Regular Paper

Contribution of the Application, Transport and Network Layers to the Self-Similarity of Internet Traffic

Peter Ivo Racz,† Takahiro Matsuda† and Miki Yamamoto††

There have been no studies evaluating the dominant factor responsible for generating self-similar network traffic, although many researchers have discussed the causes of self-similarity. In this paper, we have isolated and compared the effects of the application, transport and network layers on self-similarity. We first focus on the interaction between the transport and network layers only. With a simple computer simulation scenario we show that in realistic bottleneck scenarios with only long greedy TCP flows present, the TCP flows fully utilize the bottleneck link, which make the aggregated TCP flow short-range dependent while each flow that share the bottleneck link itself is long range dependent. We also show that in special situations, when the TCP flows become synchronized, the bottleneck link may become underutilized giving space to long-range dependence to surface. We show the connection between the network layer parameters necessary for synchronization to occur, and verify our results with simulation. To examine the effect of the application layer, we introduce web traffic besides the long greedy flows into the bottleneck link. We show that link utilization is again a key factor determining the self-similar properties of the aggregated flow, but has much less impact on the self-similar properties of the aggregated web sessions and long greedy flows. Finally, we point out the most influential effects and the conditions when they dominate the self-similar properties of aggregated TCP flows.

1. Introduction

Self-similarity is a ubiquitous property of network traffic. It represents a correlation property of traffic volume over a wide range of time scales, i.e., Long Range Dependence (LRD). Traffic with self-similar nature is detrimental. Its effects range from high queue buffer overflow rates to large delays and persistent periods of congestion [1]. Therefore understanding the nature of self-similarity and its reduction is a focus of attention of the networking community [2].

Recent research efforts have identified various origins of self-similarity on different protocol layers. On the application level, Internet applications like HTTP have shown self-similar traffic, due to the heavy-tailed file-size distributions of the transmitted files [3]. The transport layer —— such as TCP —— also actively participates in modifying the self-similarity of the traffic arriving from the traffic source [4),5]. Latest research has revealed that even a single TCP flow can induce and show significant self-similar properties [6]. This is due to the congestion avoidance and retransmission schemes built into the TCP congestion control. And finally, the network itself plays an important role by amplifying or reducing the degree of self-similar traffic. The queuing techniques of routers and switches [2),7] and the applied traffic filters [8] all affect self-similarity. It has also been shown that self-similar flows induce self-similarity in other flows which originally showed no self-similar property when sharing the same queue [9].

Despite the large number of papers published on self-similarity, the complexity of the subject limits the focus of most papers. The originality of our research is that through experiments, we separate and rank the effects of the application, transport and network layers on the self-similarity of both single and aggregated TCP flows.

The rest of the paper is organized as follows. Section 2 gives a brief overview of self-similarity and the Hurst parameter. Section 3 describes the simulation environment. Section 4 investigates the interaction between the transport and network layers. Section 5 focuses on transport, network and the application layers as well. Section 6 summarizes our findings.

2. Definition of Self-Similarity

Self-similarity was found in different types of network traffic under various traffic conditions [1),3),10] since its discovery [11]. To quantify the degree of self-similarity in network traffic

† Graduate School of Engineering, Osaka University
†† Faculty of Engineering, Kansai University

throughout this paper we use the Hurst parameter. In this section, we give a short introduction on self-similarity, the Hurst parameter and the applied measurement method.

Traffic is self-similar with Hurst parameter H ($0 < H < 1$) if for all $k > 0$ and $t \leq 0$,

$$Y(t) = k^{-H} Y(kt) \tag{1}$$

where $Y(t)$ is the traffic volume in the function of time and $=$ means equality in the sense of distribution. If $H > 0.5$, the traffic is also long range dependent which means that it has considerable fluctuations over a long period of time.

Various estimators can be used to provide estimates of self-similarity and long-range dependence [12),13)]. We have used the variance time plot method, which is defined by an aggregated series

$$X^{(m)}(k) = \frac{1}{m} \sum_{i=(k-1)m+1}^{km} X_i, k=1,2....., \left[\frac{N}{m}\right] \tag{2}$$

based on the time series X_i. We have calculated the variance of this aggregated series for block sizes of $m = 1$, 10, 100 and 1000, where $m = 1$ corresponds to a period of 50 ms. We plot on a log-log plot the sample variance versus m. If the series is self-similar with long-range dependence then the plot is a line with slope $\beta > -1$. If the series is self-similar with short-range dependence then $\beta \leq -1$. The estimation of Hurst parameter H is given by $H = 1 + \beta/2$.

3. Simulation Environment

This section describes the simulation environment used throughout this paper. We used the discrete event network simulator ns2 [14)] to run our simulations.

On the application layer, we examine the effect of UDP applications along with two of the most popular TCP application in today's networks: web traffic and long greedy traffic sources. Besides popularity, we choose these applications because they use TCP as transport protocol, which is in the focus of attention when examining the self-similar properties of transport layer [15)]. UDP, the other widely used transport layer protocol is an open-loop protocol and was shown to have less impact on the self-similarity of internet traffic [16)].

There are many TCP variants and their optimizations implemented in the ns2 simulator.

Table 1 TCP Reno settings.

TCP model	One-way
RTT clock (tcpTick_)	10 ms
Initial/reset value of cwnd (windowInit_)	1
Maximum limit of congestion window (maxcwnd_)	infinite (0)

The goal of this paper is to compare the effect of the currently widely used version of TCP against the influence of the network and application layers. We have therefore used the Reno version of TCP with options that are common in today's networks. **Table 1** describes the various settings of TCP Reno used throughout the simulations.

We used the one-way TCP implementation of the ns2 simulator. TCP flows with one-way TCP senders can carry traffic only from the senders towards the receivers. The returning ACK packets have no payload as described in the manual of the ns2 simulator [17)]. The restriction of the one-way TCP senders imposes no limitation through the bottleneck link scenarios in this paper.

The size of the packets is set to 1 kbytes which does not closely reflect reality. However, the packet size of 1 kbytes only has numerical effects, and doesn't influence the conclusions in this paper. The packet size of 1 kbytes has been also frequently used by other reserchers [15)].

The granularity of the internal clock used to measure the Round Trip Time (RTT) (tcpTick_) of the packets has changed from 500 ms to 10 ms in recent years. We have validated our simulation results against clock granularity of the ideal 1 ms, the default 10 ms and the legacy 500 ms as well and found no significant impact. Throughout the paper the simulation results are shown with the clock granularity set to the current default of 10 ms.

Presently the ns2 simulator defaults the initial/reset value of the congestion window (windowInit_) to the size of 2 packets. Values above the original size of 1 packet are aimed at helping the transmission of traffic that is composed of short-lived bursts on links with small bandwidth or long delays. We have investigated the effect of the initial/reset value of the congestion window size on the self-similarity of network traffic and found no measurable difference.

