1995

Reliable Stream Transmission in Mobile Computing Environments

Houzhi Xu

Bharat Bhargava
Purdue University, bb@cs.purdue.edu

Report Number:
95-002

RELIABLE STREAM TRANSMISSION IN MOBILE COMPUTING ENVIRONMENTS

Houzhi Xu
Bharat Bhargava

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

CSD TR-95-002
Janurary 1995
(Revised January 1996)

Reliable Stream Transmission in Mobile Computing Environments

Houzhi Xu
Bharat Bhargava
Department of Computer Science
Purdue University
West Lafayette, Indiana 47907
{hxu, bb@cs.purdue.edu}

CSD-TR-95-02
January 1995

Abstract

Providing reliable stream transmission in the IP-based mobile computing environment presents a significant obstacle to the development of a practical mobile Internet system [B3C94], [MS93], [I93]. In this paper, we propose a protocol for reliable stream transmission in the mobile computing environment. By caching packets passing through the base station, we can not only establish a reliable stream transmission connection between the mobile host and its partner in the Internet, but can also guarantee the normal behavior of the TCP protocol in the Internet. Minimizing the computational overhead related to packet interception at the base station is also addressed in section "Implementation". Despite an increase in caching overhead at the base station. our mathematical model prodicts an overall improvement in communicaiton performance.

Key Words: Reliable Stream Transmission, TCP, Packet Loss Rate, Communication Latency, Data Caching.

1.0 Introduction

In 1991, an IP-based protocol for mobile internetworking [IDM91] was proposed. The proposed protocol was intended to efficiently support communication among mobile computers and the existing Intenet without requiring Internet modification. The mobility of mobile hosts was expected to be transparent to both the transport and application layers, enabling the current TCP protocol to be extended to the mobile Internet environment as the reliable stream transmission protocol. Several research groups [DFM91], [B3C94], [IM93], [I93], [MS93] have addressed the issues on the basis of the procedures similar to that proposed in [IDM91]. These efforts all encountered difficulties in supporting the TCP protocol in the mobile environment [MS93], [B3cC93], [I93], a probelm which had been predicted in [IDM91]. A network layer break of about ten seconds often resulted in a transport layer delay of thirty to forty seconds, excessive for the proper support of the current TCP protocol and its corresponding application program [MS93]. As pointed out by [IM93], [B3cC94], and others, these delays arise from a combination of the frequent disconnection of mobile hosts and the nature of the TCP protocol shceme. Packets intended for mobile hosts could be lost in transmission while the mobile host "disappeared" from base station. The source host of this connection would then retransmit these lost packets following a delay generated by Karn's alorithm in the TCP protoocl suit [C91], [RFC793]. Further theoretical and experimental work on a realible stream transmission protocol in the IP-based mobile Internet is therefore necessary.

At present, the TCP protocol, as a reliable stream transmisison protocol, is widely used at the transport layer in the Internet. The TCP must therefore be adequately suported if the applications are to be extended to the mobile computing environment. In [CDJM91], the statistical data collected in 1989 indicated that TCP packets make up roughly 80% of all wide-area network traffic. In [Pv93], It was noted that "at a site where the number of Internet hosts grew linearly, wide-area TCP traffic for a number of protocols grew exponentially both in the number of connections made and (at even higher rates) the amount of data transferred.". It is therefore critical to provide reliable stream transmission protocol in the Internet.

In this paper, we analyze a variety of approaches in order to determine the best method of data caching at the base station for reliable stream transmission in the mobile Internet environment. In section 3, we present additional validation for the selected approach. In section 4, the implementation algorithm for this approach is presented. Its computational and space requirements are analyzed. Section 5 presents conclusions and directions for further work.

2.0 Examination of possible approaches

We will now examine two alternative reliable stream transmission protocol options which may be employed in the mobile Internet environment. From this examination, we conclude that data (packet) caching offers the best performance..

