Asian Journal of Information Technology 5 (11): 1291-1294, 2006

Performance Evaluation of Logarithmic Congestion Control Algorithm in TCP Networks

[1]P. Chitra and [2]M. Rajaram
[1]Department of Computer Science and Engineering, NH:47, Salem Main Road, Valayakaranoor, Komarapalayam Namakkal (Dt) India
[2]AP/EEE, Thanthai Periyar Government Institute of Technology, Vellore

Abstract: This study proposes a new window-based congestion control algorithm for TCP networks. This algorithm is named as TCP-Log. In window-based congestion control schemes, increase rules determine how to probe available bandwidth, whereas decrease rules determine how to back off when losses due to congestion are detected. The proposed algorithm generalizes the AIMD algorithms used for the TCP connections. These algorithms provide additive increase using logarithmic inverse of the current window size and provide multiplicative decrease using logarithmic inverse of the current window size. TCP compatibility of the algorithm is evaluated using the simulation of the proposed model. Simulations are done by ns2, a discrete event simulator. The simulations explore the benefits of using TCP-Log in transport layer. The results are compared with TCP/Tahoe,TCP/Reno TCP/Newreno, TCP/Sack, TCP/Fack. The comparisons show that the proposed algorithm performs better in terms of throughput and bandwidth utilization.

Key words: Congestion control, TCP, AIMD, ns2

INTRODUCTION

During the last decade, computer networks have been growing very massively. Large numbers of computers get connected to both private and public networks. Reliable end-to-end transmission of data is a much needed service for many of today's applications running over the Internet which makes TCP an essential component of today's Internet. TCP is a connection-oriented, end-to-end reliable protocol designed to fit into a layered hierarchy of protocols which support multi-network applications.

Network congestion control remains as a critical issue and give high priority due to the growing size, demand and speed (bandwidth) of the increasing integrated services networks. Designing effective congestion control strategies for these networks is difficult because of the complexity of the structure of the networks, nature of the services supported and the variety of the dynamic parameters involved. However, there is still a fundamental TCP performance bottleneck for one transmission regime: Paths with high bandwidth and long round-trip delays. TCP implements reliable data delivery by measuring the RTT, i.e., the time interval between sending a segment and receiving an acknowledgment for it and retransmitting any segments that are not acknowledged within some small multiple of the average RTT. TCP's end to end congestion control mechanism reacts to packet loss by adjusting the number of outstanding unacknowledged data segments allowed in the network[1-3]. Congestion control is the major requirement for multicast to be set up in the current Internet. Such algorithms are implemented in the protocol, TCP[4-6]. In the existing algorithms, increasing the congestion window linearly with time increases the bandwidth of the TCP connection and when the congestion is detected, the window size is multiplicatively reduced by factor of two[7].

This study proposes a new window-based congestion control for Internet Transport Protocols and applications. TCP compatibility of the algorithm is evaluated using the simulation.

Window based congestion control for tcp networks: The most widely used congestion control mechanism in the current TCP is window-based congestion control mechanism[8]. A TCP connection controls its transmission rate by limiting its number of transmitted-but-yet-to-be-acknowledged segments. Ideally, TCP connections should be allowed to transmit as fast as possible as long as segments are not lost due to congestion. In very broad terms, a TCP connection starts with a small value of *w(slow start)* and then "probes" for the existence of additional unused link bandwidth at the links on its

Corresponding Author: P. Chitra, Department of Computer Science and Engineering, SSM College of Engineering, Komarapalayam, India 638-183

end-to-end path by increasing *w*. A TCP connection continues to increase *w* until a segment loss occurs (as detected by a timeout or duplicate acknowledgement). When such a loss occurs, the TCP connection reduces w to a "safe level" and then begins probing again for unused bandwidth by slowly increasing *w*[12].

The AIMD algorithm my be expressed as given in[6]

$$I: W_{t=R} \leftarrow W_t + (\alpha/W_t);\ \alpha > 0$$
$$D: W_{t=at} \leftarrow W_t - (\beta/W_t);\ 0 < \beta < 1 \quad (1)$$

Where

I → Increase in window as a result of the receipt of one window of acknowledgement in a Round Trip Time (RTT).

D → Decrease in window size on detection of congestion by the sender.

W_t → Window size at time t.

R → Round Trip Time of the flow

α and β → Increase and Decrease Rule constants

Initially, the congestion window is equal to one MSS (Maximum-Sized Segments). TCP sends the first segment into the network and waits for an acknowledgement. If this segment is acknowledged before its timer times out, the sender increases the congestion window by one MSS and sends out two maximum-size segments. If these segments are acknowledged before their timeouts, the sender increases the congestion window by one MSS for each of the acknowledged segments, giving a congestion window of four MSS and sends out four maximum-sized segments[8]. This procedure continues as long as

- the congestion window is below the threshold and
- the acknowledgements arrive before their corresponding timeouts.

