An Evaluation of Multimedia Data Downstream with PDA in an Infrastructure Network

Youn-Sik Hong*, and Hye-Sun Hur*

Abstract: A PDA is used mainly for downloading data from a stationary server such as a desktop PC in an infrastructure network based on wireless LAN. Thus, the overall performance depends heavily on the performance of such downloading with PDA. Unfortunately, for a PDA the time taken to receive data from a PC is longer than the time taken to send it by 53%. Thus, we measured and analyzed all possible factors that could cause the receiving time of a PDA to be delayed with a test bed system. There are crucial factors: the *TCP window size, file access time* of a PDA, and the inter-packet delay that affects the receiving time of a PDA. The window size of a PDA during the downstream is reduced dramatically to 686 bytes from 32,581 bytes. In addition, because flash memory is embedded into a PDA, writing data into the flash memory takes twice as long as reading the data from it. To alleviate these, we propose three distinct remedies: First, in order to keep the window size at a sender constant, both the size of a socket send buffer for a desktop PC and the size of a socket receive buffer for a PDA should be increased. Second, to shorten its internal file access time, the size of an application buffer implemented in an application should be doubled. Finally, the inter-packet delay of a PDA and a desktop PC at the application layer should be adjusted asymmetrically to lower the traffic bottleneck between these heterogeneous terminals.

Keywords: Multimedia data, downstream, PDA, TCP window size, inter-packet delay

1. Introduction

A wireless LAN (WLAN) service is a form of telecommunication that uses electromagnetic waves to communicate among mobile terminals within a 50~100 meter radius of their AP (Access Point), where it serves as a base station. It is cost effective to use WLAN instead of CDMA/GSM when one wants to utilize its multimedia based service within a fixed area.

Most Internet users spend a lot of time downloading data from a web server, instead of uploading. We think that this situation occurs not only in a wired network but in a wireless network as well. In a wired-cum-wireless network based on WLAN, a so-called *infrastructure network*, different types of terminals such as desktop PCs serving as fixed hosts and PDAs (Personal Digital Assistants) serving as mobile hosts can be connected to such a network. Assuming that a PDA is frequently used for downloading data from its stationary server like a desktop PC, a desktop PC acts as a fast sender and a PDA as a slow receiver due to a substantial difference in their computational capabilities (CPU, memory, etc).

Because the greater part of data transmission is downstream, the overall performance of multimedia data transmission depends heavily on the performance of the downstream with PDA. Thus, we will focus on the analysis of the performance related to multimedia data downstream between these two heterogeneous terminals. We will propose a method to improve its performance through an analysis based on the performance metrics: file access time, window size, and inter-packet delay.

This paper consists of the following steps. In Chapter 2, we discuss the research motivation and previous works. In Chapter 3, we briefly explain file access time, window size and inter-packet delay, which are the performance metrics of multimedia data transmission in a WLAN environment. In Chapter 4, the experimental results are shown, and we conclude our work in Chapter 5.

2. Related Work

2.1 The Motivation

In the case of WLAN environments that provide multimedia-based services, a PDA is used mainly for the purpose of receiving rather than sending. For example, within the hot spot that is an area for utilizing a WLAN service, a PDA user can download multimedia contents such as sports news, movies or music videos and then play them with his/her PDA [1]. When there is data transmission between a PDA as mobile host and its stationary server, the time taken to receive data at a PDA from the server is longer than the time taken to send it by

Manuscript received September 29, 2005; accepted December 20, 2005
This work was supported by the MOICE and ITEP through the Multimedia Research Center at University of Incheon.
Corresponding Author: Youn-Sik Hong
* Dept. of Computer Science and Engineering, University of Incheon, Incheon, Korea (yshong@incheon.ac.kr, mshush@incheon.ac.kr)

53% [3], as will be explained in Section 3.2. Thus, we will analyze the factors that cause the excessive receiving time and propose a method to improve it.

