Impact Factor 2.147

ISSN 2349-638x

Reviewed International Journal

AAYUSHI INTERNATIONAL INTERDISCIPLINARY RESEARCH JOURNAL (AIIRJ)

Monthly Publish Journal

VOL-III | ISSUE- III | Mar. | 2016

Address
•Vikram Nagar, Boudhi Chouk, Latur.
•Tq. Latur, Dis. Latur 413512
•(+91) 9922455749, (+91) 9158387437

Email
•aiirjpramod@gmail.com

Website
•www.aiirjournal.com

CHIEF EDITOR – PRAMOD PRAKASHRAO TANDALE

Introduction Of Tcp Over Wireless Networking System

Prof. Kazi A.A.
Dept. Of Computer Science
Azad Mahavidyalaya Ausa.
Dist. Latur

Abstract:-

Transmission Control Protocol (TCP) was designed to perform well in wired networks where packet losses are mainly due to congestion. It has been observed that TCP performance degrades while operating over mobile wireless networks. In these networks packet losses occur also due to radio signal errors and disconnections in case of handover or loss of signal and not always due to congestion. TCP interprets packet loss on these networks as an indication of congestion and invokes congestion control mechanism as in a wired network. This reduces transmission rate drastically reducing throughput over these networks. In this paper we propose a simple TCP algorithm for wireless network environment. The proposed algorithm estimates the available bandwidth in the network using a forecasting method based on the rate of ACK's received by the sender. Further based on this estimated bandwidth the congestion control parameters adapt dynamically to the network state and thus preventing drastic throughput degradation. The proposed algorithm will be then compared to the existing TCP New Reno and TCP Westwood.

Key words: TCP, computer networks, congestion Control

I. Introduction

TCP was specifically designed to provide a reliable end to end byte stream over an unreliable internetwork. Each machine supporting TCP has a TCP transport entity either a user process or part of the kernel that manages TCP streams and interface to IP layer. A TCP entity accepts user data streams from local processes breaks them up into pieces not exceeding 64KB and sends each piece as a separate IP datagram. Client Server mechanism is not necessary for TCP to behave properly. The IP layer gives no guarantee that datagram will be delivered properly, so it is up to TCP to timeout and retransmit, if needed. Duplicate, lost and out of sequence packets are handled using the sequence number, acknowledgements, retransmission, timers, etc to provide a reliable service. Connection is a must for this service. Bit errors are taken care of by the CRC checksum. One difference from usual sequence numbering is that each byte is given a number instead of each packet. This is done so that at the time of transmission in case of loss, data of many small packets can be combined together to get a larger packet, and hence smaller overhead. TCP connection is a duplex connection. That means there is no difference between two sides once the connection is established.

The Transmission Control Protocol is a reliable, connection-oriented, full duplex, byte stream transport control protocol. This protocol supports flow and congestion control, and is used by many end-user applications, including web browsers and e-mail clients. TCP is designed with a assumption that packet loss means congestion in the network. It detects the packet loss either by the timer out of retransmission timer or reception of 3 duplicate

