Impact of Round Trip Delay Self-Similarity on TCP Performance

Tatsuya Hagiwara, [†]Hiroshi Majima, Takahiro Matsuda, and Miki Yamamoto

Graduate School of Engineering, Osaka University 2–1, Yamadaoka, Suita-shi

Osaka 565-0871, Japan

[†]Faculty of Engineering, Osaka University
2–1, Yamadaoka, Suita-shi Osaka 565-0871, Japan

E-mail: {hagiwara,matsuda,yamamoto}@comm.eng.osaka-u.ac.jp

Abstract

Recent measurement showed that self-similar nature is found not only in network traffic volume but also round trip packet delay. In this paper, we discuss three issues of the self-similarity of round trip time (RTT), which is one of the most important parameters to determine TCP throughput performance. First, we discuss the origin of the packet delay self-similarity. A recent study anticipated that the queueing delay of self-similar traffic is the reason for packet delay self-similarity. With computer simulation, we evaluate the correlation between traffic and RTT self-similarity. Next, we investigate the impact of RTT self-similarity on TCP throughput performance. Computer simulation results show that RTT self-similarity gives high variability to file transfer time. Finally, we investigate the impact of RTT self-similarity on RTO (Retransmission Time Out). We discover that the bigger the Hurst parameter of RTT is, the more frequent unnecessary timeouts occurs. Furthermore, we propose a new RTO calculation algorithm to improve these unnecessary timeouts.

1 Introduction

Since empirical studies showed the self-similar nature of network traffic[1], numerous studies on the influence of self-similarity on network performance have been made. Some papers reported important properties about the selfsimilar traffic[2, 3]. For example, the self-similar traffic degrades network performance [4, 5] such as buffer overflow probability. As a result, the network utilization decreases remarkably due to retransmitting the overflowed packets.

Recent studies have shown that round trip delay in the Internet also has self-similar nature[6, 7]. Packet delay characteristic is an important network parameter especially in feedback-control based network protocol such as ACK (Acknowledgement) or NACK (Negative ACK)based flow control. However, there has been published no evaluation of impact of packet delay self-similarity on network performance. In this paper, we study the impact of packet delay selfsimilarity on TCP. It is important to evaluate the impact because TCP is the most dominant transport protocol in the current Internet, and its performance depends on the round trip packet delay, which is referred as RTT (Round Trip Time) in the case of TCP.

We discuss the following three issues of RTT self-similarity. First, we discuss the origin of the packet delay self-similarity. A recent study anticipated that the queueing delay of selfsimilar traffic is the reason for packet delay self-similarity[8]. With computer simulation, we evaluate the correlation between traffic and packet delay self-similarity in order to confirm the anticipation.

TCP has two mechanisms to determine its throughput performance; congestion avoidance[9] and slow-start mechanisms. Next, we evaluate the impact of RTT selfsimilarity on TCP throughput performance when retransmission timeout does not occur, that is, TCP throughput is determined by congestion avoidance phase. Finally, in order to evaluate the impact on slow-start mechanism, we investigate the effect of self-similarity on RTO estimation algorithm. We discover that the higher the Hurst parameter of RTT is, the more frequent unnecessary timeouts occur. If unnecessary timeout occurs continuously, the network utilization will decrease remarkably due to TCP rate control algorithm. Furthermore, we propose a new RTO calculation algorithm to improve these unnecessary timeouts. It was shown that our improved RTO estimation algorithm performs significantly better.

The rest of the paper is organized as follows. In section 2, we briefly introduce the the packet delay self-similarity. Section 3 describes the mechanism of TCP and its RTO estimation algorithm. In section 4, we evaluate the correlation between traffic and packet delay self-similarity. In section 5, we evaluate the influence of RTT self-similarity on TCP performance from two viewpoints, throughput performance and RTO estimation algorithm. Finally, in section 6, we summarize this paper.

2 Packet Delay Self-Similarity

In recent years, empirical studies [6, 7, 8] on packet delay characteristics showed that packet delay characteristics have self-similar characteristics.

In the literature [6], 12 traces of round trip Internet packet delay (UDP packets) was analyzed and it was found that these traces, when viewed as time series data, often exhibit Hurst parameter estimates greater than 0.5, indicating long-range dependence. The paper indicates that Internet delay is bursty across multiple time scales, which implies that end-user quality of services in the Internet is likely to be impacted by long period of very large and/or highly variable delays.

