On the Autocorrelation Structure of TCP Traffic *

Daniel R. Figueiredo, Benyuan Liu, Vishal Misra, Don Towsley †
{ratton, benyuan, misra, towsley}@cs.umass.edu

Department of Computer Science
University of Massachusetts
Amherst, MA 01003-4610

Computer Science Technical Report 00-55

Nov 2, 2000

Abstract

Considerable attention has recently focussed on the characteristics of network traffic and related issues. In particular, the observation that data traffic in networks exhibits long range dependence or self-similarity has led to a myriad of research efforts in this area. In this paper we show that the TCP protocol can generate traffic with correlation structures over a finite range of timescales that are similar to characteristics observed in Internet traffic. This is independent of application-level characteristics and takes effect under a wide range of network conditions. We point out and analyze two of the mechanisms inside TCP that contribute to this: the exponential timeout back-off and the congestion avoidance. We support our claim using Markovian models that can be numerically analyzed. The models are validated against simulation results and measurements obtained in the Internet.

Keywords: self-similarity, TCP congestion control, Markov model

*A version of this paper was submitted for review

†This material is based upon work supported by the National Science Foundation. This research was also supported by CAPES (Brazil)

1 Introduction

The existence of non-degenerate correlation structures over a range of timescales in network traffic has been observed in a variety of network environments, such as Ethernet traffic [1], Wide Area Networks traffic [2], and World Wide Web traffic [3]. There have been recently many research efforts to investigate various aspects of this characteristics, for example, modeling techniques ([4, 5, 6, 7]), impact on network performance ([8, 9, 10]), and explanations for the presence of such correlation structures ([3, 11, 12, 13]).

Several efforts to explain this correlation structure of the network traffic have been made from different perspectives. Crovella et al. [3] conjectured that the strong correlated behavior in network traffic is caused by the heavy-tailed distribution of WWW documents sizes, the effect of caching, and human thinking time. All these causes come from the application or user level perspectives and can generate self-similar data traffic through the underlying transport protocols. In [11], the authors studied the chaotic behavior of TCP congestion control and concluded that the assumptions in [3] are not necessary to explain the origin of self-similarity in TCP data traffic. The authors argue that the "chaos" created by the TCP congestion control alone can generate self-similarity. The conclusions reached by the authors of [11] are puzzling as they report on TCP behavior that is periodic in nature, as the authors themselves point out (albeit with a possibly long period), and thus, not chaotic. Consequently, this behavior cannot generate self-similarity in network traffic. The work also raises the question as to how and why TCP congestion control exhibits the long term temporal correlations that they report.

In this paper we provide another possible explanation for the non-degenerate correlation structure in network traffic. In particular, we analyze the TCP protocol and show how its mechanisms can generate such traffic patterns. These mechanisms do not by themselves give rise to self-similar traffic. However, we show that the traffic induced by these mechanisms has a similar correlation structure within similar timescales to that of measured Internet traffic of previous studies. Since most applications use TCP as their data transport protocol and the majority of the network traffic is carried by TCP, one can easily misinterpret the origin of these correlation structures. The efforts that have tried to explain self-similarity from an upper level perspective can just as well be explained by TCP itself. They are not necessary in order to generate the correlation structures in the network traffic if TCP is used as the transport protocol. As a result, we point out that one should be careful when claiming to identify the cause of correlation structures observed in network traffic.

We now describe two different related works as follows. In a pioneering paper [12], Manthorpe, et al. point out that real network traffic is not strictly self-similar, but exhibits a correlation structure only over a finite range of timescales. They then introduce the terms pseudo self-similarity and local Hurst parameter to describe the correlation structure of data traffic over a this range of timescales instead. A process is pseudo self-similar if the variance of the aggregated process over a finite range of timescales decays linearly in a log-log plot. The local Hurst parameter is defined to be a simple function of the slope in this range of timescales. Formal and precise definitions of a pseudo self-similarity and local Hurst parameter are given in [6] and [12], respectively. In this paper, we will adopt these terminologies to refer to the correlation structure of the network traffic. This will allow a comparison between the local Hurst parameter and the Hurst parameter previously used in the literature. Also in [12], the authors discover through simulation that under certain conditions, TCP protocol can generate pseudo self-similar traffic. They claim that the combination of TCP with the heavy-tailed distribution of round trip delays and the wide range of transmission rates seen in real networks are responsible for introducing pseudo self-similarity in data traffic.

In another related work [13], Guo et al. claim that the TCP congestion control mechanism generates heavy-tailed off periods in the traffic transmission pattern, which then introduces long-range dependence in the overall traffic. The authors try to explain the self-similarity of TCP using the framework of multiplexing ON/OFF processes with heavy-tailed ON or OFF period, and also in terms of its chaotic behavior. We note that formally, the off periods in the TCP congestion control mechanism are not heavy-tailed, since the maximum off period is bounded by the maximum timeout value. Hence it is erroneous to conclude that the TCP congestion control mechanism produces self-similar traffic in the absence of other mechanisms. A limitation of their analysis is that the pseudo self-similarity only occurs when the loss probability is between 1/8 and 1/4. This is not consistent with observations of pseudo self-similarity in TCP traffic under lower loss probabilities, as was also observed by the authors.

In this paper we show that TCP protocol itself, in particular, its exponential back-off and congestion avoidance mechanisms, gives rise to pseudo self-similar traffic over a range of timescales. We construct separate Markovian models for each of these mechanisms and present analytical results that support our claim. Simulation is also used to support our claim and validate the models. We show that the local Hurst parameter predicted by the models for typical network packet loss probabilities lies in the same range as the values obtained from previous measurements, for example [1, 3, 14]. Our analysis applies to a wide range of loss probabilities, including very low loss probabilities. We further demonstrate that the analysis of real TCP traces agrees with the range of timescales predicted by the analytical models. Moreover, under certain network conditions, we show that the local Hurst parameters obtained from the model are similar to the ones calculated from real traces. We relate the parameters of the timeout and congestion avoidance mechanisms to the local Hurst parameter and the timescales each mechanism exerts. Finally we provide a possible argument for the pseudo self-similarity of the aggregate TCP flows in real networks based on our model analysis.

The rest of the paper is organized as follows. Section 2 describes the TCP timeout and congestion avoidance mechanisms and presents two Markov chains to model them. Section 3 presents the mathematical framework used to analyze the models, and demonstrates that both models give rise to pseudo self-similar traffic over a certain range of timescales. Section 4 provides simulation results to validate our models and support the analytical results. Discussions on model parameters and remarks on the aggregation of TCP flows are given in Section 5. Finally, in Section 6 we present our conclusions.

2 TCP Timeout and Congestion Avoidance Models

In this section we describe the behavior of TCP timeout and congestion avoidance mechanisms via Markov chains. Our purpose here is not to provide a detailed or complete model of the TCP protocol, but rather to highlight and investigate the timeout and congestion avoidance mechanisms of TCP using different models to demonstrate that both features of TCP can lead to strong correlation over a finite range of time scales.

TCP is a network transport protocol that provides reliable end-to-end data communication [15]. It is a window based protocol and performs flow control and congestion control, by regulating its sending window size through an additive increase/multiplicative decrease mechanism. TCP includes different mechanisms inside its complex suite, among which Congestion Avoidance and Timeout Back-off can have an important effect in the traffic pattern.

