TCP's Role in the Propagation of Self-Similarity in the Internet*

A. Veres[1], Zs. Kenesi[2], S. Molnár[2], and G. Vattay[3]
[1]Traffic Laboratory, Ericsson Research, Budapest, Hungary
and COMET, Columbia University, New York, USA
e-mail: veres@comet.columbia.edu
[2]HSN Laboratory, Budapest University of Technology and Economics
Budapest, Hungary
e-mail: kenesi,molnar@ttt-atm.ttt.bme.hu
[3]Department of Physics of Complex Systems, Eötvös University
Budapest, Hungary
vattay@complex.elte.hu

Abstract

This paper analyzes how TCP congestion control can propagate self-similarity between distant areas of the Internet. This property of TCP is due to its congestion control algorithm, which adapts to self-similar fluctuations on several timescales. The mechanisms and limitations of this propagation are investigated. It is demonstrated that if a TCP connection shares a bottleneck link with a self-similar background traffic flow, it propagates the correlation structure of the background traffic flow asymptotically, above a characteristic timescale. The cut-off timescale depends on the end-to-end path properties, e.g., round-trip time and average window size, and the receiver window size in case of high speed connections. It is also shown that even short TCP connections can propagate long-range correlations effectively. In case when TCP encounters several bottleneck hops, the end-user perceived end-to-end traffic is also long-range dependent and it is characterized by the largest Hurst exponent. Through simple examples, it is shown that self-similarity of one TCP stream can be passed on to other TCP streams that it is multiplexed with. The limitations of propagation depend on network conditions. Propagation complements the widespread scaling phenomena reported in the literature. Our arguments are supported with a combination of analytic techniques, simulations and statistical analyzes of a number of wide area Internet traffic measurements.

Keywords

TCP congestion control, self-similarity, long-range dependence, TCP adaptivity

I. INTRODUCTION

Statistical self-similarity and long-range dependence are important topics of recent research studies. Both phenomena are related to certain scale-independent statistical properties. Statistical self-similarity can be detected when traffic rate fluctuates on several timescales and its distribution scales with the level of aggregation. Long-range dependence means that the correlation decays slower than in traditional traffic models (e.g., Markovian), i.e., it decays hyperbolically. A number of authors have argued that self-similarity in data networks can be induced by higher layer protocols [5] [6] [25] [27] [30] [32]. In this paper we do not discuss the roots of self-similarity, instead, we demonstrate how the induced self-similarity is propagated and spread in the network by lower layer adaptive protocols, in particular, by TCP, which represents the dominant transport protocol of the Internet. This paper also analyzes the impact of self-similarity on end-to-end TCP dynamics, and how the end-user perceived rate fluctuations depend on the end-to-end path properties.

The phenomenon of self-similarity was observed in data networks in [14] [15], followed by several experimental papers showing fractal characteristics in other types of networks and traffic, (e.g., in video traffic [3] [11] or in ATM networks [18]). A comprehensive bibliographical guide is presented in [33]. These observations have questioned the validity of previous short memory models when applied to network performance analysis [25]. The impact of self-similar models, e.g., Fractional Gauss Noise on queuing performance has been investigated in a number of papers [4] [7] [21].

In addition to self-similarity and long-range dependence it was also observed that TCP can exhibit a more complex scaling behavior, which can be captured by multifractal characteristics [9] [20].

* This is an extended version of a paper presented at SIGCOMM 2000 [31].

Considerable effort has been applied to investigating the causes of these phenomena. In [5] the authors argue that self-similarity is induced by the heavy-tailed distribution of file sizes found in Web traffic. In [32] Ethernet LAN traffic was modeled as a superposition of independent On/Off processes with On and Off periods having heavy-tailed distributions. An important related theoretical result [27] proves that the superposition of a large number of such independent alternating On/Off processes converges to Fractional Gaussian Noise.

To prove the validity of this model in TCP/IP networks, several papers have investigated the connection between application level file sizes, user think-times, and the On/Off model. As there are several layers between the application and the link layer, it is of primary importance to investigate how protocols convert and transfer heavy-tails through the protocol stack down to lower layers. The effect of TCP and UDP transport protocols are investigated in [8] [23] [24] and it is found that TCP preserves long-range dependence (LRD) from application to link-layer.

Based on this result, the authors of [8] and [9] argue that *transport mechanisms* strongly affect the short timescale behavior of traffic, but they *have no impact in large timescales*. In this paper, we demonstrate that this statement is valid *only for the local behavior of TCP* when only the traffic of a single link is investigated. In contrast, in the *network case* a surprisingly complex mechanism is present.

The On/Off model is attractive because it captures the heavy-tailedness property. Nevertheless it assumes that during On periods the traffic rate is constant. In reality, TCP uses an end-to-end congestion control algorithm to continuously adapt its rate to actual network conditions. If network conditions are governed by large timescale fluctuations, then TCP will "sense" this and react accordingly. This paper shows that TCP efficiently adapts to traffic rate fluctuations on several timescales. In addition, we demonstrate that TCP can be modeled as a linear system above a characteristic timescale depending on the round-trip time and the maximum window size ($T = RTT * W/2$). The asymptotic linearity implies that the correlation structure of a background traffic stream is taken over faithfully by an adaptive TCP flow. In particular, we show that *TCP can inherit self-similarity from a self-similar background traffic stream*. Since TCP has an end-to-end control, while adapting to these fluctuations, it *propagates self-similarity* encountered on its path between the source and the destination host.

We also demonstrate that if a TCP stream is multiplexed with another one, then, depending on the network conditions, it can pass on self-similar scaling to the other TCP stream. In our model the network is regarded as a mesh of end-to-end adaptive streams. Intertwined TCP streams can *spread self-similarity* throughout the network contributing to global scaling. By analyzing the effects from a network point of view we argue that, on one hand, TCP plays an important role in balancing and propagating global scaling. On the other hand, it keeps local scaling intact where it is already strong. This way we complement results reported in [8]. The main purpose of this paper is to analyze the basic mechanisms behind these phenomena.

To clarify our terminology, we briefly summarize the definition of a few basic concepts. Let $X = (X_k : k \geq 0)$ be a weakly stationary process representing the amount of data transmitted in consecutive short time periods. Let $X_k^{(m)} = \frac{1}{m}\sum_{i=(k-1)m+1}^{km} X_i$ where $m \geq 1$ denote the m aggregated process. X is called *exactly self-similar* with self-similarity parameter H if $X_k \stackrel{d}{=} m^{1-H} X_k^{(m)}$ and the equality is in the sense of finite-dimensional distributions. In the case of *second-order self-similarity*, X and $m^{1-H}X^{(m)}$ have the same variance and autocorrelation. Second-order self-similarity manifests itself in several equivalent ways, one of them is that the spectral density of the process decays as f^{1-2H} at the origin as $f \to 0$. This latter property also defines the class of Long-Range Dependent (LRD) processes.

In practice, exact self-similarity cannot be tested because it would require testing scale invariance of all finite dimensional distributions. We restrict ourselves to second-order self-similarity, which can be tested between some finite upper and lower cut-off timescales, see [2]. Therefore, in this paper we also use the term "self-similarity" to refer to scaling of second-order properties over some specific timescales, see also [17] [28]. We note that certain statements of the paper are also valid in the sense of exact statistical self-similarity. The mathematical analysis part of the paper proves some asymptotic results relevant to LRD processes.

The ns-2 simulator [22] is used for the network simulations. Several variants of TCP are investigated

(Tahoe, Reno, SACK), however, we found that our conclusions are invariant to the TCP version.

The paper is organized as follows. In Section II, several wide-area TCP measurements are analyzed showing self-similar scaling and a possible explanation is presented based on a few simple assumptions. Section III investigates how TCP adapts to fluctuations on different timescales. We show that TCP in a bottleneck buffer can be modeled as a linear system above a characteristic timescale. Section IV investigates how an aggregate of TCP sessions with durations of heavy-tailed and light-tailed distributions propagates self-similarity of a background traffic stream. We also analyze the case of high speed connections, when the connection rate is limited by the advertised window of the receiver. Finally, Section V presents results concerning the spreading of self-similarity in the network case when TCP has to pass multiple hops and compete for resources with other TCP streams.

II. Adaptivity of TCP: a Possible Cause of Widespread Self-Similarity

In this section, we demonstrate how TCP adaptation leads to the propagation of self-similarity using wide area network measurements. The test methodology is also explained using a single measurement, then the test results of several other WAN TCP measurements are presented. Finally, we introduce a simple analytic model of TCP adaptation.

A. Scaling Analysis of Wide Area TCP Measurements

During the first experiment a large file was downloaded (a traffic trace file from the Internet Traffic Archive) from an FTP server (*ita.ee.lbl.gov*) to a client host 15 hops away in Hungary (*serv1.ericsson.co.hu*), passing several backbone providers and a trans-Atlantic link. At the client side there was no other traffic present. The client was directly connected to an ISP by a 128 kbps leased line. All packets were captured at the client side with the *tcpdump* utility. The total amount of bytes received was 50 Mbyte and it was logged with a resolution of 50 ms during the file transfer which lasted for 6900 s. The average throughput, which takes into account the retransmissions and the TCP/IP overhead, was about 58 kbps (i.e., some congestion was experienced in the network). The average round-trip delay between the server and the client was 208 ms. From the packet trace we concluded that the version of the TCP was Reno.

