ORIENTAL JOURNAL OF COMPUTER SCIENCE & TECHNOLOGY
An International Open Free Access, Peer Reviewed Research Journal
www.computerscijournal.org

ISSN: 0974-6471
December 2011,
Vol. 4, No. (2):
Pgs. 447-450

Packet Loss Detection Using Constant Packet Rearranging

ANKUR LAL, SIPI DUBEY and BHARAT PESSWANI

Computer Science and Engineering, RCET Bhilai (India).
*Corresponding author: E-mail: ankur.at.cit@gmail.com

(Received: August 12, 2011; Accepted: September 04, 2011)

ABSTRACT

When we rearrange the packet the most standard implementation of the TCP gives poor performance. In this paper loss of packets in TCP is detected using two diverse methods CPR (Constant Packet Re-arranging) and WCPR (Without Constant Packet Re-arranging). Constant packet rearranging does not depend or rely on the duplicate acknowledgement to detect the packet loss. Instead the timer is used to maintain how long packet is transmitted.

Key words: Packet rearranging, CPR, WCPR.

INTRODUCTION

Here the two methods TCP old and TCP Constant packet rearranging is compared .The main idea behind retransmit packet this is to improve the performance of TCP throughput by avoiding sender to timeout. Using fast retransmit can continuously improve the TCP's performance in the presence of irregular rearranging but it still operates under the assumption of that out-of-order packet which indicate the packet loss and which leads to congestion. As a result its performance degrades in the presence of constant rearranging. This is procedure for rearranging both data and acknowledgment packet. Packet rearranging is generally attributed to transient conditions pathological behavior and erroneous implementation. As per the design format of the TCP's errors and congestion control mechanism which is based on the principle that packet loss is an indication of the network congestion. As per TCP senders backs off transmission rate by decreasing its congestion control windows. TCP uses two strategies for the detection of the packet loss the first one is based on the sender's retransmission timeout which is also referred as coarse timeout. When the senders timeout which is responded by the congestion control by slow start which leads into decreasing congestion window to one segment. The packet detection loss is detected at the receiver side by using the sequence number. In this case receiver checks the sequence number of received packet. The hole in the sequence indicates that there is loss of the packet in such case the receiver generates the duplicate acknowledgement for every "out-of-order" segment it receives. Until the lost packet received, the entire reaming packet with higher sequence number is consider as out of order and will cause to creation of duplicates packets. After that sender retransmit the lost packet without waiting for timeout which helps to reduction of congestion windows.

Packet rearranging

This paper presents a methodology for simulating and measuring TCP Rearranging, providing an insight into the behaviors of the congestion and retransmission algorithms, and demonstrating that Rearranging has a measurable effect on performance. These measurements illustrate that there is a maximum Rearranging delay threshold that should be applied to packets, regardless of percentage rearranging, below which Rearranging has negligible effects. Determination of this threshold, on a specific path, is key to ensuring that a specific switch or router does not introduce Rearranging to such an extent that it causes unnecessary retransmissions and an associated reduction in throughput. figure 1.1.1 show a graph between Rearranging and packet rate is drawn and as shown Packet Rearranging does increase as packet rates increases.

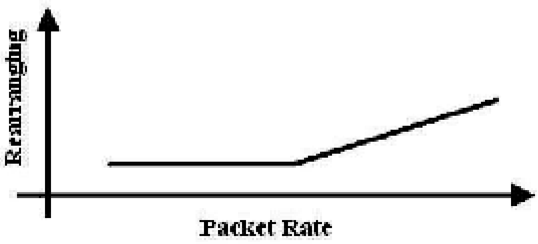

Fig. 1: Shows graph between Rearranging and packet rate

Existing work

TCP uses two strategies for detecting packet loss. The first one is based on the sender's retransmission timeout expiring and is sometimes referred to as coarse timeout. When the sender times out, congestion control responds by causing the sender to enter slow-start, drastically decreasing its congestion window to one segment. The other loss detection mechanism originates at the receiver and uses TCP's sequence number. Essentially, the receiver observes the sequence numbers of packets it receives; a "hole" in the sequence is considered indicative of a packet loss. Specifically, the receiver generates a "duplicate acknowledgment" for every "out-of-order" segment it receives. Note that until the lost packet is received, all other packets with higher sequence number are considered "out-of-order" and will cause duplicate acknowledgment to be generated. Modern TCP implementations adopt the *fast retransmit* algorithm which infers that a packet has been lost after the sender receives a few duplicate acknowledgments. The sender then retransmits the lost packet without waiting for a timeout and reduces its congestion window in half. The basic idea behind fast retransmit is to improve TCP's throughput by avoiding the sender to timeout

