BANDWIDTH ESTIMATION IN WIRELESS NETWORKS (TCP-BWIW): TO IMPROVE TCP PERFORMANCE IN WIRELESS COMMUNICATION NETWORKS

A. Ezil Sam Leni, M.S.
Research Scholar
Sathyabama Deemed University, India
Corresponding Author: email: Lenisatish99@lycos.com

S. K. Srivatsa, Ph.D.
M.I.T- Chennai University, India

Abstract - As wireless networks with high data rate get widely deployed, improving the performance of TCP over these networks plays vital role. Wireless link losses have dramatic adverse impact on TCP performance due to the difficulty in distinguishing the congestion losses from wireless link losses. We first, studied the various existing Bandwidth Estimation Algorithms over the wireless networks. In this paper, we propose an enhanced bandwidth estimation technique with congestion control algorithms to improve TCP performance over wireless link with random loss. We propose, a new technique called TCP-BWIW to estimate the bandwidth and the congestion window is set accordingly. The performance of TCP-BWIW is analyzed, and it mitigates the Network congestion.

Keywords - Bandwidth Estimation, Wireless Links, Congestion, Wireless Link Errors, Congestion Errors.

INTRODUCTION

Transmission Control Protocol (TCP) [1] performance over wireless links have been studied for the last several years. The TCP has proved efficient in wired networks, showing an ability to adapt to modern, high-speed networks and in Internet. One of the big challenges is improving the TCP performance in Internet Traffic, including the traffic generated by web accesses, e-mails, bulk data transfers, remote terminals and multimedia traffic. Internet Protocol (IP)[2] is a network layer protocol which is connectionless, best effort and variable length packet delivery. IP does not guarantee the reliable, timely and in-order delivery of packets between end stations. On the other hand, TCP is a Transport layer protocol that uses the basic IP services to provide end-end connection oriented services with reliable and ordered delivery of data.

The performance degradation of TCP in wireless networks, as in much research [3,4], is mainly due to its lack of ability to differentiate the packet losses caused by network congestion from the losses caused by wireless link errors. Therefore, the TCP congestion control mechanism reduces the transmission rate, even when not necessary. TCP Reno uses Additive Increase Multiplicative Decrease (AIMD) as the congestion control mechanism which uses three various phases namely, slow-start phase, congestion phase and congestion avoidance phase and maintains two state variables to regulate the transmission rate: congestion window *(cwnd),* and slow start threshold (ssthresh). The *ssthresh* sets *cwnd* that discriminates the slow start phase and congestion avoidance phase. At the beginning of transmission (slow start phase), the source increases the *cwnd* exponentially until congestion occurs. Once congestion is recognized, ssthresh is set to one half of the bytes in flight. When *cwnd* reaches the new *ssthresh* value, TCP enters a congestion avoidance phase during which *cwnd is* increased linearly as shown in Figure 1.

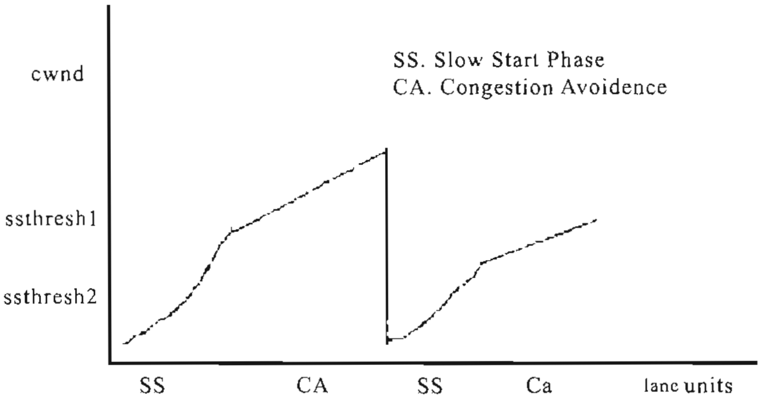

Figure 1: TCP congestion contron algorithm.

There are many bandwidth estimation schemes available. The rest of the paper is organized as follows: First, the characteristics of TCP and the challenges the TCP faces in wireless environments are discussed. Then, various bandwidth estimation schemes to improve TCP performance are reviewed. After that the proposed scheme is presented which is followed by a discussion of the simulation results of various schemes with our scheme. The paper is closed with conclusions.