2.1 Timeout (TO) Model

TCP uses a timeout mechanism to achieve reliable transmission of data and to avoid congestion collapse [15]. Each time a packet is sent out from the source, TCP starts a retransmission timer and waits for the acknowledgement from the receiver. TCP keeps a run time average of the round trip time (RTT) and sets the duration of its timer (RTO) based on the average and variance of the RTT estimate. If the packet is lost, the sender will not receive the acknowledgement for that packet and the timer will eventually expire, if no triple duplicate acks are received. Upon a timeout the packet is retransmitted and the sending window is set to one. This reduces the sending rate and hopefully relieves the network congestion. If the congestion continues and the retransmission is lost, TCP doubles the current timeout value of the next retransmission to further reduce the sending rate and adapt to the congested network. This exponential back-off continues for each potential retransmission after the first timeout occurs. When the subsequent segment of data is successfully transmitted, TCP recomputes the RTT estimate, resets the timeout value accordingly and resumes the normal mode of operation. In most TCP implementations, the timeout is backed off to a maximum of 6 times, giving timeout values that are equal to 64 times the original timeout. After the timeout value reaches its maximum, it does not further increase and remains the same if loss continues.

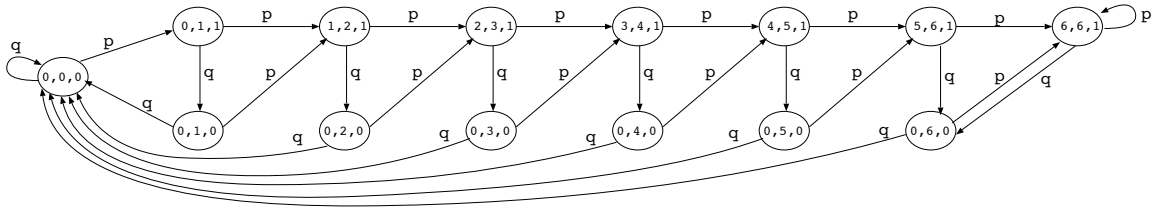

Figure 1: TCP timeout model

For both the TO model and the following congestion avoidance (CA) model, we assume the packet loss is described by a Bernoulli process with parameter p and that acknowledgement packets are never lost in the network. Based on the TCP timeout mechanism, we can construct a discrete time Markov chain illustrated in Figure 1 to model the timeout behavior. The single parameter of the model is the packet loss probability p ($q = 1 - p$). The detailed explanation of the chain is given as follows.

- **Definition of the States:** A state in the Markov chain is represented by the tuple (T, E, R), where T indicates the time ($RTT * 2^T$) until the next packet transmission; E indicates the current value of the back-off exponent; and R indicates if the packet being sent is a retransmission ($R = 1$) or a new packet ($R = 0$). Note that both T and E range from 0 to 6, and that the time spent in each state is always a multiple of the RTT.
- **Transition probability matrix:** Once TCP enters the back-off mode of operation, if a retransmitted packet is lost, then the time until the next transmission is given by the current timeout value. If a packet is successfully transmitted, then the time until the next transmission is given by a single RTT. Note that it takes two consecutive successful packet transmissions in order for TCP to resume its normal mode of operation and reset its back-off exponent. Moreover, every time a packet is lost, the back-off exponent is increased by one to double the timeout value. Figure 1 illustrates all the possible transitions in the model.

2.2 Congestion Avoidance (CA) Model

We now focus on the congestion avoidance mechanism and ignore the presence of timeouts to better capture the behavior of the additive increase/multiplicative decrease mechanism of the window size. By changing the congestion control window size, TCP regulates the amount of data it injects into the network and, thus, adapts to different network conditions. TCP exits the slow start phase and enters congestion avoidance after the window size exceeds a certain threshold, which is set to one half of the maximal window size value that was reached before the previous congestion. In the congestion avoidance phase, the window size increases by one packet when all packets in the current window are successfully acknowledged. Hence, the window size grows linearly during the congestion avoidance phase. When a packet is lost, TCP decreases the size of the current congestion window to reduce the amount of traffic it injects into the network. In most versions of TCP that are currently deployed, such as TCP Reno and TCP Sack, the window size is reduced by half if three duplicate acknowledges for the same packet are received. If a timeout occurs, the window size is reduced to one and, after resuming from the timeout mode of operation, TCP starts the window growth cycle again.

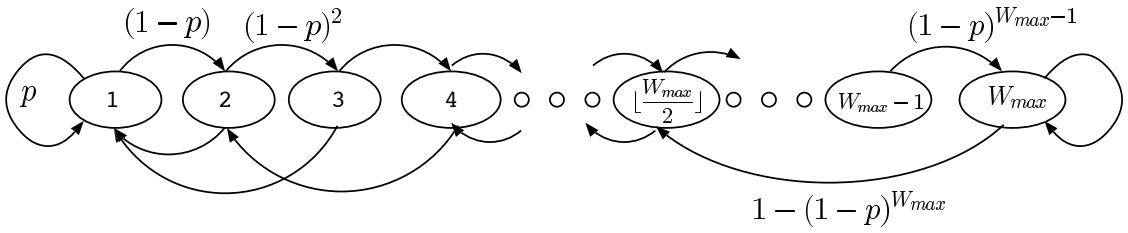

Figure 2: TCP congestion avoidance model

In the CA model, we only consider window reduction events that are due to triple duplicate ACKs, ignoring the timeout events and the slow start phase. Hence, the window grows linearly when no loss occurs and is reduced to half upon congestion. We also make the common assumption that the RTT is larger than the time required to send all packets in a window [16]. The model is very simple and has two parameters: the maximum window size (W_{max}) and the packet loss probability (p). The Markov chain for this model is illustrated in Figure 2 and is explained as follows.

- **Definition of the states:** The state is represented by a single variable that indicates the current window size of the TCP sender. This variable ranges from one to the maximum window size (W_{max}). The initial state depends on the value of the CA threshold, and can range from 1 to $\lfloor W_{max}/2 \rfloor$. Note that the time spent in each state is equal to one RTT.
- **Transition probability matrix:** During the CA phase, the window size increases by one packet when all packets in the current window have been successfully acknowledged, provided that the current sender window lies below its maximum value. Thus, the transitions occur after the window is fully transmitted. The probability that all w packets of the current window are successfully acknowledged is $(1-p)^w$. Once the window size reaches its maximum value, it remains at this value until a packet is lost. Whenever a packet is lost, the TCP sender reduces the window size to half, causing a transition from state w to state $\lfloor w/2 \rfloor$. The probability that at least one packet is lost among the w packets in the current window is just $1-(1-p)^w$. Therefore,

$$p_{i,j} = \begin{cases} (1-p)^i & j = i+1 \leq W_{max} \text{ or } j = i = W_{max} \\ 1-(1-p)^i & j = \lfloor i/2 \rfloor \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

3 Model Analysis

In this section we first describe the mathematical framework used to analyze the correlation structure of both the TO and CA models. We then demonstrate various analytical results that support our claim of sustained correlation in TCP traffic over a finite range of timescales. To construct and analyze both models, we use the TANGRAM-II modeling tool [1] [17], which allows us to easily obtain, among other measures, the autocorrelation function for each model [18]. Using the autocorrelation function, we then perform the numerical procedure described below to obtain the local Hurst parameter estimate plot. With this plot, we can visually inspect the timescales over which the model generates pseudo self-similar traffic. For simplicity, we let the value of the RTT be 1 in both the CA and TO models.

