.254.208.21	IP/TCP/HTTP	ACK
0.	1514	129.254.208.21	129.254.214....	IP/TCP/HTTP	ACK
0.	1514	129.254.208.21	129.254.214....	IP/TCP/HTTP	ACK
0.	1514	129.254.208.21	129.254.214....	IP/TCP/HTTP	ACK
0.	1514	129.254.208.21	129.254.214....	IP/TCP/HTTP	ACK
0.	1514	129.254.208.21	129.254.214....	IP/TCP/HTTP	ACK
0.	1514	129.254.208.21	129.254.214....	IP/TCP/HTTP	ACK
0.	1514	129.254.208.21	129.254.214....	IP/TCP/HTTP	ACK
0.	60	129.254.214.230	129.254.208.21	IP/TCP/HTTP	ACK
0.	1514	129.254.208.21	129.254.214....	IP/TCP/HTTP	ACK

	Time	FMT	Level
3	4185,711394	ULSDUarrivescid=147,len=60	Med
3	4185,727945	BESDUsenttoTES(CID:148,len:1514)	Low
3	4185,848293	BESDUsenttoTES(CID:148,len:1514)	Low
3	4185,931718	ULSDUarrivesCID=147,len=60	Med
4	4185,994507	BESDUsenttoTES(CID:147,len:1514)	Low
4	4186,139055	ULPKTarrivescid=147,len=60	Med
4	4186,151146	BESDUsenttoTES(CID:148,len:1514)	Low
4	4186,167311	BESDUsenttoTES(CID:148,len:1514)	Low
4	4186,196308	BESDUsenttoTES(CID:148,len:1514)	Low
4	4186,211782	BESDUsenttoTES(CID:148,len:1514)	Low
4	4186,237947	BESDUsenttoTES(CID:148,len:1514)	Low
4	4186,288090	BESDUsenttoTES(CID:148,len:1514)	Low
4	4186,309153	BESDUsenttoTES(CID:148,len:1514)	Low
4	4186,351346	ULSDUarrivesCID=147,len=60	Med
5	4186,345321	BEpdusenttoTES(CID:148,len:1514)	Low

Figure 3. Example of delayed ACK packets monitored in SS

Currently most TCP implementations are using delayed ACK scheme in spite of a few shortcomings introduced in section II. In previous tests with TCP option as the streaming protocol, these delayed ACKs are monitored every 200 ms like in Figure 3 (This value can be changed from system to system). The left part of Figure 3 shows TCP segments captured at the NIC (network interface card) in the laptop computer by a kind of protocol analyzer, e-Watch Pro and the right of Figure 3 shows SDUs monitored at the MAC layer of IEEE 802.16 SS, which is located at the other processor and interfaced with the laptop computer by PCMCIA (Personal Computer Memory Card International Association) interface, by a kind of debugging tool by developed by our lab. 'Time' column in the right side of figure shows that delayed TCP ACKs indicated by the green color are delivered to BS every near 200 ms. These packets's connection identifier (CID) is 147 and their length are 60 bytes.

Our basic idea to remove $T_{random\_access\_delay}$ factor from RTT of long-lived multimedia TCP flows in the IEEE 802.16 SS is based on using this characteristic. If the delayed ACK timer is known to the MAC layer before a few number of frames, N_{look_ahead}, within that delayed ACK packets come to this layer (e.g. about 8 or 9 frames in our testbed), it can execute the random access procedure in advance. Therefore, as soon as the delayed ACK packets arrive, they can be transmitted to BS without any delay.