acknowledgments. After detecting packet loss, it reduces its sending rate as per congestion control mechanism. As per congestion control mechanism, TCP maintains two variable *ssthresh* and *cwnd* when TCP detects packet loss by timer-out, the congestion control mechanism reduces the *ssthresh* to half of present effective transmission window (effective window is minimum of advertised window by receiver and senders congestion window *cwnd*) and make *cwnd* equal to one segment and TCP enter the slow-start phase. During slow-start TCP increases its *cwnd* by one segment for every acknowledgment it receive. When *cwnd* become larger than *ssthresh*, TCP enter congestion avoidance phase during which it increment *cwnd* by one segment for one round trip time (rtt). When TCP detects packet loss by the reception 3-duplicate acknowledgment, congestion control mechanism make *ssthresh* equal to half of transmission effective window and enter the fast recovery mode. During fast recovery mode, *cwnd* is set to *ssthresh* plus 3 segments. Each time another duplicate ACK, increment *cwnd* by the segment size and transmit a packet (if allowed by the new value of *cwnd*) and when the next ACK arrives that acknowledges new data, TCP set *cwnd* to *ssthresh* and enter congestion avoidance. The existing versions of TCP, like Reno or NewReno, experience heavy throughput degradation over channels with high error rate, such as wireless channels. The main reason for this poor performance is that the TCP congestion control mechanism cannot distinguish between packet losses occurring randomly in wireless channels and those due to network congestion. In other words, the assumption that packet loss is always an indicator of network congestion does not apply especially to heterogeneous networks that include wireless links, in which packet loss may be induced also by noise or any other reason than congestion. There, random loss due to bit corruption is misinterpreted: upon loss detection, the TCP sender reduces its transmission rate unnecessarily thus degrading the throughput performance [2, 3]. In this work an accurate and efficient algorithm is developed to solve the fundamental Transport estimation problem - Bandwidth estimation, related to the development of end-to-end algorithms that measure the current TCP transmission rate and estimate the capacity available along the path from the source to the destination.

In the proposed algorithm, a simple filter is used for available bandwidth estimation and dynamic adjustment of filter parameter according to the state of wireless link improves the throughput of small-sized file transmission. The bandwidth available is estimated on the rate ACK packets received by the sender. Using this dynamically calculated bandwidth the values of *ssthresh* and *cwnd* are calculated.

The implementation is done at the senders end in the network and can be used in wired-wireless networks. We can investigate the performance of the proposed algorithm by simulation and show how effective the proposed algorithm is for the small-sized file transmission.

Ii. Related Work

In this section, we summarize the previous works related to the improvement of TCP throughput for wired-wireless networks. The approaches to improve TCP performance can be classified into (1) link layer protocols, (2) split-connection protocols, (3) snoop protocols, and (4) end-to-end protocols.

In the link-layer protocol, a wireless link is considered as a reliable link using link-layer techniques such as automatic repeat request (ARQ) and forward error correction (FEC) [8]. The link-layer protocols for CDMA primarily use ARQ technique. These protocols have advantage that those fit naturally into the layered protocol stack of the Internet. The link-layer protocol like ARQ or FEC operates independently of transport layer protocol like TCP. These protocols can improve TCP performance in wired-wireless network better than other protocols. However, these need special support of network infrastructure such as router. It is difficult for link-layer protocol to be deployed widely due to the expensive cost for special support of infrastructure.

The split-connection protocol splits TCP connection into wireless and wired parts at base station, and hides wireless loss from wired part connection. In MTCP [10], a specialized protocol in wireless link called selective repeat protocol (SRP) or UDP is used over wireless link. However, [10] reported that MTCP obtains no significant advantage in using SRP in comparison with TCP. In [11], M-TCP is used for wireless link. Another example is Split TCP[12] for mobile adhoc networks as shown in figure 2.1.

Indirect-TCP (I-TCP) [11] uses standard TCP over wireless link. However, like other split-connection protocol, using TCP over wireless link results in performance degradation because original TCP sender often experiences timeout and data transmission stalls frequently. In addition, base station maintains two TCP connections; one is wireless part connection and the other is wired one. So the overhead of base station is twice in comparison with non-split connection protocol and the base station needs higher resources such as buffer capacity and CPU power. Furthermore, split-connection protocol violates the end-to-end semantics of TCP. In addition, since base station maintains two sets of TCP status information for one end-to-end TCP connection, handoff procedures become complicated and take long time. As a result, the split connection protocols can hardly improve TCP performance in wired-wireless network.