The literature [7] is the extended version of the literature [6]. The paper implies the long-range dependency of a round-trip delay increases with network load and seems to be positively correlated with the packet loss for that path. These results have implications for the design of real-time, interactive multimedia applications. When Hurst parameter is high, the combination of round-trip delay and loss rate may be too extreme for delay- and loss-sensitive applications, such as Internet telephony and video-conferencing.

In the literature [8], round trip time of NTP(TCP) packet traces were analyzed and it was found that RTT has self-similar nature. The Hurst parameter on NTP packet delay was between 0.6 and 0.8. Also the literature [8] indicates that the queueing delay of self-similar traffic is the reason for self-similar nature of packet delay. The main findings of literature [8] are that: (i) The packet round-trip delay processes of the five routes in Internet are all LRD processes; the intensity of LRD measured in terms of Hurst parameter H is different for routes. (ii) The measured packet round-trip delay process is only a sampled process with certain sampling interval. (iii) The probability distribution P (round-trip delay > x) decays with x more slowly than exactly exponential rate. Based on the findings, they think that the LRD in a packet round-trip delay process is caused by the LRD in the arrival packet traffic in Internet.

3 Transmission Control Protocol

We study the Reno version of TCP, which is the most popular TCP in the current Internet. TCP generally uses window flow control and TCP-Reno congestion control is based on the additive increase and multiplicative decrease mechanism containing Fast Retransmit and Fast Recovery algorithms[10].

TCP employs an ACK(Acknowledgement)-based window control, that is, a TCP sender updates its congestion window size every RTT. Therefore, TCP throughput performance depends on the RTT. It is well known that the theoretical throughput of TCP connection is given by the following expressions[11].

$$B(p) = \frac{1}{E[RTT]} \sqrt{\frac{3}{2bp}} \tag{1}$$

where E[RTT] is average of RTT, p is packet loss probability and b is the number of packets that are acknowledged by a received ACK.

TCP provides reliable data transfer to higher-layer applications. In order to do this, TCP must detect segment losses and retransmit them. TCP-Reno has two methods to detect the losses: Fast Retransmit-based and RTObased methods. With RTO-based method, which is the most basic mechanism to detect the losses, the sender retransmit a segment if its ACK has not been received in the expected amount of time (this is RTO based on the RTT). On the other hand, with Fast Retransmit, after receiving three duplicate ACKs for the same TCP segment, the sender infers that a packet has been lost and retransmits it without waiting for a retransmission timer to expire, leading to higher channel utilization and connection throughput. If segment losses are detected by RTO, TCP congestion window, which specifies the amount of data the TCP sender can transmit before receiving an ACK, is reduced to 1 segment, that is, RTO based method effectively reduces TCP's throughput. Therefore, the goal of a good RTO estimator should be to minimize the number of unnecessary retransmissions due to RTO. If RTT drastically changes, RTO may not be able to adapt to the RTT and the TCP sender may end up retransmitting segments.

In Jacobson's algorithm[12], RTO is calculated by

$$RTO = SRTT + kRTTVAR$$
(2)

$$RTTVAR = RTTVAR$$

$$+ h \times (|SRTT - RTTmeas| - D)$$
(3)

with k = 4, h = 0.25 in many cases, where SRTT is a smoothed estimate of RTT and RTTVAR is a smoothed estimate of the variance of RTT. Both variables are updated every time an RTT measurement RTT_{meas} is taken. SRTT is updated using an EWMA (exponentially-weighted moving average) with a gain of α

$$SRTT = (1 - \alpha)SRTT + \alpha RTT_{meas} \tag{4}$$

and the literature [12] recommends $\alpha = \frac{1}{8}$, which leads to efficient implementation using fixed-point arithmetic and bit shifting.

The widely-used BSD RTO implementation has several possible limitations as follows; (i) the measurements are made using a clock with a 500 msec granularity; (ii) the adaptive RTT and RTT variation estimators are updated



Figure 1: Simulation Environment based on NS

with new measurements only once per round-trip, so they adapt fairly slowly to change in network conditions; (iii) the resulting RTO estimate has a large minimum value of 1 sec, which may make it inherently conservative. But, with the advent of higher precision clocks and the TCP "timestamp" option, all three of these limitations might be removed. So, we adopt "timestamp" option for our evaluation. As described before, measuring RTT and estimating RTO gives large impact on TCP performance. So, there are many studies concerning measuring RTT and estimating RTO. The literature [13] deals with measuring RTT and estimating RTO from some different viewpoints. The literature [13] showed that they analyzed the impact of varying a number of estimator parameters, finding that the one with the greatest effect is the lower bound placed on RTO, followed by the clock granularity, while other parameters have little effect.

