WIDE: Wireless Information Delivery Environment in Distributed Hot Spots*

Mehmet Yunus Donmez, Sinan Isik, and Cem Ersoy

Bogazici University, Department of Computer Engineering,
Istanbul, Turkiye
{donmezme, isiks, ersoy}@boun.edu.tr

Abstract. We developed an information delivery system, namely WIDE (Wireless Information Delivery Environment), in client-server architecture using 802.11b infrastructure. WIDE aims to deliver popular information services to registered mobile clients in WLAN hot spots. We present the proposed system architecture, related delivery mechanism and communication protocols. We also give a brief overview of mechanisms required for secure and reliable communication in WIDE system. Performance evaluation results of the proposed system using the implemented prototype are also included in this paper.

1 Introduction

Current advances in computer technology lead to the emergence of battery-operated, low-cost and portable computers such as personal digital assistants (PDAs) or laptop computers equipped with wireless communication peripherals. The increasing demand to access data stored at information servers even while the users are on the move, coupled with the continuing advances in telecommunications, interconnectivity and mobile computing, make information delivery to mobile clients a broadly studied subject.

The motivation of the system proposed in this paper is the Infostation concept [1]. An Infostation is seen as a small cell providing a high bandwidth radio link for data services, which takes on the concept of discontinuous service provision for certain types of services. There are ongoing projects on Infostation concept. One of these project, which is being carried on by Rutgers University, focuses primarily on the data link layer and below [2]. Another project, which is being carried on by Polytechnic University, focuses on developing Infostation applications and transport layer mechanisms to support those applications [3,4]. Rover Technology [5] is another project, which is studied at MIND Laboratory in University of Maryland employing location tracking to decide the services to deliver to the users.

In this paper, we describe the design of a system, namely WIDE (Wireless Information Delivery Environment), which delivers popular or personal information

* This work is partially supported by the State Planning Organization of Turkey under the grant number 98K120890, and by the Bogazici University Research Projects under the grant number 02A105D.

R. Battiti, M. Conti, and R. Lo Cigno (Eds.): WONS 2004, LNCS 2928, pp. 315–328, 2004.

services to registered mobile clients in wireless hot spots. The system design includes protocols that use the IEEE 802.11b WLAN technology to distribute data within isolated coverage areas in a reliable and secure manner.

WIDE resembles gas stations or ATM machines [4], which can be found in locations where there is the appropriate user density, but where users drive or walk to in order to actually access the service. As users pass through the coverage area of the system the most recent version of the subscribed information services will be automatically downloaded to their mobile terminals without any user intervention. Received data may be used in a later time.

In a campus environment, as the user passes through the hot spot of a department building with his PDA or laptop computer, the system may be used to download a wide range of information that might be useful to him. This information may include the most recent data about course locations, course announcements, course web pages and course notes as well as the events in the building and on the campus. As the user walks out of the building and arrives to the cafe, information relevant to that environment such as administrative, departmental, student club and cultural organization announcements are delivered to the user as well as newspaper articles, e-books, etc.

2 WIDE System

2.1 Requirements of WIDE

WIDE is a client-server system, which aims to deliver wireless information services to registered mobile and stationary clients in distributed hot spots. The design of the WIDE system has to meet some basic requirements. First of all, clients of WIDE must be authenticated by the system before getting any services. For this purpose, a secure authentication mechanism should be employed in WIDE. In addition, a global security mechanism should be employed on WIDE, so that the network packets of WIDE system should only be identified and processed within WIDE components.

The transfer of any information services has to be arranged in a way that requires little or no human-computer interaction while users pass through the coverage area of a server. Also, transmissions to and from clients should be arranged so that the desired transactions can be completed while clients are in the coverage area. In addition, any updates of the information services offered to the client shall be transmitted to clients. Hence, we need to have information about the interests of the authenticated clients. A publish/subscribe mechanism should be designed to create a user profile for each client of WIDE system. Subscriptions of clients to information services are recorded in their user profiles. Clients receive information services in case of any updates with the help of their user profiles as they pass through the wireless coverage area of a server.

Since the system offers popular information services, it should perform in an acceptable level in terms of reception time for individual clients when there are many users demanding the same service. Hence, the design should be based on data broadcasting, or, more precisely data multicasting to provide scalability and efficient use of the wireless channel.

The protocols included in the system have to be designed with the idea of battery energy conservation. In addition, they must satisfy the reliability needs of the wireless medium. The residence time of a client in the coverage area may be very short which may lead to an incomplete data transfer. The completion of any incomplete data transfer should be dealt by the system infrastructure using some recovery and error correction mechanisms. In addition, the protocols for data transfer should be designed in a way that allows the coexistence of other communication traffic on the wireless channel.