TCP Westwood [6] performs available bandwidth estimation for the update of cwnd and ssthresh when fast retransmission or timeout occurs. By using available bandwidth estimation for setting cwnd and ssthresh, the sender does not reduce transmission rate needlessly and the throughput performance and link utilization is greatly improved. TCP Westwood achieves higher performance than TCP Reno in wired and wireless network environments. However, available bandwidth estimation algorithm is complex and the algorithm for available bandwidth estimation cannot follow the rapid change of network condition. Furthermore, TCP Westwood does not take the transmission of small-sized file such as Web document into consideration.

Iii. Proposed Works

The proposed algorithm consists of two parts; one is the estimation of available bandwidth and the other is the update of slow start threshold and congestion window size using the bandwidth estimate.

A. Available Bandwidth Estimation:

The proposed algorithm estimates network available bandwidth as follows. Let t(k) denote the time at which original sender host receives k ACK's and d(k) the amount of kth data segment sent to destination host. Whenever ACK reaches the sender host, the sender host calculates sample available bandwidth estimation B(t) using the following

B(k) = d(k(t(k) t(k-1)) (1)

Because ACK interval t(k) – t(k-1) fluctuates under TCP window flow control, the value of B (k) also fluctuates and is inaccurate estimate for available bandwidth. Therefore, the proposed algorithm smoothes Bold(k) using smoothing filter like other TCP variants including TCP Westwood. The filter implemented in TCP Westwood is well designed and provides more accurate estimates than other TCP variants proposed before. However, the filter of TCP Westwood is complicated and its computation time is not negligible, that is, computation overhead is larger than other TCP variants. Furthermore, it cannot capture the rapid change of available bandwidth well as stated in the previous section.

We focus on the well-known simple smoothing filter, exponential weighted moving average (EWMA). The EWMA filter is expressed with α (0 ≤ α ≤ 1) as

Bnew(t+1) = Bold(t) + – Bold(t)) (2)

The idea based on the moving average filter is that most recent samples are given higher weights and hence they reflect the current status of the filter. Note that a very large α provides that past estimates influences the present estimated bandwidth to a larger extent. This is good when we consider stable networks that do not change rapidly like a wired network. Conversely smaller α indicates that the estimate bandwidth depends on the current sample bandwidth.

Actually, it is difficult to obtain accurate estimates at the rapid change of network condition if the filter uses fixed α. For example, if the filter uses small α, the estimated value captures instantaneous rapid change well and hence small α is appropriate for the network whose condition changes instantaneously and rapidly as seen in wireless handoff. On the other hand, if α is set large, the estimate is less updated and this is preferable for the case in which the network condition does not change so much. In actual use of TCP over wired-wireless network, small-sized file transmission such as Web documents is a typical application. When a small-sized file transmission starts, TCP needs accurate available bandwidth estimate as soon as possible. This is the same situation as cellular handoff.

In order to obtain the high performance for the small-sized file transmission over wired-wireless network, we dynamically adapt α in (2) to the current network state. By dynamically updating α, Bnew is quickly adapted when TCP session starts or network condition changes rapidly, and Bnew is updated moderately when the network condition is stable.

When tth ACK is received, we first update α with Bold(t) and B(t) in the following manner:

if (B(t) ≥ Bold(t)) (3)

old(t) / B(t)

if (B(t) ≤ Bold(t)) (4)

t) / Bold(t)

Finally, updated α is substituted into (2) and tth estimate of available bandwidth Bnew(t) is calculated. Then Bnew(t+1) is used for setting slow start threshold (*ssthresh*) and congestion window (*cwnd*) in TCP congestion control algorithm.

Iv. Congestion Control Algorithm

Next we update *ssthresh* and *cwnd* and controls transmission rate when either timeout occur or three DUPACK (Duplicat ACK packets) are recieved. The proposed algorithm updates *ssthresh* and *cwnd* using available bandwidth estimate value Bnew.

if (3 DUPACK received)
    ssthresh = (Bnew*RTTmin);
if (*cwnd* ≥ *ssthresh*)
    cwnd = *ssthresh*;
    Algorithm: 1
if(Retransmission Timeout occurs)
    ssthresh = (Bnew*RTTmin);
    cwnd = 1;
    Algorithm: .2

When three DUPACK arrive, the proposed algorithm updates *ssthresh* and *cwnd* but does not drastically reduce the transmission rate as the congestion in network is minimal. When a timeout occurs, the *cwnd* is set to 1 as timout is an indicator of heavy congestion in the network as the ACK packets are not received. The transmission rate needs to reduce drastically.