CHARACTERISTICS AND CHALLENGES IN TCP
- CHARACTERISTICS

The mobile connectivity provided by the wireless networks allows users to access the information anytime anywhere. The basic characteristics of wireless networks are described below.

- - Bandwidth

Bandwidth need of wireless networks increases with increase in data transfer rate. Mobile users share the bandwidth within a cell and can move to another cell with higher or lower bandwidth, leading to variable wireless link data rates.

- - High Bit Error Rate

As wireless link errors occur in wireless networks, the bit error rate varies between 1%-10% of transmitted data, even though retransmission algorithms are used.

- - Varying Delay

Delay changes because deviation *of RTT* (Round Trip Time) causes spurious TCP Timeouts.

- - Asymmetrical Transmission Link

Wireless networks have asymmetric uploading and downloading rates. In most TCP schemes, packet loss occurs in the forward link.

- CHALLENGES THAT TCP FACES

- - ACK Compression

In TCP, arrival of ACK packets in the sender side, advances the congestion window, and thus increasing the data transfer rate. Instantaneous arrival of bursts of ACK in the reverse path degrades the Data Transfer rate because of the higher number of retransmissions. This forces the forward path to carry heavy traffic and exacerbates the forward path conditions.

- - Correlated Errors

In wireless networks, as the transmission technology is Radio transmission, the error process is caused by path loss, noise, fast fading, slow fading and interference from other devices. Moreover, there is no linear relationship between bit error and the distance.

- - Rerouting

During a TCP connection, if the routing-path is changed, it is not informed to the hosts and varies the *RTT* resulting discrepancies to the sender.

- - Clustering

In TCP, the packets from different TCP connections can share the same link. So, to correctly estimate the bandwidth in use, a TCP source must observe its own link utilization for a longer time than for the entire cluster transmission. The observation time depends on how many connections share the link and the cluster size. The cluster size depends on Bandwidth-delay Product (BDP).

BANDWIDTH ESTIMATION

Bandwidth Estimation schemes are used to estimate the Bandwidth available for the sender to transmit the data with better and fairer utilization of network resources. The literature proposes several Bandwidth estimation algorithms for TCP congestion control. In this section, first we study about the various, existing algorithms pointing with their performance.

Bandwidth Estimation algorithms are classified into Reactive schemes and proactive schemes. TCP New Reno and TCP SACK use the reactive mechanism and TCP Veno, TCP Westwood, TCP Jersey, TCP New Jersey and TIBET [6] are the proactive mechanism. The algorithms are explained with their own characteristics and advantages.

- TCP RENO

It is the simple and efficient algorithm for wired networks. It uses Additive Increase and Multiplicative Decrease as the algorithm as stated in, "Introduction", TCP Reno is incapable of handling multiple packet losses in one transmission window, which is the very likely situation in wireless links. In TCP Reno, it uses fast recovery algorithm for retransmission. After one packet drop is discovered, Reno terminates the fast recovery algorithms. So, multiple packet losses make the fast recovery algorithm again and again and slow down the recovery of the lost packet. To overcome this difficulty, TCP New Reno is implemented.

- TCP NEW RENO

In New Reno, the fast algorithm does not terminate until multiple losses are recovered. It cannot distinguish the cause if the packet loss and so more effective fast retransmission algorithm cannot be used.

- TCP VENO

TCP Veno estimates the backlogged packets in the buffer of the bottleneck link, and uses this estimation to differentiate the random error losses from the congestive losses. According to this estimation, if the number of backlogged packets is below a threshold, the loss is considered to be random and above a threshold, the loss is considered to be congestive loss. Even though this algorithm works well in some situations, the indication is irrelevant under some critical circumstances.

- TCP VEGAS

TCP vegas is the more sophisticated estimation scheme. TCP Vegas computes the difference between the expected and the actual flow rated that are defined by *cwnd/RTT* min (min value measured by the TCP source) and *cwnd/RTT,* respectively. In this when the network is not congested, the actual flow rate is close to the expected one, while, when the network is congested actual rate is smaller than the expected flow rate. It computes the quantity using the Equation Eq. (1):

$$\textit{diff= (expected\_ rate-actual rate) .RTT rain.} \qquad (1)$$

Because of the minimum RTT value, it suffers from rerouting and persistent congestion problems that are stated in the section related to “Characteristics and Challenges in TCP”. Due to this limitation it is not introduced in Internet.