Tests were performed for the presence of self-similarity. Here we present three tests, the first and second ones are based on the scaling of the absolute moments (also called absolute mean and variance-time plots [26]), and the third one is a wavelet-based analysis [1], see Figure 1. The result of the tests suggests asymptotic self-similarity with Hurst parameter approximately 0.75. Non-stationary effects are often present in actual data traffic, which can deceive the self-similarity and LRD tests, see [19]. To avoid such pitfalls we applied the wavelet based test developed by Veitch [29] to test whether the scaling exponents are constant throughout the trace.

During the experiment, there was only one connection active on the link, so explanations based on the superposition of heavy-tailed On/Off processes, exponential backoff [10] [12], or chaotic behavior [30] are not applicable *locally*. However, the investigated TCP connection traversed several backbone links where, due to the large traffic aggregations, self-similarity could arise either because of heavy-tails or chaotic competition. Presumably, whatever the reason for self-similarity was, the TCP connection adapted to the background traffic stream at the bottleneck link, and the effect of the adaptation was that self-similarity was propagated to the measurement point.

We repeated the measurement between different hosts in the Internet, covering very different networking situations. Because the scaling region has to cover at least two orders of magnitude for the scaling analysis, the measurements have to be long enough, at least several hundred seconds long. During such relatively long measurements, daily trends or other types of non-stationarity may disturb the tests. Stationarity analysis was performed to evaluate whether the time series can be considered to be stationary. As it was shown in [10] and [12], if the packet loss probability exceeds a certain value, it may also cause TCP traffic to become LRD even if background traffic is SRD and packet losses happen independently. Although this phenomenon may be very important in the emergence of LRD in TCP traffic in highly congested links, since we are interested

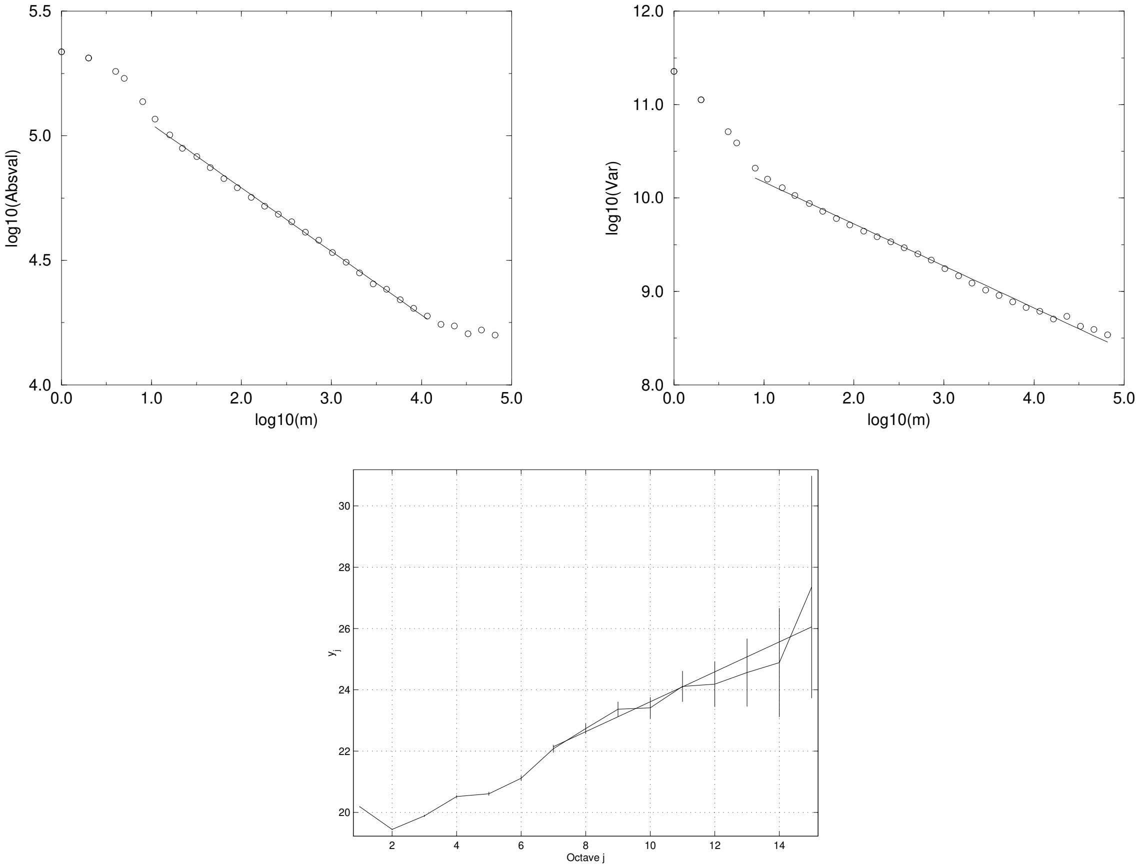

Fig. 1. Scaling analysis of the traffic generated by a file transfer logged at the client side. a) Absolute mean method $H \approx 0.76$. b) Variance-time plot $H \approx 0.77$. c) Wavelet analysis $H \approx 0.744$ [0.71 0.788].

in TCP's adaptation, we only considered measurements where TCP timeouts were not excessively long.

The basic statistics of the analyzed traces are shown in Table I. The first group of measurements consists of "international" measurements between hosts outside of US and Columbia University, New York. In this group we have traces *fr3* (France), *uk* (UK), *au* (Australia). The next three measurements are trans-Atlantic measurements, but in these cases the end-hosts were connected via relatively small speed leased lines, in case of *hu* the speed of the leased line was 128k, while *leasejan26* and *leasejan28* were performed through a 256k leased line of a small home-ISP, all in Budapest, Hungary. The next measurement, *cablejan*, was performed between Columbia University and a host connecting with a cable modem to a public ISP, also in New York. The last group consists of high-speed "backbone" measurements, all within the US: *us3*, *usFeb3*, *usRog*. All these measurements were done between Columbia University and WWW or FTP servers connecting to the Internet via high speed links.

Self-similarity was present mostly in traces where the test TCP connection possibly passed one or more bottlenecks, (i.e., in the "international", "cable modem" and "leased line" traces). In case of backbone measurements, self-similarity was present during peak hours. Otherwise, during low load periods, we usually measured $H = 0.5$. In almost all the cases when self-similarity was not present, we could find that the TCP rate was limited by the advertised window and not the free capacity on the path. Another possible reason is that the access line limits the TCP rate, in which case the resulting traffic flow is smoothed. The effect of paths with large bandwidth delay products and small windows on the propagation of self-similarity is further analyzed in Section IV-C.

B. Simple Analytic Model

In this section, we introduce a simple analytic model supporting the argument that TCP can adapt and inherit self-similarity from background traffic. Later in the paper we analyze the limitations of the simple model and we are going to refine it further.

All relevant components of the simplified network model are illustrated in Figure 2. A single greedy TCP connection sends data between host A and host B. The path of the connection consists of three parts: a network cloud before and after router R and a bottleneck buffer in router R, where the connection has to share service capacity and buffer space with a self-similar background traffic flow. Self-similarity of the background traffic can be induced, for example, by large aggregations of infinite variance On/Off streams as suggested in [5]. In the analytic model it is assumed that TCP can adapt ideally to a background traffic stream in a bottleneck buffer. Under "ideal adaptivity" we mean that the TCP connection is able to consume all remaining capacity unused by the background traffic stream. It is also assumed that the TCP connection does not have any effect on the background traffic. The generality of this assumption covers several practical cases, for example, if the background flow is a large aggregate consisting of a large number of connections. The limits of these assumptions are analyzed later in the paper.

Denote the background traffic rate by $B(t)$, $0 \leq B(t) \leq C$, where C is the service rate of the bottleneck buffer in bit per seconds. If TCP congestion control is "ideal" and its effect on the background traffic is neglected, then the TCP connection will utilize all unused service in the bottleneck. The rate of the "ideal" TCP flow is denoted by $A(t)$:

$$A(t) = C - B(t).$$

The resulting process is simply a shifted and inverted version of $B(t)$, which implies that the correlation structure of processes $A(t)$ and $B(t)$ are the same. In other words, TCP "inherits" the statistical properties of the background process. In particular, let us model the background traffic rate as Fractional Gaussian Noise (FGN):

$$B(t) = m + \sqrt{a} N_H(t) \tag{1}$$

where m is the average rate in bit per seconds [bps], a is the variance, and $N_H(t)$ is a normalized FGN process with Hurst parameter H. Note that FGN is a discrete time process, so the rate at time t is approximated by the amount of bytes sent during sufficiently small constant duration time periods. Based on the arguments above, the adapting TCP will also be an FGN with the same statistical self-similarity exponent H. As TCP congestion control works end-to-end, the same traffic rate can be measured along the path *before* and *after* router R as well. This implies that TCP propagates self-similarity or LRD to parts of the network where otherwise it would not be present.