4 Correlation between Traffic Volume and RTT Self-Similarities

A packet round-trip delay is the sum of delays on each subnet link traversed by the packet. A packet delay in the Internet consists of four components (processing delay, queueing delay, transmission delay and propagation delay). Being fixed the length and routing of a packet, the packet delay only changes with the queue delays on the route. In the Internet, a packet route generally includes a tandem of many queues on routers, so the packet round trip delay process is mainly determined by the queueing performances along the route.

Let express packet round trip delay T(t) the form

$$T(t) = T_0 + \sum_{t=1}^{N} Q_i(t)$$
(5)

where T_0 is constant part of RTT, N is the total number of queues along its route and $Q_i(t)$ is the delay of the packet experiencing in queue *i*. Recent study[8] anticipated that $Q_i(t)$ behaves as a LRD (Long Range Dependence) processes when the arrival packet traffic is a LRD processes, because the complementary distribution of a queue length decays more slowly than strictly exponential rate. Put another way, we expect that there is a correlation between the traffic volume and RTT self-similarities. With computer simulation, we evaluate the correlation in order to confirm the anticipation. We employ ns2[14] simulator and set up the simulation environment as illustrated in Fig.1, where TCP connections share one bottleneck link. We make the following assumptions,

- All connection are TCP connections.
- The number of connection is n = 32.
- Node i (i = 2,...,n) sends packets according to Pareto file size distribution. Aggregated Traffic based on Pareto file size distribution behaves as a selfsimilar process[2].
- Node 1 always has data to send (i.e., greedy model).
- Packet size is 1000bytes.
- R is drop-tail router.
- Hurst parameters of input traffic are High(H = 0.89), Middle(H = 0.79) and Low(H = 0.68).

Various techniques for estimating Hurst parameter in a time series are available[15] such as Variance Time Plot(VTP), R/S analysis, Periodgram. We employ VTP method, which is a simple but effective estimator[1].

Table 1 shows long-range dependency of RTT evaluated by VTP. Here, H_{Traffic} and H_{RTT} represent the Hurst parameter of traffic volume and RTT, respectively. We can see that H_{RTT} increases as $H_{Traffic}$ increases. That is, there is some correlation between traffic volume and RTT self-similarities. Self-similarity of traffic volume is one of the factors, which leads RTT self-similarity.

5 Influence of RTT Self-Similarity on TCP

5.1 Throughput Performance

In this subsection, we evaluate TCP throughput performance in order to investigate the influence of RTT selfsimilarity on TCP throughput performance. We set up the simulation environment as illustrated in Fig.2. We assume that RTT is given by FGN(Fractional Gaussian

Table 1: Long range dependency of RTT

$H_{Traffic}$	0.68	0.79	0.89
H_{RTT}	0.56	0.72	0.83



Figure 3: TCP throughput traces

Noise)[16] with mean and variance of 0.05 sec and 0.0015 sec^2 , respectively. FGN can provide a self-similar process according to Hurst parameter. We also assume that a packet is lost randomly with probability of 0.01 and RTO expiration does not occur. Figure 3 shows TCP traffic traces, where we plot throughput against times where time granularity is 0.1, 1, and 10 sec. In Fig.3, the vertical axis shows average throughput per each observing time scale.

In the case of $H_{RTT}=0.5$, the coarser the time scale is, the flatter the throughput dynamics is. On the other hand, in the case of $H_{RTT} = 0.9$, the high variability is kept even if timescale is 10sec. Next, to see the degree of variability of the throughput, the probability density function of throughput is displayed on Fig.4.