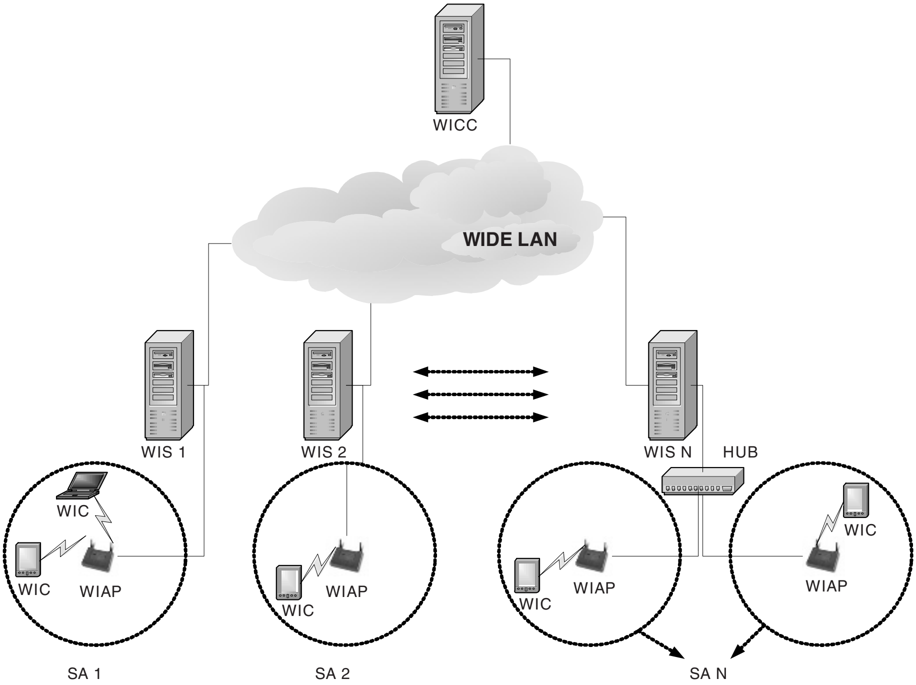


Fig. 1. WIDE system architecture

2.2 WIDE System Architecture

There are three main components in WIDE system as shown in Figure 1. These components are clients, data delivery servers and a server controller. We call a client of the system as WIDE Client (WIC), which is a battery operated handheld or laptop PC with necessary equipments that provide wireless connectivity to servers of the system via 802.11b WAPs, namely WIDE Access Points (WIAPs). The servers of the system are called WIDE Server (WIS) and these are responsible for preparing and delivering information services to clients. The information services, which are available for delivery to clients, are assumed to be stored on local disk of each WIS. The delivery management information such as service identifier, class, version, name and location on the local disk is recorded in a database called WIDE Server Database (SDB).

The component called WIDE Cluster Controller (WICC) keeps and manages system management database called WIDE Cluster Controller Database (CCDB), which consists of a number of tables. These tables are user authentication table, servers information table, user profiles table and information services table.

In WIDE system, each WIS communicates with WICC through the WIDE LAN. The communication between a WIS and a WIC is established via a WIAP. There can be one or more WIAPs connected to a WIS, but a WIAP can only be connected to one WIS. We define the Service Area (SA) of a WIS as the geographical area covered by WIAPs that are connected to a WIS. Figure 1 shows the system architecture, its components and the WIDE LAN.

2.3 Communication Protocols in WIDE

Network Layer Protocols. The system is designed on top of Internet Protocol (IP) stack. WICs must have an IP address that is valid in the SA of a WIS to communicate with that WIS. WIDE is designed to operate independent from how the IP level connectivity of WICs with WISs is established.

Since WICs of WIDE are mobile, they may roam into SA of different WISs, which are probably in different subnets. The IP address assigned to a WIC in one SA may not be valid in another one due to network addressing. DHCP [6] may be used as a valid addressing of WICs in SAs. When WICs enter the SA of a WIS, a DHCP server assigns an IP address to that WIC and the configuration of address is done automatically on the WIC. Here, WIS may be configured as a DHCP server or another machine can be employed as a DHCP server.