Logarithmic congestion control algorithm: The proposed algorithm mainly aims in increasing the window size faster and to gain bandwidth quicker to transfer the large amount of data[9]. Please note that window adjustment policy is only one component of the congestion control protocol derived congestion detection (loss,ECN etc.),retransmissions, estimation of Round trip time etc., remain the same as TCP[10]. The proposed algorithms mainly aim in increasing the window size faster and to gain bandwidth quicker. These rules generalize the class of all congestion control algorithms based on window size adjustment. Now for analysis of the algorithms the model is given below

$$I: W_{t+R} \leftarrow W_t + (\alpha / \log(W_t))^k;\ k = 0.0$$
$$D: W_{t+at} \leftarrow W_t - (\beta / \log(W_t))^l;\ l = 1.0 \quad (2)$$

Analysis of algorithm: The algorithm is implemented and is represented by Eq. 2 and this is called as Logarithmic congestion control algorithm (TCP-Log). The algorithm is implemented to study the variation of window size and the resulting throughput and window size with respect to time. The algorithm begins with slow start state. In this state, the congestion window size is doubled for every window of packets acknowledged. Upon the first congestion indication, the congestion window size is cut in half and session enters in to the logarithmic congestion control state[7,11,12]. In this state the congestion window size is increased by $(\alpha/\log(Wt))^k$ for each new acknowledgement received. Congestion is detected by two events:

- triple-duplicate ACK and
- time-out. If by triple-duplicate ACK, the algorithm reduces the window size by $(\beta/\log(Wt))^l$. If the congestion indication is by time-out, the window size is set to 1. The window size, throughput and total throughput of the algorithm is compared with the standard congestion control algorithms of TCP[1,8].

The topology in Fig. 1 consists of set of sources (n0,n1,n2,n3,n4,n5,n6,n8,n9,n11,n12) and set of sinks (n15,n16,n17,n18,n19,n20,n21). The sources and sinks are connected by four routers namely n7,n10,n13 & n14 as shown in Fig. 1. We have attached TCP agents (TCP,TCP/Reno,TCP/Newreno,TCP/Sack1,

TCP/Fack,TCP/Vegas,TCP-Log) to the source nodes and sinks(TCPSink,TCPSink/DelAck,TCPSink/Sack1, TCPSink) to the receiver nodes. The links are assumed to be bidirectional with a different bandwidth. The link connecting the four routers is assumed with a bandwidth of 0.5Mb, buffer capacity of 20 and a Drop Tail buffer management algorithm. The data packets are generated by an FTP. Sources are connected to the FTP source. The TCP agents implemented with logarithmic increase and decrease rules. These rules are implemented into the TCP agent program tcp.cc. The algorithm is selected by choosing the values for the TCP/Agent variable WindowOption_. If WindowOption_ is 7 then logarithmic algorithm is chosen and if it is other than 7 then the other algorithms are applied. The bandwidth utilization of TCP-log is higher than the other existing algorithms.

The data from various sources are being transmitted to the sinks in the form of packets. When the transmission of packets exceeds the capacity of the link, the packets will be stored in the queue. When there is an overflow in the queue, it will lead to packet losses due to congestion. This problem can be eliminated by using the newly constructed congestion control algorithm. The simulations are done by ns-2 Network Simulator, which is a discrete event simulator used by lot of researchers [13]. The throughput (λ) is modeled by