In Fig.4, the horizontal axis shows random variable x as throughput, the vertical axis shows probability density function f(x). It is confirmed that, in the large time scale, the distribution of throughput is wide when H of RTT is large. From the above results, we can expect that the long-range dependence of RTT influences the file forwarding time because the coarse time scale corresponds to the large



Figure 4: pdf of TCP throughput

Table 2: Time for completing file forwarding

H_{RTT}	0.5	0.7	0.9
Max(s)	9.5	11.3	13.2
Min(s)	6.5	5.7	4.3
Standard Deviation	0.51	0.98	1.99

file transfer. Table 5.1 shows the transfer time of a file with file size of 1MB. RTT self-similarity results in high variability of file transfer time. The difference between maximal and minimal transfer time is 9sec in the case of $H_{RTT} = 0.9$, while it is 3sec in the case of $H_{RTT} =$ 0.5. This is not a desirable result from a viewpoint of fairness among TCP connections, because, a file transfer time depends heavily on start timing of each connection.

5.2 Unnecessary Timeout

In this subsection we evaluate RTO estimation with different Hurst parameters of RTT in order to investigate unnecessary timeouts, and we propose a new RTO estimation algorithm to improve its unnecessary timeouts.

We use the same simulator as described in section 5.1. We use two kinds of RTT, which have different Hurst parameter. We assume k = 4, and $\alpha = 1/8$ in Eqns.2 and 4. Let RTO series with $H_{RTT} = 0.5$, and $H_{RTT} = 0.9$ be $RTO_{H=0.5}$ and $RTO_{H=0.9}$, respectively. From those re-



Figure 5: Timeout Index



Figure 6: Average Ratio RTO/RTT_{measured}



sults, we can see $RTO_{H=0.9}$ is more bursty than $RTO_{H=0.5}$. That is, RTO may keep small value continuously in the cause of $RTO_{H=0.9}$. In this case, when RTT changes drastically, RTT exceeds RTO. That is, unnecessary timeouts occur frequently when Hurst parameter of RTT is high.

Next, we introduced the following evaluation index to investigate unnecessary timeout.

$$Timeout \ index = \frac{estimated \ RTO}{measured \ RTT} \tag{6}$$

It means that a timeout did not occur, when *Timeout* index is larger than 1 and a packet is not lost. However, when *Timeout* index is extremely larger than 1, in the case that a packet is lost, the sender must wait long time until timer expiration. On the other hand, when *Timeout* index is smaller than 1, it means that the unnecessary timeout occurred. Figures 5 and 6 show the number of unnecessary timeout normalized by the number of unnecessary timeout at $H_{RTT} = 0.9$ and k = 4 and average *Timeout* index vs k, respectively.

In Fig.5, many unnecessary timeouts occur when Hurst parameter of RTT is 0.9 in the case of k = 4, which is commonly used in BSD implementation. We can see that the number of timeout occurring at the $H_{RTT} = 0.9$ is 10 or more times large as compared with that at the $H_{RTT} =$ 0.5. This reason is as follows. RTTVARS corresponds to variance of RTT. Usually, LRD process is evaluated by ensemble average and variance[1]. But, in the case of TCP, average and variance of RTT is local average and variance. Because time correlation is high when a series of RTT is bursty (H=0.9), RTTVAR tends to become small. That is, the setup of k = 4 is not good when Hurst parameter of RTT is large.

From Fig.5, we can see that unnecessary timeouts decrease as k increases. On the other hand, from Fig.6, we can see that as k increases, TCP senders must wait for long time to detect timeout, that is, utilization of network resource is reduced. In other words, there is an adequate value of k to realize the reduction of unnecessary timeouts and effective resource utilization. Therefore, we propose a new RTO estimation algorithm to set the value of k according to the Hurst parameter of RTT. That is, we set k = 4 when $H_{RTT} = 0.5$ and k = 7 when $H_{RTT} = 0.9$. In this case, when $H_{RTT} = 0.9$, we can see that the number of unnecessary timeouts and an average time index is almost the same as $H_{RTT} = 0.9$.

In order to use this method, a practical Hurst parameter estimator is required. It is difficult to employ commonly used estimators such VTP, because these methods require large amount of computational costs. However, tendency of long range dependence can be estimated with RTTVAR as follows. We calculate RTTVAR with different two values of α . Since RTTVAR and α correspond to the variance of RTT and granularity of smoothed RTT, RTTVAR is sensitive to α when H_{RTT} is large while it is not sensitive when H_{RTT} is small. Therefore, we can estimate the tendency by:

$$RTTVAR_{\alpha_1}/RTTVAR_{\alpha_2} \tag{7}$$

where $\alpha_1 > \alpha_2$. Figure 7 shows Eqn.7. According to Eqn.7, we can reduce long range dependency at H_{RTT} by changing k = 4 to k = 7.