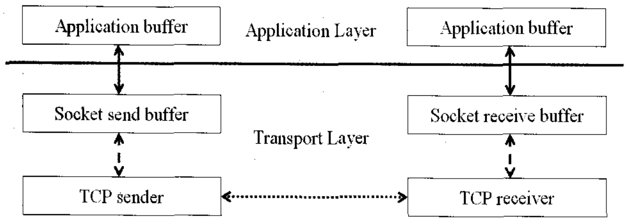


Fig. 1. The buffers used in TCP

The data transmission protocol adopted in this paper is TCP (Transmission Control Protocol) [8]. With the inherent characteristics of TCP, it is easy to find the lost data packet during transmission. Besides the TCP parameters, the window size at a receiver and the ACK delay are chosen for our analysis. As shown in Figure 1, we consider a manipulation of the send buffer and the receive buffer at the transport layer as well as an application buffer at the application layer to enhance the performance of a PDA by tuning the TCP [12] [13].

2.2 Related Works

Balakrishnan et al [2] used a laptop PC as a server with a WLAN NIC (Network Interface Card) to measure TCP performance in a wireless network. However, when *Pilosof* [5] performed data transmission between mobile hosts, he just simulated it using NS (Network Simulator) software instead of real devices. However, we built a test bed system consisting of a PDA as a mobile host and a desktop PC as a fixed host for a real application, and measured the actual data. In addition, *Pilosof* used the devices with the same or similar performance capacity in his experiments, while setting an inter-packet delay between each packet to 1-2 ms [5] as a nearly fixed value. In this paper, since there is a significant difference in performance between a PDA and a desktop PC, we will vary the inter-packet delay from none to 10 ms to take it into account.

Many of the previous works conducted their experiments based on Unix or Linux Operating Systems [2] [4], whereas we have carried out all of our experiments based on Windows 2000/CE.

3. Performance Metrics for evaluating multimedia data transmission in WLAN

3.1 A Test-bed Infrastructure Network Based on WLAN

We have designed and implemented VMS (Voice Messenger Systems) to measure the performance of multimedia data transmission. VMS is an infrastructure network that integrates a wired LAN, based on Ethernet with a WLAN based on the IEEE 802.11b standard. The WLAN contains only one BSS (Basic Service Set) for simplicity.

VMS is a kind of file transfer system which consists of a VMS server and VMS clients. A desktop PC and PDA represent FH (Fixed Hosts) and MH (Mobile Hosts) respectively. One of the desktop PCs is used as the VMS server. The rest of them and all of the PDAs are used as VMS clients. A PDA with a PCMCIA NIC can be connected to a wired LAN through a BS (Base Station). The hardware specification of the hosts used in the VMS is listed in Table 1.

Table 1. The hardware specification of the hosts used in the VMS

Host type		CPU	RAM	NIC
MH	PDA	Strong Arm(206MHz)	64MB	PCMCIA (11Mbps)
FH	PC	Pentium 4(2.4GHz)	512MB	PCI (100Mbps)

Let us briefly explain how a VMS works: Each VMS client records one's voice and sends it to the VMS server after converting it into a wave file format. The server receives this voice message sent from its client and stores it in its hard disk. It transfers this message to the authenticated client that requests it.

Before we discuss performance metrics, we should define the terminology used: 'PDA upstream' is the process of transferring data from a PDA as a MH VMS client to the server. During this process, the PDA client reads and sends its internal data and then the server receives and writes them into its file system. Conversely, we call the process of receiving data for a PDA client from the server 'PDA downstream'. It moves in an opposite direction to the upstream.

PDA sending time is the time that has elapsed to complete data transmission during the upstream. PDA receiving time is the time that has elapsed to complete data transmission during the downstream.

3.2 The Preliminary Factors Considered for Performance Evaluation

3.2.1 Packet size

To find the packet size at the application layer that gives the best performance in terms of transmission time, we varied it from 512 bytes to 7,128 bytes at the application layer, transport layer. Figure 2 shows the PDA sending time and the PDA receiving time measured while varying its packet size, when the file size transmitted was about 469 Kbytes.

Figure 2 shows that as the packet size becomes larger, the corresponding transmission time becomes faster. That is, as the file to be transmitted is divided by a larger packet size, both the total number of data packets segmented and the number of ACK packets get smaller. As a result, with a larger packet size its transmission time becomes faster.

