Influences of TCP Congestion Control Mechanisms to Multi-Fractal Nature of Generated Traffic

¹Hiroki Doi, ²Takahiro Matsuda, and ²Miki Yamamoto,

¹Central Research Institute of Electric Power Industry
Communication & Information Research Laboratory
2-11-1, Iwado kita, Komae-shi, Tokyo 201-8511 JAPAN.

²Graduate School of Engineering, Osaka University,
2-1 Yamada-oka, Suita, Osaka, 565-0871, JAPAN.

E-mail:doi@criepi.denken.or.jp

Abstract—TCP traffic has been reported to have not only self-similarity but multi-fractal nature. Multi-fractal nature of traffic volume, in general, is well known to have significant influence to queueing behavior in short time scale. In the paper, we investigate about "what component of TCP congestion control mechanism affects to multi-fractal nature of generated traffic?". We artificially divide TCP congestion control mechanism to several components and evaluate multi-fractal nature of generated traffic based on multi-fractal scaling analysis. Our simulation results show that Retransmission Time Out mechanism plays an important role for multi-fractal nature of TCP traffic.

I. INTRODUCTION

Recent studies have shown that the Internet traffic has not only self-similarity[1] but also multi-fractal nature[2]. Self-similarity of network traffic represents Long Range Dependence (LRD), which means a correlation property over time scales on the order of a few hundreds of milliseconds and larger. LRD has significant influence on queueing behavior. Generally, LRD is characterized by only one parameter i.e. Hurst parameter H where $0.5 < H < 1$ and higher H means correlation property over a longer time period. With the same arrival rate, increasing H causes increasing an average queue length.

On the other hand, multi-fractal has more complex scaling property than self-similarity. Multi-fractal nature represents spikiness of traffic and a correlation property of traffic in short time scales. Therefore, traffic has different behavior from a multi-fractal point of view while having the same Hurst parameter.

Multi-Fractal has a significant influence on queueing behavior as well as LRD. A queueing analysis demonstrated in [3] has shown that in the case of lower link utilization, multi-fractal affects on the queueing behavior

while LRD affects in the case of higher link utilization. This indicates that at intermediate link utilization, we should carefully design networks with taking account of both of multi-fractal and LRD.

TCP traffic has been reported to exhibit multi-fractal nature for a specific time scale of the order of the round trip time(RTT)[2][4]. This means TCP congestion control mechanism plays an important role for generating multi-fractal nature because ACK clocking mechanism is adapted to TCP congestion control[5]. However, it is still unclear about what part of TCP congestion control mostly affects to multi-fractal nature, because TCP congestion control has complicated mechanism including Fast Retransmission, Retransmission Time Out[5]. Additionally, lots of improvements of TCP congestion control have been proposed, e.g. Selective Acknowledgment[6].

In the paper, we focus on TCP congestion control from the viewpoint of multi-fractal nature of generated traffic, especially on a cause of multi-fractal nature. As far as the authors know, there is no literature discussing what part of complicated TCP congestion control mechanism plays an important role for generating multi-fractal nature. TCP has experienced several improvements. In each improvement, a new component of congestion control mechanism is added to TCP. For example, old style TCP has Retransmission Time Out mechanism. TCP Tahoe is added a Fast Retransmit mechanism to improve throughput degradation due to redundant waiting time for retransmission. TCP Reno which is widely used in the current Internet is added a Fast Recovery mechanism to improve window recovery under moderate congestion. To evaluate influences of each component to multi-fractal nature, we evaluate TCP without Fast Retransmission and Fast Recovery(called default TCP), TCP Tahoe only with Fast Retransmission and TCP Reno with both of them. Our simulation results show the following inter-

Fig. 1. Generated traffic patterns by MWM (Top : $p=2$, Middle : $p=20$, Bottom : $p=100$)

esting results.

- Multi-fractal nature is mainly caused by Retransmission Time Out mechanism.
- Fast Recovery and Fast Retransmit mechanisms improve (reduce) multi-fractal nature of generated traffic.

There have been many researches to improve TCP performance based on the fact that Retransmission Time Out degrades throughput performance of TCP. Furthermore, it is well-known that LRD is also caused by Retransmission Time Out mechanism[8]. Our results show that avoiding occurrence of Retransmission Time Out is the best approach to improve TCP performance, i.e., throughput degradation, LRD, and multi-fractal nature.