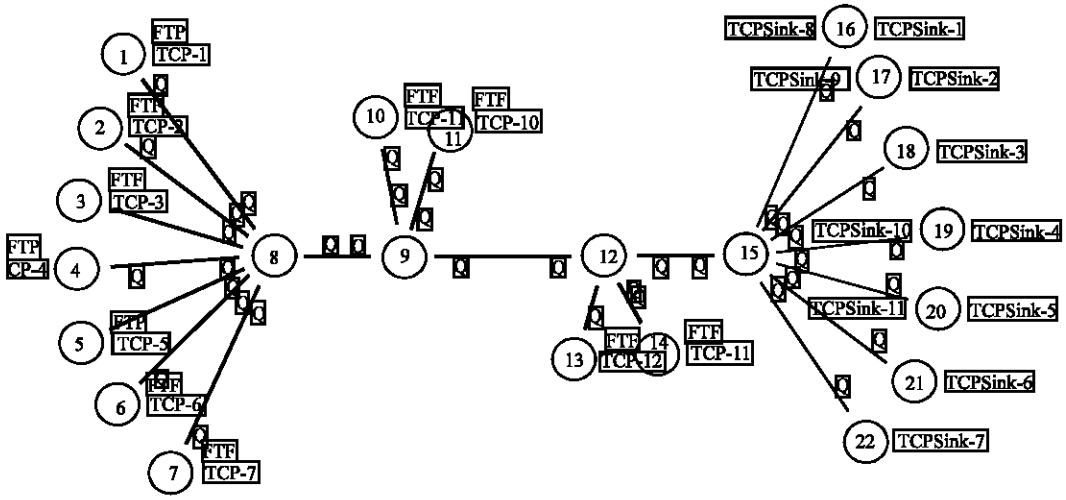

Fig. 1: Topology

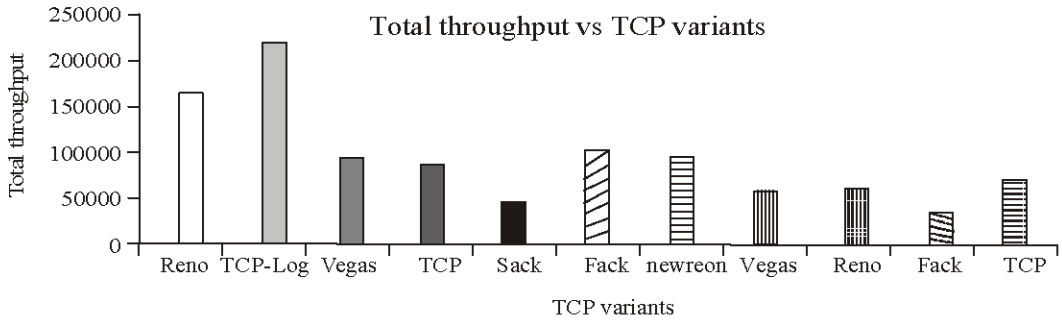

Fig. 2 : Total throughput vs TCP Sources where $\alpha = 1.0$ and $\beta = 0.5$

$$\lambda = \frac{C.S}{R.\sqrt{P}} \quad (3)$$

Where S is the segment size and R is the round trip time, P is the packet loss rate and C is the constant value commonly approximated as $1.5*\sqrt{2/3}$

Total throughput is modeled by

$$\text{Totthroput} = \frac{\text{highest\_ack\_*size\_}}{\text{t\_rtt\_}} \quad (4)$$

Where highest_ack_ is the highest acknowledgement number, size_ is the packet size, t_rtt_ is the total round trip time.

The Fig. 2 shows that the total throughput of TCP-Log is high than the other TCP variants.

CONCLUSION

In this study, the presentation and evaluation is done for a family of window-based congestion control algorithms, called logarithmic congestion control algorithms. The logarithmic variation is considered in the window size for our experimentation. These algorithms generalize the familiar AIMD algorithms. The throughput is shown and total throughput of TCP sources. The simulation result reveals that good performance in band width utilization, throughput and total throughout than the existing algorithms.

REFERENCES

1. Chandrasekaran, M. and R.S.D. Wahidhabanu, 2006. Performance evaluation of polynomial congestion control algorithms mimd-poly and PIPD-Poly in TCP networks, Asian J. Inform. Tech., 5: 346-357.
2. Van, J., 1988. congestion Avoidance and Control ACM Computer Communication Review, Proceedings of the Siggcomm'88 symposium in Stanford, CA, 18: 314-329.
3. Widmer, J., R. Denda and M. Mauve, 2001. A Survey on TCP Friendly congestion control . IEEE Network Magazine.

4. Congestion Control Schemes for, 1991. TCP/IP networks current implementation in BSD unix, Douglas E. Homer, Internetworking with TCP/IP, Englewood cliffs,New jersey, Prentice Hall.
5. Stevens, W.R., 1994. TCP/IP Illustrated Addision-Wesley, Reading MA.
6. Richard, Y. and S.S. Lam, 2000. General AIMD Congestion Control. In: proceedings of ICNP.
7. Sally, F. and K. Fall, 1999. Promoting the use of end-to-end congestion control in the internet.
8. Shundong, J., L. Guo, I. Matta and A. Bestvaros, 2001. A Spectrum of TCP-Friendly Window based Con gestion control algorithms Tech Rep BU-CS-2001-015. computer science Departmernt, boston university.
9. Zheng, W. and J. Crowcroft , 1991. A New Congestion Control Scheme: Slow start and Search (Tri-S), ACM Computer Communication Review, 21: 32-43.
10. Rejaie, R., M. Handley and D. Estrin, 1999. RAP: An end to end Rate based congestion control mechanism for real time streams in the internet in Proc IEEE INFOCOM, 3: 1337-1345.
11. Georgi, K., 2005. A Simulation Analysis of the TCP Control Algorithms. International Conference on Computer Systems and Technologies-Com. Sys. Technology.
12. Kevin, F. and S. floyd, 1996. Simulation-based Comparisons of Tahoe, Reno, SACK TCP. Computer Communication Rerview.
13. ns-2 Network Simulator",http://www.isi.edu/nsnam/ns/,2000.