- TCP WESTWOOD

TCP Westwood gives end-to-end bandwidth estimation in the sender side by measuring the rate of acknowledgements. This rate is used to set *ssthresh* and *cwnd* after congestion events, i.e. the receipt of three duplicate ACKs. This avoids reducing the data transfer rate to half in TCP Reno after packet loses and achieves fairer link utilization. TCP Westwood has various versions Westwood 1, Westwood 2 and Westwood 3.

- TCP JERSEY

TCP Jersey adopts slow start and congestion avoidance from New Reno, but implements rate based congestion window control procedure based on Available Bandwidth Estimation (ABE). It uses two components Congestion Warning (CW) and Available Bandwidth Estimation (ABE). The purpose of CW is to give the Sender the image of bottleneck queue. ABE uses the same concept of Westwood rate estimation with different implementation. ABE estimates the available Bandwidth according to the Equation Eq (2):

$$\begin{aligned} &R_n=RTTxR_{n-1}+L_n \\ &(t_n - t_{n-1}) + RTT \qquad (2) \end{aligned}$$

Where Rn, is the estimated Bandwidth when the n packet arrives, a time tn, tn-1 is the previous ACK arrival time, Ln, is the size of the data that the nth ACK and RTT is the TCP’s Round Trip Time Delay at time tn.

- TCP NEW JERSEY

TCP New Jersey [5] uses different rate estimation algorithm called Timestamp based Rate Estimation Algorithm (TABE). In TABE, we consider traffic pattern in both forward and reverse direction including congestion and error. The packet arrival time, which is stamped by the receiver and echoed back by the ACK, is used in Bandwidth estimation. The Bandwidth is given in Equation (3)

$$R_n = \frac{RTT \times R_{n-1} + L_n}{(t_n - t_{n-1}) + RTT} \qquad (3)$$

Where R ,, is the estimated Bandwidth, t is the arrival time of the nth packet at the receiver, ${}_{n_]}$is the previous ACK arrival time, L „ is the size of the data that the nth ACK and RTT is the TCP's Round Trip Time Delay at time t. The timestamp option is widely implemented in TCP and so there is no additional overhead to the overall protocol implementation.

PROPOSED SYSTEM

In Bandwidth Estimation schemes, the only quantity ensured efficiently with the ender side with one algorithm is the bandwidth used by the TCP source i.e. the used bandwidth. The available bandwidth is the maximum rate at which a TCP connection, experiencing congestion control could transmit data. We use this as the basic concept in our proposed system. TCPBWIW uses the bandwidth estimation on the sender side based on the time at which the last packet is transmitted, and in the receiver the receiver, receives side based on the time of the packet. By using these time durations, it computes the available bandwidth for so many packets transmitted and received. The Algorithm is given below:

- SENDER SIDE:

```
Algorithm[1 ]:-
if (Packet is sent)
{
Length [m] (packet_size *8);
SBWE [m] _ (nack *Length [m])/ (tnow - tlsend);
BWE [m] (1-fp) * (SBWE [m] *
SBWE [m-1])12)
+fp *BWE [m-1]; }
```

Where, Length is the packet length, nack is the number of segments acknowledged by the last ACK, tnow is the current time, tlsend is the time at which the last packet is sent, m and (m-1) indicates the current and the previous values of their respective variables, fp is the filtering factor and is taken as 116.

- RECEIVER SIDE:

```
-Algorithm [2]: -
if(Packet is received)
{
Length [m] =(acked* packet_size *8);
SBWE [m] = (nack*Length[m])/(tnow-
tlack);
```

BWE [m] =(1-fp) * (SBWE [m] *
SBWE [m-1])12 +fp *BWE [m-l];

Where nack is the number of segments acknowledged so far, tnow is the current time, Clack is the time at which the last ACK was received, m and (m-1) indicates the current and the previous values of their respective variables, fp is the filtering factor and is taken as 116.

During congestion occurrence, *cwnd* and *ssthresh* are updated according to the TCP-slow start congestion algorithm using the Equation [4] as

ssthresh = BWE [m] * RTT $_{\text{min}}$
(4) BWE is the estimated Bandwidth and RTT
1,,;,, is the minimum Round Trip Time.