This paper is organized as follows. In section II, we explain the definition of multi-fractal scaling and discuss its influence on queueing performance. In section III, we explain the TCP congestion control mechanism and difference of the three versions of TCP. In section IV, we evaluate multi-fractal nature of these TCPs and clarify what component of the TCP congestion control affects the multi-fractal nature. In V, we conclude the paper.

II. MULTI-FRACTAL NATURE OF NETWORK TRAFFIC

A. Definition of Multi-Fractal Scaling

Let us consider a traffic arrival process $A(0, t)$ during the interval time $(0, t)$, and its associated increment process $X_{\Delta}(i)$ duration Δt , from $i\Delta t$ to $(i-1)\Delta t$, defined by

Fig. 2. $f(\alpha)$ for $p = 2, 20$ (we did not evaluate $f(\alpha)$ in the case of $p = 100$. In this case, MWM behaves as a almost pure mono-fractal traffic)

$$X_{\Delta t}(i) = A(0, i\Delta t) - A(0, (i-1)\Delta t). \quad (1)$$

The scaling hypothesis is that the q -th ($q = \dots, -2, -1, 0, 1, 2, \dots$) moment of the increment process behaves as

$$\sum_i X_{\Delta t}(i)^q \sim C(q)\Delta t^{-\tau(q)} \text{ for } \Delta t \rightarrow 0, \quad (2)$$

where $\tau(q)$ is called structure function. In general, the structure function $\tau(q)$ will be decreasing and nonlinear in terms of q . When $\tau(q)$ is a linear function of q , the scaling behavior is said to be mono-fractal (self-similar). Multi-fractal scaling can be described in terms of an α -spectrum $f(\alpha)$. $f(\alpha)$ is defined by

$$f(\alpha) = \inf_q [\alpha q + \tau(q)]. \quad (3)$$

Multi-fractal spectrum $f(\alpha)$ measures the burstiness of traffic through the scaling of its higher-order moments.

When $f(\alpha)$ has a wide spectrum, traffic contains the bursty or spikiness behavior at small scales. On the other hand, when $f(\alpha)$ spectrum has a narrow spectrum, traffic behavior closes to the mono-fractal.

Figure 1 shows traffic traces generated by MWM(Multi-fractal Wavelet Model)[9] which can generate a traffic pattern with multi-fractal scaling. p represents a shaping parameter of MWM and from the figure, we can see that smaller p corresponds to more spiky traffic pattern. Figure 2 shows $f(\alpha)$ for $p = 2$, and 20. From Figs. 1 and 2, we can see that more spiky traffic trace has a narrower spectrum.

B. Impact on Queueing Behavior

Traffic with rich multi-fractal nature has significant influence on queueing performance because multi-fractal

TABLE I
TCP VERSIONS AND COMPONENTS OF CONGESTION CONTROL.

TCP Version	Fast Recovery	Fast Retransmission	Retransmission Time Out
Default			✓
Tahoe		✓	✓
Reno	✓	✓	✓

nature represents spikiness of packet arrivals. In Er-ramilli et al.[3], performance of single server queue is evaluated for three types of input traffic (1) with both LRD and multi-fractal scaling, (2)with LRD and without multi-fractal scaling, and (3)without LRD and with multi-fractal scaling. This results show that LRD and multi-fractal nature affect on the queueing behavior in the case of high and low link utilization, respectively. Furthermore, at intermediate link utilization, where it would be reasonable to operate networks, both LRD and multi-fractal nature are important. Therefore, we should carefully design networks with taking account of not only LRD but also multi-fractal.

III. TCP CONGESTION CONTROL

A. Congestion Control of TCP-Reno

We study the Reno version of TCP, which is the most popular TCP in the current Internet. TCP-Reno congestion control is based on the additive increase and multiplicative decrease mechanism containing Fast Retransmit and Fast Recovery algorithms [5]. Its window update algorithm for congestion window $cwnd$ is described as follows.

- (1) After every non-repeated ACK (acknowledgment),

(Slow Start Phase)

$$cwnd = cwnd + 1 \quad \text{if } cwnd < W_t$$

(Congestion Avoidance)

$$cwnd = cwnd + 1/cwnd \quad \text{otherwise}$$

where W_t is a window threshold.