3.1 Estimation of local Hurst parameter

A crucial step in validating our model is demonstrating the fact that it generates pseudo self-similarity. A technique frequently used in literature for estimating the (local) Hurst parameter is that based on wavelet analysis. Wavelet based estimators [19] are computationally very efficient, and robust in the presence of non-stationarities in extracting out the "self-similar" nature of a given signal. The plot most frequently used to both demonstrate and estimate self-similarity in a given trace or time series is the energy scale plot. In these plots, the logarithm of the variance of the wavelet coefficients of the analyzed signal in a particular scale, is plotted vs that particular scale. The x-axis in these plots is on a $\log_2$ scale. Linear regions in these plots indicate "self-similarities", and the slope of the linear region gives an estimate of the (local) Hurst parameter. The variance of the wavelet coefficients measures the "energy" in the signal in the given scale. If we go back to the analysis presented in [20], [19], we see that this "energy" is really an estimate of the power spectral density of the process about a frequency determined by the particular scale. The frequency progression in scales is logarithmic, i.e. the j^{th} wavelet space, denoted by the scale 2^{-j}, represents a frequency $2^{-j}\nu_0$, where ν_0 is determined by the sampling rate of the time series and the particular choice of the analyzing wavelet. The (local) Hurst parameter is then estimated by first estimating the local polynomial decay coefficient of the power spectral density ($\approx 1/f^{\alpha}$). Recall that in the energy scale plots the variance ("energy") is plotted on a log scale, and since the scales themselves represent a logarithmic progression of frequencies, the variance scale plot is essentially a double log plot of the power spectral density. Linear regions in such a plot represent polynomial decay of the power spectral density with a constant coefficient α. This α is related to the local Hurst parameter via the relation $H = (\alpha+1)/2$. Thus, the wavelet based estimators calculate the Hurst parameter implicitly, with the intermediate step of estimating the slope of the power spectral density on a double log plot.

The power spectral density of a (stationary) stochastic process is defined as the Fourier transform of its autocorrelation function. Since we have a technique of calculating the autocorrelation functions of our Markovian models numerically, we can compute the power spectral

[1] TANGRAM-II is publicly available at http://www.land.ufrj.br

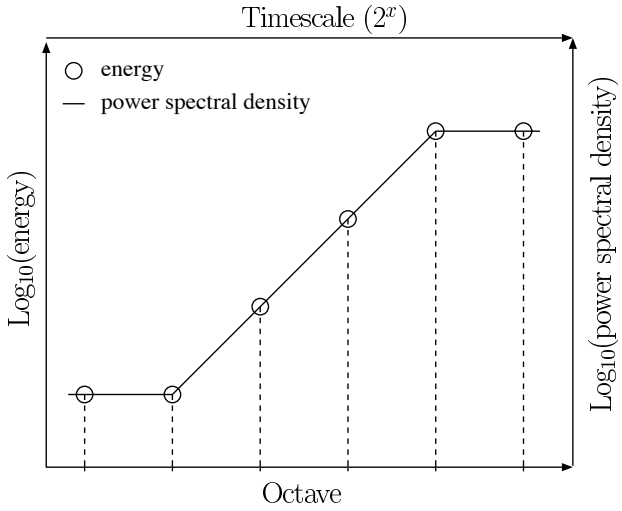

Figure 3: Relating the variance scale plot to the double log plot of power spectral density

density of the process by taking a discrete time Fourier transform of the autocorrelation function. We can then plot the resultant power spectral density on a double log plot, revealing pseudo self-similarities and we can identify local Hurst parameters by calculating the slopes of the linear regions of those plots. We use wavelet based estimators later on in the paper to estimate local Hurst parameters of time series obtained from experiments and simulations, and to maintain consistency of presentation we plot the analytically obtained power spectral density using similar axes. That is, the x-axis is plotted using a $\log_2$ scale of decreasing frequencies (recall that a wavelet space j corresponds to a frequency of $2^{-j}\nu_0$, and the energy scale plots have wavelet spaces and thus timescales increasing along the x-axis), and the y-axis plots $\log_{10}$ of the power spectral density (the energy scale plots use $\log_{10}$ of the energies, which in turn are estimates of the power spectral density). We label the x-axis in the power spectral density plots by timescales rather than frequency, to both maintain consistency with the wavelet plots as well as to retain the intuitive nature of plots, in which quantities increase from left to right (timescale is the inverse of frequency, hence decreasing frequencies represent increasing timescales). Thus, the reader should visually interpret these plots the same way as the energy-scale plots commonly used in literature and elsewhere in this paper. An illustration is shown in Figure 3. Interpreting the left and bottom labelling as the y and x axes, the energy-scale plot is drawn by connecting the discrete points corresponding to the logarithm of the energies in scales 2^{-j} by straight lines. Those energies and scales map on to power spectral densities and frequencies (timescales) and interpreting the right and top labelling as the axes, we obtain a smooth curve which goes along the points plotted in the energy-scale plot.

3.2 Analysis of the TO model

We begin by presenting the results of the TO model. Figure 4 illustrates the local Hurst estimates obtained under different loss probabilities. The x-axis represents the timescales in units of RTT and in $\log_2$ scale (i.e., $RTT * 2^x$). The y-axis represents the log of the power spectral density as discussed in the previous section. From the plot, we observe that all curves have a linear increasing part and then gradually become a flat horizontal line. As described in the previous subsection, the local Hurst parameter can be related to the slope of the linear increasing part of the curve by $H = (\alpha+1)/2$, where α is the slope of the linear part. The values of H for different packet loss probabilities can be observed in Figure 4, and they range from 0.5 to 0.8. These

values are consistent with the results obtained from previous studies of real network traffic [1, 3], where the value of H (greater than 0.5) is used to indicate the degree of self-similarity in traffic. Note that as the loss probability increases, the local Hurst parameter also increases, provided the loss probability is kept under 30%. An intuitive explanation for the behavior is that, as the loss probability increases, the system is more likely to move to higher values of the back-off exponent (right-hand side of the model), instead of mostly staying in the first few states when the probability is low. This is likely to introduce more correlation in the generated traffic. From the results in Figure 4, we also observe that, as the loss probability increases, the timescales at which the process exhibits pseudo self-similarity increases. For low loss probabilities the model exhibits little correlation structure, which can be observed by the presence of a flat horizontal line in the local Hurst parameter plot.