Since the residence time of WICs in SAs may be short, a rapid address configuration, such as DRCP [7], may be employed. If IPv6 [8] is chosen as a network layer protocol, stateless autoconfiguration solves the addressing problem of WICs. If Mobile IP [9] is employed in WIDE, each WIS may be configured as a foreign agent or another machine can be employed as a foreign agent. The care-of address assigned by foreign agent will enable WICs to receive service from the system.

Transport Layer Protocols. In WIDE system, WIC-WIS communication is constructed on top of UDP. We cannot deliver popular data to multiple users simultaneously using TCP since it does not support broadcasting and multicasting.

We used IP unicast, IP broadcast and IP multicast mechanisms which are implemented over UDP. IP unicast is required for control messages concerning only the WICs that they are sent to or initiated from. IP broadcast mechanism is employed on the WIS to send control messages that concerns all WICs in the SA. IP multicast mechanism is employed to transfer data simultaneously to multiple users, which are interested in that information.

For the WIS-WICC communication, TCP protocol is used. Between these two components, control messages concerning the administrative databases of the system are transferred. However, these messages are crucial for the system integrity and we have to make sure that they reach to the recipient.

2.4 WIDE Communication Design

Communication between a WIS and WICs proceeds in cycles called Communication Cycles (CCs). In each CC, there are specific time periods in which certain tasks are performed. These time periods, which are named as index broadcast period (IBP), reception preparation period (RPP), data period (DP), authentication period (AUP) and request period (RQP), sequentially follow each other in this order in time. DP is also divided into time slots, which are called as communication slots (CS). Figure 2 illustrates the timing diagram of a CC.

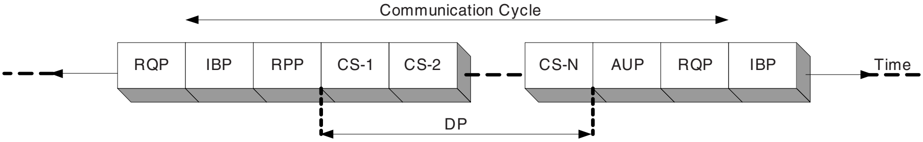


Fig. 2. Timing diagram of a CC

A WIC entered to the SA of a WIS, sends its authentication request to WIS in AUP to be able to receive service from the system. WIS sends the response to authentication request in AUP of a following CC.

Requests of WIC for subscription to information services or requests for unsubscriptions are transmitted to WIS on RQP. In addition, retransmission request for the information services, whose packets are missed, and polling request for updates of information services on the user profile are also transmitted to WIS on RQP. WIS sends the corresponding response messages to WIC on the same RQP.

A scheduler running in WIS decides data to transmit during each CC and prepares the index. The scheduling of an information service in a WIS, requires at least one WIC in the SA of that WIS who previously subscribed or has just subscribed to that service. If WIC has just subscribed to that information service or a retransmission is requested for that service from WIS due to incomplete reception, then that service is queued for delivery. In addition, if WIC has made an authentication or polling request and if there exists a version of that information service that is newer than the one recorded in the user profile of that WIC, then that service is queued for delivery. At the time of delivery, the service appears on the index.

When a WIC is within the SA of that WIS, it listens to the index sent on IBP to see which information services are offered by WIS during that CC. This index message also informs the clients interested with the information service about the multicast group of transmission and the version of the data to be transmitted. Each multicast group is coupled with a CS in DP. Application program running on WIC examines the index and determines whether there are any available items of interest by examining the user profile existing in the mobile terminal. If items of interest are available, WIC performs the necessary operations such as joining to the announced multicast group and preparing the buffers for receiving an information service in the RPP. Information services are delivered to WICs in the form of packets of fixed size. Data packets of each item announced for that CC in the index are delivered in the corresponding CS in DP. Consequently, WIC will receive data packets of the interested service from the joined multicast group, while other data packets will be dropped by the IP layer.

Data communication between WICs and WISs is established by using IP broadcast, IP multicast and IP unicast technologies together. There are several virtual channels utilizing the above technologies. These channels do not physically exist and they share the same physical channel of 802.11b. Virtual channels are defined in the transport layer and can be classified as point-to-point channels, broadcast channel and multicast channels as identified by their corresponding IP addresses and port numbers.

There is only one Broadcast Channel (BCH). Point-to-point channels used in the system are named as Uplink Authentication Channel (UACH), Downlink Authentication Channel (DACH), Uplink Request Channel (URCH) and Downlink Request Channel (DRCH). Data Channels (DCHs), whose number is a predefined system parameter, are multicast channels and each DCH has its own communication slot. The virtual channels, time periods and the messages between WIC and WIS are illustrated in Figure 3.