Limitations

- ´ TCP detects packet loss through duplicate Acknowledgement.
- ´ It performs poorly when packets are reordered.
- ´ Its Throughput decreases whenever packet is reordered.
- ´ Not easier to deploy.
- ´ Decreased robustness

Proposed system

The basic idea behind TCP constant packet rearranging is to detect packet losses through the use of timers instead of duplicate acknowledgments. This is prompted by the observation that, under constant packet rearranging, duplicate acknowledgments are a poor indication of packet losses. Because TCP constant packet rearranging relies solely on timers to detect packet loss, it is also robust to acknowledgment losses as the algorithm does not distinguish between data (on the forward path) or acknowledgment (on the reverse path) losses.

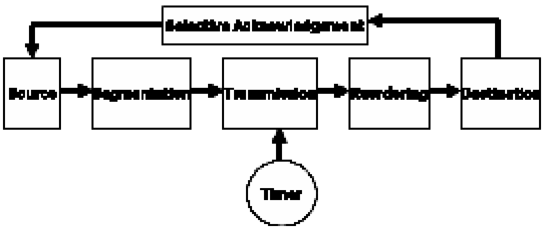

Fig. 4(a): Packet rearranging

Advantages of proposed system

- ´ Proposed system works perfectly when packet is reordered.
- ´ It uses Timer Control to detect the packet Loss.
- ´ Its performance will be same even the packet is reordered.
- ´ Proposed system not only depends on the

duplicate acknowledgements and packet rearranging to detect the packet losses.

- This system performs consistently better than existing mechanisms that try to make TCP more robust to packet rearranging.
- Easier to deploy since no changes are required at the sender side.

Problem formulation modules

Transmission without Rearranging

If we transmit a message without packet rearranging, then If part of a message is lost during the transmission then we need to retransmit the entire message or we need to retransmit from that particular part. Therefore, upon detecting loss, the TCP sender backs off its transmission rate by decreasing its congestion window.

Transmission with Packet Rearranging

If we transmit a message as packets then we need to retransmit only the packet which is lost and not the entire message. The message is send from the source to the ingress router and then to the intermediate routers and then to the outgress router and the destination. The basic idea behind TCP constant packet rearranging is to detect packet losses through the use of timers instead of duplicate acknowledgments. This is prompted by the observation that, under constant packet rearranging, duplicate acknowledgments are a poor indication of packet losses. Because TCP constant packet rearranging relies solely on timers to detect packet loss, it is also robust to acknowledgment losses as the algorithm does not distinguish between data (on the forward path) or acknowledgment (on the reverse path) losses.

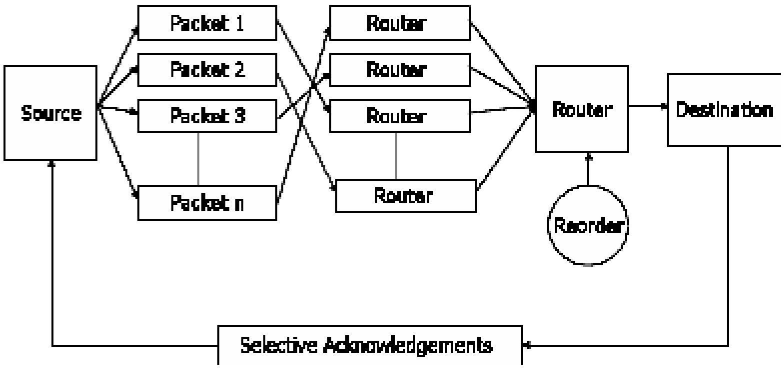

Fig. 4(b): Packet rearranging

Segmentation

Segmentation is the process of dividing the source code into small number of packets and transmitting the packets through the routers. We define certain limits for the size of the packets. The header information includes source machine name, destination machine name, position of the packet and the related information.