The result above is based on a simple scenario using a few assumptions, such as ideal TCP adaptivity, single bottleneck, and assuming that the TCP flow does not modify the background traffic characteristics. However, if the implications of this simple scenario are valid in real TCP/IP networks, the consequences for traffic engineering are far reaching. Regarding this, we are going to address the following important questions:

1. What are the limitations of TCP adaptation, i.e., how "ideal" is TCP congestion control when propagating self-similarity or other statistical properties?
2. A single long-lasting connection was used in the simple network model and in the measurements. Can self-similarity be propagated by short duration TCP connections?
3. We assumed that the background self-similarity traffic flow is non-adaptive. Is self-similarity propagated if the background traffic flow is an aggregate of adaptive flows?
4. We considered a single bottleneck on the TCP path. On the other hand, TCP connections may traverse multiple bottleneck routers and buffers where they can mix with multiple self-similar inputs. What are the characteristics of the end-to-end TCP flow in this case?
5. Is self-similarity propagated between adaptive connections, in particular, can self-similarity be inherited from one TCP to another one that has no direct contact with the source of self-similarity?

III. TCP AS A LINEAR SYSTEM

In the previous section it was assumed that TCP congestion control is "ideal", which, as a matter of course, cannot be the case in real networks. The consequence of self-similarity is that fluctuations are not limited to a certain timescale. When analyzing how "real" TCPs propagate self-similarity, the adaptation of TCP to fluctuations on several timescales should be investigated. In this section it is shown that TCP in a bottleneck buffer can be modeled as a linear system, i.e., TCP takes over the correlation structure of the background traffic through a linear function.

TCP is an adaptive mechanism that tries to utilize all free resources on its path. Adaptation is performed as a complex control loop called the congestion control algorithm. Of course, full adaptation is not possible, as the network does not provide prompt and explicit information about the amount of free resources. TCP itself must test the path continuously by increasing its sending rate gradually until congestion is detected, which is signaled by a packet loss, and then it adjusts its internal state variables accordingly. Using this algorithm, TCP congestion control is able to roughly estimate the optimal load in a few round trip times. Since congestion control was introduced in the Internet [13], it has proved its efficiency in keeping network-wide congestion under control in a wide range of traffic scenarios.

In this section we analyze the adaptivity of TCP, and conclude that a simple network configuration, which consists of a single bottleneck buffer shared by a "generator" flow and a "response" TCP flow, can be modeled as a linear system above a characteristic timescale. The lower cut-off timescale of linear behavior depends on the path properties of the connection. The linear system transforms certain statistical properties, e.g., autocovariance, between the "generator" stream and the "response" traffic stream through a transform function, which is characteristic of the network configuration.

A. Measuring the Adaptivity of TCP on Several Timescales

In the first analysis a single, long, greedy TCP stream is mixed with random background traffic streams. See Figure 3 for the configuration. The background streams are constructed in a way, such that they fluctuate on a limited, narrow timescale. To limit the timescale under investigation, the background traffic approximates a constant amplitude sine wave of a given frequency f: $A_{background}(f,t) = a\sin(2\pi ft + \alpha) + m$ where α is a uniformly distributed random variable between $[0, 2\pi]$. The process $A_{background}(f,t)$ is a stationary ergodic stochastic process with correlation $R(\tau) = a^2/2 \cdot \cos(2\pi f\tau)$. The power spectrum of this process consists of a single frequency component at f. In the simulation the background process has to be approximated by a packet stream (packet size of 1000 bytes). The resulting spectrum of the "packetized" time series is not an impulse but a narrow spike, see Figure 4.

If TCP is able to adapt to the fluctuations of the background traffic flow, the same frequency f should appear as a significant spike in the power spectrum of the TCP traffic rate process as well. The ratio of the amplitudes of this frequency component in the spectra is a measure of the success of TCP adaptation on this timescale. Denote the *measure of adaptivity* at frequency f by $D(f)$

$$D(f) = S_{tcp}(f)/S_{background}(f) \tag{2}$$

where $S_{background}(f)$ is the spectral density of the background traffic rate process at frequency f and $S_{tcp}(f)$ is the spectral density of the adapting TCP rate process at the same frequency.

Figure 4a shows an experiment with a background signal of $f = 0.01[1/s]$. The top part of the figure shows the spectrum of the background traffic approximating a sine wave of frequency f. The bottom part is the measured spectrum of the TCP response. The spectrum of the response has a significant spike at f, but it also contains a few smaller spikes at higher frequencies caused by the congestion control mechanism.

Conducting the experiment for a wide range of frequencies f, it is possible to plot the adaptivity curve of TCP. Figure 4b shows the result for several versions of TCP. Note that the shape of the function only slightly depends on the TCP version. It can be seen that in this configuration, TCP adapts well to frequencies below $f_0 \approx 0.15[1/s]$, but it cannot adapt efficiently to fluctuations on higher frequencies.

server	bin size	duration	avg. rate	VT-H	Wavelet-H
fr3	100ms	1200s	471k	0.77	0.874 [0.808, 0.940]
uk	100ms	1800s	115k	0.66	0.704 [0.621, 0.786]
au	100ms	3346s	566k	0.8	0.94 [0.854, 1.038]
leasjan28	10ms	3591s	131k	0.8	0.95 [0.875, 1.019]
leasjan26	100ms	1000s	154k	0.82	0.855 [0.782, 0.929]
hu	50ms	6900s	58k	0.77	0.72 [0.68, 0.77]
cablejan	10ms	3601s	474k	0.85	0.831 [0.759, 0.903]
us3	10ms	495s	1.4M	0.53	0.562 [0.458, 0.666]
usFeb3	10ms	1868s	552k	0.5	0.511 [0.466, 0.556]
usRog	10ms	2115s	98k	0.89	0.889 [0.848, 0.931]

TABLE I

SUMMARY OF WAN MEASUREMENTS. HURST ESTIMATES: VARIANCE-TIME PLOT (VT-H), WAVELET METHOD (WAVELET-H)

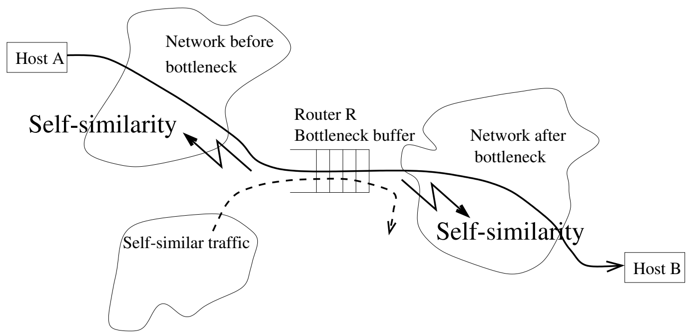

Fig. 2. Network model

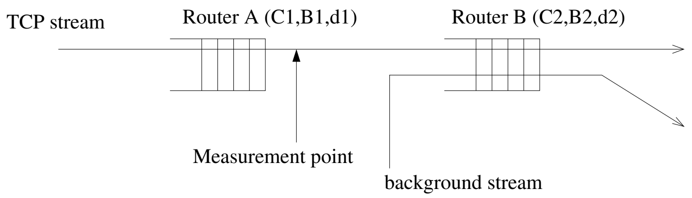

Fig. 3. Simulation model for the test of TCP adaptivity. The two buffers are identical: service rates $C_1 = C_2 = 1$Mbps, propagation delays $d_1 = d_2 = 5$ms, buffer sizes $B_1 = B_2 = 40$ packets.

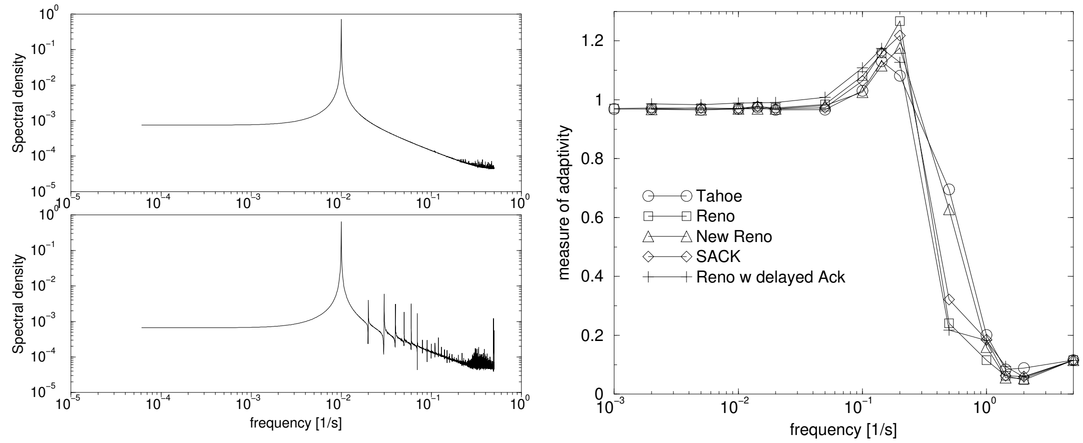

Fig. 4. a) Frequency response to a sine wave of $f = 0.01[1/s]$ (top: background sine wave, bottom: TCP response). In this configuration the measure of adaptivity is $D(0.01) \approx 1$. b) Measure of adaptivity $D(f)$ as a function of the frequency for several TCP variants.

