Enhanced Algorithm of TCP Performance on Handover in Wireless Internet Networks

Dong Chun Lee[1], Hong-Jin Kim[2], and Jae Young Koh[3]

[1] Dept. of Computer Science Howon University, Korea
ldch@sunny.howon.ac.kr
[2] Dept. of Computer Information, KyungWon College, Korea
[3]Director, Principal Member of Eng. Staff, National Security Research Institute, Korea

Abstract. In the TCP over Wireless Internet Networks (WINs), TCP responds to losses such as high bit errors and handovers by invoking congestion control and avoidance algorithms. In this paper we propose new handover notification algorithm that is to send an explicit handover notification message to the source host from mobile host when occurring to handover. Upon receipt of explicit handover notification, the source host enters persist mode. This way, data transmissions at the source host during handover are frozen. In numerical result, proposed algorithm provides a little performance improvement compared with general TCP, and expects to greater performance improvements while having frequent handover in WINs.

1 Introduction

The existing Internet environment has been changed to single networks, integrated wire and wireless, due to appearance of wireless networks. The first issue we faced from this Internet integrated networks is terminal mobility. Accordingly, it has been studying very actively about mobile IP which can correspond to terminal mobility, through reinforce of addressing and routing. Except this network hierarchy issue, we still have issues such as validation of TCP efficiency to guarantee reliability of connection between End-to-End [2,3]. TCP dominated in communication environment is suitable to traditional network which is composed of wire network and fixed host. Application of TCP to wireless network which is different from wired network such as bandwidth, high delay, discrete, bit error, disconnection and handover will cause falling in efficiency of End-to -End throughout by unnecessary calling of mechanism [4, 7].

TCP treats packet loss caused by bit error rate in network as congestion control mechanism; because it considers that packet loss is from congestion. This mistreatment of TCP results in efficiency falling in throughout. If TCP sender finds packet loss, he lessens transport window size and resend lost packet. Disorder control or avoidance and packet loss in network caused by disconnection in handover high bit error rate. If the packet loss recovery mechanism mentioned in advance is applied to network, it causes unnecessary performance degradation. In addition, if you apply TCP protocol to mobile host, it results in unnecessary decrease in terms of bandwidth usage and performance degradation through handling rate decrease and high delay. To make environment for mobile computing, we should find location of host which change its connection points, and keep connection with changeable connection during communication. The new revised activation method was invented and was studied. This activation isn’t impacted to the existing TCP, however it considers character of wireless networks.

There are several methods to improve End-to-end throughput in wireless network in different perspectives. They are End-to-End protocol, Split connection protocol and link hierarchy protocol. End-to-End protocol is the protocol which sender knows existence of network, and it uses mainly Selective Acknowledge which can recover several packet in window not to depend on timeout while resending packet and ELN which can protect call of disorder control mechanism, by informing packet loss reason is not disorder but network characters. As we may guess the meaning of terms, Split-connection is the way to apply proper protocol after splitting the connection of wire and wireless network. In other words, it is split into wire network between mobile host and Base Station (BS) and wireless network between FH and BS, Link hierarchy protocol as providing local reliability, uses combination of ARQ (Automatic Repeat request) and FEC (Forward Error Correction) which bring TCP improvement by hiding loss related wireless network in network hierarchy such as TCP.

This paper proposes the way to improve TCP performance on handover in WINs. When handover happens, the Mobile Host (MH) which recognizes handover sends explicit handover notification packet formatted as ICMP to the fixed host and informs handover. Through this process, it protects not to happen retransmission timeout and over crowded control process on force.

2 Related Work

Fast retransmission algorithm is the way to take lesser time of handover by quick resending process of fixed host through process that MH sends three overlapped Acks allocated last received pack to Fixed Host (FH) after handover completion [5]. The existed solution for improvement of End-to-End efficiency is for transported protocol to re-start sending data after handover completion. By this, re-sender timer of fixed host can escape time out. The advantage of quick re-sending algorithm is to be requested a least change for software at end host. Mobile IP is revised to send available handover at the one layer in protocol hierarchy, and also be revised TCP to call quick re-sending procedure. It doesn't depend on other medium router in wireless network [7]. After fist resending, it goes through complicated control procedure by closing window and using slow start algorithms. Therefore both rest networks and mobile host can escape disorder of cell.