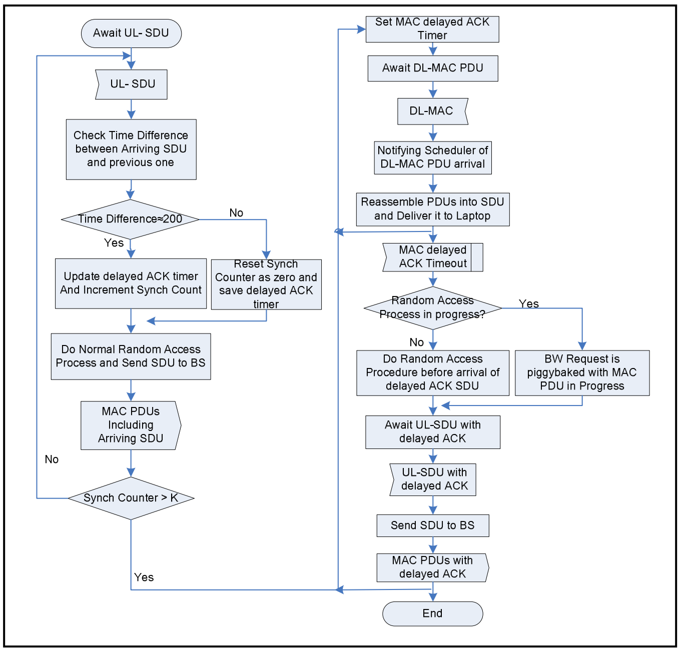


Figure 4. Look-ahead random access algorithm using delayed ACK

The simplest approach to achieve this goal is cross-layer design between the TCP layer and the MAC layer. However, the shortcoming of this approach is to require the TCP implementation, which is already distributed, to be revised. Another approach is like the snoop module algorithm in which the snoop module (agent) in BS observes the flows of TCP packets and TCP ACKs and stores them for local retransmission of the wireless portion in wired-cum-wireless environment [14].

In our algorithm this snoop agent is located at SS and observes the flows of TCP packets and TCP ACKs like [14] but does not store them and bypasses them. The algorithm is depicted in Figure 4 in detail. This algorithm consists of two phase. In the first phase of the algorithm corresponding to the left part of Figure 4, the delayed ACK timer in the MAC layer is synchronized with the delayed ACK timer in TCP layer, which goes off at every 200 ms relative to when the kernel was bootstrapped, using UL-SDU. This synchronization is regarded as achieved if a few numbers of successive UL-SDUs including TCP ACKs arrives at every 200 ms . For this one incrementing counter depicted as Synch Counter is added.

At start, Synch Counter and delayed ACK timer of MAC layer is initialized as zero. Whenever UL-SDUs with TCP ACK packets arrive at SS from the laptop computer, the arrival time is recorded and compared with the delayed ACK timer. If the time difference is not equal to about 200 ms, Synch Counter is reset and the new arrival time is saved for the delayed ACK timer. If the time difference is about 200 ms, Synch Counter is incremented and the delayed ACK timer is updated with new arrival time. Then UL-SDU is transmitted according to the normal UL-transmission procedure.

If the specified number of successive delayed ACKs which mean that Synch Counter is over K are observed, synchronization between the delayed ACK timer of MAC layer and that of TCP layer is assumed to be achieved. In the second phase, the delayed ACK timer is set to expire every 200 ms and before N frames during that UL-random access procedure can be completed. And the snoop agent looks over all the DL-MAC PDUs from BS. If a reassembled SDU from multiple PDUs is a complete TCP packet, the snoop agent notifies the scheduler of the arrival of the TCP packet and bypasses it without additional intervention and modification.

Whenever the delayed ACK timer goes off, the scheduler of SS checks whether there are DL-TCP packets or not. If there are, the scheduler checks if another random access process is in progress again. If there is a procedure in progress, the request for the wireless bandwidth is piggybacked. If not, the scheduler initiates the new random access procedure before delayed TCP ACK comes. After UL-random access procedure is completed and a waiting UL-SDU comes, this packet can be transmitted without any delay.