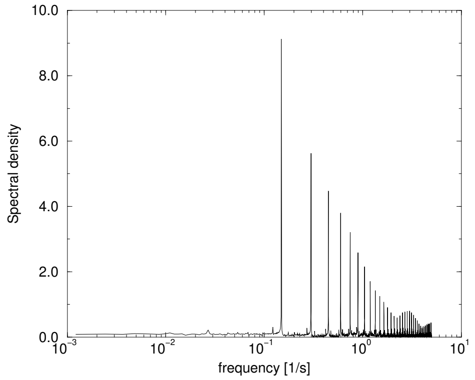

Fig. 5. Spectrum of TCP congestion window process when no background traffic is present.

At f_0 a resonance effect can be observed. At this frequency TCP is more aggressive, and gains even higher throughput than what is left unused by the non-adaptive background flow ($D(f_0) > 1$). This frequency is equal to the dominant frequency of the TCP congestion window process when there is no background traffic present (idle frequency), see Figure 5.

In [16] a macroscopic model for TCP connections was published. It is derived that if every p^{th} packet is lost for a TCP connection, then the congestion window process traverses a periodic sawtooth and the length of the period is $T = RTT * W/2$, where RTT is the round-trip time of the path in seconds and W is the maximum window size in packets. In our case we can approximate $RTT = B/C + d$, where B is the buffer size in packets, C is the service rate in packets per second, and d is the total round-trip propagation delay in seconds. The maximum window size is $W = B + Cd$, which is the maximum number of packets in the pipe (buffer and link). This gives an estimate of $T = 6.81$s and $f_{cycle} = 1/T = 0.15$[1/s]. The result agrees with the measured resonance frequency f_0, and confirms our argument that the resonance effect observed in the measure of adaptivity function $D(f)$ is due to the TCP window cycles (see Figure 4b).

The characteristic timescale of the TCP window cycles spans a relatively wide range in real networks, and the relation of $T \approx RTT*W/2$ can be used for an approximation. For example, if the round-trip time, which in the previous simulation was approximately 0.33 s, is rather in the range of a few tens of milliseconds, the cut-off timescale drops below 1 s. Even below this timescale TCP adapts to fluctuations, though the effectiveness is limited, as shown by the transmission curve. The characteristic frequency f_0 separates traffic dynamics to "local" and "global" scales. Above f_0 the background process shapes the spectrum, while below f_0, the spectrum is a result of TCP control dynamics and external stochastic processes have less impact on it.

B. Tests for Linearity

In real networks background traffic is not limited to a single timescale. In the following, we analyze the case when several frequencies are present and test whether TCP is able to adapt to fluctuations on these timescales or not. The motivation is to prove that TCP can adapt to fluctuations on several timescales independently of each other, more precisely, we want to show that TCP control is linear.

By linear system we mean that if the background traffic rate is given by $B(t)$, and the adapting TCP traffic rate $A(t)$ is expressed using a function Ψ, then $A = C - \Psi(B)$, where Ψ is a linear function of B: $\Psi(a_1B_1 + a_2B_2) = a_1\Psi(B_1) + a_2\Psi(B_2)$. Note that the linear system does not imply that A is a linear function of B. In case of ideal adaptivity, Ψ takes the simple form of $\Psi(x) = x$, and the TCP rate is obtained simply as $A(t) = C - B(t)$, see Section II. If the background traffic is a superposition of streams

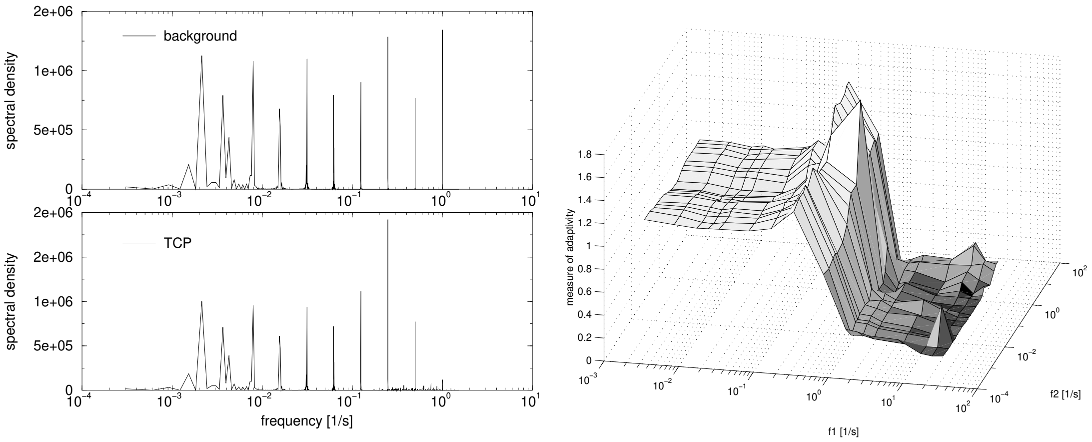

Fig. 6. a) TCP frequency response to the superposition of 10 random phase sine waves. top) background traffic, bottom) TCP response. b) Measure of TCP adaptivity $D(f_1)$ when the background process is composed of two frequencies f_1 and f_2.

$B_i(t), \quad i = 1 \ldots N$

$$B(t) = \sum_{i=1}^{N} B_i(t)$$

then the rate of TCP is given by

$$X(t) = C - \Psi(B(t)) = C - \sum_{i=1}^{N} \Psi(B_i(t)).$$

This construction provides us with a simple test on linearity: we investigate the response to the superposition of several $B_i(t)$ streams and investigate the spectrum of the response. Figure 6a shows the spectral density of the background and the TCP response when the background is a composition of 10 random phase sine waves equidistantly spaced on a logarithmic scale (the nonzero widths of the spikes are due to the fact that the background mix only approximates sine waves with varying packet spacing). It can be observed that, below $f = 1$, TCP is able to adapt to all frequency components in the mix.

To test whether TCP really adapts to fluctuations independently, a wide range of traffic mixes were simulated consisting of two frequencies f_1 and f_2. A large number of simulations were performed, covering a whole plane with the two frequencies, in the range of $[0.05, 500][1/s]$. We then calculated the adaptivity measure for *one* of the frequencies ($D(f_1)$). If the system is linear, the measure of adaptivity function at frequency f_1 should be independent of the other frequency f_2. Our simulation results support this argument, see Figure 6b.

C. Response to White Noise

In the previous analysis the background processes were limited to superpositions of sine wave processes. In real networks background traffic streams cannot be modeled by just a few frequency components, it is more appropriate to model background traffic streams as "noises".

Two types of special noises are most relevant in traffic modeling: the White Noise (WN) process and the Fractional Gaussian Noise (FGN) process. The White Noise process is the most commonly used process in the analysis the frequency response of a system and the Fractional Gaussian Noise process frequently appears as the limit process of traffic aggregations [27].

If TCP is a linear system, then it should transform the correlation of any complex stochastic process (e.g., WN or FGN) through the same transform function. In this section the response of TCP to a WN process is analyzed. WN is a special noise that has constant spectral density. If TCP is linear, then it should respond with the characteristic curve obtained previously. The result is depicted in Figure 7. The similarity of the

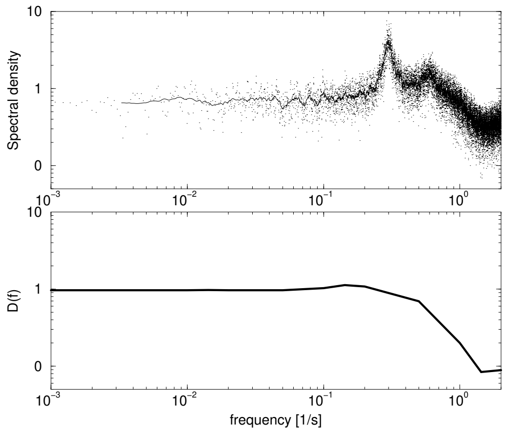

Fig. 7. a) TCP's frequency response to white noise, spectral density (dots) and its smoothed version (line). b) Measure of adaptivity $D(f)$, see also Figure 4b