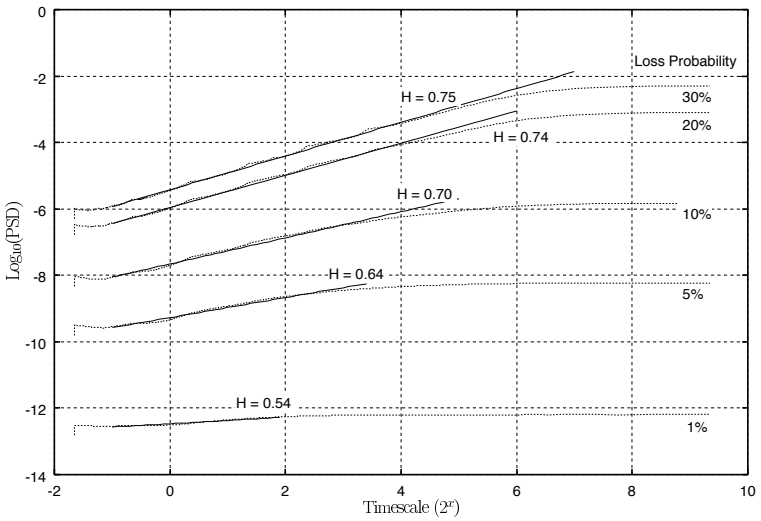

Figure 4: Analysis of the Timeout model

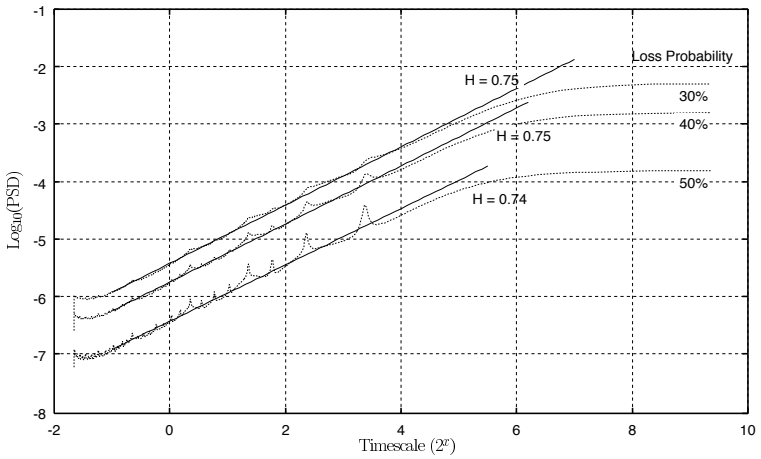

Figure 5: Analysis of the Timeout model under high loss probability

One could ask what would happen if the model was exposed to severe packet loss conditions and conjecture that the correlation structure would be more pronounced. Figure 5 illustrates this situation and presents the results seen by the model subjected to high loss rates. Note that as the loss probability increases past 30%, the time scales over which the model generates pseudo self-similarity decreases slowly. However, the local Hurst parameter obtained from the slope of the linear rise of the curves remain around 0.75. Intuitively, when the loss probability is high, the dynamics of our model drift to the states with high inter-packet delay. However, due to the

finite state space of the model, the largest inter-packet delay is $RTT * 2^6$. The peaks in the linear rise of the local Hurst parameter curves are due to periodic behavior of the autocorrelation function. The autocorrelation function has periodic oscillations with period of 64 RTT, which is caused by the structure of the Markov chain. The reason that these peaks do not appear in wavelet analysis of measured traces is that wavelet analysis uses an average value of the energy around each octave.

3.3 Analysis of the CA model

We now focus on the analysis of the CA model. Figure 6 illustrates the results obtained with the model under different loss probabilities. Here the maximum window size, W_{max}, was set to 30. This is a typical value in TCP connections in real networks.

From the results, we observe that all local Hurst estimate curves have a linear increasing part, similar to the previous model. Again, the slope of this linear part can be related to the local Hurst parameter. For all loss probabilities analyzed, the values for H range from 0.5 to 0.8, and are larger for smaller loss probabilities. It's worth noting that in all of the measures obtained, the local Hurst estimate plot stops increasing at a given timescale, and becomes a flat horizontal line. This illustrates that the process is pseudo self-similar only over a certain range of timescales. This range of timescales also increases as the loss probability decreases.

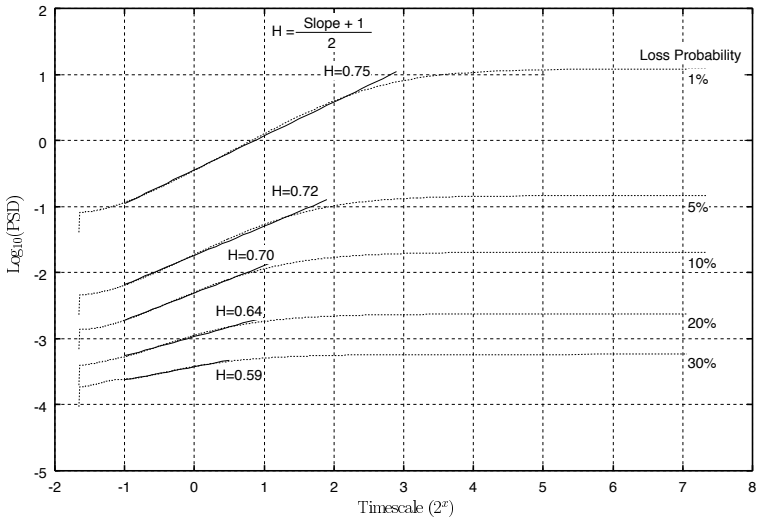

Figure 6: Analysis of the Congestion Avoidance model

An interesting question is what happens to the range of timescales and the local Hurst parameter when the model is exposed to very low loss probabilities. Figure 7 illustrates the results of models with different low loss probabilities and different values of W_{max}. We start by inspecting the results when the loss probability is 1%. In this case, the W_{max} parameter has very little impact, for values of 30, 60 or 120. All three values exhibit exactly the same local Hurst estimate curve, as shown in Figure 7. As the loss probability decreases to 0.5%, we notice that the values for W_{max} of 60 and 120 are indistinguishable, but the value of 30 cannot achieve that larger timescale, over which the traffic exhibits pseudo self-similarity. Decreasing the loss probability even further to 0.1%, we see that the timescales for W_{max} of 60 and 120 differ, and that the latter has a longer range of timescales at which pseudo self-similarity exists.

Intuitively, this behavior arises from the fact that model tends to drift to states with a larger

window size when the loss probability is low. But since the chain is finite, this drift is inherently limited. Thus, larger values of W_{max} allows the model to perform this drift to rightmost states of the chain. However, this drift only occurs with very low loss probabilities, since in this case the probability that the window increases past some large value is not negligible.