2.1 Examing mobility transparancy of mobile hosts at transport layer

As pointed out in the previous section, several previous research efforts have been directed toward ensuring the mobility transparency of mobile hosts with respect to the transport layer.These investigations indicated the presence of an exponential delay at the transport layer in the process of handoff. Our own research suggests that it is not possible to ensure complete transparency of mobility with regard to the transport layer. First, we haven't found a way to completely eliminate the temporary disconnection at the IP layer while a mobile host hands off from one base station to another. Second, even if this temporary disconnection could be eliminated, the special characteristics of the mobile environment require that mobility remain visible to the transport layer. A reliable stream transmission protocol in the mobile computing environment must consider frequent and temporary disconnections which may arise when a mobile host is out of range of a base station, or is in an area of high interference or from battery expiration and predictalbe disconnection [X94], [IBm93]. Proper support of an efficient reliable stream transmission protocol for a mobile computing system requires moving beyond the IP layer to the addition of new protocols or modification of the existing TCP.

2.2 Modifying the TCP for the mobile environment

Reliable stream transmission in the mobile computing environment may also be achieved by using a modified version of the TCP on both the Internet and the mobile cells. There are, however, two obstacles to this approach. First, the TCP has been employed on the Internet for about two adecades and has been shown to produce excellent Internet performance. Modifying the TCP to suit the new mobile environment with the Internet may result in a loss in Internet performance.mpromise the new mobile environment with the Internet may cause mal-performance in the Internet. Second, even if the modified TCP protocol proved effective across the wide variety of environments found on the Internet and mobile cells, its overall performance would be very inefficient. Frequent disconnection, noise and interference, and low bandwidth [X94] will seriously impact the performance of the TCP in the mobile cell.

2.3 Separating the mobile cells from the Internet

The preceding analysis of possible approaches to the reliable stream transmission leads us to conclude that the Internet TCP protocol must be isolated from the reliable stream transmission protocol employed in the mobile cells. This approach is highly practical, permitting the TCP protocol to be used in the Internet and thus leaving Internet communications unimpacted while still supporting mobile computing under the Internet. Since the base station is the last stop for a packet enroute to a mobile host, a new protocol can easily be tailored to suit the specific conditions of mobile cells without impacting Internet performance [X94].

In the following section, we will present packet caching at the base station as a solution to reliable stream transmission in the mobile computing environment which provides optimal performance in the mobile Internet.

3.0 Packet caching at the base station

We concluded above that the separation of mobile cells from the Internet is necessary to ensure reliable stream transmission. To support this separation, each TCP packet passing through the base station must be copied and intecepted. Packet caching brings with it the unexpected additonal benefit [X94] of reduced packet loss rate between a mobile host and its partner in the mobile Internet. Suppose that the packet loss rates in the Internet and in a mobile cell are L1 and L2, respectively. The total packet loss rate in a mobile Internet situation without packet caching will be $1-(1-L1)\times(1-L2)$, which is approximately $L1 + L2$. However, the packet loss rate will reduce to $max(L1, L2)$ if we utilize packet caching at the base station.

We modify the two mathmatical models given in [X94] to reflect the absence of data sharing in pakcet caching. These modified versions ignore the data sharing parameter:

$$D = \frac{(2+n)\times(l1+l2+1)}{(2+o)\times(1+l2)+n\times(1+l1)} - 1$$

$$T = \frac{(n+1)\times(l1+l2+1)}{(l2+1)+(l1+1)\times n} - 1$$

Here D denotes the improvement in communication latency with data caching at the base station (BS) versus without data caching, T denotes the improvement in traffic load with data caching, n denotes the number of intermediate nodes between the BS and the resource host of the packet (including the resource host). *l1, l2* denote the packet loss rates in the Internet and in the mobile cells, respectively, and o denotes the rate of communication latency $o = \frac{\text{time spent in a base station with data caching}}{\text{time spent in a base station without data caching}} - 1$. Furher details can be found in [X94].

Consequently, packet caching at the base station actually improves the performance of the mobile Internet, although it does generate computational overhead at the base station. The following conclusions are obtained from the mathematical models:

- Data caching always improves the traffic load T.
- Data caching will have virtually no impact on communication latency if the packet loss rates in the Internet and on the air are approximately 10% and 3%, respectively, the overhead o at the base station is 40%, the intermediate node number n is 3, and there is no data sharing.

From this, we see that parameter n has a significant effect on communication latency, whil o has little impact. This implies that, even if the computational overhead at the base station is very high, the total latency will not greatly increase.

4.0 Implementation

In this section, we present an algorithm that efficiently supports the reliable stream transmission protocol in the mobile computing environment. The mobility of mobile hosts is hidden both from the mobile hosts themselves and from the base stations from the stand point of the transport layer. The TCP protocol remains the reliable transmission protocol, with change made only at the base station. In addition to being a router, a base station is now funneling reliable packet transmission via MTCP (Mobile TCP) protocol.