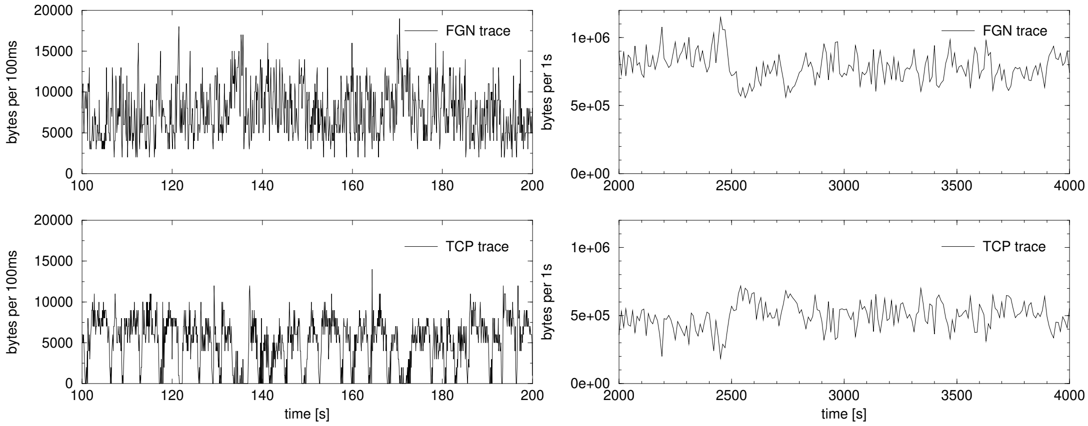

Fig. 8. Traces of FGN ($H = 0.8$) and adapting TCP flows at two aggregation levels. a) 100ms aggregation b) 10s aggregation

curve to our previous test-signal based test ($D(f)$) supports the linearity argument. In addition, the constant flat range, which starts at a characteristic timescale and spans several timescales upwards, provides us with information about the timescale limitation of TCP adaptivity. Note that this mechanism behaves like a low-pass filter.

IV. TCP Adaptation to Self-Similar Background Traffic

Once we have investigated the linearity of TCP and have shown that the transform function is flat below a characteristic frequency, it is quite obvious to expect that TCP, while adapting to signals of complex frequency content, reproduces the same spectral density as the original signal above a timescale, which depends on the path properties (round-trip time, size of the pipe, etc.). If, for example, TCP traverses a link where the traffic shows self-similarity, it will adapt to it with a spectral response equal to the spectrum of the self-similar traffic (asymptotically). As TCP's control algorithm works end-to-end, this property is "propagated" all along the TCP connection path.

A visual test can be seen in Figure 8, where traffic rates of a self-similar $H = 0.8$ FGN background stream and an adapting TCP are depicted. The figure shows that on larger timescales the TCP trace mirrors the FGN trace.

In the experiment we simulated a single TCP sharing the bottleneck buffer with a synthetic FGN traffic flow of $H = 0.8$. Figure 9 shows the power spectrums of both the TCP and FGN traces at an aggregation level of 10ms. As suggested in the previous section, TCP shows the same spectrum as FGN asymptotically, there is a difference only around and below the cut-off timescale. Consequently, the TCP traffic flow is also

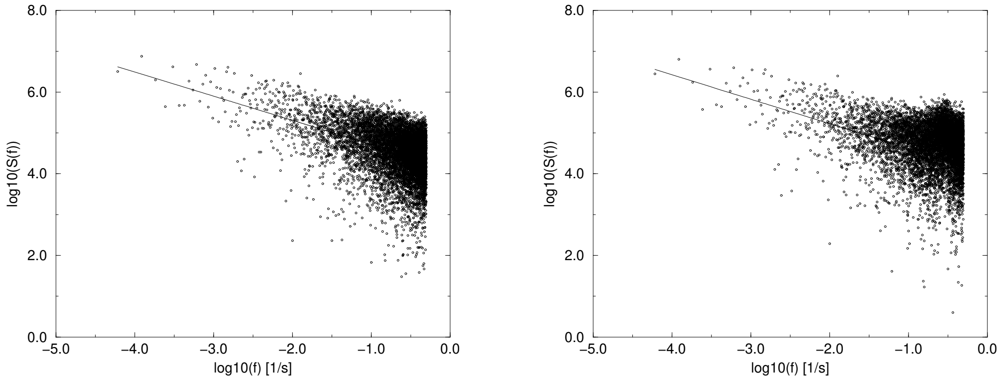

Fig. 9. a) Power spectrum of background traffic $H = 0.8$. b) Power spectrum of TCP traffic adapting to the FGN, estimated $H = 0.8$.

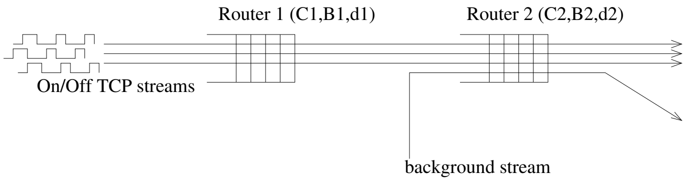

Fig. 10. Simulation model of SRD driven TCP traffic multiplexed with self-similar background traffic (FGN with $H = 0.8$). $C_1 = C_2 = 1$Mbps, $d_1 = d_2 = 5$ms, $B_1 = B_2 = 40$ packets. $k = 10$ parallel sessions with exponentially distributed On and Off periods with means $T_{ON} = T_{OFF} = 10s$.

asymptotically second-order self-similar with the same scaling parameter as the FGN process ($H = 0.8$).

A. *Can Adaptive SRD Traffic Propagate Self-Similarity?*

So far we have analyzed cases when long greedy TCP sessions were mixed with background traffic. In this section, we analyze the more general case of short files. It is demonstrated that the aggregation of short TCP sessions with ligh-tailed On and Off distributions may still exhibit self-similarity property. This self-similarity phenomenon cannot be explained by the self-similarity generation mechanism investigated in [6] [27]. In [6] [27], it is reported that in case of light-tailed On and Off periods the aggregated traffic is SRD.

We have carried out a simulation study with the aggregation of light-tailed On/Off TCP sources. During the simulation we established k parallel sessions. Within each session TCP connections were generated independently and the durations of TCP connections were exponentially distributed (with mean T_{On}) followed by exponentially distributed silent periods (T_{Off}). The simulation was started from the equilibrium state of the process. (See Figure 10.) Let's denote the number of active TCPs at time t by $N(t)$, $0 \leq N(t) \leq k$. With this construction $N(t)$ is a stationary Markov process and it is short-range dependent. See the self-similarity tests for $N(t)$ in Figure 11a ($H \approx 0.5$).

On the other hand, if these sessions are mixed with self-similar background traffic, the aggregate TCP traffic, i.e., the amount of bytes transmitted by all TCPs, is self-similar (Figure 11a). The reason is that the superposition of short duration TCPs can efficiently adapt to a background self-similar process just like one long duration TCP connection.

A real network measurement also supports our argument. Short files (90 kbyte) were downloaded using the *wget* utility from *serv1.ericsson.co.hu* to *locke.comet.columbia.edu* (round-trip time $RTT \approx 180$ ms, average download rate $r \approx 160$ kbps, SACK TCP)[1]. Whenever the download ended, a new download was initiated for the same file. The experiment lasted for an hour, and the file was downloaded about 800 times.

[1]Note that the access speed at the *serv1.ericsson.co.hu* side was increased to 256 kbps during this measurement.

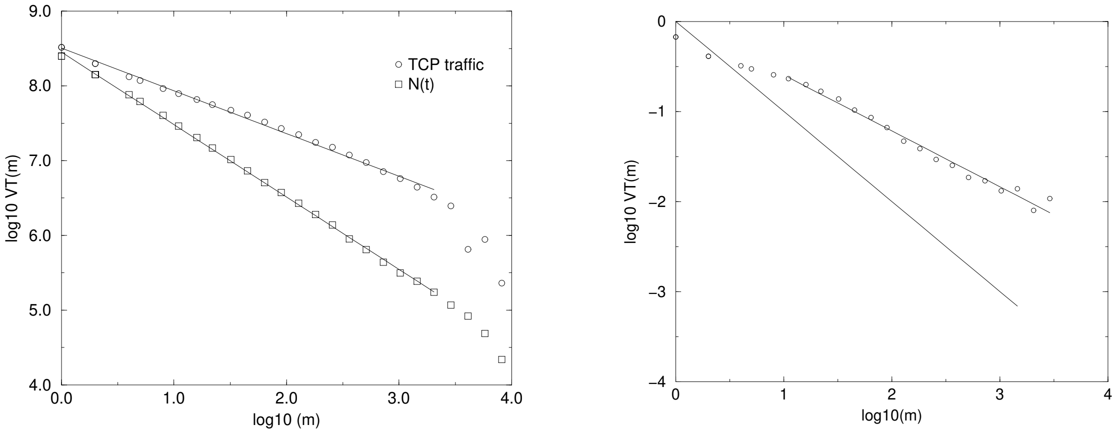

Fig. 11. a) Variance-time test for the simulated On/Off process $N(t)$ ($H \approx 0.5$) and the simulated aggregate TCP traffic ($H \approx 0.72$). b) Variance-time plot of traffic generated with equal size short file transfers from *serv1.ericsson.co.hu* to *locke.comet.columbia.edu*, logging resolution 100 ms, $H \approx 0.7$.