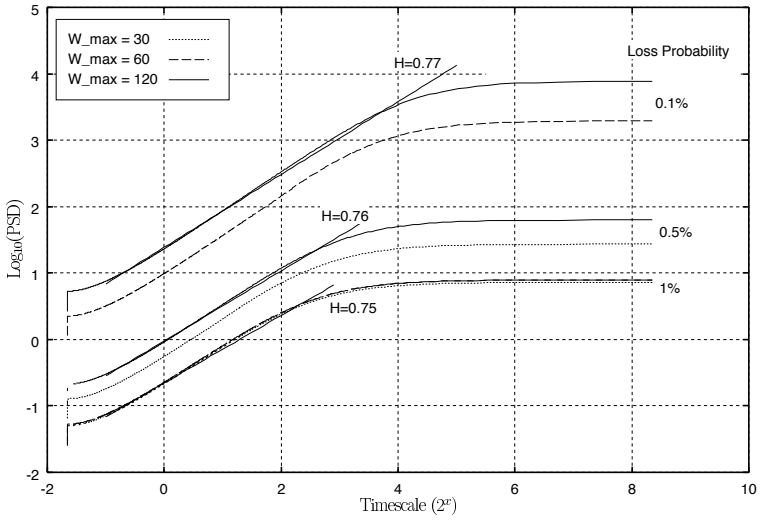

Figure 7: Analysis of the Congestion Avoidance model under very low loss probability

We finish this section by outlining the behavior of both models under different loss probabilities. The local Hurst parameter and the range of timescales over which the TO model exhibits pseudo self-similarity increases as the loss probability *increases*. In contrast, the CA model predicts that the local Hurst parameter and the range of timescales increases as the loss probability *decreases*. A final comment is that in the general case (i.e., $W_{max} = 30$), the range of timescales of the TO model is much larger than the range of timescales of the CA model. Thus, these two mechanisms combined play important roles at different timescales in the TCP protocol.

4 Simulation Scenario

In this section we use simulation to support our claim that the TCP internal mechanisms can generate pseudo self-similar data traffic over a finite range of timescales. We also validate both TO and CA models with results obtained from simulation.

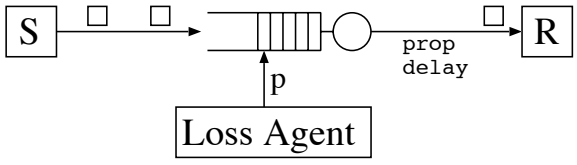

Figure 8: Simulation scenario

All simulations in this work were performed using the NS-2 simulator [21]. A simple network topology, consisting of a single source, a queue and a receiver, was simulated to investigate the traffic correlation structure generated by a single TCP session over a lossy link. A single TCP session allows us to easily control the loss rate of the packet flow. We note that in a multiple session scenario, a specific TCP flow will have similar behavior to the single session case. This holds if the loss process applied to both flows is the same. Thus, it suffices to investigate the single session case.

Figure 8 illustrates the model being simulated. The sender object acts as an infinite source; thus it always wants to send as much data as possible. The queue object stores packets from the source and forwards them to the receiver. We assume that the queue has an infinite buffer space to minimize the correlation of packet losses. The loss agent is attached to the queue and drops packets randomly at the time of their arrival according to a Bernoulli process with parameter p. The receiver simply collects the data packets sent. The data transfer between sender and receiver is done using the SACK version of the TCP transport protocol [22]. We assume that the *ack* packets sent by the receiver are never lost. The parameters varied during our experiments are the loss probability, the link propagation delay (which is important to determine the RTT), and W_{max}, the maximum window size of the TCP protocol. All other parameters were kept constant throughout the experiments.

In all our simulation experiments, a single TCP flow is investigated and there is no background traffic. All packets traversing the queue belong to the TCP session. In our experiments, the simulations were usually executed for 1 or 2 hours of simulation time, corresponding roughly to the transmission of 10000 to 300000 packets, but this strongly depends on the loss probability.

After a simulation run, two time series of packet versus time can be collected from the traces. One time series corresponds to the link between the sender and the queue (before the loss agent), and the other between the queue and the receiver (after the loss agent). In all results presented, we analyze the first time series. For purposes of our study, we verified that both time series have very similar behavior, thus our conclusions also hold for the latter time series.

We used the Wavelet Analysis Method to analyze the time series generated by the simulation. Before performing the wavelet analysis, the time series was aggregated into bins with size smaller than one RTT. The aggregated time series was analyzed using the publicly available tool developed by Veitch and Abry [19]. It's important to note that the tool generates a wavelet graph where the x-axis represents timescales ($AGR * 2^x$, where AGR is the aggregation level of the time series), and the y-axis represents the "energy" at that timescale with a 95% confidence interval.

4.1 Observations from Simulation

Figure 9 shows the results of the wavelet estimator for the single TCP session model shown in Figure 8 with different loss probabilities. The link propagation delay was set to 80ms, the link bandwidth to 1000 packets/sec and W_{max} was set to 30. Notice that the time to transmit the maximum window (30ms) is smaller than the RTT (around 160ms), which agrees with previous assumption. The results shown are for TCP SACK, which is the deployment trend of TCP in the Internet []. However, we also simulated and analyzed TCP Tahoe, which generated results very similar to the ones shown here. Note that as the loss probability increases, the timescales over which pseudo self-similar traffic is present increases. We also note that even for a low loss probability, the simulation results still exhibit some correlation structure for a small range of timescales. These two observations will be further discussed in the next section.

Our first observation is that a single TCP flow can exhibit pseudo self-similarity on the traffic over a finite range of timescales. This is observed in Figure 9. The same observation was also made in [12] where the authors use a different simulator and a more complex network model (i.e., more protocol layers). In [13], the authors also made similar observation through NS simulations, using a sequence of short-lived (15 packets) TCP flows. We point out that the TCP TO and CA mechanisms are responsible for this pseudo self-similarity in the data traffic

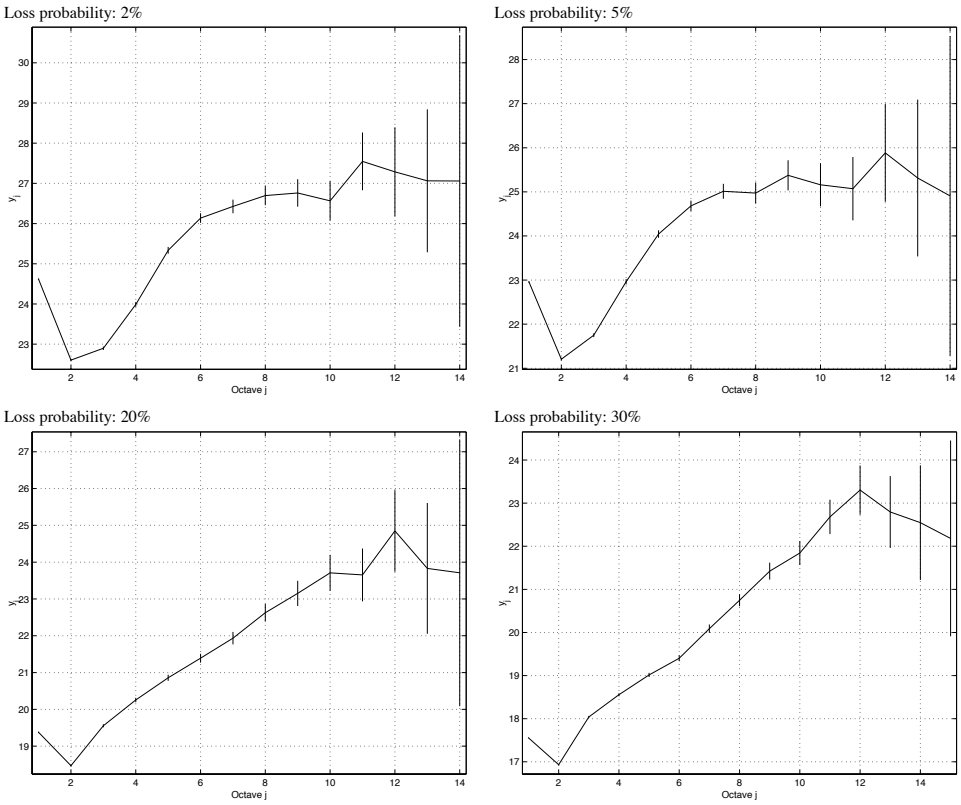

Figure 9: Wavelet analysis of the simulation traces

over some range of timescales.