BCH can be considered as a control channel. WIS sends the index message on the period IBP; start and finish probes announcing the beginning and end of AUP and RQP periods. When a WIC enters to the service area of a WIS, it is informed about the existence of that WIS and then synchronizes with the communication cycle of that WIS by listening to BCH.

UACH and DACH are used for the authentication of WICs. WICs send their authentication requests over UACH on the periods AUP and WISs send authentication notification messages to WICs over DACH on the AUP periods.

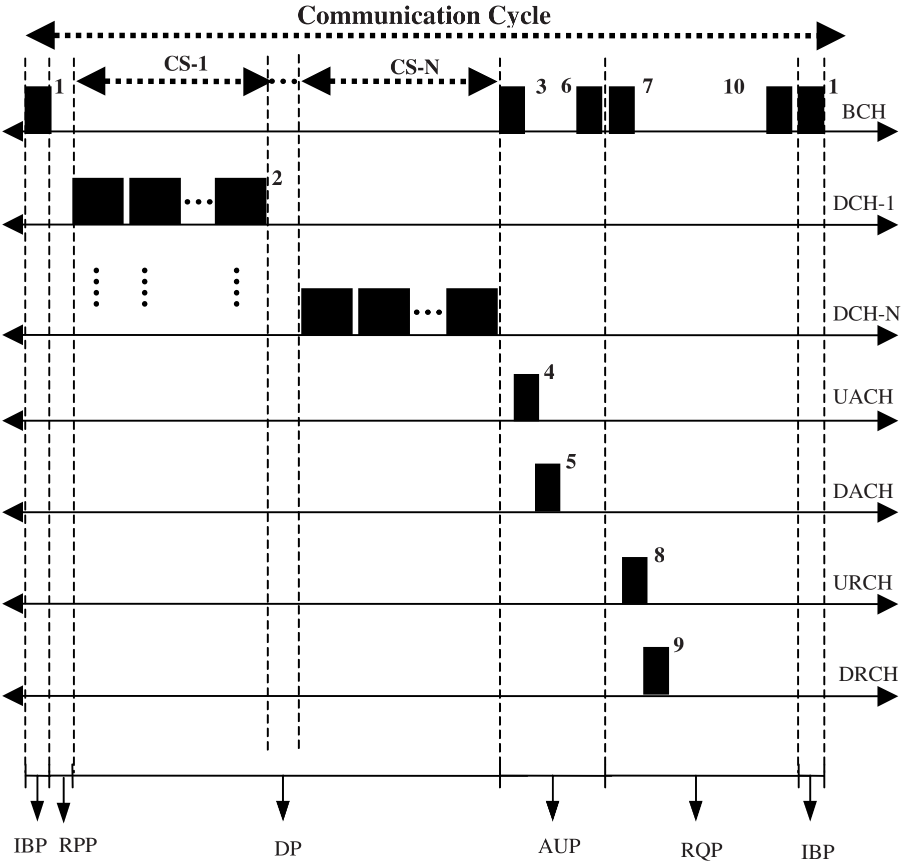
URCH and DRCH are used for subscription, unsubscription, retransmission requests and notification purposes. WICs send subscription to an information service request, unsubscription from an information service request, periodic polling request, retransmission of an incomplete transfer request, over URCH on the periods RQP. WISs send notification of these requests in the same RQP.

The time period of each DCH corresponds to a CS, which appear in a DP one after the other. DCHs are used to deliver information services to WICs on the corresponding CS in period DP. Index message informs WICs about the DCH over which the information services will be delivered in that CC. WICs join the IP multicast groups in the period RPP, if the information services announced are the interested ones. They start to receive UDP packets including the bytes of the interested information service on the corresponding period in DP in that CC.

WIS and WICC communicate with each other over a full duplex TCP channel. The communication between these two components of the WIDE system is realized by messages exchanged over this channel.

2.5 Mechanisms of WIDE System

Publish / Subscribe Mechanism. Subscription to information services is provided with a publish/subscribe mechanism in WIDE system. The list of the information services offered by the system is called the table of contents (TOC), which is also offered as a service. In WICs, a user interface is provided to view the local copy of



1. Index Message
2. Data Packets
3. AUP Start Probe
4. Auth. Request Message
5. Auth. Notification Message
6. AUP Finish Probe
7. RQP Start Probe
8. Request Messages
9. Request Notification Messages
10. RQP Finish Probe

Fig. 3. Communication cycle and virtual channels

TOC to users. Figure 4 shows the TOC GUI on a PDA client. Users may create a subscription request anytime and anywhere with the help of this user interface. Subscription requests are transmitted automatically to WICC via a WIS when WIC roam into the SA of that WIS. A local and a remote list of subscriptions are kept in WIC and WICC respectively. Initially the TOC service is automatically in the subscription list for each WIC. Remote subscription list is kept as a user profile for each WIC.