The maximum limit on the TCP congestion window (maxcwnd_) has profound effect on the simulation results. We discuss the implications

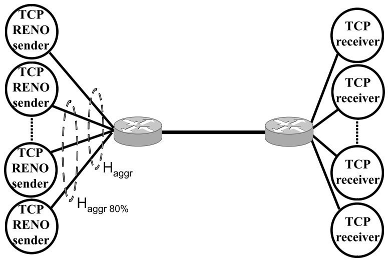

Fig. 1 Bottleneck topology with multiple greedy TCP flows.

Table 2 Parameters and assumptions for simulation.

Number of flows	4–150
Link delay	50–100 ms
Link speed	1–5 Mbps
Buffer size	15–55 packets
Packet size	1 kbytes
Queuing scheme	FIFO, RED

of maxcwnd_ on the self-similarity of network traffic in Section 4.3. The simulation results in all other sections in this paper assume that there is no limit on the congestion window as indicated in Table 1.

The network layer contains 3 functions; aggregating multiple flows, queuing and routing. We discuss the effect of each of these functions by employing a simple network topology with a bottleneck router as shown in **Fig. 1**. The TCP receivers are attached directly to the bottleneck, while the senders are connected via a feeder link. These feeder links impose no bandwidth restriction on the individual TCP flows, but give the opportunity to introduce different end-to-end propagation delay for each flow. The propagation delays are uniformly distributed between 50–100 ms as described in **Table 2**.

Each simulation is run for the duration of 2,200 sec, where we have discarded the first 400 sec of data to avoid transients. The flows are started randomly during the first 30 s of the simulation.

We have measured the Hurst parameter with Variance-Plot method, at 4 timescales: 50 ms, 500 ms, 5 s, and 50 s. We conducted linear regression on the variance of the aggregated series on a log-log scale using the linear regression function of the gnuplot software. We have also verified selected points of our measurements with the wavelet based estimator of the SELFISH [12] self-similarity analysis tool.

Most of the Hurst parameter and other curves in the figures throughout this paper were drawn with least squares approximation. In a few cases, the data was only smoothed with the Bezier function to reveal finer details.

4. Interaction between the Transport and Network Layers

In this section we examine how the interaction between the transport and network layers affect the self-similarity of TCP flows. In order to focus on the effects of the transport and network layers only, we employ greedy data sources at the application layer, because the greedy data source reflects the self-similar properties of the TCP congestion control. This is because the greedy source constantly generates packets which must wait for TCP to open its usable window before transmission.

We examine the properties of both the single and aggregated TCP flows. The aggregated flow consists of between 4–150 single TCP flows and crosses the simple bottleneck topology shown in Fig. 1. Each aggregation level give different packet loss rates at the bottleneck link as the TCP flows compete for the available bandwidth. The buffer size, transmission speed and link delay of the bottleneck link are set according to Table 2.

4.1 Effect of Aggregation

To measure the effect of aggregation we measure the Hurst parameter of each single and the aggregated TCP flow first at fixed network layer parameters of $D = 50$ ms, $S = 3$ Mbps, and $B = 55$ packets, where D is the propagation delay between the sender and receiver, S is the transmission speed of the bottleneck link, and B is the size of the queue buffer. In each experiment we also measure the level of packet loss of both the single and aggregated TCP flows. **Figure 2** shows the Hurst parameter of both the single and aggregated TCP flows against the measured packet loss level. The single flow curve represent the average Hurst parameter of the single TCP flows of the aggregated TCP flow. We can see the following facts from Fig. 2:

(A) As a result of aggregation, the Hurst parameter of the aggregated TCP flow has lower values compared to its composite single TCP flows. The aggregated flow is predominantly short-range dependent ($H < 0.5$) while it shows an increasing trend on high packet loss levels.

This finding confirms the finding of the literature [18]. Sikdar and Vastola [18] analytically proves that the Hurst parameter of aggregated

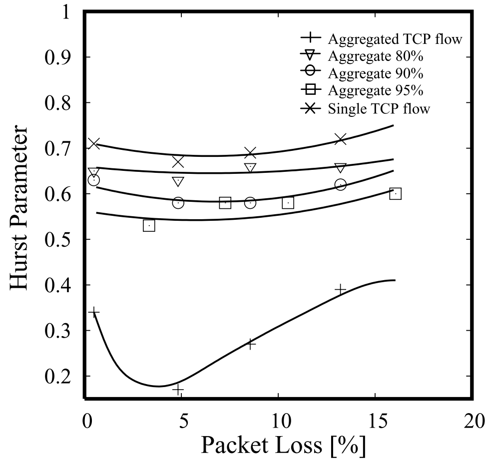

Fig. 2 Hurst parameter of the single TCP flow and the aggregated TCP flows formed out of randomly chosen 80% 90% 95% and 100% of the single TCP flows crossing the bottleneck link.

TCP flows is only dependent on the level of packet loss and shows increasing trend.

Traffic with Hurst parameter $H < 0.5$ is short-range dependent. The variance of short-range dependent traffic decreases faster than exponential when measured at increasing timescales. That is, short-range dependent traffic is amenable to statistical multiplexing and does not have the detrimental properties of long-range dependent traffic on network performance.

Long-range dependent traffic ($H > 0.5$) however has considerable fluctuation on large timescales compared to smaller timescales. These fluctuations on large timescale are responsible for the detrimental effects of long-range dependent traffic. In the following we show that the fully utilized bottleneck link in our experiment prevents significant fluctuations to occur, and is therefore the main reason behind the short range dependence of the aggregated flow.

The fully utilized bottleneck link shapes the arrival process into a constant bit rate departure process, which makes the possible fluctuation of the arrival process very limited. During time interval dt, the number of packets arriving to a fully utilized bottleneck link is,

$$N_{in} = N_{out} + N_{drop} + Q_{t2} - Q_{t1} \qquad (3)$$

where N_{in} is the number of packets arriving at the bottleneck link router, N_{out} is the number of transmitted packets, N_{drop} is the number of dropped packets at the bottleneck link during time interval dt, Q_{t2} and Q_{t1} are the queue size at the bottleneck link at time t_2 and t_1 respectively, and $dt = t_2 - t_1$. Eq. (4) is derived by substituting N_{drop}=$N_{in} * p$ and $N_{out} = S * dt$ into Eq. (3).

$$N_{in} = \frac{S * dt + Q_{t2} - Q_{t1}}{1 - p}, \qquad (4)$$

where S is the link speed of the bottleneck link, and p is the average packet loss during time interval dt. According to Eq. (4) the variance of the arrival process depends on the variance of two parameters, the variance of the queue size and the variance of the packet loss.

The queue size related fluctuation is bounded by the buffer size: $abs(Q_{t2} - Q_{t1}) \leq B$, where B is the size of the buffer at the bottleneck link. The buffer size on the other hand is bounded by the queuing delay. This is because on a typical link in today's internetworks, the maximum queuing delay is in the range of 0.1 sec, which gives $B < 0.1 * S$, where S is the link speed. Referring back to the counter of Eq. (4), the queuing related fluctuation is below 0.5% on the minute timescale, i.e., when $dt \approx 60$ s. That is, the queuing related fluctuation of the arrival process on large time scale is negligible at bottleneck links with realistic link delays, when the bottleneck is fully utilized.