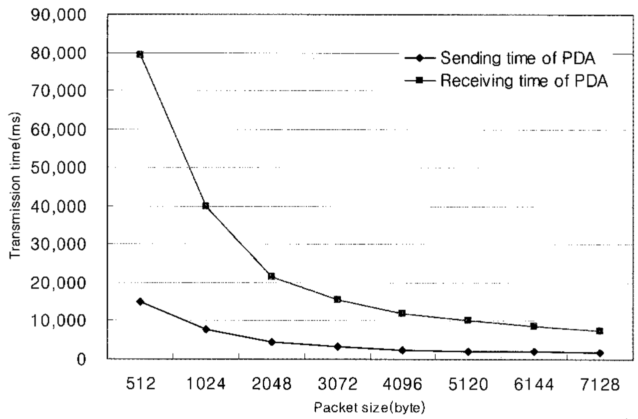


Fig. 2. The transmission time measured while varying its packet size when the file size transmitted is 469 Kbytes

3.2.2 The sending time versus the receiving time

As you saw in Fig.2, the PDA receiving time is always slower than the PDA sending time. For more thorough analysis, the ratios of receiving time to sending time for PDA client with the file size are listed in Table 2 when the packet size is set to 4,096 bytes. The sample wave files have been generated by recording one's voice for 30 to 600 seconds.

Table 2. A comparison of the sending time with the receiving time when the packet size is set to 4,096 bytes

Recording time (seconds)	File size (KB)	Ratio of receiving time to sending time	
		PC client	PDA client
30	235	1.0	1.53
60	469	1.0	1.63
120	938	1.0	1.66
300	2,344	1.0	1.70
600	4,688	0.99	1.70

For the desktop PC client, the receiving time is almost the same as the sending time independent of the file size transmitted. However, for the PDA client, the receiving time is longer than the sending time by 53% to 70%. Notice that the ratio increases as the file size becomes larger.

3.3 Performance Metrics of PDA Downstream

As we see in Figure 2, the time that elapsed during the PDA downstream is longer than during the PDA upstream. Especially, as the data to be transmitted become larger, the receiving time is much longer than the sending time (Table 2). Thus, our main concern is that all possible factors that cause the receiving time to be delayed during PDA downstream should be listed and analyzed. The performance metrics that we should consider are the following: TCP window size, file access time of PDA, and inter-packet delay between packets.

3.3.1 Window size

Window size is the value that a receiver advertises to a sender on receiving a request to begin data transmission. It is a 16-bit field, limiting the window to 65,535 bytes [6]. The receiving process can usually control the size of the window offered by the receiver. This can affect TCP performance.

Eq. (1) shows the relation between the window size and the rate of packet transmission [9]. In Eq (1), X(t) is the rate of packet transmission at time t and W(t) is the window size at time t. RTT is an abbreviation of the round trip time. It represents the status of transmission delay in a given network. Units of RTT are measured by seconds.

X(t) is inversely proportional to RTT, whereas it is proportional to W(t). That is, as the window size gets bigger, the rate of packet transmission increases. The network analyzing software [10] has been used to trace the variation of TCP window size during the upstream and the downstream.

3.3.2 File access time of PDA

As shown in Table 1, there are distinct differences between PDAs and desktop PCs in the throughput of data processing due to the substantial gap between the H/W and the firmware processing time [4].

$$X(t) = \frac{W(t)}{RTT} \quad (1)$$

PDA uses memory chips as storage devices instead of the hard disc. With an iPAQ 3660 model chosen as a PDA client in the VMS, it consists of 16 MB flash memory and 64 MB SDRAM [7]. A program buffer allocated by the VMS is to be stored into SDRAM. On the other hand, a data file is to be stored into the flash memory. When a file is read from the PDA, it reads from the flash memory and stores it in the buffer that resides in the SDRAM. On the contrary, when a file is written to the PDA, it writes from the SDRAM and stores it in the flash memory.

Comparing the file access time during data transmission between a desktop PC and a PDA, we carried out an analysis to see how this could affect the overall performance.