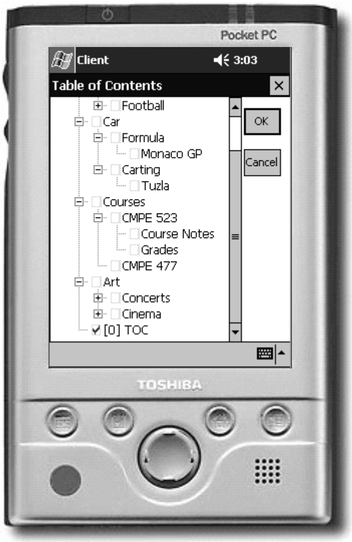


Fig. 4. Local copy of Table of Contents on a PDA client

Similarly, if user does not want to receive or update a service any more, he can create an unsubscription request with the help of TOC user interface. Unsubscription requests are also transmitted automatically to WICC via a WIS when WIC roam into the SA of that WIS. The entry for the service that the user wants to unsubscribe from is deleted from the corresponding user profile.

Reliable Data Delivery Mechanism. The messages initiated by the WIC have to be acknowledged by the WIS. These messages are the request messages related to information services. WIC has to be sure that its requests are received by the WIS and they are being processed in the system. The functionality of the system depends on the reliable delivery of these messages to WIS. If acknowledgement messages for information service related requests are not received by the WIC in the same request period, then it repeats its requests in the next request period. Similarly, if acknowledgement messages for authentication requests are not received by the WIC in a predefined duration, then it repeats its requests in the next authentication period after the expiration of the duration.

We choose to employ a reliability mechanism, which employs a mixed type of carousel [10], erasure code and automatic retransmission request (ARQ) [11] techniques. We do not require the reliable transmission of each data packet individually. Before transmission of data packets of an information service, these packets are encoded using a forward error correction (FEC) technique called erasure codes in which the reception of any k packets out of $k+m$ transmitted packets is sufficient for reliable reception [12]. After this phase, a packet number is given to each packet in sequential order. WIC keeps track of the packets that it received using these packet numbers. This helps WIC to discard the duplicate packets caused by the carousel mechanism. When enough number of packets is received for the FEC decoding process, these packets are decoded to form the actual data packets in sequential order. If the received number of packets is not enough to recover the actual data, the missing packets can be captured in the next carousel cycle, if exists. If there are still missing packets to be captured, then an ARQ request is prepared by WIC to request the retransmission of that information service.

Security Mechanism. Security mechanisms include confidentiality, entity authentication, data origin authentication and data integrity. Symmetric-key encryption schemes are used to provide security in WIDE system. These encryption schemes have low complexity and high data throughput, providing fast and power-efficient processing [13]. The contents of each packet exchanged between WISs and WICs has to be kept secret from the public. Hence, each packet has to be encrypted with a key, which is only known by the endpoints of the communication and WICC. Each component in the system has a distinct key. WICC key is known by all of the WICs in the system. The headers of each packet initiated from a WIS are encrypted with the WICC key to be identified by each WIC including unauthenticated ones in the SA. The payload parts of these packets are encrypted with the WIS key which serves as a service key. The WIS key is acquired by WICs at the end of entity authentication operations. Entity authentication is accomplished in WICC by comparing the user password ciphered with WIC key in authentication request message with the correspondent in the user authentication table.

Authentication also applies to information itself. WIS and WICs entering into a communication should identify each other. Information delivered over a channel should be authenticated as to origin, date of origin, data content, time sent, etc. We used time stamping for each packet passed between WICs and WISs. Each party in communication checks the time stamps of the messages, which are received from other parties, and keeps the last received time stamp for each different party. The packets, which have time stamps earlier than or equal to the last encountered time stamp, are discarded. This prevents the situations in which any fake server captures the packets and delivers these copies or modified versions of these packets. Additionally, a two way challenge-response mechanism is applied to all requests and responses. WIS announces a challenge for AUP and RQP in start probes. WIC puts the response of that challenge together with its own challenge in the request message. In the notification message, WIS sends the response of challenge in the request message back to WIC. This mechanism ensures that the responding party is the one that is expected to respond. A challenge is a random number generated for each CC. There are distinct challenge functions in both WIC and WIS known by each other. If the response sent to the initiator of a challenge is the same as the result of the challenge function of the responding party, then that packet is recognized as a WIDE packet.