According to Eq. (4), the variability of packet loss is the other factor, which affects the fluctuation the aggregated flow. A p% fluctuation of the packet loss translates into a $1/(1-p)$% fluctuation of the arrival process of the aggregated flow, which makes the packet loss dominant compared to the queue related fluctuation discussed above. With the increase of the average packet loss rate, the range of possible fluctuation increases. **Figure 3** shows that the Hurst parameter indeed increases with the increase of packet loss. However, on large timescales this fluctuation is also very limited. This is because TCP Reno adjusts its transmission speed according to the packet loss rate and such large timescale fluctuation of packet loss would create large timescale fluctuation of the transmission rate of the TCP flows. The criteria of the bottleneck link being fully utilized prevents the large timescale fluctuation of the TCP flows.

To investigate further the properties of fully utilized bottleneck links, we have randomly selected a group of the single TCP flows crossing the bottleneck link and in a following experiment measured the Hurst parameter of the aggregate of these selected flows. We run simulations where we selected 80%, 90% 95% etc.

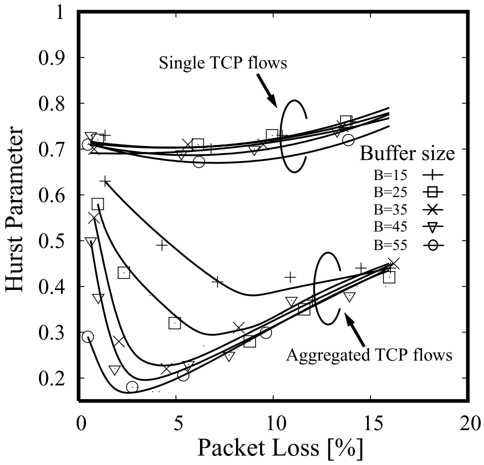

Fig. 3 Hurst parameter of aggregated and single TCP flows at different buffer size ($S = 3$ Mbps, $D =$ 50–100 ms).

of the flows crossing the bottleneck link. This kind of experiment is of interest, because similar conditions apply when doing live measurement over the internet. The results are also shown on Fig. 2.

(B) Figure 2 shows that the Hurst parameter of the aggregate of the selected flows rapidly increases as we include less and less flows in the aggregate. When the selected flows make up about less than 70% of all the flows crossing the bottleneck link, the Hurst parameter reaches its maximum showing long-range dependency at every packet loss level.

Even though the bottleneck link is fully utilized, the aggregate of the selected flows can fluctuate on both short and long time scales. This is because while competing with the rest of the flows at the bottleneck link, the aggregate of the selected flows can gain or loose bandwidth against the rest of the flows. The smaller the selected aggregate compared to the rest of the flows is, the bigger the variance of these fluctuation become, which may lead to higher Hurst parameter values, if self-similarity is present.

4.2 Effect of Buffer Size

In this section we focus on three parameters that influence the effect of queuing. First we examine the effect of the packet buffer, then in the next subsections we examine the effect of the maximum congestion window and finally the effect of the queuing technique by comparing the effects of FIFO tail-drop queuing and RED queuing.

Using the network setup of the previous subsection, Fig. 3 shows the Hurst parameter curves of the aggregated TCP flows and their component single TCP flows for the buffer size range of $B =$ 15–55 packets. All other network parameters are constant. We can see the following facts from Fig. 3:

(C) The single TCP flow's Hurst parameter curves are long-range dependent ($H > 0.5$) and identical for all buffer sizes. This indicates that the Hurst parameters of the single TCP flows are independent of the buffer size of the network layer. They solely depend on the packet loss level.

(D) On low packet loss levels the Hurst parameter of the aggregated TCP flows show an increased level. The smaller the buffer, the higher the Hurst parameter is.

Veres, et al. [5] concluded experimentally, that the self-similarity of the aggregated TCP flow increases as B/N decreases, where B stands for the buffer size of the bottleneck link, and N stands for the number of TCP flows crossing the bottleneck. However, according to Fig. 3 the Hurst parameter is related to B only on low packet loss levels, and related to $1/N$ only on high packet loss levels. Note that Fig. 3 depicts the Hurst parameter as the function of packet loss rather than the number of TCP flows. The packet loss and the number of flows are interchangeable in this situation, because increases in the number of flows exhibit proportionate increases in packet loss at the bottleneck link router.

In the following, we discuss how the buffer size affects the Hurst parameter of the aggregated TCP flows at low packet loss levels (i.e., Fact D), which was not clarified in Ref. 5) and Ref. 19). We create a model and later show that not only the buffer size, but other network layer parameters like the link speed and delay variance have similar effects on the Hurst parameter of the aggregated TCP flow. Finally we confirm our results with simulation.

In the previous simulation we have measured the utilization of the bottleneck link along with the Hurst parameter of the aggregated flow at different buffer sizes. **Figure 4** shows that the smaller the buffer size, the lower the utilization of the bottleneck link is. In the previous subsection we have shown that fully utilized bottleneck links have negligible variance on large timescales and found that such aggregated flows are short range dependent. Contrary to fully utilized links, aggregated flows crossing under-

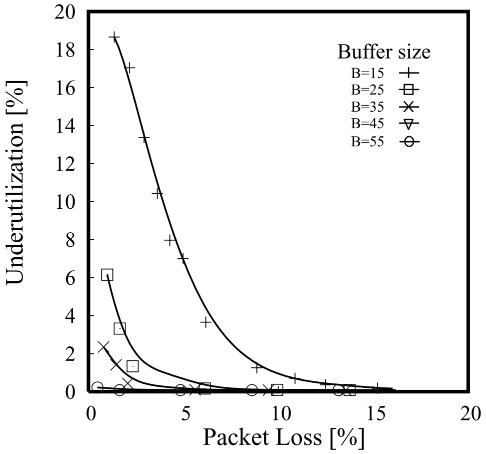

Fig. 4 Underutilization of the bottleneck link for different buffer size values (Utilization = 100% − Underutilization).

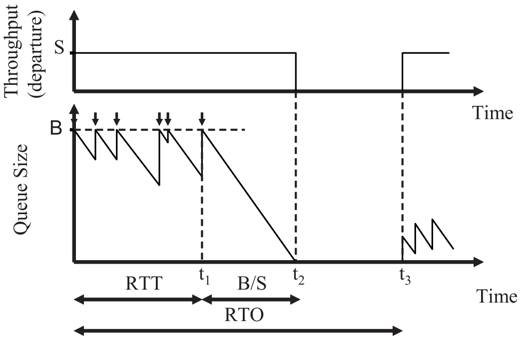

Fig. 5 Throughput (above) and queue size (below) of the bottleneck link during synchronization.

utilized links may have strong long-range variance increasing the Hurst parameter of the flow. This happens when the utilization of the bottleneck link shows strong long-range variance. Comparing the corresponding flows on Fig. 3 and Fig. 4 confirms that lower link utilization leads to higher Hurst parameter of the aggregated flow, that is the underutilization is the reason for the elevated Hurst parameter of the aggregated flow at small buffer sizes.