3.3.3 Inter-packet delay

In the application layer, an inter packet delay is the interval between the time when a previous packet was sent and the time the current packet was first sent. In general, the transmission time becomes faster with a shorter inter-packet delay. However, if there is a distinct difference in processing capability between sender and receiver, inter-packet delay at normal should be adjusted to give enough time intervals to complete its internal processing for low-end devices. Therefore, it is necessary to consider it more carefully.

4. Experiment Results

We performed our experiments to measure the above metrics and to find a way to improve the receiving time during the downstream. In these experiments, a packet size of 4,096 bytes at the application layer was used.

4.1 Variation in the Window Size

During the upstream, the window size of the PDA and desktop PC was 32,581 bytes and 17,520 bytes respectively. Both of them were kept constant. During the downstream, the window size of the desktop PC was 17,453 bytes, almost the same as before, whereas the window size of the PDA was reduced remarkably to 686 bytes from 32,581 bytes (Table 3).

Table 3. The least value of TCP window size (unit: bytes)

PC window size		PDA window size	
Upstream	Downstream	Upstream	Downstream
17,520	17,453	32,581	686

Due to the substantial differences in performance between them during the downstream, the desktop PC and PDA represent the characteristics of fast sender and slow receiver respectively [6]. The data transmission time and ACK response time measured in both the upstream and the downstream are shown in Figure 3.

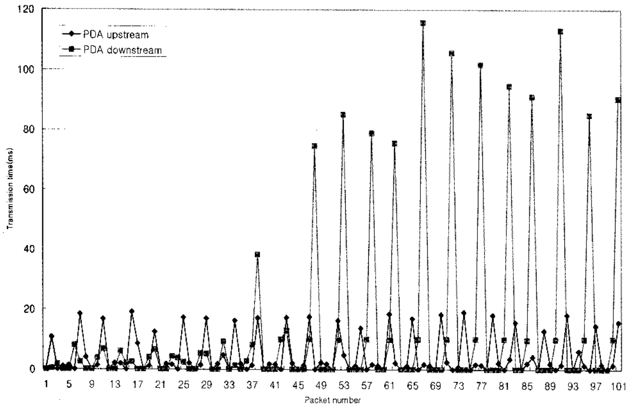


Fig. 3. Data transmission time and ACK response time

When the packet size is 4,096 bytes, the number of packets at the transport layer will be 3, consisting of two of 1,460 bytes and one of 1,176 bytes. In the PDA upstream, every time the first segment is to be sent, its transmission time will be delayed. However, because the time to send it will not exceed 20 ms, the window size of both the PDA and the desktop PC will be kept constant.

In the case of the PDA downstream, on the first occasion it will be more stable than the upstream. However, after the transmission of the 38th packet the response time of the ACK packet will be delayed, and then the window update packet should be sent. It is the ACK packet that informs the updated window size at the receiver to the sender. Thus, the transmission time will be increased dramatically by up to 115ms. This phenomenon will occur frequently until the receiving process is completed. This is the reason why the receiving time is delayed. The problem occurs because the time taken to receive data from the socket send buffer of a desktop PC (*fast sender*) and then process it in the socket receive buffer of a PDA (*slow receiver*) was delayed. Thus, to improve the performance during the downstream, it is necessary to increase both the size of the send buffer of the desktop PC and the size of the receive buffer of the PDA.

4.2 File Access Time with its Size

To measure the file access time with the specified file size, the size of an application buffer that stores data varies from 512 bytes to 10,240 bytes. Besides, we selected the 5 sample files with sizes ranging from 800 KB to 4,000 KB. We measured it 100 times while varying the size of its application buffer, and took the mean figure.