4.2 Model Validation

In order to verify that the models proposed in section 2 actually capture the general behavior of the TO and CA mechanism, we validate some measures obtained from the models with simulation results. We start by validating the TO model and subsequently we present a validation for the CA model.

Using the TO model, we can compute the limiting probabilities of the value of the timeout back-off exponent just before a packet is transmitted. Let B be a a random variable denoting this value. The probability density function, $P[B = k]$ $k = 0, \ldots, 6$ can be obtained by solving the model numerically and analyzing the traces from the simulation. The values for B can be easily obtained in the simulation since TCP maintains this variable as part of its current state. Note that the timeout mechanism is well captured by the random variable B, and its value is independent of other internal mechanisms.

Figure 10 presents the results obtained for the model and the simulation with a 95% confidence interval. The simulation setting had $W_{max} = 30$, and propagation delay equal to 80ms. A total of 10 simulation runs were executed, each run lasting a few hours in simulation time, in order to obtain accurate estimates. The results from the model match the simulation results quite well for higher values of loss probability. But as the loss probability decreases, the simulation results starts to starts to deviate from the model.

An explanation for this small discrepancy is that for low packet loss probabilities, it is likely

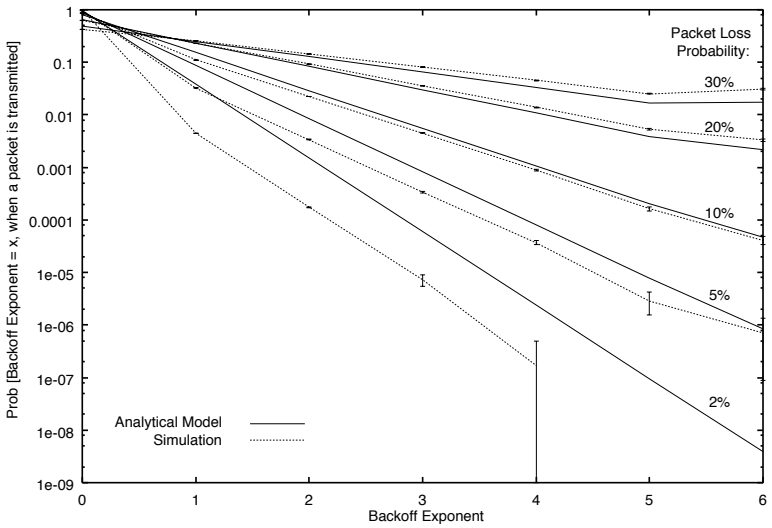

Figure 10: Comparison between TO model and simulation results

Loss probability	$E[W], W_{max} = 30$	$E[W], W_{max} = 60$	$E[W], W_{max} = 120$	$E[W]$, classical formula
30%	1.9	1.9	1.9	2.2
20%	2.4	2.4	2.4	2.7
1%	12.4	12.4	12.4	12.2
0.1%	26.6	38.6	40.8	38.7

Table 1: Expected value for window sizes in the CA model

that a lost packet will not lead to a timeout in the simulation, while in the model every packet loss leads to a timeout. Note that in the simulation (as in the real TCP protocol) packets can be retransmitted due to triple duplicate acks, which occur more frequently under low loss probability. Thus, the model overestimates the limiting probabilities for values of B greater than zero. The discontinuity and the large confidence intervals for the low packet loss probability are due to the fact that for low loss probabilities, the probability that B assumes a larger value is very small (i.e., 10^{-6}), and cannot be easily obtained through traditional simulation techniques.

The validation of the CA model with simulation results is more subtle, since the inner mechanisms of the TCP protocol that affect the window size cannot be easily isolated (i.e., slow start, TO mechanism, fast retransmit). Therefore, the validation with NS simulation is difficult and not meaningful, since many functionalities of TCP would have to be disabled in order to obtain a correct and fair estimate of the window size. To cope with this problem, we investigate the expected window size when the model is in steady state, namely, E[W]. This result can be compared with the classical TCP formulas for throughput and expected window size: $E[W] = k/\sqrt{p}$, where p is the packet loss probability and k is a constant between 1 and 1.5 [23, 24]. We use the traditional value of $k = \sqrt{3/2} = 1.22$. Note that this classical result models solely the CA phase of the TCP protocol and ignores the presence of slow start, timeouts and other enhanced features. The formula also assumes that W_{max} will be large enough not to interfere or limit the expected value. Since the model isolates the CA mechanism from the complete protocol, we expect the model to agree with the classical formula.

In Table 1 we present the values of $E[W]$ obtained from our CA model under different loss probabilities and with different values for W_{max}, together with the results from the classical formula above. Note that the model and the classical formula yields similar results ($W_{max} = 30$)

for various loss probabilities. However, for very small loss probabilities, W_{max} parameter has a significant impact, since it establishes an upper bound for the window size. But by increasing the W_{max} parameter, out model is able to capture the value of $E[W]$ as predicted by the classical formula. In all cases where the maximum window size was large enough, the model presented results very similar to the classical $E[W]$ formula.

The expected window size will also have a significant impact on the range of timescales over which the model generate pseudo self-similar traffic. This will be discussed in detail in the next section. For now, note that for very low loss probabilities, $E[W]$ depends on the value of W_{max} parameter, and that decreasing the loss probability increases the $E[W]$. This behavior is expected and can be observed both from the model and the classical formula results.

5 Discussions

In this section we first investigate different aspects of the models contributing to the shape of the local Hurst estimate curve. In particular, we identify the timescales at which the models produce pseudo self-similar traffic. We then provide a possible explanation for the pseudo self-similarity observed in real network traffic based on our results for a single TCP flow.

We start by revisiting the CA model and relate the knee of the local Hurst estimate curve where the linear increase starts to flatten to a horizontal line. Figure 11 presents the same results obtained in Section 3 but with a cross at the end of the range of timescales over which the local Hurst estimate curve shows linear growth. These points were approximated by $\log_2(RTT * E[W])$, with RTT equal to 1. We observe that these cross points do establish a timescale beyond which the traffic correlation structure disappears. This observation holds true for all the results obtained from our model. Since $E[W]$ from the model agrees with the classical TCP formula, as shown in Section 4, the classical formula can be used to predict the timescales at which one expects to see pseudo self-similar traffic, provided that W_{max} is large enough not to limit the TCP window growth.

Interestingly, in [25], the authors analyze the congestion avoidance behavior of TCP from a control theoretic standpoint. One of their findings is that the characteristic timescale [2] of TCP is directly proportional to $E[W]$. Our observations are consistent with that result and could very well be explained by it.