Probe [3] is the way to make resending quickly about related packet by mobiles sending three Negative Acks to the FH. Probe algorithms is the Split-connection protocol hiding MH movement from TCP to decrease impact of handover in TCP function.

Probe chooses sign notifying handover in a layer of network as fast retransmission algorithms. This algorithm tries to solve disconnection over handover by using information from Mobile IP to find completion of handover.

Snoop protocol is the split connection way providing new routing protocol to decrease data loss on handover. If handover happens, it can skip the process forwarding data as new BS from primary BS, not like other protocol. In a result, it can remove data handover delay.

In previous handover algorithms, Fast Retransmission algorithm [4] as End-to-End protocol which fixed host, sender node, knows wireless existence reopens sending quickly to decrease stopping time over re-opening of communication in the network level after hand off completion, by measurement of impact from mobile movement over throughput and delay and recognition of reasons of function loss. In other words, it is the way to decrease time on hand off after completion of handover procedure. MH starts resending process of lost segment not waiting for re-sender timer after sending three overlapped Ack allocated in last receiving packet or for re-sender timer after sending to the FH. Because of fast retransmission call resending only when informed handover, it is only least change at software in existed FH. In addition, it doesn't depend on based network or medium router. However, it is only providing solution for packet loss due to disconnection in handover and disregards packet loss by high bit error rate of network in itself. Besides, after completion of handover, it takes round trip time to arrive completed bandwidth and it makes sending rate decrease by shrinking sending window size. Probe algorithms is the split connection protocol hiding movement of MH from TCP to decrease impact of handover in TCP function. It is the way to rise quick resending to corresponded packet by MH sending three Negative Acks to the FH after completion of handover. This algorithm has problems which it should always storage Ack packet and sends them and it cannot guarantee of transparency in hierarchy in the BS. Snoop protocol [2] is the split connection way to provide new routing protocol to decrease data loss on handover. This algorithm can remove delay over handover and improve efficiency in case of high bit error rate. However it has some deficits such as too much information related with handover process and heavy burden of unnecessary data and buffering,

The above algorithms to recover TCP efficiency decrease due to handover tried to improve End-to-End function through minimization of FH TCP timeout case and decrease delay time after completion of handover. However it still takes several round trip time to get completed bandwidth, and it has issue to resend lost data due to time out during sending data buffered in BS.

3 New Handover Algorithms

When the handover occur in a cell, MH send explicit handover notification packet with Internet control message protocol structure to FH and old BS, and indicate to handover start explicitly. Then it cannot be taken place the timeout of FH retransmission timer and the procedure of congestion control. The explicit handoff notification packet include the window size where window size mean to buffer size of received host and the address of new BS which it is after the handover in MH. This make sender FH to persist mode, and sending host cannot transmit all packets basically. Procedure of packet change in handover is the following figure 1.