3 Implementation of a WIDE Prototype and Its Performance

We implemented a WIDE prototype, which delivers services in a campus environment. Initial prototype uses the Bogazici University, Computer Engineering (CMPE) WLAN access points as WIAPs. The components are implemented using

Microsoft Visual C++ 6.0 and Microsoft Platform Software Development Kit (SDK) for Visual C++ 6.0. WIC prototype is designed to run on laptop computers, which have Windows 98 Second Edition or later operating system on them for full functionality. WIS and WICC prototypes run on desktop machines which have a Windows 2000 Family operating system on them. In addition, we have a WIC prototype having partial functionality that runs on a Toshiba E740 PDA with Pocket PC operating system.

We designed some experiments to evaluate the performance of WIDE prototype. In these experiments, we executed the WIC application on a laptop computer with a Pentium III processor operating at 500 MHz and 192 MB RAM. The operating system on WIC machine is Windows 2000 Professional Edition.

WIS and WICC applications are executed on the same desktop computer with a Pentium IV processor operating at 1800 Mhz and 1 GB RAM. The operating system on server machine is Windows 2000 Advanced Server Edition.

The wireless connectivity between server and the client is provided with a Cisco AiroNet 350 Series WAP connected to the server machine via an Ethernet adapter and a 3Com AirConnect WLAN PCMCIA card installed on the client.

In our experiments, we set the data payload size to a value of 1400 bytes because the maximum throughput of UDP on wireless medium is achieved when the UDP packet size is 1472 bytes and rapidly drops after this value [14]. Each data packet contains the header of its own which is added by the WIS application. The sum of data packet payload, data packet headers and UDP headers should be less than 1472 bytes.

Typically, number of carousel cycles is set to a value of two. However, in our tests, we set this parameter to a value of one to be able to detect the number of cycles for the completion of the reception of an information service.

The length of the time periods in a CC are also kept fixed in the tests. The durations of RPP, AUP and RQP are set to 100 ms, 10 ms and 10 ms respectively on WIS. For these experiments, we prepared files of size varying from 100 KB to 1000 KB with an increment of 100 KB.

3.1 Effect of Socket Buffer Size and DBDP

The client machine is placed indoor, to five meters apart from the WAP, in the line of sight, to evaluate the effect of buffer size and *DBDP* on the performance. *DBDP* is the delay placed between each data packet. The requested information service was a file of size 100 KB. The socket buffer sizes were varied between the Windows default value of 8 KB to 256 KB increasing with powers of two. We measured the time between the reception of the first packet and the last packet of the file on the application layer. The experiments were repeated for different values of *DBDP* varying between 0 ms to 30 ms with an increment of 10 ms. In Figure 5, socket buffer sizes are plotted versus reception times for different *DBDP* values. Each data value on the graph corresponds to the average of 15 experiment results.

Figure 5 presents that the buffer sizes of sockets used for information service reception in client prototype significantly affect the performance. With the default buffer size, even if a delay of 20 ms is placed between data packets, packet losses occur because of buffer overruns. For a delay of 30 ms, read operations from the buffer is faster than write operations to the buffer. In this case, we get a straight line because there are not any buffer overruns. However, reception of 100 KB in two seconds is a poor performance. We tried to find a value pair of *DBDP* and buffer size such that there are not any buffer overruns but the reception time is kept as low as possible. It can be observed from the figure that the best possible solution in our experiments is to increase the buffer size to 256 KB and keeping the *DBDP* parameter value at 0 ms when the size of information service is 100 KB.

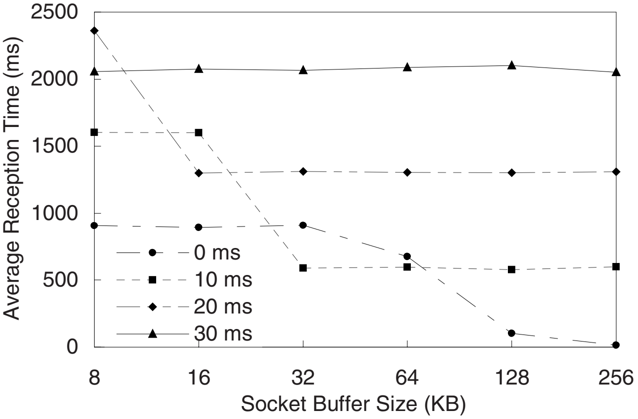


Fig. 5. The effect of buffer size and DBDP on average reception time

For higher file sizes, we could not achieve the same performance in terms of reception time with *DBDP* set to 0 ms. In Figure 6, the effect of *DBDP* values on performance in terms of reception time is presented. The setup for this experiment is the same as the previous one. The file size of the information service used in our experiments was 200 KB. Here, we fixed the buffer size of sockets to 256 KB. Each data value on the graph corresponds to the average of 15 experiment results.