While we intend to minimize computational overhead at the base station, three new components must be introduced: packet hashing table (PHT), the MTCP process, and the MTCP timer process.

4.1 Packet Hashing Table (PHT)

A packet hashing table (PHT) is used to maintain the cached packets. Each table entry is a pointer to a connection record. The key of the table is KEY = <source-IP, source-port, destination-IP, destination-port>. And the access time to the table is a constant.

A connection record has the following pseudo data structure:

```
const int Max_Win_Size  8;
struct CONN_REC {
     List_of_Packets;
     Start_Packet;
     End_Packet;
     Is_On_Duty;
};
```

This data structure is actually a sliding window suit similar to that of TCP. The following example will be used to explain each field in this structure (Figure 1). List_of_Packets is the body of the window and consists of two parts. Part I contains the set of consecutive packets which have arrived from the sender. The aknowledgements for the packets in Part I are sent out by the base station on behalf of the mobile host prior to any actual response by the mobile host. Start_Packet and End_Packet point respectively to the beginning and end of Part I. Is_On_Duty will be used by the timer process and will be discussed later. The constant Max_Win_Size provides an upper limit on the number of packets between Start_Packet and End_Packet. We restrict that the number of packets between Start_Packet and End_Packet.

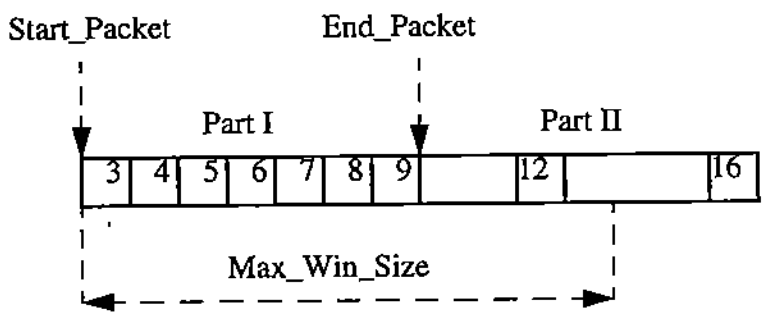

Figure 1: A schematic connection record

4.2 MTCP Process

The MTCP process handles all packets coming through the base station and emits control messages between base stations. Packets passing through the base station are of three varieties: associative packets, data packets, and acknowledgment packets (ACKs). In the remainder of this section, we present the MTCP process algorithms for each of these packets varieties.

4.2.1 Associative packets

Associative packets concern the creation and termination of connection record in the PHT of the base station. Such packets may address the stabilization and the termination of the connection between thesender and the requester (SYN, AXCK, FIN and their combination) or may take the form of control messages between two base stations.

When a base station receives connection stabilization packets (SYN, SYN+ACK), the MTCP process allocates an entry in the PHT for the connection and sets each field in the record as follows: Is_On_Duty = TRUE; Start_Packet = End_Packet = the sequence number contained in FIN packet. When the base station receives termination packets (FIN, FIN+ACK), the MTCP process sets IS_On_Duty = FALSE.

Control messages between two base stations are synchronize the physical transfer of the mobile host between base stations. The original base sattion must send the value of Start_Packet to the new base station (this message is called HANDOFF) via a packet created by its MTCP. It also sets Is_On_Duty = FALSE. The new base station, on up receiving the HANDOFF, will allocate an entry in the PHT and perform any related functions as if it received a connection stabilization packet. It also creates and sends a packet destined for the original base station (called HANDOFF_ACK) to ensure reliablility. The original base station will retransmit the HANDOFF until it receives the HANDOFF_ACK.

4.2.2 Data Packets

If the incoming packet is a data packet, the MTCP process forwards it and then checks the PHT for a preexisting connectin record, If no connectin record is found, it takes no further action, If a connection record does exist in th PHT, it will insert the packet into List_Of_Pakct and update End_Pakcet. it checks the PHT to see if the connection record has already existed. If the connection record is not in the PHT, do nothing else. If the connection record is in the PHT, it will insert the packet into List_Of_Packets, and update End_Packet. The MTCP must verify that End_Packet - Start_Packet is less than Max_Win_Size. If the updated End_Packet is still in the allowed range, it will send out the ACK for the received packet.