A similar observation can be drawn from the TO model. Revisiting the results presented in section 3, we would like to identify the range of timescales that the local Hurst estimate curves grows linearly, and relate this to the parameters of the model. From observation, we notice that for high loss probabilities, this value is $RTT * 2^6$, which is the largest timescale associated with the model. Note that the probability of an inter-packet delay being greater than $RTT * 2^6$ is zero. Thus, all events in the model occur within this timescale, and this value can be seen as an upper bound. However, in the TCP protocol, the exponential back-off uses the RTO value as its basic unit, instead of the RTT value as used in our model. Therefore, the largest timescale associated with the pseudo self-similarity behavior is $RTO * 2^6$. It has been observed [16] that the RTO value ranges from 2.5 to 10 times the RTT value. We are still trying to obtain a simple relationship between the parameters (i.e., loss probability) and the range of timescales over which the pseudo self-similarity is present in the data traffic. Until we do so, we use this intuitive upper bound for the range of timescales that the TO mechanism generates pseudo

[2] they refer to it as the "pole of TCP dynamics"

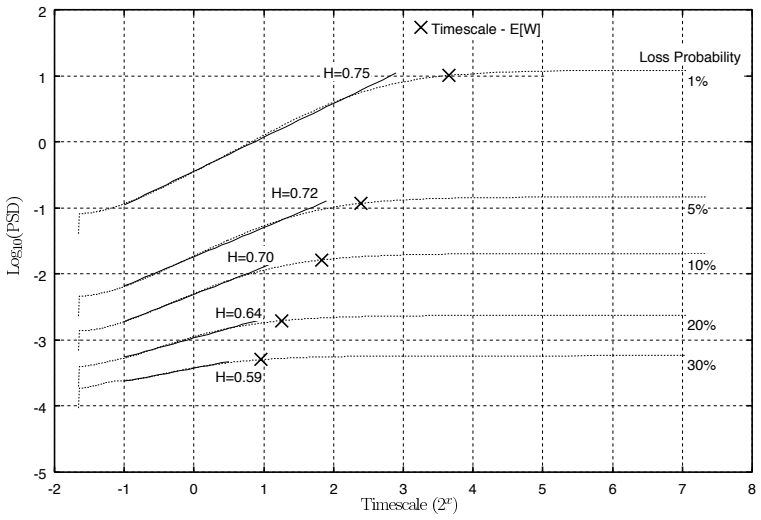

Figure 11: Range of timescales of the CA model

self-similar traffic.

In previous sections, we show that the exponential back-off and congestion avoidance mechanisms in TCP protocol lead to pseudo self-similarity in a single TCP flow. However, real Internet TCP traffic can contain many TCP flows, each with different source-destination pairs. Our analysis does not immediately explain why aggregated TCP flows could exhibit pseudo self-similarity. Thus, we provide a plausible argument for the pseudo self-similarity of real aggregated TCP traffic in Internet. If different TCP flows experiences different bottleneck links, the traffic pattern of the TCP flows will then independent of each other and exhibit pseudo self-similarity. The assumption of independent bottleneck links is reasonable in the backbone traffic, since most congestion occur at the edges of the network and the backbone links are usually underutilized. Intuitively, it is expected that pseudo self-similarity will be preserved when the independent flows are aggregated together. It has been shown that the aggregation of exact (asymptotic) self-similar processes is still an exact (asymptotic) self-similar process with Hurst parameter equal to the maximum among those of the individual processes [26]. We think this is one of the factors that contributes to the pseudo self-similarity in the data traffic observed in real networks.

In [14], the authors show how TCP congestion control can propagate pseudo self-similarity between flows traversing the same bottleneck link. Basically, if one of the flows traversing the link generates pseudo self-similar traffic, then pseudo self-similarity can be passed on to other TCP flows sharing that common bottleneck. In their paper, they measured a long-lived TCP flow over a trans-Atlantic link, and claim the TCP flow became pseudo self-similar by adapting to the background traffic at the bottleneck link.

Using the values of their experiment setting and the formula (eq. 29) in [16], also assuming the RTO to be in the range of 2.5 to 10 times the RTT, which is a common range as noticed in [16], we estimate the loss rate seen by the TCP flow to be in the range of 0.08 to 0.14. With this loss probability range our TO model predicts a local Hurst parameter of 0.69 to 0.72, which is comparable to the Hurst parameter reported by their trace analysis, 0.74. Moreover, our model predicts that the timescale for pseudo self-similarity goes up to 2^6 RTO, which corresponds to the range of 8 to 10 on the x-axis in Figure 1(c) in [14]. These timescales agree with the knee point where the wavelet analysis curve starts to turn flat. It is surprising that the TO model can predict the local Hurst parameter and its timescales quite well. Therefore, another equally

plausible explanation for the pseudo self-similarity observed in their experiment is the TCP TO mechanism itself.

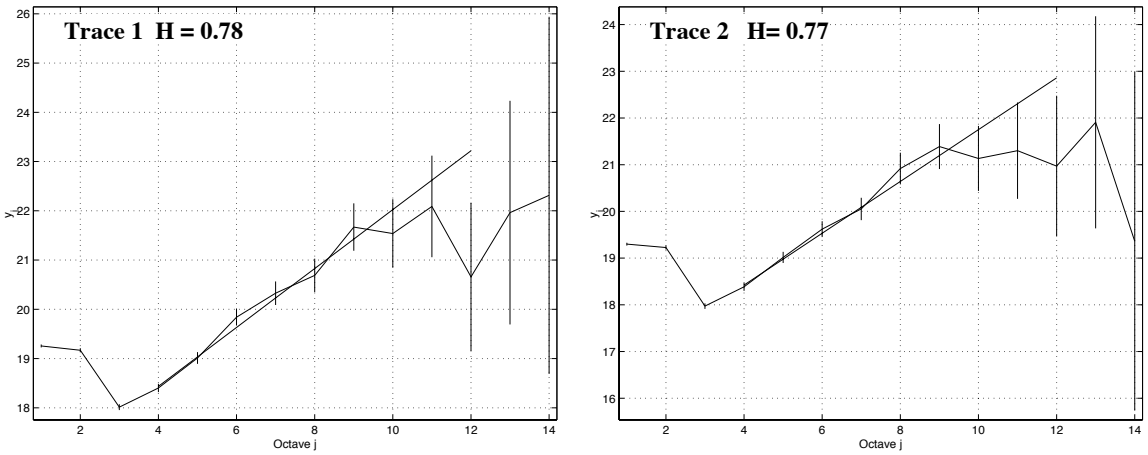

Figure 12: Wavelet analysis of traces between North and South America

We now investigate if our models are suitable in predicting the correlation structure of TCP traffic in real networks. To accomplish this, we collect and analyze TCP traces from the Internet and compare the traffic characteristics to the ones obtained from the models.

A set of one-hour long TCP traces from east coast to west coast over vBNS and from North America to South America over international link were collected for seven consecutive days using *tcpdump* [3]. The packet loss probability observed in some of the traces from North to South America is around 0.13. Figure 12 illustrates the wavelet analysis for two representative traces with this loss probability. From the figure, we observe that the traces exhibits pseudo self-similarity with a local Hurst parameter of around 0.77 and largest timescales of 2^7 and 2^8 RTT, which corresponds to 9 and 10 on the x-axis. For a loss probability of 0.12, our TO model predicts a local Hurst parameter of 0.72 and largest timescale of $RTO * 2^6$, which corresponds to 2^7 to 2^9 RTT, assuming that RTO takes values from 2.5 to 10 RTT. Note that the model again gives a good estimate for the local Hurst parameter and the largest timescales with respect to the trace analysis. The TO model is used here due to the high loss probability, under which the congestion avoidance mechanism seldom takes effect.