- (2) When three duplicate ACKs (dup ACKs) are received,
 - (i) set $W_t = cwnd/2$ and $ndup = 3$,
 - (ii) retransmit a packet which dup ACKs indicate,
 - (iii) set $cwnd = W_t + ndup$,
 - (iv) each time another dup ACK arrives, $ndup = ndup + 1$,
 - (v) $cwnd = W_t$ when an ACK for new data is received,

where $ndup$ is the number of dup ACKs.

- (3) When a retransmission timer expires (Retransmission Time Out),

Fig. 3. Simulation model.

- (i) set $W_t = W/2$

- (ii) set $cwnd = 1$.

In this case, the algorithm goes into slow start phase.

With fast retransmit, after receiving three dup ACKs for the same TCP segment, the sender infers that a packet has been lost and retransmits the packet without waiting for a retransmission timer to expire, leading to higher channel utilization and connection throughput. Fast Recovery is a congestion avoidance algorithm to avoid going into Slow Start, which reduces the connection's throughput because it sets the congestion window size to 1. Once three dup ACKs are received, the sender enters into the fast recovery phase, retransmits one packet which dup ACKs indicate, and reduces its congestion window by one half. In fast recovery, the sender's usable window becomes $cwnd = W_t + ndup$. The sender sends a new packet for each additional dup ACKs received ($ndup$ is incremented). Upon receipt of an ACK for new data, the sender exits fast recovery by setting $ndup$ to 0, and resumes normal congestion avoidance.

B. TCP versions and Components of Congestion Control

In the next section, we use three types of congestion control, namely, TCP-Reno, TCP-Tahoe, and default TCP, to investigate what component affects on the multi-fractal nature. Table I shows components of congestion control included in these TCP versions. Default TCP and TCP-Tahoe can be regarded as subsets of TCP-Reno. Default TCP has only Retransmission Time Out mechanism to detect packet losses. TCP-Tahoe has Retransmission Time Out and Fast Retransmission mechanisms.

IV. COMPUTER SIMULATION RESULTS

A. Simulation Model

Figure 3 shows a network model for computer simulation. We use OPNET simulator[10]. We use a simple network topology with one server and one client. We

Fig. 4. $f(\alpha)$ of default TCP for packet loss probability of 1,2,3, and 4%

Fig. 5. $f(\alpha)$ of Tahoe TCP for packet loss probability of 1,2,3, and 4%

assume that the server always has data to send (i.e., greedy model). Link delays are constant and the congestion control is triggered by randomly dropping TCP segments with probabilities of 1%, 2%, 3%, and 4%. In a real network, TCP segments are dropped at a router. So, burstiness is observed in packet loss in general. This burstiness of packet loss may cause multi-fractal nature or LRD on traffic pattern. However, our goal in the paper is to analyze multi-fractal nature caused by transport protocol, not by network protocol (packet dropping at a router). Thus, we assume that packets are lost randomly in order to remove influence of burstiness of packet loss.

The increment process $X_{\Delta}(i)$ is measured from arrival packets to the client every 5msec (i.e., $\Delta t = 5\text{msec}$), and $f(\alpha)$ is calculated from the measured process.

B. Evaluation of $f(\alpha)$

Figures 4,5, and 6 show $f(\alpha)$ of default TCP, TCP-Tahoe, and TCP-Reno, respectively. We can see that default TCP has the widest spectrum and TCP-Tahoe and TCP-Reno have almost the same spectrum. This means that Retransmission Time Out mechanism significantly affects on the multi-fractal nature, because default TCP

Fig. 6. $f(\alpha)$ of TCP Reno for packet loss probability of 1,2,3, and 4%

has only the Retransmission Time Out mechanism to detect congestion while other two TCPs have the Fast Retransmission or Fast Recovery as well as the Retransmission Time Out.

In order to evaluate contribution of Retransmission Time Out to the multi-fractal nature, we ran simulations for TCP-Tahoe and TCP-Reno in a situation where Retransmission Time Out does not occur. Because the TCP sender detects congestion with the Retransmission Time Out when multiple packets are lost in one window of data, we artificially adjust the packet dropping pattern to avoid occurrence of Retransmission Time Out.