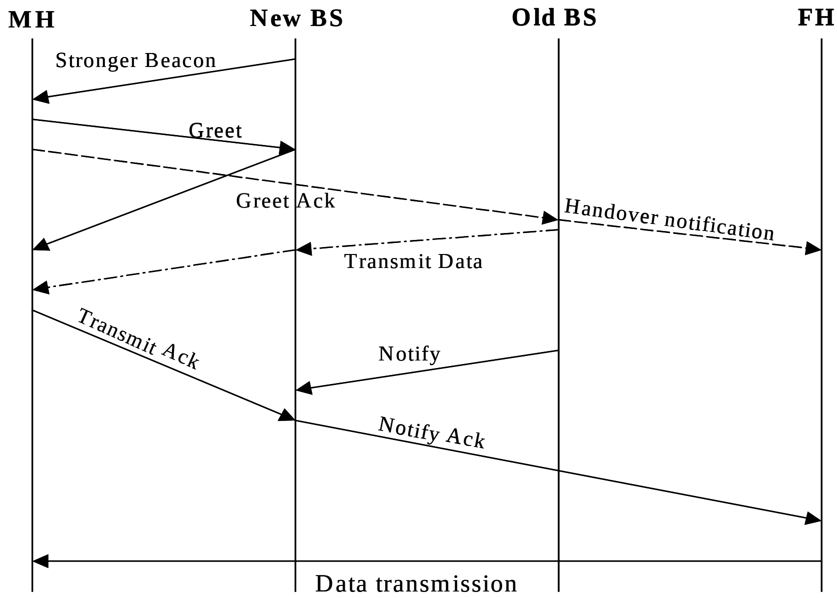

Fig.2. Message changing procedure during the handover

In the handover process, the MH receive new beacon, and if it know handover start, new BS send Greet packet including IP address of FH with old BS and its IP address. FH send explicit handover notification packet with window information and address of new BS of MH. New BS send greet Ack packet to the MH, and old BS transmit buffering data to MH through new BS. If the handover is over, the MH sends information of the great sequence number that received from old BS, and the FH can be received packet data. When the FH receive explicit handover notification packet, it changes value of sending window size to 0, and waits for the Ack message of MH that handover will over after change to persist mode in itself. In persist mode, it halts to retransmission timer and states that is in FHs, and sends Probe packet period that inquire to increasing window size for sending packet to receiver node using the persist mode, which Probe packet process in the new BS. And when Ack message from MH send to new BS, FH change persist mode to normal mode, which have value advertising window before handover. Then the FH send packet as sending window size that use before handover. Explicit handover notification packet happens to one time during handover, and it has no much effect on traffic of the path. This algorithm is following

Algorithm Handover
  Procedure MH
  while
  If Handover happen to Then

```
   make explicit handover notification packet;
   explicit handover notification packet;
   mobile host IP, fixed host IP, New BS IP,
   mobile host's advertising window = 0;
   send explicit handover notification packet;
 If handover is over  Then
   make Ack packet;
   include information of sequence number from old BS;
   send Ack packet;
 Procedure New BS
 while
 If old BS's buffering packet receive  Then
  send buffering packet;
 If Probe packet receive Then
  Persist reply mode;
Procedure FH
 while
 If explicit handover notification packet receive  Then
  enter Persist mode;
 If Ack packet receive from MH  Then
  come out form the Persist mode;
  end FH;
  end New BS;
  end
```

4. Simulation and Analysis

The simulation was performed using NetSimulator and IBM compatible PCs with Pentium. For the purpose of this model, we have assumed that simulation is performed base on network model that configure switching system with linking to nodes. Fig. 2 shows that this network model consists of FH, MH, and BS in mobile environments. In this model, it assume that wire link has 56 Kbps, wireless link has 19.2Kbps, and maximum window size is 32, and packet size is 128 Bytes, and TCP in FH make use of Tahoe TCP protocol. In order to ensure proper packet processing, we applied the information used for actual simulations, which was computed using the Poisson distribution, equally to mobile networks. Simulation carried to 7 times handover by seconds during 100 seconds to compare to performance analysis.
Fig. 3 shows throughput that MH is in handover. The proposed algorithm has packet loss less than the general TCP. Because it transmits buffering packet to MH through old BS during the handover. The proposed method shows that the more handover time takes long, all throughput decrease, and staying time in cell take time more, the more throughput decrease. Also, the more holding time in cell and handover time take long, the more performance has difference.