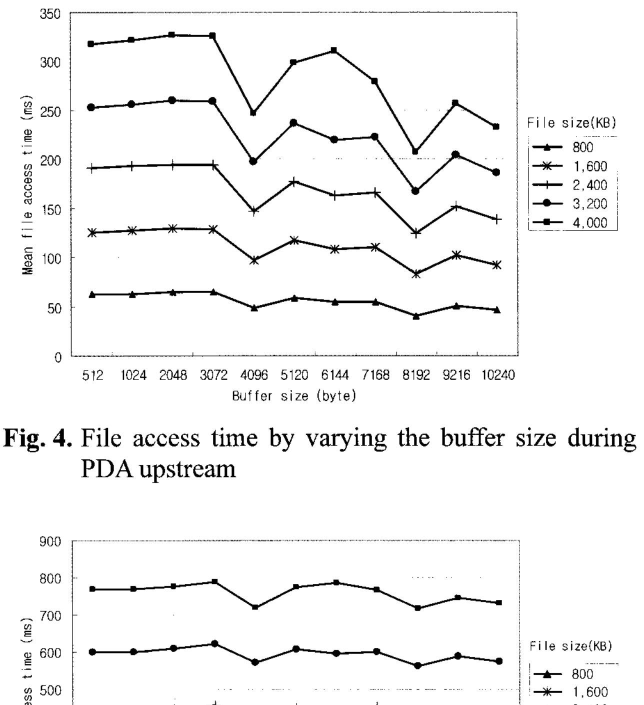


Fig. 4. File access time by varying the buffer size during PDA upstream

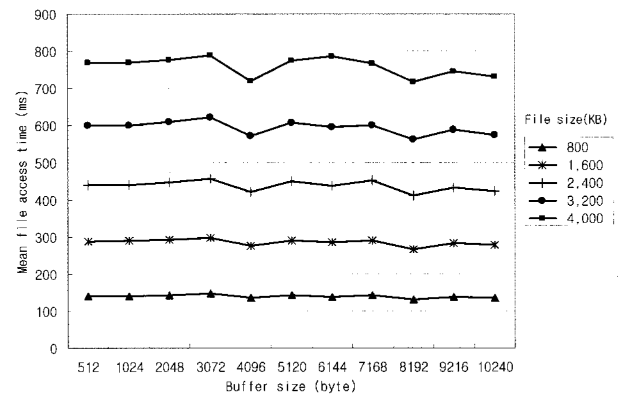


Fig. 5. File access time by varying the buffer size during PDA downstream

The mean file access times during the upstream and the mean file access time during the downstream are shown in Figures 4 and 5 respectively. The reason why the PDA downstream takes 2 times longer than the PDA upstream is that the time taken to write data into the flash memory takes 2 times longer than the time taken to read data from it. However, independent of eitherthe upstream or the downstream, the file access time is fastest when the size of the application buffer is set to 8,192 bytes. Besides, it is faster by 10% in the case of the buffer of 3,072 bytes.

4.3 Adjustments of Inter-Packet Delay

An inter-packet delay can be adjusted by using the sleep function, which is in one of the Visual C++ 6.0's libraries [11]. Assuming that devices with the same performance are used, it is generally true that their transmission time becomes faster with a shorter inter-packet delay. However, because we are using devices with great differences in performance, a shorter delay does not guarantee higher performance.

Table 4. Comparison results before and after applying the proposed method

	Before applying the method				After applying the method			
Process type	Upstream		Downstream		Upstream		Downstream	
Direction of trans-mission	Send from PDA	Receive at PC	Send from PC	Receive at PDA	Send from PDA	Receive at PC	Send from PC	Receive at PDA
Inter-packet delay	1	10	1	10	10	10	none	None
Trans-mission time	2,502		11,993		2,249		4,622	

Table 4 shows that during the upstream the transmission time on average is fastest when it set to 10 ms. This is because the socket receive buffer (8,192 bytes) of the desktop PC is smaller than the socket send buffer (16,384 bytes) of the PDA by a ratio of 2:1. Thus it is necessary for the desktop PC to give enough time to maintain the balance of the processing time due to its smaller socket buffer size. In other words, if we increase its size to that of the PDA, the overall throughput will be enhanced. During the downstream the best performance was achieved when applying no inter-packet delay. During the downstream the receiving time recorded while applying the method decreased by 61% compared to that recorded without applying it. Notice that the send time during the upstream shows little change between application and non-application.