Timer control

Whenever each and individual packet starts sending a timer is started. The system current time is taken as a start time and added with delay and it acts as a threshold time and if the threshold time exceeds the maximum elapsed time of the packet then the packet is retransmitted. If the time doesn't exceed then the packet may arrive safe. If so the next packet is transmitted else the current packet is transmitted until it arrives safely. Thread concept is used to implement the timer.

RESULTS

The performance of the transmission control protocol with packet rearranging is tested on Windows XP. Comparison chart compares the throughput of TCP without Packet Rearranging with TCP With packet Rearranging. The performance is shown by comparing the Transmission rate of existing system with proposed system.

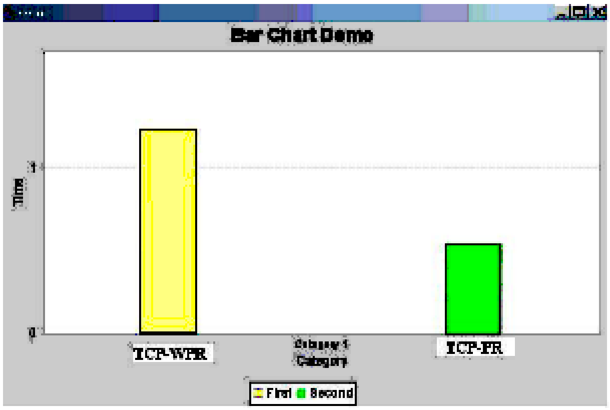

Fig. 5.2: Comparison chart of TCP without packet rearranging and with packet rearranging

CONCLUSION

In this paper we proposed and evaluated the performance of TCP constant packet rearranging, a variant of TCP that is speciûcally designed to handle constant rearranging of packets (both data and acknowledgment packets). Our simulation results show that TCP constant packet rearranging is able to achieve high throughput when packets are reordered and yet is fair to standard TCP implementations, exhibiting similar performance when packets are delivered in order. Such mechanisms include proposed enhancements to the original Internet architecture such as multi-path routing for increased throughput, load balancing, and security; protocols that provide differentiated services; and traffic engineering approaches. As shown from result tat the delay time for Linux experiment is much more then of Solaris

REFERENCES

1. Luo X. and Chang R. K. C., "Novel Approaches to End-to-end Packet Rearranging Measurement" Proc. ACM/ USENIX Conf. Internet Measurement, (2005).
2. Bare, A. A., "Measurement and Analysis of Packet Rearranging," Masters Thesis, Dep. Computer Science, Colorado State University, (2004).
3. Colin M. Arthur, Andrew Lehane, David Harle, "Keeping Order: Determining the Effect of TCP Packet Rearranging," icns, pp.116, International Conference on Networking and Services (ICNS '07) (2007).
4. Sharad Jaiswal, G. Iannaccone, C. Diot, J. Kuorose, and D. Towsley. Measuring and classiûcation of out-of-sequence packets in a Tier-1 IP Backbone, International Measurement Workshop(IMW) (2003).
5. Piratla, N. M. , Jayasumana A. P. ,and Bare A. A., "A Comparative Analysis of Packet Rearranging Metrics," Proc. IEEE/ACM 1st Int. Conf. Communication System Software and Middleware (COMSWARE 2006), New Delhi, (2006).
6. Jaiswal, S., Iannaccone, G., Diot, C., Kurose, J. and Towsley, D., "Measurement and Classification of Out-of-sequence Packets in Tier-1 IP Backbone," Proc. IEEE Infocom, 1199- 1209 (2003).
7. R. Teixeira, K. Marzullo, S. Savage, and G. M. Voelker, "Characterizing and measuring path diversity of Internet topologies," presented at the ACM Sigmetrics, San Diego, CA, (2003).
8. S. Bohacek, "A stochastic model of TCP and fair video transmission," in *Proc. IEEE INFOCOM*, 1134–1144 (2003).
9. M. Franklin, T. Wolf. A Network Processor Performance and Design Model with Benchmark Parameterization. First Workshop on Network Processors, Cambridge, MA (2002).