V. Implementation

We plan to evaluate the performance of the proposed algorithm by simulation using network simulator ns2. We would then compare the proposed algorithm with the existing two TCPs: TCP Reno and TCP Westwood. To simulate TCP Westwood, we use TCP Westwood module for ns2.

Vi. Performance Evaluation

The outcomes of the proposed algorithm will be tested for efficiency and correctness using the following test cases

a. Effectiveness of bandwidth estimation: This would be used to test how long it took to estimate bandwidth compared to other TCP implementation like TCP Westwood. A graph of bandwidth estimated vs time would be used to plot the results.

b. Fairness Issue: The proposed algorithm will be tested if its implementation gives all connections the same opportunity to share data. A segment number vs time graph will be plotted to determine the same.

c. Friendliness Issue: The implementation will be also test if bandwidth is shared fairly among different TCP protocols.

The throughput in wired and wireless network would be determined and any significant changes would be analyzed

Vii. Conlusion

In this paper we have discussed about the congestion control mechanism in TCP. We have analyzed the problems related to congestion control which leads to the reduction of throughput in the network. Various other works related to providing a solution to the problem is discussed and its shortcoming have been identified. We have proposed a new algorithm for estimating the bandwidth using a simple low pass filter and using this estimated bandwidth to control congestion in the network.

Viii. References

[1] J. Postel, "Transmission control protocol," RFC 793, Sept. 1981.
[2] Mathis, Matthew, et al. "The macroscopic behavior of the TCP congestion avoidance algorithm." ACM SIGCOMM Computer Communication Review 27.3 (1997): 67-82.
[3]Lakshman,T. V.,and Upamanyu Madhow."The performance of TCP/IP for networks with high bandwidth-delay products and random loss." Networking, IEEE/ACM Transactions on5.3 (1997):336-350.
[4] Balakrishnan, Hari, et al. "A comparison of mechanisms for improving TCP performance over wireless links." Networking, IEEE/ACM Transactions on 5.6 (1997): 756-769.
[5] Brakmo, Lawrence S., and Larry L. Peterson. "TCP Vegas: End to end congestion avoidance on a global Internet." Selected Areas in Communications, IEEE Journal on 13.8 (1995): 1465-1480.
[6] Mascolo, Saverio, et al. "TCP westwood: Bandwidth estimation for enhanced transport over wireless links." Proceedings of the 7th annual international conference on Mobile computing and networking. ACM, 2001.
[7]Wang, Ren, et al. "Adaptive bandwidth share estimation in TCP Westwood." Global Telecommunications Conference, 2002. GLOBECOM'02. IEEE. Vol. 3. IEEE, 2002.
[8] Luby, Michael, et al. Forward error correction (FEC) building block. RFC 3452,December, 2002.
[9] Ayanoglu, Ender, et al. "AIRMAIL: A link-layer protocol for wireless networks." Wireless Networks 1.1 (1995): 47-60.
[10] Yavatkar, Raj, and Namrata Bhagawat. "Improving end-to-end performance of TCP over mobile internetworks." Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on. IEEE, 1994.
[11] Bakre, Ajay, and B. R. Badrinath. "I-TCP: Indirect TCP for mobile hosts." Distributed Computing Systems, 1995., Proceedings of the 15th International Conference on. IEEE, 1995.
[12] Kopparty, Swastik, et al. "Split TCP for mobile ad hoc networks." Global Telecommunications Conference, 2002. GLOBECOM'02. IEEE. Vol. 1. IEEE, 2002