Note that List_Of_Packets will not be longer than Max_Win_Size + the window size of the TCP sliding window in the sender site as long as we stipulate that the End_Packet be within the allowed the range. This guarantees that the update time and that memory usage in the base station is therefore constant for each connection.

4.2.3 ACK Packets

The handling of ACK packets from either the mobile host or the new base station is straight forward. The MTCP deallocates all sources for the left side of Part I starting from the acknowledged packet (shrinks the sliding window in the base station) and transmits all the packets in the right side of Part I starting from the acknowledge packet. It is possible that no connection record has been entered in the PHT after receiving an ACK from the mobile host since the HANDOFF message has not yet been received. The MTCP will then forward this ACK to the original base station.

4.3 Packet retransmission

Like the sender of a TCP connection, the base station must provide a timer for cached packet retransmission. The different components of this timer are the period time between two retransmissions and the deletion of the connection record from the PHT.

If the sender is a fixed host in the Internet and the receiver a mobile host, the period time between two retransmissionswill be fixed. This strategy prevents the exponential delay in packet transmission and does not impact traffic congestion in the Internet. In the case of reversed data flow from mobile hosts to the Internet, the Internet can be adapted for the retransmission period (Karn's Algorithm). The strategy is objective to the TCP protocol.

The MTCP must also delete the connection record from the PHT. When the timer activates a PHT entry, it will retransmit the packets between Start_Packet and End_Packet. If Start_Packet is equal to End_Packet, it ascertains whether Is_On_Duty is FALSE, in which case the MTCP will deallocate all sources for this connection entry.

4.4 An example

In this section, we provide and example to illustrate the cooperation between the MTCP and TCP in guranteeing. We only consider how the MTCP intercepts TCP data packets and ACK packets. The following example, which is shown in Figure 2, assumes a scenario in which a mobile host MH retrieves a file from a file server in the Internet using FTP.

When a base station BS receives a data packet "data" from the sender (in this case, the file server), the BS simply forwards the packet to the MH. If a connection record exists for this packet, it either makes a local copy of the packet and updates End_Pakcet or simply proceeds to access the next TCP packet. If the packet is still within the allowed range (between Start_Packet and Start_Packet + Max_Win_Size), the BS either sends and ACK back to the file server or simply waits for the next TCP packet. If the data packet "data" is in transmission, the BS will retransmit the lost data after a fixed timeout until receiving an ACK from the MH. When BS2 receives an ACK from the mobile host (MH in FIgure 3), the MH tries seeks the relevant entry in the PHT. If it finds the connection record, BS2 shrinks its window by deallocating the related sources or forwards the ACK to BS1 for deletion. BS1 will follow the same procedure, deletingthe previous packet which was located in one of the previous base station. The base station which responded to the ACK will transmit the right side of the Part I in the sliding window.

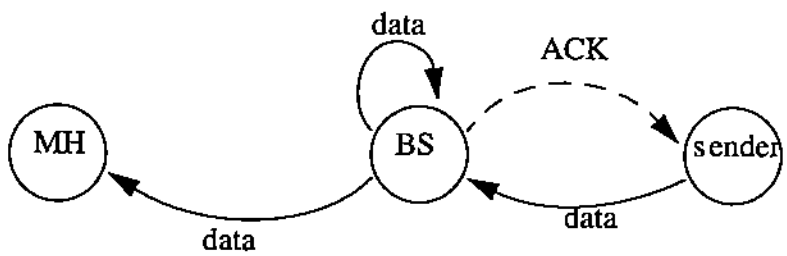

Figure 2: TCP data packet forwarding

4.5 Computational complexity

The MTCP protocol reaches its maximum performance under conditions of constant computing time, when a mobile host maintains a constant Internet connection without disconnections or correctly performs a handoff.. There is a constant delay in packet delivery between the IP and the MTCP as well as delays in computing time for the MTCP code. If the MTCP process is called upon less often than the IP process, we can assume that the delay for delivery between the IP and the MTCP is constant.

The worst delay occurs when the acknowledged packet remains in the original server while the mobile host has passed n base station. In this case , the delay is (n * the constant delay in each base station). This circumstance is, however, unlikely to be common. FIrst, when the mobile host performs a handoff, the packets left in the previous base station are also intended to be transferred to the new base station. Second, wireless communication is much slower than are communications in a loca network area, and two successive base stations are likely to be linked in a local area network.