The traffic was captured with *tcpdump* at the client host. The Variance-Time plot shows that the traffic rate dynamics was self-similar, in spite of the short file-sizes, see Figure 11b.

As a new download does not use any memory from a previous TCP connection, long-range correlations in short TCP connection rates can be explained only by the long-memory dynamics of the network. In case of smaller files, TCP's capability to adapt to changing network conditions decreases. Although 90 kbyte is larger than the current average file size in the Internet, it has to be emphasized that a subset of connections is enough to propagate self-similarity. Furthermore, if HTTP 1.1 replaces HTTP 1.0, persistent TCP connections will be able to adapt better to traffic fluctuations, eventually improving the propagation effect; similarly, if a TCP implementation preserves some state from a previous connection, the propagation effect is improved.

B. Discussion on Short-Range Dependent TCP Streams

For simplicity, first assume that there is only one session with On/Off TCP connections multiplexed with self-similar traffic. In this case $N(t)$ takes the values 0 or 1 for exponentially distributed durations. Assuming ideal adaptivity, when the session is active (a TCP is active), it can grab all capacity left unused by the background self-similar traffic. The traffic rate during the *active* periods of the On/Off session can be expressed by $A(t) = F(t)$ where $F(t)$ is the free capacity left by the self-similar background traffic ($F(t)$ is an FGN process, see Section II). During *inactive* periods $A(t) = 0$. Thus the traffic rate of the TCP controlled On/Off session for all t can be written in explicit form as

$$A(t) = N(t)F(t) \tag{3}$$

Assuming that the sessions are independent of the background process ($N(t)$ and $F(t)$ are independent), the autocovariance of $A(t)$: $\gamma_A(\tau) = \mathrm{cov}(A(t), A(t+\tau))$ is

$$\gamma_A(\tau) = E\left[(N(t)F(t) - m_N m_F)\left(N(t+\tau)F(t+\tau) - m_N m_F\right)\right] \tag{4}$$

where $m_N = E[N(t)]$ and $m_F = E[F(t)]$. Factorizing:

$$\gamma_A(\tau) = E[N(t)N(t+\tau)]E[F(t)F(t+\tau)] - m_N^2 m_F^2 \tag{5}$$

The left hand side of the product is

$$E[N(t)N(t+\tau)] = E[(N(t) - m_N + m_N)(N(t+\tau) - m_N + m_N)] = \gamma_N(\tau) + m_N^2 \tag{6}$$

The same holds for $F(t)$, and so the covariance can be written as

$$\gamma_A(\tau) = (\gamma_N(\tau) + m_N^2)(\gamma_F(\tau) + m_F^2) - m_N^2 m_F^2 \tag{7}$$

Finally,

$$\gamma_A(\tau) = \gamma_N(\tau)\gamma_F(\tau) + m_F^2\gamma_N(\tau) + m_N^2\gamma_F(\tau) \tag{8}$$

If $F(t)$ is LRD, its autocovariance decays asymptotically as $\gamma_F(\tau) \sim \tau^{-\beta_F}$ as $\tau \to \infty$, where $0 \leq \beta_F < 1$. On the other hand, if $N(t)$ is SRD, its autocovariance decays asymptotically faster than $\tau^{-\beta_N}$ where $\beta_N \geq 1$.

Consequently, the covariance of $A(t)$ decays asymptotically at the lower rate, in this case, at the rate of the background LRD process since $\beta_F < \beta_N$:

$$\gamma_A(\tau) \sim \tau^{-\beta_F} \quad \text{as} \quad \tau \to \infty \tag{9}$$

If the On/Off process is LRD as well, e.g., the On and/or Off times are heavy-tailed, then asymptotically the larger Hurst exponent characterizes the path. In practice, the border of the scaling region depends on the actual shape of the covariances and the means m_A and m_F.

If there are more than one On/Off streams sharing the bottleneck buffer with a self-similar background traffic stream, $N(t)$ takes higher values than 1 as well. However, for the adaptivity of the aggregate it is sufficient to have at least one active connection as it was shown in Section IV-A. The aggregate traffic of multiple On/Off streams adapting to a background stream may be approximated by

$$A_{\text{aggr}}(t) = \Theta[N(t)]F(t) \tag{10}$$

where $\Theta(.)$ is the Heaviside-function, ($\Theta(x) = 1$ if $x > 0$ and 0 otherwise). $\Theta[N(t)]$ itself is also an On/Off process. If the On/Off processes are independent and they are exponentially distributed, then $N(t)$ forms a Markov process ($\Theta[N(t)]$ is the indicator process for the empty state of this Markov chain) and it is SRD.

The conclusion of this section is that if the end-to-end service uses TCP connections, then the traffic generated by the service is also adaptive, and in this case the adaptivity of the end-to-end service is sufficient to "propagate" LRD to other parts of the network. Moreover, if $N(t)$ is LRD, then the larger Hurst exponent $\max(H_N, H_F)$ is propagated.

C. TCP Connections Limited by the Receiver Window

In section II-A it was found that several high speed connections within the US do not show self-similarity in the tests. Common to all of these measurements is that the bandwidth delay product on the connection's path was large, and the TCPs traffic rate was limited not by free link capacity, but the advertised window of the receiver. Even in this case, it is most likely that the connection shares buffers with self-similar background traffic, but since TCP is not able to utilize the full bandwidth, the adaptation effect between the background traffic flow and the TCP connection is weak.

We performed a series of simulations that reproduce the above backbone scenario. We modified the previous simulation model: the link speed was increased to 4 Mbps and the round-trip link propagation delay to 100 ms. The buffer size remained 40 packets as in the previous simulations. The background traffic was a synthetic FGN traffic flow with $H = 0.8$, average 0.5 Mbps, and variance 0.5 Mbps. With these settings, the average link utilization was just 12.5 %, similarly to an overprovisioned backbone link.

We performed several simulations with a single TCP passing through the buffer. In consecutive simulations, the TCP receiver had different window sizes ranging between 10 to 45 packets. Each simulation lasted for 7200 sec, long enough to perform self-similarity tests. When the window is small, TCP is limited by the advertised window, thus it is not able to utilize the maximum available throughput. We expect that in this case, the impact of background traffic on the TCP flow's autocorrelation is weak. As the window increases, the average TCP rate also increases, finally, after some point, the average rate flattens out as the network load becomes the limit for the TCP throughput (see Figure 12a). From the end-user perspective, this is the desired operating point of the system.

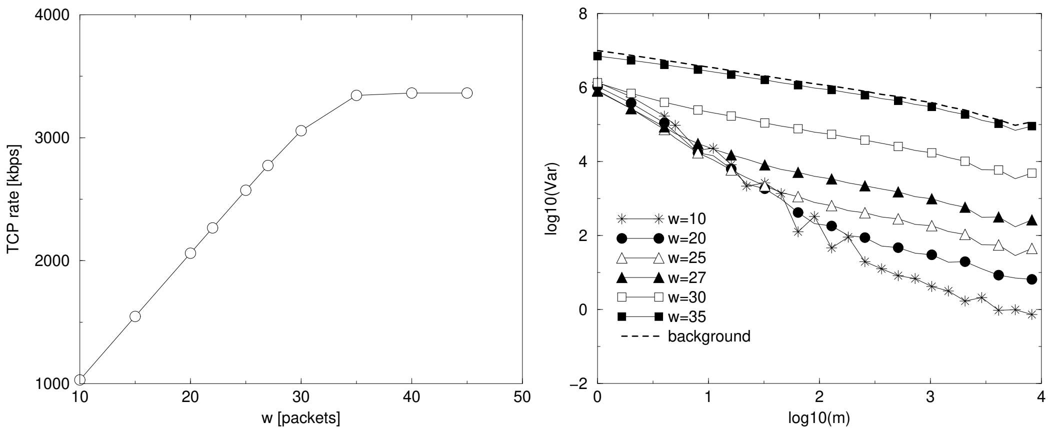

Fig. 12. a) TCP throughput vs. receiver window size in packets. b) VT plots for several windows sizes, the topmost graph is the VT plot of the FGN trace.

After each simulation run, we performed the tests for self-similarity. Figure 12b shows the results of the Variance-Time tests. It can be seen that all traces have the same slope asymptotically as the VT plot of the background traffic independently of the size of the receiver window. The difference is in the timescale where self-similarity scaling starts: as the window increases, scaling starts at smaller and smaller timescales. This phenomenon is the possible explanation why we observed self-similarity mostly in case of non-window limited scenarios in our real-life tests in Section II-A.

Today, there are still many operating systems and applications that do not support large windows. In case of high speed access speeds, the price is sub-optimal TCP performance. As high speed access technologies (e.g, DSL or cable modem) become more widespread, it is likely that operating systems and applications will demand larger windows, moving the cut-off timescale of propagation closer to the estimated analytic value derived in Section III-A.