Investigating further the source of the underutilization phenomena, we have found that it is due to synchronization between the TCP flows of the aggregate. **Figure 5** shows the throughput rate and the queue size of the bottleneck link during such a synchronization scenario. The small arrows on the top of Fig. 5 mark the time, when a single TCP flow overfloods the buffer and enters timeout. Since most of the single TCP flows timeout in a RTT interval of time, there will be very few packets entering the queue of the bottleneck link between time t_1 and t_3. The underutilized period of the bottleneck link starts at t_2, when the queue gets empty and lasts until t_3, when the first flows recover from timeout and start transmitting again.

According to Fig. 5, underutilized periods appear in the departure process of the bottleneck link when

$$RTO > RTT + \frac{B}{S}. \tag{5}$$

and the length of these underutilized periods is

$$T_{OFF} \approx RTO - \left(RTT + \frac{B}{S}\right), \tag{6}$$

where RTT is the longest round-trip time among the TCP flows crossing the bottleneck, B is the size of the buffer of the bottleneck, S is the transmission speed of the bottleneck link, RTO is the time between the beginning of the timeout of the first TCP flow and the time when the first flow starts retransmitting.

With the help of Eq. (6) we now can explain the effect of the buffer size on the self-similarity of TCP flows (i.e., Fact D). With the increase of the buffer size, the B/S ratio of Eq. (6) increases and T_{OFF} decreases, that is, the underutilized periods are getting shorter decreasing the long-range variance of the throughput. Above a certain buffer size Eq. (5) no longer holds, the queue is large enough so that it never empties and the underutilized periods disappear. The buffer size does not only affect the length of these periods, but also the frequency of their occurrence. This is because at small buffer sizes the single TCP flows overflow the buffer more frequently.

4.3 Effect of Maximum Window Size

In this section we examine the effect of the maximum congestion window (maxcwnd_) on the self-similarity of network traffic. Using the network topology described in Fig. 1 and the simulation parameters of the previous section, we measure the Hurst parameter of a fixed number of flows ($N = 7$) at different maxcwnd_ values. The maxcwnd_ limitation applies to all single TCP flows. **Figure 6** shows the simulation results.

According to Fig. 6, reducing the maxcwnd_ reduces the Hurst parameter of both the single and aggregated flows. The effect of maxcwnd_ can be devided into three qualitatively different regions based on the packet loss level.

In the range where the packet loss level is at

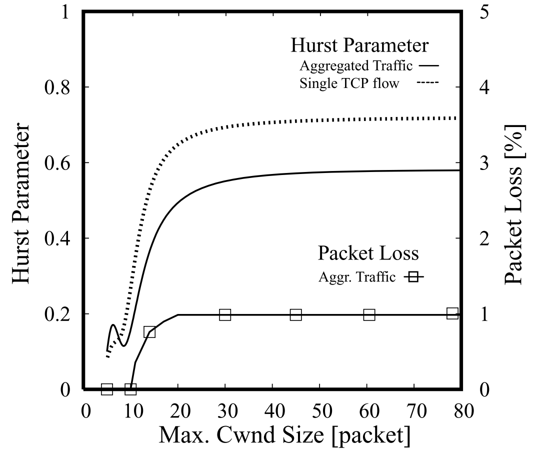

Fig. 6 Hurst parameter of the single and aggregated TCP flows at different maximum congestion window size.

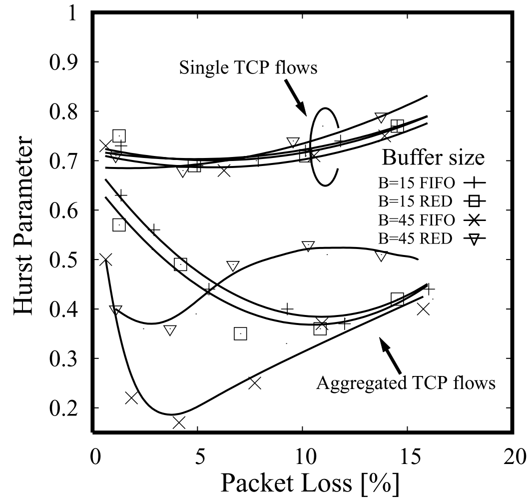

Fig. 7 Hurst parameter of aggregated and single TCP flows at different buffer size using tail-drop FIFO and Random Early Detection schemes.

its maximum ($p_{max} \approx 1\%$), the maxcwnd_ of the TCP flows is large enough not to impose any limit on their congestion windows. On this range, Fig. 6 shows self-similarity because the TCP flows are synchronized. All simulation results in the other sections of this paper are based on this assumption that the TCP flows do not limit the congestion window of the TCP flows.

In the range where $0 < p < p_{max}$, the maxcwnd_ limits the transmission of some of the individual flows. These limited flows transmit with fixed transmission rate set by the maxcwnd_ until a packet drop occurs. Packet drops occur when the momentarily not limited flows ($cwnd < maxcwnd\_$) overload the buffer. By lowering the maxcwnd_ more flows become limited and less flows compete for the available bandwidth. This is reflected in the decreasing packet loss values. In this packet loss range, the Hurst parameter of both the aggregated and single TCP flows show decreasing trend, converging to zero. This is due to the decreasing long-range variance of the aggregated flow. The limited flows transmit with fixed rate due the limit on their congestion window introducing no long-range variance. With the decrease of the maxcwnd_ the number of competing flows that intruduce variance into the aggregate's throughput rate is decreasing and therefore lower the Hurst parameter of the aggregate flow.

Finally, the zero packet loss $p = 0\%$ indicates that there is no competition between the flows for the available bandwidth, because the maxcwnd_ in each TCP flow is not large enough to cause congestion at bottleneck link. All single flows transmit with their congestion windows open to the maximum. Neither the aggregated nor the single flows show long range dependence ($H < 0.5$) in this range. The Hurst parameters converge to zero, because the large scale variance of the flows are zero due to the constant transmission rate of every single flow. On small time scales, only a small amount of variance can be measured, because the TCP senders transmit in a bursty fashion due to the compression of TCP acknowledgment packets.

4.4 Effect of Queuing Scheme

Our goal in this subsection is to compare the effects of the RED algorithm on the self-similarity of network traffic against the drop-tail queuing mechanism and the other factors discussed in this paper. RED is in the focus of interest, because it is also a widely used queuing algorithm in today's networks. RED is designed to slow down TCP senders to prevent buffer overflow by randomly dropping packets from the queue before overflow occurs.

We re-ran the simulation of the previous subsection this time using Random Early Detection (RED) queuing mechanism at the bottleneck link. **Figure 7** shows the simulation results for RED at two different buffer size values. We can see the following facts in Fig. 7:

(E) When the buffer is large ($B = 45$), RED increases the Hurst parameter of the aggregated flow compared to tail-drop queuing on a wide packet loss range ($p > 1\%$).