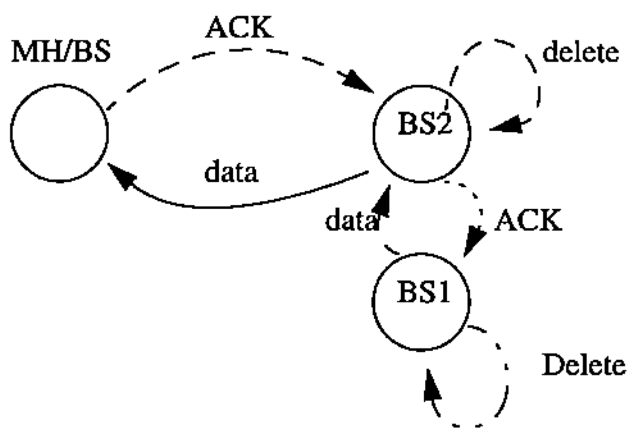

Figure 3: The response to ACK

4.6 Overall performance analysis

From the above section, we observe that the TCP packet passing through a base station has a constant delay that arises from the interception of the MTCP. However, packet caching at the base station also redues the packet loss rate, implying that the average communication latency for a TCP packet is at leaset linearly reduced and is possibly exponentially reduced due to the high varablility of the Internet [CDJM91]. On balance, then, the average performance is actually improved when comjpared to that at base station without packet caching.

5.0 Conclusion and Future Work

The combination of the TCP and the MTCP protocol provides a feasible solution to the support of reliable stream transmission in the IP-based mobile computing environment. Retention of the TCP protocoll as a part of the reliable stream transmission protocol of the mobile IP guarantees that all existing application programs will be available to a mobile host.

Data caching at the base station provides the infrastructure for the reliable data transmission protocol. Data caching introduces both a lower packet loss rate and a higher space and computational overhead, two factors which are essentrially in balance. The relaible stream tarnsport protocol can be thus supported via data caching without extra cost.

In the future work, we plan to design and implement a mobile computing system with reliable stream transmission under the Internet.

References

[B3C94] Trevor Blackwell, Kee Chan, Koling Chand, Thomdas Charuhas, etal. *Secure Short-cut Routing for Mobile IP*. Usenix Summer 1994 Technical Conference, Boston, MA, June 1994.

[C91] Douglas E. Comer. *Internetworking wiht TCP/IP. Vol. 1: principles, Protocols, and Architecture*. 2nd edition, Prentice-Hall, Englewood Cliffs, New York, 1991.

[CDJM91] Ramon Caceres, Peter B. Danzig, etal. *Characteristics of Wide-Area TCP/IP Conversations.* ACM SIGCOMM'91.

[DFM91] Daniel Duchamp, Steven K. Feiner, Gerald Q. Maguire Jr. *Software Technology for Wireless Mobile Computing*. IEEE Network Magazine, November 1991, Page 12-18.

[I93] John Ioannidis. *Protocols for Mobile Internet Working*. Ph.D. Thesis, Columbia University, 1993.

[IBm93] Tomasz Imielinski, B.R. Badrinath. *Mobile Wireless Computing: Solutions and Challenges in Data Management*. Technical Report DCS-TR-296, Dept. of Computer Science, Rutgers University.

[IDM91] John Ioannidis, Dan Duchamp, Gerald Q. Maguire Jr.. *IP-based Protocols for Mobile Internetworking*. SIGCOMM'91.

[IM93] John Ioannidis and Gerald Q. Maguire Jr. *The Design and Implementation of a Mobile Internetworking Architecture*. 1993 Winter Usenix, January, 1993, San Diego, California.

[MS93] Andrew Myles and David Skellern. *Comparison of Mobile Host Protocols for IP*. Macquarie University 2109, School of Mathematics, Physics, Computing , Electronics, April 1993.

[Pv93] Vern Paxson, *Growth Trends in Wide-Area TCP Connections*. Technical Report, Lawrence Berkeley Laboratory, November, 1993.

[rfc793] *Transport Control Protocol.* DARPA Internet Program Protocol Specification, RFC 793, Sept. 1981.

[X94] Houzhi Xu, *Caching for Better Performances in a Mobile Computing Environment, M.S. Thesis*. Purdue University, December 1994.