If we compare the reception time values for 100 KB and 200 KB when the buffer size is 256 KB and *DBDP* is zero, we observe that there is a large gap between the reception time values. At this point, we analyzed the packets transmitted by IS and packets transmitted by WAP with a network analyzer [15]. We observed that although all the packets of the file were transmitted by the IS, some of the packets were not transmitted by the WAP. We also analyzed the statistics provided by the WAP to see that the existence of buffer overruns. Hence, we concluded that we should decrease the packet arrival rate to WAP by increasing *DBDP*. In this experiment, we observed that the best performance was achieved when *DBDP* value is around 3 ms. In the range between 0 ms to 3 ms decreasing packet loss led to increase in performance. After 3 ms, in a condition when there is no packet losses, increase in *DBDP* value led to extra delay between packets and hence, a decrease in performance.

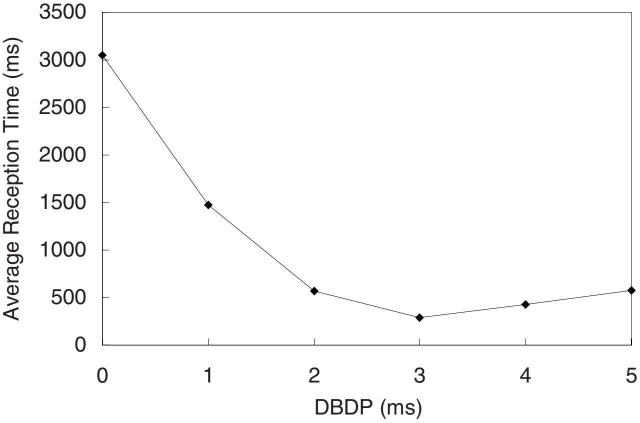


Fig. 6. The effect of DBDP on average reception time for 200 KB file

3.2 Effect of Information Service Size

We repeated experiments for all file sizes between 100 KB and 1000 KB by setting *DBDP* value to 3 ms to see the effect of file size on reception time. Results of these experiments are presented in Figure 7. In these experiments, we measured the reception time at the application layer, which is the elapsed time between reading the first packet and the last packet from the socket buffer, which is represented by application layer data reception (ADR) line. In addition, we measured the elapsed time between the occurrence of the first packet and the last packet on the WLAN with the help of the network analyzer, which is represented by network layer data reception (NDR) line. To be able to compare the achieved performance with another protocol, we downloaded the same files from the FTP server installed on the IS machine. The reception time results given by the FTP client is represented by FTP line. Each data value on the graph corresponds to the average of 15 experiment results.

The results show the expected behaviour for these three cases, which is the linear increase of reception time with the file size. We observe that the packets are not read from the socket buffer as soon as they arrive to the buffer. The most reasonable cause of this behaviour is the context switches to or from the data reception threads forced by the operating system. Because of the latency in reading the packets from the buffers, the packets accumulate in the buffers prior to any read operations. At the application layer, the reading speed is faster than the arrival rate of packets. Hence, the reception time in NDR is observed to be higher than the reception time in ADR. In addition, we observe that NDR performs similar to FTP. The average throughput is calculated as 3.84 Mbps for NDR and 3.83 Mbps for FTP.