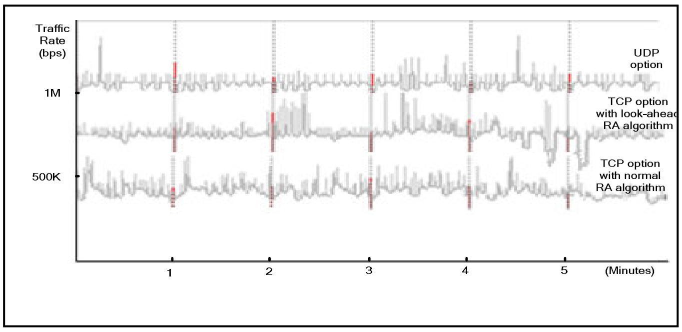
IV. RESULT

When we estimated the throughput of streaming service from a general Internet multimedia server using media player (e.g., MS windows media player 9) with two transport layer options over IEEE 802.16 testbed in our lab. As shown in the Figure 5, the average data rate of downlink streaming service with UDP protocol among three options is about 1.1 Mb/s while that of DL-streaming service with TCP protocol with normal random access delay which is specified in IEEE 802.16 d/e specifications is about 450 kb/s. The traffic of TCP option with normal random access (RA) algorithm option has much fluctuation because it needs to wait for a few frames in order to obtain the opportunity for UL-transmission. However, traffic rate with our algorithm is about 780 kb/s and this is close to that of TCP option without random access process which is ideal throughput without any uplink random access delay as explained section III. These results shows the UL-random access delay specified by IEEE 802.16 d/e TDMA MAC has the significant effect on the TCP performance.

V. CONCLUSION

In this paper, we describers the random access protocol of IEEE 802.16 d/e and analyze the delay factors considering the implementation complexity. In order to speed TCP throughput up without revising existing TCP implementations, RTT of wireless portion in the wired-cum-wireless environment among many factors needs to be reduced by the MAC layer. So we propose the algorithm to minimize that delay factor using the characteristic of TCP delayed ACK which is generally adopted by most of the TCP versions. This algorithm is like having no random access procedure in the best, in particular, in case of long lived streaming services. We will extend this algorithm to short-lived flows and dynamic channel state which means dynamic p_s and N_{look_ahead} proposed in section III in the future.

Figure 5. The throughput of multimedia streaming service over IEEE 802.16 d/e TDMA MAC



REFERENCES

[1] Draft IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Broadband Wireless Access Systems(IEEE802.16 REVd/D5-2004), May, 2004.

[2] Draft IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Broadband Wireless Access Systems-Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands(IEEE P802.16e/D7), April, 2005.

[3] Seong Chul Cho, Yuro Lee, Dong Seung Kwon, "Turbo-Coded OFDM/TDMA System for Beyond 3G Mobile Multimedia Communications," ICWN'03, Las Vegas, Nevada, USA, pp.418-425, June 2003.

[4] Jun-Bae Seo, Nam-Suk Lee; Nam-Hoon Park; Choong-Ho Cho, "Performance Analysis of IEEE802.16d Random Access Protocol," Communications, 2006 IEEE International Conference, Volume 12, pp5528 – 5533, June 2006

[5] Douglas E. Comer, "Internetworking with TCP/IP," Fourth Edition, Prentice Hall

[6] W. Richard Stevens. " TCP/IP Illustrated, volume I", Addison Wesley

[7] M. Allman, V. Paxson, and W. R. Stevens, "TCP Congestoin Control,", RFC2581, April 1999

[8] In-ho Roh, Young Yong Kim, "Improving TCP Performance using BADA(Base-station Aided Delayed ACKs) algorithm in Wired-Cum-Wireless Environment," Volume 3, pp897-901, Oct. 2002

[9] N. Brownlee and K.C. Claffy, "Understanding Internet Traffic System: Dragonflies and Tortoises," IEEE Communications, Vol. 40, No. 10, pp. 110-117, 2002.

[10] Xudong Wu, Nikolaidis, I; " On the Advantage of Lifetime and RTT Classificaiton Schemes," Performance, Comuting and Communications Conference 2003, pp301-308, April 2003

[11] In-ho Roh, Young Yong Kim, "Improving TCP Performance using BADA(Base-station Aided Delayed ACKs) algorithm in Wired-Cum-Wireless Environment," Volume 3, pp897-901, Oct. 2002

[12] Nak Woon Sung, Nam Hun Park , "Hardware/Software Codesign of TDMA MAC applied to IEEE 802.16 Subscriber Station," PIMRC2005, Berlin, Germany, CD: Wireless Networks Session, Sept 2005

[13] Xiaohua Li, "Contention resolution in random –access wireless networks based on orthogonal complementary codes," Communications, IEEE Transactions, Vol 2, Issue 5, Sept, 2003

[14] Hari Balakrisshnan, Srinivasan Seshan, Randy H. Katz, "Improving Reliable Tranport and Handoff Performance in Cellular Wireless Networks," Wireless Networks, Vol1 Issue4, Dec 1995