Although larger windows cause increased variances in all timescales, see Figure 12b, the end-user perceived throughput also increases significantly in case of "fat pipes". If other performance parameters are taken into account, however, such as delay or loss, long-range dependence (and higher TCP rate variance) may be undesirable.

V. Spreading of Self-Similarity in the Network

Previously we analyzed the case when a TCP connection shares a single bottleneck buffer with self-similar background traffic, and it was only this bottleneck that affected the rate of TCP. In this section the network case is discussed.

Two aspects are analyzed. The first one deals with the case when the path of an adaptive connection passes through several buffers with self-similar inputs. These buffers are candidates to become bottlenecks occasionally during the lifetime of the connection. The second one investigates whether self-similarity can spread from one adaptive connection to the other causing widespread self-similarity in a network area. The presented results are intended to highlight the basic mechanisms, so the investigated scenarios are simplified for the ease of discussion.

A. *Discussion of the Multiple Link Case*

A wide area TCP connection usually spans 10-15 routers along its path, out of which there are usually several backbone routers with high level of aggregated traffic, see Figure 13. A TCP connection has to adapt to the whole path. The capacity of the end-to-end path, at time t, depends on which buffer is the bottleneck at this time. Because of traffic fluctuations, the location of the bottleneck moves randomly from one router to the other.

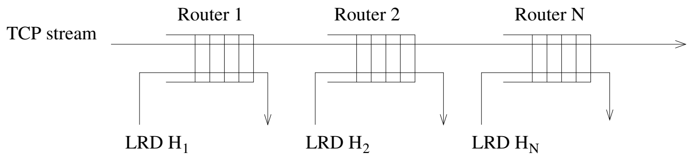

Fig. 13. A TCP connection traversing multiple hops with independent background LRD (H_i) inputs.

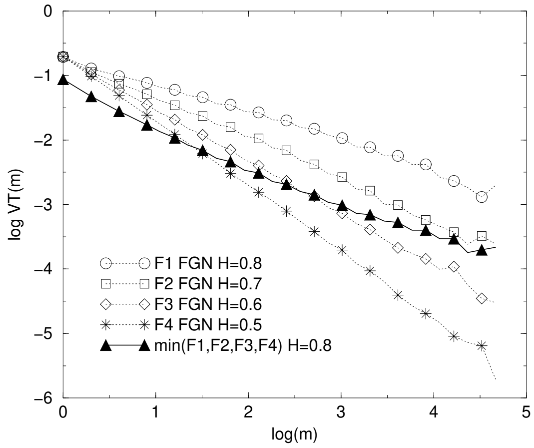

Fig. 14. Variance-Time plots of F_i FGN processes of $H = 0.8, 0.7, 0.6, 0.5$ respectively (identical mean rates and variance), and the end-to-end process $A(t) = \min_{i \in N} F_i(t)$. The end-to-end path is characterized by $H \approx 0.8$ asymptotically.

Assuming ideal end-to-end adaptivity, the rate of the adaptive TCP connection is equal to the free capacity of the bottleneck link at time t:

$$A(t) = \min_{i \in N} F_i(t) \tag{11}$$

where N is the number of links and $F_i(t)$ denotes the free capacity of the i^{th} link on the path.

For simplicity, assume that the crossing background LRD streams on the links are independent and the link at time t is either empty: $F_i(t) = 1$, or full: $F_i(t) = 0$. With this simplification the rate of the adaptive connection can be written as

$$A(t) = \prod_{i=1}^{N} F_i(t) \tag{12}$$

In the previous section it was shown that the product of independent LRD processes is also LRD and it is asymptotically characterized by the largest exponent:

$$\gamma_A(\tau) = \tau^{-\min_i \beta_i} \quad \text{as} \quad \tau \to \infty \tag{13}$$

Thus, in the multiple link case it is the largest Hurst exponent among the background LRD streams on the links that characterizes the TCP connection.

For a numerical example using more complex processes, four FGN background samples were generated with equal mean rates, but with different Hurst exponents, to model $F_i(t)$, $i = 1 \ldots 4$, see Figure 14. The end-to-end process $A(t)$, which is the minimum of the FGN processes, is asymptotically second-order self-similar, and it has the same Hurst exponent as the largest Hurst exponent among the $F_i(t)$ processes, i.e., the result is the same as in the simple full/empty case.

Another possible interpretation of (11) is that we consider the $F_i(t)$ not as rate processes, but as indicator processes of congestion. From the end user perspective it is important to analyze whether the network is able to support the expected service level requirements, for example, whether the file transfer rate degrades below an acceptable level or not. Let $F_i(t)$ be the indicator process of link i indicating whether the link is congested and it cannot support the expected service rate for the connection ($F_i(t) = 0$), or is not congested

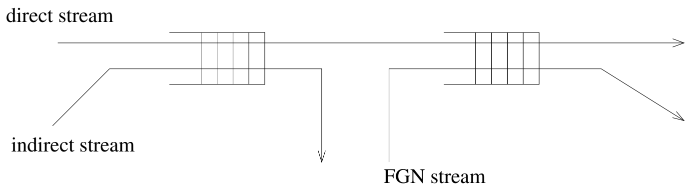

Fig. 15. Network model for the investigation of self-similarity spreading.

($F_i(t) = 1$). Thus, if the background congestion indicator processes are LRD, then it is the largest Hurst exponent that characterizes the end-to-end service characteristics of the investigated TCP connection.

B. Spreading of Self-Similarity among Adaptive Connections in Multiple Steps

So far, in all analyzed cases, adaptive traffic was in direct contact with self-similar background traffic. In this section it is investigated whether self-similarity caused by adaptation can be passed on to adaptive traffic streams that have *no direct contact with the source of self-similarity*. A few simple conditions are given as well. Assuming that our argument is valid, self-similarity can spread out from a localized area, consequently, strong self-similarity is balanced throughout a wider area of the network.

A simple network scenario is used for the investigation. An adaptive traffic stream (*direct stream*) shares a link with self-similar FGN ($H = 0.8$) background traffic. The *direct stream* is mixed with another adaptive stream on a second link, which itself has no direct connection with the FGN traffic (indirect stream), see Figure 15. The data rate of the direct stream is thus affected by two other streams, and also the two adaptive streams have an effect on one another. We are going to investigate the statistical properties of both the direct and the indirect streams.

Assume ideal adaptivity and max-min fairness among the adaptive streams. Also assume that the service rates of both links are equal (C). If the background stream was inactive, the bottleneck would be the first buffer and the adaptive streams would share simply half the service rate, both sending at a rate of $C/2$.

$$A_{dir}(t) = A_{indir}(t) = C/2 \tag{14}$$

In the presence of the FGN stream flowing through the second buffer, the rates can still remain $C/2$, unless it is the second buffer which becomes the bottleneck, i.e., when the capacity left unused by the FGN stream is $C - A_{FGN}(t) < C/2$. In this case the direct stream can use at most $A_{dir} = C - A_{FGN}(t)$, so the indirect stream can grab all remaining service capacity in the first buffer $A_{indir} = C - A_{dir} = A_{FGN}$. In short:

$$A_{dir}(t) = \min(C/2, C - A_{FGN}(t)) \tag{15}$$
$$A_{indir}(t) = \max(C/2, A_{FGN}(t)) \tag{16}$$

Calculation of the autocovariance of A_{dir} and A_{indir} is difficult because of the *min* and *max* operators. We consider two simple, extreme cases. In the first case, the rate of the background LRD stream is always greater than $C/2$, simplifying the expressions to $A_{dir}(t) = C - A_{FGN}(t)$ and $A_{indir}(t) = A_{FGN}(t)$, i.e., *spreading of self-similarity is ideal*. In the second extreme case the rate of background process is always smaller than $C/2$, leading to $A_{dir}(t) = A_{indir}(t) = C/2$, i.e., *self-similarity disappears* from both adaptive streams. These results has been verified by simulations as well.

The investigated scenario demonstrates the simplest mechanism of how adaptive connections may have effect on each other. We simulated a more complex scenario, where the synthetic *FGN* stream is replaced by an aggregate stream of randomly generated short TCP file transfers. The distribution of the file sizes is heavy-tailed. The direct and indirect TCP streams are also replaced by aggregates, but the file sizes within these aggregates are light-tailed.