On higher packet loss levels the aggregated flow may even show small long-range dependence ($H > 0.5$).

Sikdar, et al.[2] has modeled the single flows of network traffic as ON-OFF processes, and shown that the source of self-similarity in the aggregated traffic is the result of the infinite variance of the length of these OFF periods. When RED is used at the bottleneck link, RED further increases the amount of OFF periods in the traffic by sending the TCP flows into timeout instead of activating their congestion avoidance technique. This is because the probability that two consecutive packets in the queue that RED drops belong to a single flow are high due to the bursty nature of TCP. After loosing two or more packets in a congestion window of data, the effected TCP flow will stop transmitting while waiting for its Retransmission Timeout timer to expire[20]. That is, applying RED at the bottleneck link increases the number of timeouts of the single TCP flows (OFF times) which is responsible for the increased Hurst parameter of the aggregated flow.

(F) When the buffer size is large ($B = 45$), RED reduces the Hurst parameter of the aggregated flow on very low packet loss levels ($p < 1\%$).

There are two opposing mechanisms determining the self-similarity of the aggregated flow on low packet loss level. RED increases the Hurst parameter of the aggregated flow by frequently dropping two or more packets from the same flow, forcing single flows into timeout. At the same time RED reduces the aggregated flow's Hurst parameter by preventing the synchronization between the single flows. According to Fig. 7 the dominant effect is the synchronization, therefore the cancellation of synchronization overweights the effect of timeouts and makes RED reduce the Hurst parameter.

When the buffer is small ($B = 15$), RED doesn't influence the Hurst parameter of the aggregated flow, because RED forces the singe flows into timeout with much lower probability. This is because the single flows can send only smaller bursts to the queue due to the small buffer size ($B = 15$) and the high number of flows. With smaller bursts in the queue, RED is less likely to drop two or more packets from the same flow. Also, in such environment RED often can not prevent the queue from saturating and drops packets in a drop tail fashion reducing its impact compared to tail-drop queuing.

In this section we have seen that RED has different effect on the self-similarity of network traffic according to the size of the packet buffer. On links with large buffers, RED mostly increases the self-similarity of traffic. Sikdar, et al.[2] has proposed some modifications to RED and shown that the modified RED algorithm can in fact reduce the self-similarity of network traffic.

4.5 Effect of Routing Scheme

To examine the effect of routing on the self-similarity of network traffic, we generalized our simple bottleneck network model decribed in Fig. 1. After crossing the bottleneck link, the single flows take different paths to their destinations through the IP network cloud as shown in **Fig. 8**.

When a link failure, metric change etc. occurs in the network cloud, traffic is temporarily disrupted while the routing protocol converges. In our discussion, we ignore the temporary effects of routing, because self-similarity is a long range property of traffic while in stable networks routing events are infrequent. Also, modern routing protocols and traffic engineering techniques tend to minimize converge time.

After the convergence, some of the flows that cross the bottleneck link will take new paths towards their destinations through the network cloud. The impact on the bottleneck link depends on how far the routing event occurs in the network from the bottleneck link. The closer it happens, routing affects more of the flows that cross the bottleneck link.

The TCP senders of the affected flows will experience different RTTs along the new paths and will adjust their transmission speed accordingly. This change of RTT is due to the different propagation delay and queuing delays along the new paths. We evaluate the impact of the propagation and queuing delays on the

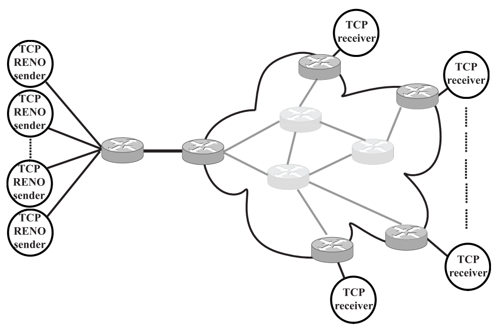

Fig. 8 Bottleneck link topology with network cloud.

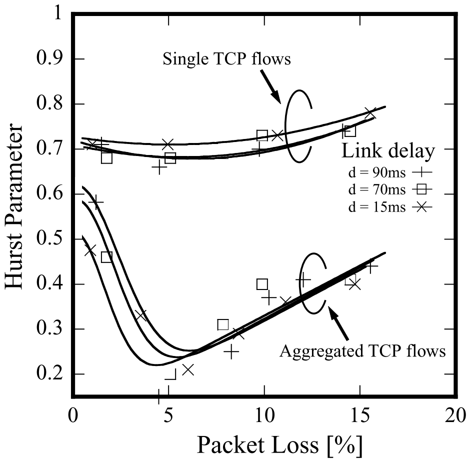

Fig. 9 Hurst parameter of aggregated and single TCP flows at different link delays.

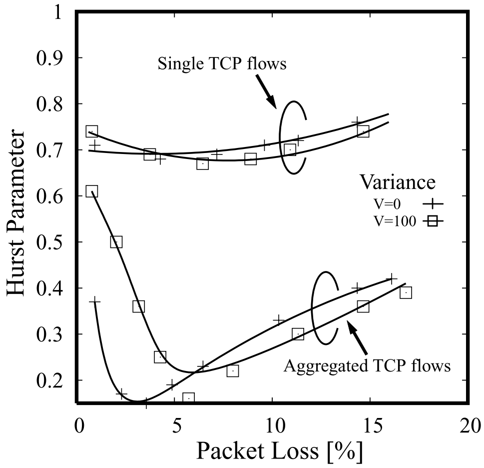

Fig. 10 Hurst parameter of aggregated and single TCP flows at different link delay variance levels.

self-similarity of network traffic in two steps.

First, to understand the effect of propagation delay, we ran simulations using the bottleneck topology shown on Fig. 1 with different bottleneck link propagation delay values. By changing the delay value of the bottleneck link shown on Fig. 1, we model that a routing event in the network cloud shown on Fig. 8 changes the RTT of some of the single flows that cross the bottleneck link. The flows originally had a propagation delay uniformly distributed between 50–100 ms using the feeder links as described in Sect.3, therefore changing the propagation delay of the bottleneck link can be interpreted as changing the propagation delay of some of the flows while leaving the rest unchanged.

Figure 9 shows the simulation results for different link delay values.

(G) The link delay has very small impact on the Hurst parameter of the aggregated flow on low packet loss level. In this range, increasing the link delay increases, while decreasing the link delay decreases the Hurst parameter of the aggregated flow.

The analyzis of the trace of the aggregated flow showed that decreasing the link delay does not or very slightly makes the T_{OFF} underutilized periods shorter. To explain this finding we refer again to Eq.(6). According to Ref. 19) RTO is calculated as

$$RTO = sRTT + 4sD, \tag{7}$$

where sRTT is the smoothed RTT and D is the smoothed mean deviation. Substituting Eq. (7) into Eq. (6)

$$T_{OFF} \approx sRTT + 4sD - \left(RTT + \frac{B}{S} \right), \tag{8}$$

shows that the change of link delay value has no or very small effect on the length of the the T_{OFF} underutilized periods assuming that RTT $\approx$ sRTT.