Figures 7 and 8 show $f(\alpha)$ of TCP-Tahoe and TCP-Reno in this situation. We can see that $f(\alpha)$ becomes narrower as compared with Figs. 5, and 6, which are results including the effect of Retransmission Time Out. We can also see that $f(\alpha)$ becomes narrower as the packet loss probability increases. This means that Fast Retransmission makes traffic close to mono-fractal because both the congestion controls have Fast Retransmission and Fast Retransmission occurs more frequently as the packet loss probability increases.

Figure 9 shows $f(\alpha)$ for each TCP version in the case with and without the Retransmission Time Out for the packet loss probability of 4%. We can see that Fast Recovery mechanism also makes traffic close to mono-fractal because TCP-Reno has a narrow spectrum than TCP-Tahoe and TCP-Tahoe does not have Fast Recovery mechanism while TCP-Reno does.

It has been reported that Retransmission Time Out is one of causes to generate LRD[8]. Therefore, the above results show that Retransmission Time Out mechanism causes not only LRD but also multi-fractal nature. In order to improve bad behaviors of TCP such as global synchronization, throughput degradation, etc., many schemes have been proposed such as SACK(Selective Acknowledgment) option[6] and RED(Random Early

Fig. 7. $f(\alpha)$ of TCP Tahoe without Retransmission Time Out for packet loss probability of 1, 2, 3, and 4%.

Fig. 9. Comparison of $f(\alpha)$ for each TCP version for packet loss probability of 4%

Detection) gateway[11]. Most of these schemes are based on the fact that Retransmission Time Out should be avoided. Our results show that these approaches to improve TCP performance are also reasonable from the viewpoint of LRD and multi-fractal nature.

Fig. 8. $f(\alpha)$ of TCP Reno without Retransmission Time Out for packet loss probability of 1, 2, 3, and 4%

V. CONCLUSIONS

In this paper, we focus on the TCP congestion control as causes of affecting on the multi-fractal nature of the Internet traffic. We artificially divide the congestion control mechanism to several components, i.e., Fast Recovery, Fast Retransmission and Retransmission Time Out mechanisms, and evaluate contributions of these components to multi-fractal nature. Our numerical evaluation, which is based on $f(\alpha)$ spectrum, shows that Retransmission Time Out plays an important role. In the future research, we will study the congestion control mechanism to effectively reduce multi-fractal nature.

REFERENCES

[1] W.E.Leland, M.S.Taqq, W.Willinger and D.V.Wilson, "On the Self-Similar Nature of Ethernet Traffic(Extended Version),"

Proc. IEEE/ACM Trans. on Networking, Vol. 2, No. 1, pp.1-15, Feb. 1994.

[2] A.Feldmann, A.C.Gilbert and W.Willinger "Data network as cascades: Investigating the multi-fractal nature of Internet WAN traffic" *Proc. ACM Sigcomm '98*, pp.42-55, 1998.

[3] A.Erramilli, O.Nrayan, A.Neidhardt and I.Saniee "Performance Impacts of Multi-Scaling in Area TCP/IP Traffic" *Proc. IEEE INFOCOM 2000*, pp.352-359, March, 2000

[4] R.H.Riedi and J.Levy-Vehel. "TCP traffic is multi-fractal: a numerical study." Technical Report No.3129, INRIA Rocquencourt, France, Feb., 1997

[5] W. Richard Stevens, "TCP/IP Illustrated, Volume1: The Protocols", Addison Wesley Longman.

[6] M. Mathis, J. Mahdavi, and S. Floyd, "TCP Selective Acknowledgment Options," RFC 2018, Apr. 1996.

[7] K. Fall, and S. Floyd, "Simulation-based Comparison of Tahoe, Reno, and SACK TCP," *Computer Communication Review*, Vol. 26, No. 3, pp. 5-21, Jul. 1996.

[8] B. Sikdar, and K. S. Vastola, "On the Contribution of TCP to the Self-Similarity of Network Traffic," *Lecture Notes in Computer Science(Proceedings of IWDC)*, Vol 2170, pp. 596-613, 2001.

[9] R.H.Riedi, M.S.Crouse, V.J.Ribeiro, R.G.Baraniuk, "A Multi-Fractal Wavelet Model with Application to Network Traffic," *IEEE Transactions on Information Theory*, Special issue on multiscale statistical signal analysis and its applications, No. 45, pp. 992-1018, April, 1999.

[10] OPNET Technologies, Inc. (<http://www.opnet.com>)

[11] S. Floyd, V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, pp. 397-413, Aug. 1993.