The initial bandwidth is estimated based on either Algorithm [1] or Algorithm [2]. For the variation of *cwnd* and *ssthresh,* the following Algorithm [3] is used.

```
Algorithm [3]: -
if( three duplicate ACKs are received) {
ssthresh = BWE [m] * RTT if (cwnd >>ssthresh) cwnd-ssthresh;
}
if(retransmission timeout expires)
{ssthresh = BWE [m] * RTT Irvin; cwnd=l;
}
```

The network architecture that we have taken for evaluation of TCP- BWIW is shown in Figure 2.

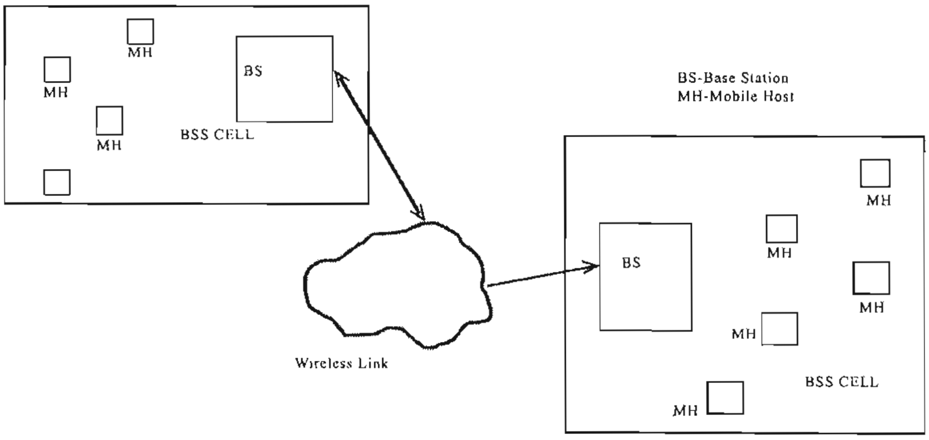

Figure 2: Simple Network Topology

In Figure 2, the Mobile Hosts (MHs) are connected to the Base Station (BS). The different Mobile Hosts, which are within a Radio Frequency, are attached to a specific Base Station and are together known as Base Station Subsystem (BSS) Cell. The Mobile Hosts are connected to their counterparts using wireless link as shown in the figure we used this simple network topology in our simulation work. The TCP packets are transmitted from one MH to another MH. We are looking for two conditions. In the first, the packets are transmitted; when there is no congestion we measured the throughput and packet loss, which are to be shown in the simulation results. By using the Algorithm [1] in the sending side and Algorithm [2] in the receiving side, the throughput is increased compared with the TCP Reno.

SIMULATION RESULTS

In the simulated Network, the MH is connected to the BS using wireless link. We have taken 2 Mbps link with ACK compression. In such situation, we have taken the simulation time and the estimated Bandwidth using TCP-BWIW and TCP-Reno are shown below in Figure 3. The Packet size is taken as 1,040 bytes. Figure 4.a and 4.b show the Bandwidth estimated in the presence and absence of RTT_{min} updating algorithm. Based on our simulation results, it is clear that during congestion, the proposed algorithm TCP-BWIW gives better performance in wireless links even when it is congested.

CONCLUSIONS

In this paper, we have studied the various existing Bandwidth Estimation algorithms for TCP congestion control. The proposed algorithm TCP-BWIW performs well during congested network. Even though the performance is good in normal condition, during fading wireless channels, the performance of this algorithm gives degraded throughput and is the limiting factor.

REFERENCES

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

[2] Postel, J. *Internet Protocol.* RFC 791, 1981.

[3] Chan, M. C. and Ramjee, R., "Improving TCP/IP Performance over Third Generation Wireless Networks." *Infocom* , 2004.

[4] Abouzeid, A. A. et al., "Comprehensive Performance Analysis of a TCP Session over a Wireless Fading Link with Queuing." *IEEE Transactions on wireless Communications,* Vol. 2, No. 2, pp. 344-356, 2003.

[5] Xu, Y. et al., "Improving TCP/IP Performance in Integrated Wireless Communication Networks." *Computer Networks,* Vol. 47, pp. 219-237, 2005.

[6] Capone, A. and Martignon, F., "Bandwidth Estimation Schemes for TCP over

Wireless Networks." *IEEE Transactions on Mobile Computing,* Vol. 2, 2004.

[7] Blanton, E. and Allman, M., "On Making TCP more Robust to Packet Reordering." *ACM Computer Communications Review,* Vol. 32, No. 1, 2002.