6 Conclusions

In this paper, first, we discussed the origin of the packet delay self-similarity. With computer simulation, we found that there is some correlation between traffic volume and RTT self-similarities. Therefore, self-similarity of traffic volume is one of the factors, to lead the RTT selfsimilarity. Second, we evaluated the influence of RTT self-similarity on TCP throughput performance. We confirmed that the long-range dependence of RTT makes the variability of file forwarding time high. That is, the higher the Hurst parameter of RTT is, the bigger the difference between maximal and minimal transfer time is. This is because, when the Hurst parameter is large, the variability is not averaged even in the case of forwarding a large file, while it is averaged when the Hurst parameter is small. It implies that it will be a severe fairness problem among TCP connections in terms of steady communication. Therefore, we will plan to develop a network control method considering RTT self-similarity to support steady communication. Finally, we investigate the impact of RTT self-similarity on RTO. We discovered that the higher the Hurst parameter of RTT is, the more frequent unnecessary timeouts occurs. If unnecessary timeout occurs continuously, the network utilization will decrease remarkably due to TCP rate control algorithm. Furthermore, we proposed a new RTO calculation algorithm to improve these unnecessary timeouts. It was shown that our improved RTO estimation algorithm perform significantly better. In the future, we will evaluate the variability of file forwarding time among various file size packets. Next, we will implement our RTO estimation algorithm and evaluate its performance.

References

 E.Leland, Murrad S.Taqque, W.Willinger, and Daniel V.Willson, "On the self-similar Nature of Ethernet Traffic(Extended Version)," IEEE/ACM Trans. on Networking, Vol.2, No.1, pp.1-15, Feb. 1994.

- [2] V.Paxson, S.Floyd, "Wide Area Traffic: The Failure of Poisson Modeling," IEEE/ACM Trans. on Networking, Vol.3, No.3, pp.226-244, June 1995.
- [3] H.Doi, T.Hagiwara, H.Tode, and H.Ikeda, "Properties of the Multiplexed self-similar Traffic on ATM Network," IEEE APCC/ICCS'98, pp.444-448, Nov. 1998.
- [4] Z.Sahinoglu and S.Tekinay, "On Multimedia Networks: Self-Similar Traffic and Network Performance,"IEEE Commun. Mag., Vol.37, No.1, pp.48-52, Jan. 1999.
- [5] K.Park, G.Kim, and M.Crovella, "On the Effect of Traffic Self-Similarity on Network Performance," 1997 SPIE International Conference on Performance and Control of Network Systems, Nov. 1997.
- [6] M. S. Borella, I. Sidhu, "Self-Similarity of Internet Packet Delay" IEEE ICN'97, pp513-517, Jan. 1997.
- [7] M. S. Borella, G. B. Brewster, "Measurement and Analysis of Long-Range Dependent Behavior of Internet Packet Delay", IEEE/INFOCOM'98,pp497-504, Feb.1998.
- [8] Q. Li, and D. L. Mills, "On the Long-Range Dependence of Packet Round Trip Delays in Internet," IEEE ICC'98, pp. 1185-1191, 1998
- [9] Mark Allman, Vern Paxson, and W. Richard Stevens, "TCP Congestion Control", Apr. 1999, RFC2581.
- [10] W. R. Stevens, TCP/IP Illustrated Volume 1, Addison Wesley,
- [11] J. Padhye, V. Firoiu, D. Towsley, J. Kurose : "Modeling TCP Throughput: A Simple Model and Empirical Validation," ACM SIGCOMM '98.
- [12] Van Jacobson, "Congestion Avoidance and Control", ACM SIGCOMM, Computer Communication Revies, 1988.
- [13] M.Allman, and V.Paxson, "On Estimating End-to-End Network Path Properties," ACM SIGCOMM, Computer Communication Review, pp263-274, Oct. 1999.
- [14] "The Network Simulator ns2 documentation", http://www.isi.edu/nsnam/ns/ns-documentation.html
- [15] M.S.Taqqu, V.Teverovsky, and W.Willinger, "Estimation for long-range dependence: an empirical study.", Fractals3(4):785-788,1995.
- [16] S. Ledesma, D.Liu, "Synthesis of Fractional Gaussian Noise Using Linear Approximation for Generating Self-Similar Network Traffic", ACM SIGCOMM, Computer Communication Review, 1999.