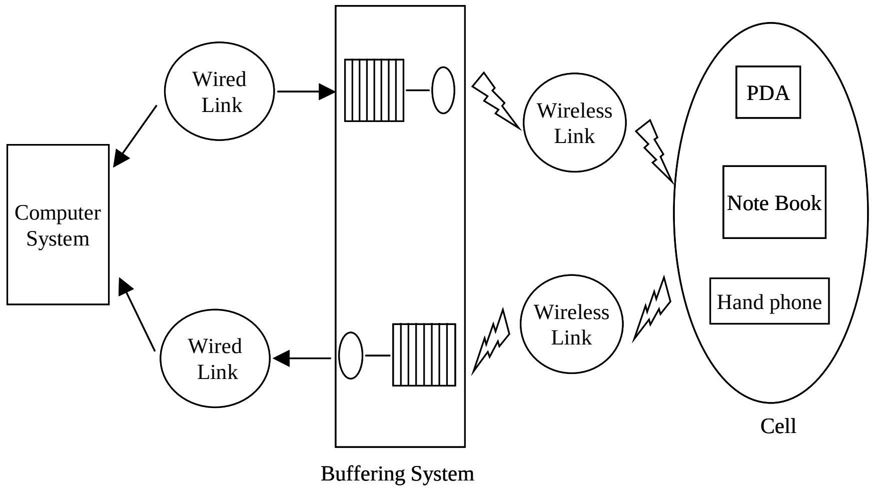

Fig. 2. Network model

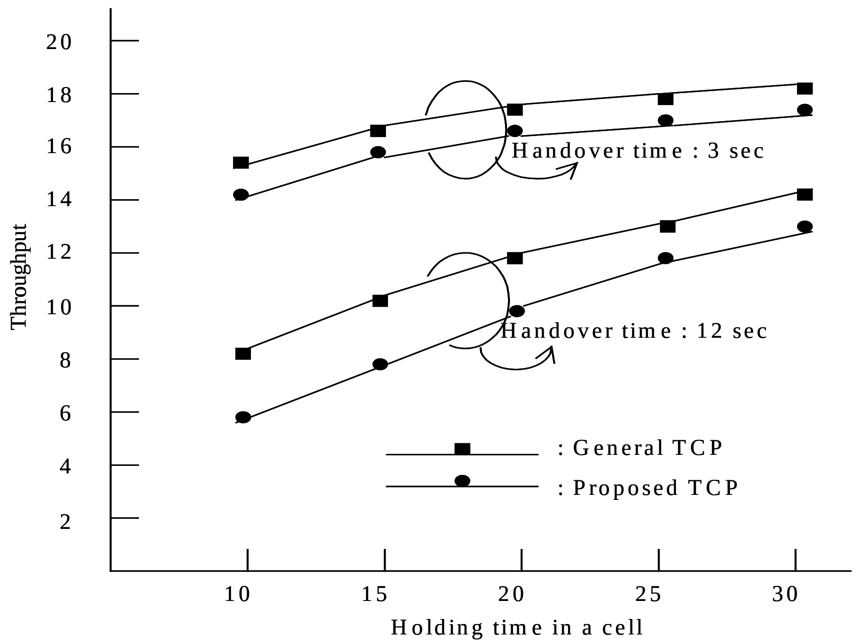

Fig. 3. Comparison of handover time

5. Conclusions

In this paper we propose new handover algorithm that improve the TCP performance degradation in handover. When starting handover, upon receipt of an explicit handover notification, the source host enters Persist mode. This way, data transmissions at the source host during handover are frozen. In numerical result, the proposed algorithm show that TCP performance provides a little performance improvement more than previous algorithm, and expect to greater performance improvements while having more handovers.

Acknowledgement

This work was supported by grant no. R05-2004-000-11267-0 from Korea Science and Engineering Foundation (KOSEF), and (in part) by the Ministry of Information & Communications, Korea, under ITRC Support Program..

References

1. Bikram S. Bakshi, P. Krishna, N. H. Vaidya, D. K. Pradhan, "Improving Performance of TCP over Wireless Networks", Technical Report # TR-01-014, Dept. of Computer Science, Texas A&M University., 2001.
2. H. Balakrishnan, S. Seshan, R. H. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks," ACM Wireless Networks, Vol. 1, Dec. 1997.
3. H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A Comparison of Mechanism For Improving TCP Performance Over Wireless Links," IEEE/ACM Tran. on Networking, Vol. 5, 1997.
4. R. Yavatkar and N. Bhagwat, "Improving End-to-End performance of TCP over Mobile Internetworks," Proc. of Moblie'94, 1994.
5. R. Caceres and L. Iftode, "Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments," IEEE J.on SAC, Vol. 14, No. 5, 1996.
6. A. Bakre and B. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts," Technical Report DCS-TR-614, Rutgers University, 1999.
7. H. Balakrishnan, S. Seshan, E. Amir and R. H. Katz, "Improving TCP/IP Performance over Wireless Networks," Proc. of Mobicom'95, 1995.
8. E. Ayanoglu, S. Paul, T. F. Laporta, K. K. Sabnani, and R. D. Gitlin, "AIRMAIL: A Link-Layer Protocol for Wireless Networks," ACM/Baltzer J. of Wireless Networks, Vol. 1, 1998.