The streams consist of $n_{heavy-tailed} = n_{dir} = n_{indir} = 100$ sessions. The file size distributions are Pareto distributions with the following parameters: the average file size is 40 kbyte for all streams, the average

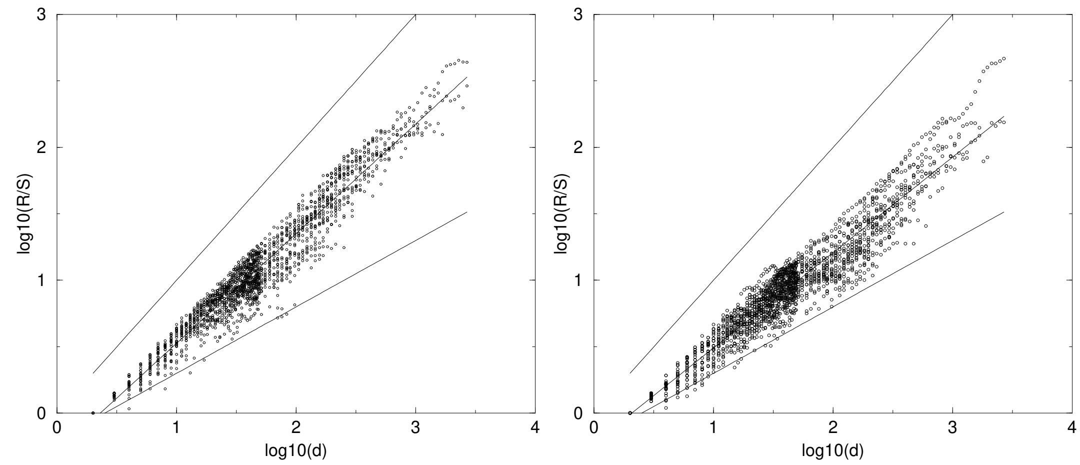

Fig. 16. a) R/S plot of *heavy-tailed* stream $H = 0.82$. b) R/S plot of *indirect* stream $H = 0.71$

waiting time between files is 20 sec. The shape parameters are $a_{heavy-tailed} = 1.1$ and $a_{dir} = a_{indir} = 3$ for both the file size and the waiting time distributions. With these parameters only one stream has heavy-tails ($a_{heavy-tailed} < 2$).

The results of the simulation experiment are shown in Figure 16. As suggested in [5] the traffic stream consisting of heavy-tailed file downloads is LRD ($H \approx 0.82$). In addition, the indirect traffic stream, although it was created using light-tailed distributions is LRD as well ($H \approx 0.71$). The cause is that long-range dependent fluctuations are propagated via the indirect stream.

Performing the previous experiment using different parameters, we have found that depending on the traffic mix, the spreading between indirect and direct streams can be strong but it can be also weak as well. In certain cases, spreading to an indirect stream does not happen at all, just like in the simple analytic example assuming ideal TCP flows and max-min fairness. The exact requirements for spreading are subjects for further study.

VI. Conclusions

It was demonstrated how a TCP connection, when mixed with self-similar traffic in a bottleneck buffer, takes on its statistical second-order self-similarity, and propagates scaling phenomena to other parts of the network. It is suggested that TCP's adaptation mechanism can be modeled as a linear system. The characteristic transform function is shown to be largely independent of the TCP version. An analytic approximation of the cut-off timescale of the propagation effect was presented, suggesting that the lower cut-off timescale depends on the end-to-end path properties. The calculations are also in agreement with the results from macroscopic TCP models published in the literature. The limitations of this property were analyzed. It was shown that TCP inherits self-similarity when it is mixed with self-similar background traffic in a bottleneck buffer through the transform function of the linear system. This property was demonstrated for both short and long duration TCP connections. We also investigated TCP behavior in a networking environment. We argue that the phenomenon of propagation is robust even when TCP connections pass several routers and also when the receiver window size limits the sending rate.

The mechanisms discussed in this paper can be considered as basic "building blocks" for a future wide-area traffic model; in real-life it is always their combined effect that we can observe. As thousands of parallel TCP connections continuously intertwine the Internet, the mechanisms described in this paper may provide us with a deeper insight into why significant and strong self-similarity is a general and widespread phenomenon in current data networks.

VII. Acknowledgements

Gabor Vattay would like to thank the support of the Hungarian Scientific Research Fund OTKA T032437 and T037903.

References

[1] P. Abry and D. Veitch. Wavelet analysis of long-range-dependent traffic. *IEEE/ACM Transactions on Networking*, 44(1):2–15, 1998.
[2] D. Avnir, O. Beham, D. Lidar, and O. Malcai. Is the geometry of nature fractal? *Science*, 279:39. 1998.
[3] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger. Long-range dependence in variable-bit-rate video traffic. *IEEE Transactions on Communications*, 43(2/3/4):1566–1579, February/March/April 1995.
[4] F. Brichet, J. Roberts, A. Simonian, and D. Veitch. Heavy traffic analysis of a storage model with long range dependent on/off sources. *Queuing Systems*, 23:197–215, 1996.
[5] M. E. Crovella and A. Bestavros. Self-similarity in world wide web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, December 1997.
[6] M. E. Crovella, M. S. Taqqu, and A. Bestavros. Heavy-tailed probability distributions in the world wide web. *Preprint: A Practical Guide To Heavy Tails: Statistical Techniques and Applications*, 1996.
[7] A. Erramilli, O. Narayan, and W. Willinger. Experimental queuing analysis with long-range dependent packet traffic. *IEEE/ACM Transactions on Networking*, 4(2):209–223, April 1996.
[8] A. Feldmann, A. C. Gilbert, P. Huang, and W. Willinger. Dynamics of IP traffic: A study of the role of variability and the impact of control. In *In Proceedings of SIGCOMM*, Cambridge, MA, USA, August 1999.
[9] A. Feldmann, A. C. Gilbert, and W. Willinger. Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic. *ACM Computer Communication Review*, 28:42–55, September 1998.
[10] D. R. Figueiredo, B. Liu, V. Misra, and D. Towsley. *On the Autocorrelation Structure of TCP Traffic* UMass CMPSCI Technical Report TR 00-55.
[11] M. Garrett and W. Willinger. Analysis, modeling and generation of self-similar VBR video traffic. In *Proc. ACM SIGCOMM*, pages 269–280, 1994.
[12] L. Guo, M. Crovella and I. Matta. *How does TCP Generate Pseudo-Self-Similarity?*, MASCOTS August 15-18, 2001, Cincinnati, Ohio, USA
[13] V. Jacobson. Congestion avoidance and control. In *Proceedings of ACM SIGCOMM*, pages 314–329, Stanford, USA, 1988.
[14] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of Ethernet traffic. In *Proc. ACM SIGCOMM*, September 1993.
[15] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1):1–15, 1994.
[16] M. Mathis, J. Semke, and J. Mahdavi. The macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communication Review*, 27(3), July 1997.
[17] S. Molnár, T. D. Dang, and A. Vidács. Heavy tailedness, long-range dependence and self-similarity in data traffic. In *Proc. 7th International Conference on Telecommunication Systems Modeling and Analysis*, Nashville, Tennessee, USA, March 1999.
[18] S. Molnár and A. Vidács. On modeling and shaping self-similar ATM traffic. In *Proc. 15th Int. Teletraffic Congress*, Washington D.C., 1997.
[19] S. Molnár, T. D. Dang Pitfalls in Long Range Dependence Testing and Estimation In *Proc. of GLOBECOM 2000,* , San Francisco, CA, USA, November 27-December 1, 2000.
[20] S. Molnár, T. D. Dang Scaling Analysis of IP Traffic Components. *ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, Monterey, CA, USA, Sept. 2000.
[21] I. Norros. A storage model with self-similar input. *Queuing Systems*, 16:387–396, 1994.
[22] UCB/LBNL/VINT Network Simulator - ns (version 2) http://www-mash.CS.Berkeley.EDU/ns
[23] K. Park, G. Kim, and M. Crovella. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *In Proceedings of the International Conference on Network Protocols*, pages 171–180, October 1996.
[24] K. Park, G. Kim, and M. Crovella. On the effect of traffic self-similarity on network performance. In *In Proceedings of the SPIE International Conference on Performance and Control of Network Systems*, pages 296–310, November 1997.
[25] V. Paxson and S. Floyd. Wide area traffic: The failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, June 1995.
[26] M. S. Taqqu, V. Teverovsky, and W. Willinger. Estimators for long-range dependence: an empirical study. *Fractals*, 3(4):785–788, 1995.
[27] M. S. Taqqu, W. Willinger, and R. Sherman. Proof of a fundamental result in self-similar traffic modeling. *Computer Communication Review*, 27:5–23, 1997.
[28] B. Tsybakov and N. D. Georganas. On self-similar traffic in ATM queues: Definitions, overflow probability bound, and cell delay distribution. *IEEE/ACM Transactions on Networking*, 5(3):397–409, June 1997.
[29] D. Veitch and P. Abry. *A statistical test for the time consistency of scaling exponents*. Preprint, 1999.
[30] A. Veres and M. Boda. The chaotic nature of TCP congestion control. In *Proc. IEEE INFOCOM*, Tel Aviv, Israel, April 2000.
[31] A. Veres, Zs. Kenesi, S. Molnár, and G. Vattay. On the propagation of long-range dependence in the Internet. In *Proc. ACM SIGCOMM*, Stockholm, Sept 2000.
[32] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson. Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ACM Transactions on Networking*, 5(1):71–86, February 1997.
[33] W. Willinger, M. S. Taqqu, and A. Erramilli. *A bibliographical guide to self-similar traffic and performance modeling for modern high-speed networks, In Stochastic networks: Theory and applications, Ed:F. P. Kelly, S. Zachary and I. Ziedins*. Clarendon Press, Oxford, 1996.