After evaluating the effect of propagation delay, we focus on the variance of RTT introduced by the queuing along the new path. To model the effect of RTT variance along the new path, we have changed the link delay value of the bottleneck link every $dt = 500$ ms during simulation with a variance of $V = 100$ while the mean link delay remained $D = 50$ ms in the network topology described in Fig. 1. **Figure 10** compares the simulation results with the constant link delay results taken from Fig. 3, where $B = 45$.

(H) The variance of RTT modifies the Hurst parameter of the aggregated flow on low packet loss level. In this range, higher variance of RTT leads to higher value of the Hurst parameter. As shown in Fig. 10, previously non self-similar ($H < 0.5$) may show long-range dependence ($H > 0.5$) as a result of change in the fluctuation of RTT values.

To explain this finding we refer again to Eq.(8). Equation 8 shows that the variance of RTT affects the self-similarity of the aggregated TCP flow through the $4sD$ product, where sD is the smoothed deviation of the RTT. An increase of the RTT variance increases the length

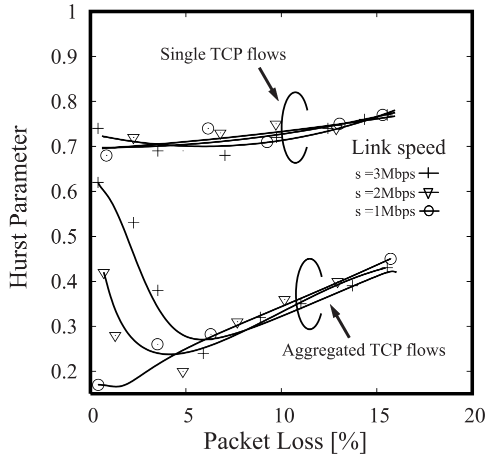

Fig. 11 Hurst parameter of aggregated and single TCP flows at different link speeds.

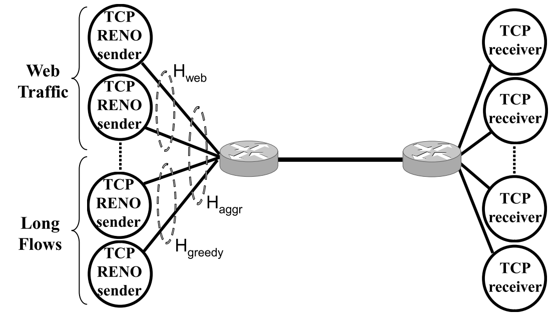

Fig. 12 Bottleneck link with web traffic and long greedy TCP flows.

of the T_{OFF} underutilized periods, which increases the Hurst parameter of the aggregated flow.

Finally, we examine the effect of direct bottleneck link failures on the self-similarity of network traffic. When the bottleneck itself fails, the router usually reroutes the traffic through another of its links. The new link may have different buffer size, delay, speed and background traffic compared to the failed link. In the previous subsections we have already discussed the effect of these parameters except the effect of the link speed. In the following we verify the effects of link speed.

Equation (6) shows that the transmission speed of the bottleneck link affects the self-similarity of the aggregated TCP flow through the B/S ratio. Increasing the link speed decreases the B/S ratio making the underutilized periods (T_{OFF}) longer. **Figure 11** shows the Hurst parameter curves for various link speeds values. In Fig. 11 we can see that slowing down the transmission speed reduces the Hurst parameter of the aggregated TCP flows while all other parameters are constant. Eventually the increased level of Hurst parameter completely disappears.

Summarizing our findings, routing may increase or decrease the self-similarity of aggregated network traffic at a bottleneck in the network when a large portion of the flows crossing the bottleneck link is rerouted to routes with higher or lower variance of delay. However, the absolute value of the delay along the new path has very small or no impact on the self-similarity of aggregated traffic.

5. Interaction between the Application, Transport and Network Layers

In this last section, we compare the effects of all 3 layers, the application, transport and network layers on the self-similarity of aggregated traffic.

The effect of the application layer depends on the applications that generate data into the networks. First, we discuss the effect of applications that use the UDP transport protocol and then we focus on the most popular TCP based applications i.e., web traffic and greedy application. Our goal is to identify how and when the self-similarity of the application layer dominates the self-similarity of aggregated network traffic.

5.1 Effect of UDP Applications

Packet loss does not affect the throughput of UDP connections because the UDP transport protocol has no built-in congestion control techniques. Therefore, when measuring the self-similarity of UDP flows, we directly measure the self-similarity of the data source at the application layer. That is, when the transport layer is UDP, the self-similarity of the application layer is dominant.

The self-similarity of popular UDP applications like streaming voice and video have been discussed in previous publications [15),16),21)].

5.2 Effect of TCP Applications

To measure the self-similarity of TCP based applications, we have modified the bottleneck link topology used in the previous sections. A subset of the source-destination pairs act as web traffic clients and servers as shown in **Fig. 12**. We conduct simulations with two kinds of web traffic, one which shows strong long range dependence, and one which does not. The web traffic with strong long range dependence was

Table 3 Parameters and assumptions for simulation.

Number of web flows	100
Number of greedy flows	0–70
Link delay	50–100 ms
Link speed	3 Mbps
Buffer size	55 packets
Packet size	1 kbytes
Queuing	FIFO

generated in the following manner. Each web traffic session consists of the download of a single page, where the inter-page request time is assumed to be heavy tailed Pareto distribution with average 30 s and shape parameter $\alpha = 1.25$. Each page consists of a single file where its size also based on a Pareto distribution, with average 50 KB and shape parameter $\alpha = 1.25$. When the web source is on, it generates data with 4 Mbps rate. When aggregating such web sessions, the aggregated traffic will exhibit long-range dependence. The non self-similar web traffic has the same parameters except that the inter-page request time and file size follows exponential distribution. It has been shown that the aggregated flow of exponentially distributed web sessions do not exhibit long-range dependence.

The rest of the flows correspond to long greedy TCP flows, which are present during the entire simulation. In each simulation, the number of web traffic flows are constant, and we increase the number of long ftp flows crossing the bottleneck link causing different levels of congestion as described in **Table 3**. We measure the Hurst parameter of the arrival process of the aggregated flow as shown on Fig. 12.

Figures 13 and **14** shows the Hurst parameter and the packet loss of the aggregated flow at the bottleneck link when the web traffic consists of flows based on Pareto and exponential distributions respectively. We can see the following facts from Figs. 13 and 14:

(I) Superimposed ON/OFF processes based on exponential distribution are known to show no long range dependency ($H = 0.5$), however in our simulation the exponential distribution based web traffic shows long range dependency ($H > 0.5$) even when no greedy flow is present ($H = 0.62$).