5. Conclusion and Further works

In the case of the WLAN environment that provides multimedia-based services, a PDA as a mobile terminal is used mainly to receive large size data from the stationary server. However, for the PDA client of the VMS, the receiving time is longer than the sending time by 53% to 70%. Notice that the ratio increases as the size of the file to be transmitted becomes larger.

There are crucial factors. The file access time of a PDA and the TCP window size affect the receiving time of a PDA. Since the time to write data into the flash memory of a PDA takes twice as long as the time taken to read data from it, the PDA downstream takes 2 times longer than the PDA upstream. To shorten it, the size of the application buffer that is implemented using an application program should be enlarged depending on the memory structure of the PDA. It will be shortened by 10% by doubling the size of the buffer.

During the upstream, both the window size of a desktop PC and a PDA are kept constant. However, during the downstream a desktop PC and a PDA represent the characteristics of fast sender and slow receiver respectively. That is, the response time of an ACK packet is to be delayed and then the window update packet should be sent. Therefore, the overall transmission time will be increased dramatically. To alleviate such a delayed response of the ACK packet, both the size of the socket send buffer of a desktop PC and the size of the socket receive buffer of a PDA should be enlarged as much as possible. With the enlarged size of the buffer, the window size at the receiver will remain stable due to faster processing.

The inter-packet delay should be adjusted to give enough time intervals to complete its internal processing for low-end terminals such as PDAs. Thus, better performance was obtained by applying distinct inter-packet delays to each of the processes. During the downstream the receiving time when applying the proposed method decreased by 61% compared with non-application of the method.

Reference

[1] Nespot web site, http://www.nespot.com.

[2] H. Balakrishnan, S. Seshan, S., E. Amir, and H. Katz, "Improving TCP/IP Performance over Wireless Networks", ACM MOBICOM, 1995.

[3] H-S. Hur and Y-S. Hong, "Performance Analysis of Multimedia Data Transmission with PDA over an Infrastructure Network", LNCS 3045, ICCSA, 2004.

[4] G. T. Nguyen, R. H. Katz, B. Noble, and M. Satyanarayanan, "A Trace-Based Approach for Modeling Wireless Channel Behavior", In Proceedings of the Winter Simulation Conference, pp. 597-604, 1996.

[5] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha, "Understanding TCP fairness over Wireless LAN", IEEE INFOCOM, 2003.

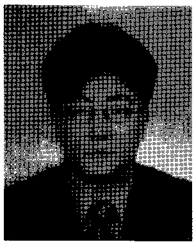
[6] W. R. Stevens, *TCP/IP Illustrated – Volume 1: The Protocols*, Addison-Wesley, 1994.

[7] S. Muench, *The Windows CE Technology Tutorial*, Windows Powered Solutions for the Developer, Addison Wesley, 2000.

[8] J. Postel, Transmission Control Protocol, RFC 793 1981.

[9] M. Hassan, R. Jain., *High Performance TCP/IP*

Networking: Concepts, Issues, and Solutions, Pearson Prentice Hall, 2004.
[10] Analyzer web site, http://analyzer.polito.it.
[11] MSDN, http://msdn.microsoft.com/library/default.asp ?url=/library/en-us/dllproc/base/sleep.asp.
[12] D. Karadia, "Understanding Tuning TCP", Sun BluePrints Online, 2004.
[13] I. Iannello, A. Pescapè, G. Ventre, and L. Vollero, "Experimental Analysis of Heterogeneous Wireless Networks", WWIC, 2004.



Youn-Sik Hong
He received the M.S. and Ph.D. degree from KAIST, Korea, in 1985 and 1989, respectively. He is a professor in Department of Computer Science and Engineering at Univ. of Incheon. His current research topics are mobile computing in wireless sensor networks and multimedia transmission in infrastructure networks.

Hye-Sun Hur
She received the M.S. and Ph.D. degrees from Univ. of Incheon, Incheon, Korea, in 1998 and 2005, respectively. Currently she is a lecturer in Department of Computer Science and Engineering at Univ. of Incheon. Her research interests are Mobile Ad Hoc Networks (MANET), Sensor Networks, and Peer-to-Peer Networks.