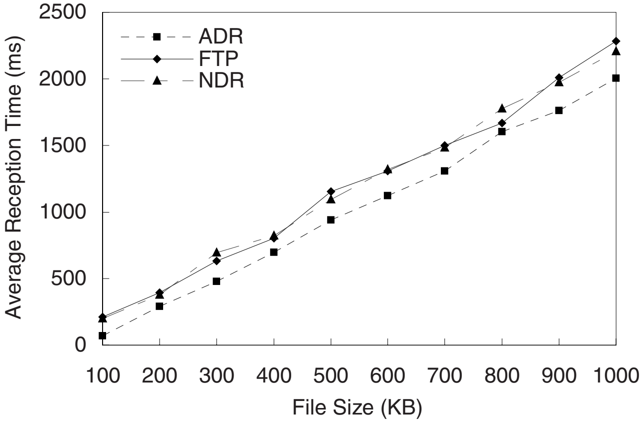


Fig. 7. Average reception time comparison with FTP for different file sizes

4 Conclusions and Future Work

WIDE is a data delivery system, which aims to offer popular information services to mobile clients using a distributed hot spot WLAN infrastructure. The requirements of the system are summarized and the system architecture is introduced. The protocols that WIDE is constructed above are discussed and the details of the communication design between components of WIDE are given. In addition, the mechanisms required for reliable and secure communication and data delivery are presented briefly. We also give some preliminary results of the performance evaluation on the implemented prototype of WIDE system. The initial prototype of WIDE client is implemented for Windows 98 SE or later platforms. The PDA version of WIC prototype should be improved to have full functionality.

Scalability and robustness of the WIDE system are not detailed in the current design. The primary goals of the current implementation were to prove the utilization of the system in a moderate-sized environment such as a university campus and find out the pros and cons of the current architecture. The future goal is to improve the architecture in terms of scalability and robustness. For this purpose, it is being planned to add backup authentication and profiling services to the current design. With these additions, the system will be able to operate under heavy load without affecting the performance dramatically.

We plan to give location based information services to clients. For this purpose we need to find the physical location of the client. We will integrate WIDE with WLAN Tracker [16] and give location based information services. Currently WIDE offers file delivery services. In the future framework, the system can be improved to give streaming and upload services such as music and video streaming, and e-mail transfer requiring special coding and security issues.

References

1. Frenkiel, R.H., Badrinath, B.R., Borras, J., Yates, R.D.: The Infostations Challenge: Balancing Cost and Ubiquity in Delivering Wireless Data. IEEE Personal Communications, pp. 66–71, April 2000.
2. DATAMAN Laboratory: NIMBLE: Many-time, Many-where Communication Support for Information Systems in Highly Mobile and Wireless Environments. http://www.cs.rutgers.edu/dataman/nimble/, 2003.
3. WICAT, Polytechnic U.: Infostation Project. http://wicat.poly.edu/ infostation.htm, 2003.
4. Frankl, P., Goodman D.: Technical Overview of the Infostation Project. http:// cis.poly.edu/research/infostation/InfostationOverview.pdf, 2001.
5. Banerjee, S., Agarwall, S., Kamel, K., Kochut, A., Kommareddy, C., Nadeem, T., Thakkar, P., Trinh, B.,Youssef, A., Youssef, M., Larsen, R.L., Shankar, A.U., Agrawala, A.: Rover: Scalable Location Aware Computing. IEEE Computer, Vol. 35, No. 10, pp.46–53, 2002.
6. Droms, R.: Dynamic Host Configuration Protocol. IETF RFC 2131, March 1997.
7. McAuley, A., Das, S., Madhani, S., Baba, S., Shobatake, Y.: Dynamic Registration and Configuration Protocol. http://hnmclab.csie.chu.edu.tw/~tmc/sip/draft-itsumo-drcp-01.txt, 28 May 2003.
8. Thomson, S., Narten, T.: IPv6 Stateless Address Autoconfiguration. IETF RFC 2462, 1998.
9. Perkins, C.E.: Mobile IP. IEEE Communications Magazine, vol. 3, pp. 84-99, May 1997.
10. Acharya, S., Franklin M., Zdonik, S.: Balancing Push and Pull for Data Broadcast. Proceedings of ACM SIGMOD Int. Conf. on Management of Data, pp. 183–194, 1997.
11. Rizzo, L., Vicisano, L.: RMDP: an FEC-based Reliable Multicast Protocol for Wireless Environments. Mobile Computing and Communications Review, Vol. 2, No. 2, pp. 1–10, April 1998.
12. Rizzo, L.: Effective Erasure Codes for Reliable Computer Communication Protocols. ACM Computer Communication Review, Vol. 27, No. 2, pp. 24–36, April 1997.
13. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC press, 1996.
14. Vasan, A., Shankar, A.U.: An Empirical Characterization of Instantaneous Throughput in 802.11b WLANs. http://www.cs.umd.edu/~shankar/Papers/802-11b-profile-1.pdf, 2003.
15. TamoSoft Inc.: CommView 4.0 Evaluation Version. http://www.tamofiles.com/ cv4.zip.
16. Komar, C.: Location Tracking and LBS in IEEE 802.11 Using WLAN Tracker. WLAN Tracker Technical Report, Bogazici University, TR-021, June 2003.