This is because the transmission process distorts the original exponential ON/OFF distribution of the data sources. By reducing the variance of each data source on small timescales, but leaving their large timescale variance unchanged, the arrival process at the

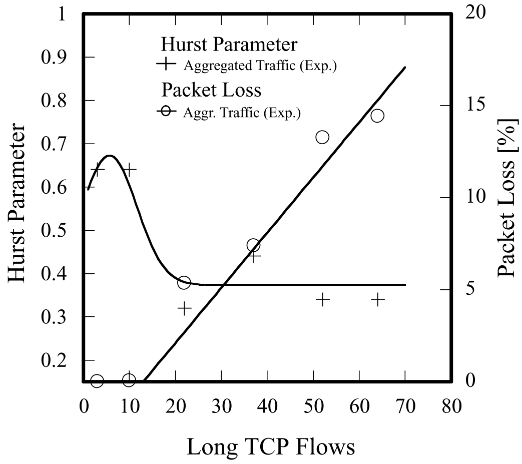

Fig. 13 Hurst parameter and packet loss of the aggregated flow when the web traffic is exponential distribution based.

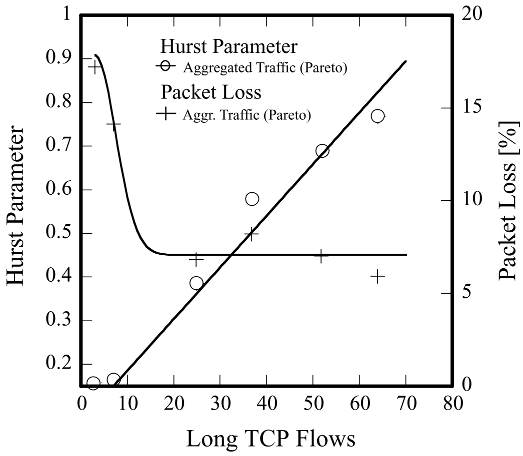

Fig. 14 Hurst parameter and packet loss of the aggregated flow when the web traffic is Pareto distribution based.

bottleneck link shows $H > 0.5$. The network reduces the variance on small timescales, because it can not transmit the packets of the data sources as fast as the data sources generate them. This is especially true when multiple sources happen to transmit at the same time. In these cases a queue builds up at the bottleneck link, which slows down the transmission rate of all active flows, which transform the short intense burst of the data sources into a longer less intense burst of the arrival process at the bottleneck link effectively reducing the variance of web traffic on small timescales.

The web-traffic passes through bottleneck links without much change in their large-scale properties regardless of the traffic conditions at the bottleneck link. When the bottleneck link is congested, the transfer of each session takes longer compared to the non-congested bottleneck link as discussed above, however it does not effect the transmitted data volume of the session and this way the large-scale variance of the aggregated flow.

(J) The aggregated flow shows long range dependence only on low packet loss levels. On this range, the aggregated flow that contains the Pareto based web traffic shows higher Hurst parameter than the aggregate with exponential based web traffic.

When there are no long greedy TCP flows present, the web traffic utilizes about half of the bandwidth of the bottleneck link and therefore the aggregated flow reflects the Hurst parameter of the web traffic. On low packet loss level, the bottleneck link is not fully utilized, because there are only a few long TCP flows that can not keep up with the fluctuation of the web traffic and can not completely fulfill the link. On this packet loss range, the effect of the web traffic is dominant on the Hurst parameter of the aggregated flow. As the number of long TCP flows increase, the long greedy TCP flows can increase their throughput more dynamically utilizing the available bandwidth more efficiently. As discussed in the previous section, the arrival process on a fully utilized bottleneck link has negligible variance on large timescales, which leads to the short-range dependence of the aggregated flow on this packet loss range ($H < 0.5$).

To further investigate this issue we have also measured the Hurst parameter of both the web traffic and the long greedy TCP flows separately at the bottleneck link in the previous experiment as shown in Fig. 12. **Figure 15** shows the Hurst parameter of the web traffic and the long TCP flows when the web traffic sources are exponential, and **Fig. 16** shows when the web traffic is Pareto distribution based. We can see the following fact from Figs. 15 and 16:

(K) The aggregate of the web traffic and the aggregate of the long TCP flows shows long-range dependency on the whole packet loss range and their Hurst parameter curves run close to each other.

The long range dependency of the web traffic on the entire packet loss range is the result of the fact that the congestion at the bottleneck link does not affect the variance of the traffic at large timescales and decreases it on small timescales as discussed in Fact I.

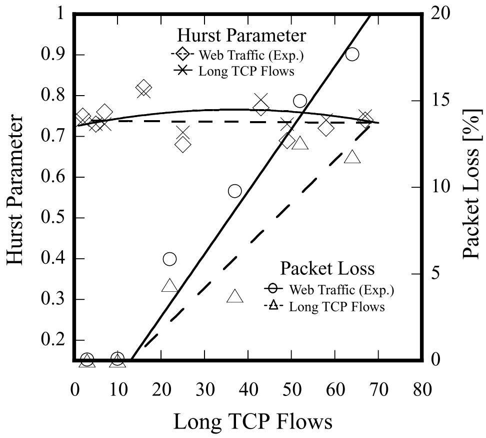

Fig. 15 Hurst parameter and packet loss of the aggregates of the web traffic flows and the long greedy TCP flows when the web traffic is exponential distribution based.

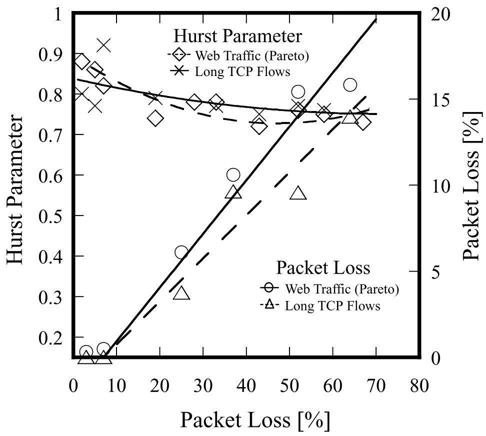

Fig. 16 Hurst parameter and packet loss of the aggregates of the web traffic flows and the long greedy TCP flows when the web traffic is Pareto distribution based.

The long range dependency of the long greedy TCP flows on the entire packet loss range is the result of the web traffic. Veres, et al. [9] has shown that the long greedy TCP traffic can be modeled as a linear system and therefore it adapts to the fluctuation of background traffic above a certain frequency. In our simulation, the long greedy TCP flows adapt to the fluctuation of the web traffic and as a result reveal a degree of similarity close to that of the web traffic.

6. Conclusions

Through experiments, we have isolated and compared effects of the application, transport and network layers on the self-similarity of aggregated TCP flows in different environments.

In lightly congested networks, the network layer is dominant when no web traffic is present. The fully utilized bottleneck link severely limits the fluctuation of the aggregated flow on large timescales, which leads to the short-range dependency of the aggregated flow. When synchronization occur, the bottleneck link gets underutilized, which allows long-range dependency to surface raising the Hurst parameter of the aggregated flow.

Most networks carry not only greedy but web and UDP based traffic as well. When these network run on low packet loss level, the dominant factor lies in the application layer and above, which include file size distribution and human factors like response time. Unfortunately, these factors can not be easily changed. When the web traffic or UDP traffic has high degree of self-similarity, the aggregated flow will show high degree of self-similarity as well.

The transport layer however can be influenced with different queuing techniques. Applying RED at the bottleneck link has positive effect as long as the traffic responds to RED and RED can keep the packet loss level low. Sikdar, et al. [2] proposed some improvements to lower the negative impact of RED on higher packet loss levels.

The network layer is also readily changeable. Avoiding too small receiving buffers at routers and up-sizing buffer memory when faster links are installed may also help to decelerate the self-similarity of the traffic at the bottleneck link by avoiding potential synchronization to occur.

In heavily congested networks, the aggregated flow shows no long-range dependence, because the fully utilized bottleneck link prevents fluctuation on large timescales both in the presence and absence of web traffic. While the aggregated flow shows no long range dependence or it is short-range dependent, smaller aggregates formed out of the single web and long greedy flows of the aggregated flow can still show strong long-range dependency.

Acknowledgments The authors would like to thank Dr. Biplap Sikdar of Rensselaer Polytechnic Institute for his valuable comments.

References

1) Paxson, V. and Floyd, S.: Wide area traffic: The failure of Poisson modeling, *IEEE/ACM Trans. on Networking*, Vol.3, No.3, pp.226–244 (June 1995).
2) Sikdar, B., Chandrayana, K., Vastola, K.S. and Kalyanaraman, S.: On Reducing the Degree of Second-Order Scaling in Network Traffic, *Proc. IEEE GLOBECOM*, pp.2594–2598, Taipei, Taiwan (Nov. 2002).
3) Crovella, M. and Bestavros, A.: Self-similarity in World Wide Web traffic: Evidence and possible causes, *IEEE/ACM Trans. on Net-working*, Vol.5, No.6, pp.835–846 (Dec. 1997).
4) Guo, L., Crovella, M. and Matta, I.: How does TCP generate pseudo-self-similarity?, *Proc. International Work-shop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*, Cincinnati, OH (Aug. 2001).
5) Veres, A. and Boda, M.: The chaotic nature of TCP congestion control, *Proc. IEEE INFOCOM*, Tel Aviv, Israel (Apr. 2000).
6) Figueiredo, D.R., Liu, B., Misra, V. and Towsley, D.: On the Autocorrelation Structure of TCP Traffic, *Tech. Rep., Computer Science Technical Report 00-55*, University of Massachusetts (2000).
7) Tinnakornsrisuphap, P., Feng, W. and Philp, I.: On the Burstiness of the TCP Congestion-Control Mechanism in a Distributed Computing System, *20th International Conference on Distributed Computing Systems* (*ICDCS 2000*) (Apr. 2000).
8) Procissi, G., Gerla, M., Kim, J., Lee, S., and Sanadidi, M.: On Long Range Dependence and Token Buckets, *SPECTS* Orlando, Fl., USA (July 2001).
9) Veres, A., Kenesi, Z., Molnar, S. and Vattay, G.: On the propagation of long-range dependence in the Internet, *ACM Computer Communication Review*, Vol.30, No.4, pp.243–254 (Oct. 2000).
10) Leland, W., Taqqu, M., Willinger, W. and Wilson, D.: On the self similar nature of Ethernet traffic (extended version), *IEEE/ACM Transactions on networking*, pp.1–15 (Feb. 1994).
11) Willinger, W.: The Discovery of Self-Similar Traffic, *Lecture Notes in computer Science*, Vol.1769, pp.513–527 (2000).
12) Karagiannis, T., Faloutsos, M. and Molle, M.: A User-Friendly Self-Similarity Analysis Tool, *Special Section on Tools and Technologies for Networking Research and Education, ACM SIGCOMM Computer Communication Review*, Vol.33, pp.81–93 (2003).

13) Abry, P. and Veitch, D.: Wavelet Analysis of Long-Range Dependence Traffic, *IEEE Transactions on Information Theory*, Vol.4, No.1, pp.2–15 (1998).
14) ns2 simulator. http://www.isi.edu/nsnam/ns
15) Park, K., Kim, G. and Crovella, M.: On the relationship between file sizes, transport protocols, and self-similar network traffic, *Tech.Rep.*, Boston University (July 1996).
16) Feldmann, A., Gilbert, A.C., Huang, P. and Willinger, W.: Dynamics of IP traffic: A study of the role of variability and the impact of control, *Proc. ACM SIGCOMM*, Boston, MA (Aug. 1999).
17) The ns2 manual. http://www.isi.edu/nsnam/ns/ns-documentation.html
18) Sikdar, B. and Vastola, K.S.: On the Contribution of TCP to the Self-Similarity of Network Traffic, *Lecture Notes in computer Science* (*Proc. IWDC*), Vol.2170, pp.596–613 (2001).
19) Stevens, W.: TCP/IP Illustrated, Vol.1: The Protocols, *Addison Wesley* (1994).
20) Fall, K. and Floyd, S.: Simulation-Based Comparison of Tahoe, Reno, and SACK TCP, *ACM Computer Communication Review,* Vol.26, No.3, pp.5–21 (July 1996).
21) Beran, J., Sherman, R., Taqqu, M.S. and Willinger, W.: Long-Range Dependence in Variable-Bit-Rate Vide Traffic, *IEEE Transactions on Communications*, Vol.43, No.2/3/4, pp.1566–1579 (1995).

(Received September 21, 2004)
(Accepted September 2, 2005)
(Online version of this article can be found in the IPSJ Digital Courier, Vol.1, pp.576–589.)

Racz Peter Ivo received the B.E., M.E. degrees in electrical and communication engineering from the Technical University of Budapest (TUB), Hungary in 1996, 1999 respectively. In 2005, he has joined Cisco Systems Japan K.K., where he is currently a customer support engineer. He is concurrently working toward his Ph.D. at the Department of Communications Engineering, Osaka University. His research interests include Quality of Service and Internet traffic analysis.

Takahiro Matsuda received the B.E., M.E. and Ph.D. degrees in communication engineering from Osaka University in 1996, 1997 and 1999, respectively. In 1999, he joined the Department of Communications Engineering, Osaka University, where he is currently an Assistant Professor. His research interests include wireless and mobile TCP/IP networks, wireless ad-hoc and sensor networks, and the Internet traffic analysis. He is a member of IEEE and IEICE.

Miki Yamamoto received the B.E., M.E. and Ph.D. degrees in communications engineering from Osaka University in 1983, 1985 and 1988, respectively. In 1988, he joined Department of Communications Engineering, Osaka University. In 2005, he moved to Department of Electrical Enginnering and Computer Science, Kansai University, where he is a Professor. During 1995 and 1996, he visited University of Massachusetts at Amherst as a Visiting Professor. His research interests include multicast communications, high speed networks, wireless networks and performance evaluation of these systems. He is a member of IEEE, ACM and IEICE.