As for the traces obtained between east and west coasts, many of them present a packet loss probability of around 0.018. Wavelet analysis of the traces show a local Hurst parameter around 0.86 and largest timescale about $RTT * 2^6$, for traces with this loss probability. However, our CA model predicts a local Hurst parameter of 0.75 and largest timescale of 2^4 RTT. This discrepancy is not surprising since even under low loss probabilities, there are a considerable number of timeouts occurring, which can impact the traffic correlation structure. Note that our CA model only considers the congestion avoidance mechanism and ignores all other aspects of TCP, including timeouts.

6 Conclusion

In this paper we demonstrate that the TCP protocol can generate traffic with pseudo self-similarity over a certain range of timescales. In particular, we point out that the timeout

[3] *tcpdump* is publicly available at http://www-nrg.ee.lbl.gov

exponential back-off and the congestion avoidance mechanisms are responsible for this correlation structure. We show that under low loss probabilities the effect of the congestion avoidance mechanism on the traffic correlation structure is more pronounced while the timeout mechanism has minimal impact. In contrast, under high loss probability the timeout mechanism has larger impact on the traffic correlation structure, while the effect of the congestion avoidance mechanism is minimal. Together, these two mechanisms can generate pseudo self-similar network traffic under a wide range of packet loss probabilities.

The results obtained from the two analytical models presented in this paper support this claim. We investigated the validity of our claim using simulation and analysis of real trace measurements from the Internet. We show that the local Hurst parameter predicted by our models is in the same range as observed in the Internet traffic and other related works. We also identified the range of timescales that both mechanisms generate pseudo self-similarity. These timescales are related to the parameters of our models.

We do not claim that the TCP protocol is the solely cause of pseudo self-similarity widely observed in network traffic. Many other studies have also tried to explain the cause and the conditions under which the network traffic can exhibit pseudo self-similarity. We believe that some of those causes are plausible and can have impact on the correlation structure of the traffic. But most likely, the origin of pseudo self-similarity is due to one or more factors, including the TCP protocol. Hence, one should be careful when attributing the origin of traffic characteristics to a specific cause.

As for future work, we plan to extend our work to a more thorough TCP model, that combines the mechanisms of the protocol suite. We would also like to investigate how can the TCP protocol be modified in order to reduce the correlation structure currently present in the traffic it induces.

References

[1] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic," *IEEE/ACM Transaction on Networking*, vol. 2, pp. 1 – 15, Feb. 1994.

[2] V. Paxson and S. Floyd, "Wide-area traffic: The failure of poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 226 – 244, June 1995.

[3] M. E. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: Evidence and possible causes," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 835 – 846, Dec. 1997.

[4] V. Misra and W. Gong, "A hierarchical model for teletraffic," in *Proc. of the 37th Annual IEEE CDC*, 1998.

[5] M. Krunz and A. Makowski, "Modeling video traffic using M/G/Infinity input processes: A compromise between markovian and lrd models," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 16, pp. 733–748, Jun 1998.

[6] S. Robert and J. Y. L. Boudec, "On a markov modulated chain with pseudo-long range dependences," *Performance Evaluation*, vol. 27 – 28, pp. 159 – 173, Oct. 1996.

[7] A. T. Andersen and B. F. Nielsen, "An application of superpositions of two-state markovian sources to the modelling of self-similar behaviour," *Proc. of the IEEE INFOCOM*, pp. 196–204, 1997.

[8] K. Park, G. Kim, and M. Crovella, "On the effect of traffic self-similarity on network performance," in *Proc. of the SPIE International Conference on Performance and Control of Network Systems*, Nov 1997.

[9] Z. Liu, P. Nain, D. Towsley, and Z. Zhang, "Asymptotic behavior of a multiplexer fed by a long-range dependent process," Tech. Rep. UMass CMPSCI Technical Report 97-16, Computer Science Dept - University of Massachusetts, Amherst, 1997.

[10] M. Grossglauser and J. Bolot, "On the relevance of long range dependence in network traffic," *IEEE/ACM Transaction on Networking*, 1998.

[11] A. Veres, B. Hungary, and M. Boda, "The chaotic nature of TCP congestion control," in *Proc. IEEE INFOCOM 2000*, (Tel Aviv, Israel), Apr. 2000.

[12] S. Manthorpe, I. Norros, and J. Y. L. Boudec, "The second-order characteristics of TCP," in *Proc. of Performance '96*, (Lausanne), Oct. 1996. Presented in the Self-Similarity hot-topic session.

[13] L. Guo, M. Crovella, and I. Matta, "TCP congestion control and heavy tails," Tech. Rep. BUCS-TR-2000-017, Computer Science Dept - Boston University, 2000.

[14] A. Veres, Z. Kenesi, S. Molnár, and G. Vattay, "On the propagation of long-range dependence in the internet," in *Proc. ACM SIGCOM 2000*, (Stockholm, Sweden), Aug. 2000.

[15] D. E. Comer, *Internetworking with TCP/IP*, vol. 1. Prentice-Hall, 3rd ed., 1995.

[16] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *Proc. ACM SIGCOMM'98*, (Vancouver, CA), Sept. 1998. A longer version is available as UMass CMPSCI Tech Report 98-08.

[17] E. de Souza e Silva and R. M. M. Leão, "The TANGRAM-II environment," in *Proc. of the 11th Int. Conf. on Modelling Tools and Techniques for Computer and Communication System Performance Evaluation (TOOLs'2000)*, (Schaumburg/Illinois, USA), pp. 366 – 369, Springer, Mar. 2000. Lec. Notes in Comp Sci 1786.

[18] R. Leão, E. de Souza e Silva, and S. de Lucena, "A set of tools for traffic modeling, analysis and experimentation," in *Proc. of the 11th Int. Conf on Modelling Tools and Techniques for Computer and Communication System Performance Evaluation (TOOLs'2000)*, (Schaumburg, USA), pp. 40 – 55, Springer, Mar. 2000. Lect Notes in Comp Sci 1786.

[19] D. Veitch and P. Abry, "A wavelet based joint estimator of the parameters of long-range dependence," *IEEE/ACM Transaction on Information Theory*, April 1999.

[20] P. Abry, P. Gonçalvés, and P. Flandrin, "Wavelets, spectrum estimation, 1/f processes," *Lecture Notes in Statistics*, vol. 105, pp. 15–30, 1995.

[21] UCB, LBNL, and VINT, "Network simulator - NS (version 2)." Lawrence Berkeley National Laboratory - University of California, Berkeley URL http://www-nrg.ee.lbl.gov/vat.

[22] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP selective acknowledgement options." RFC 2018, Apr. 1996.

[23] P. S. Center, "The TCP-friendly website." URL: http://www.psc.edu/networking/tcp_friendly.html.

[24] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *Communication Review*, vol. 27, July 1997.

[25] C. Hollot, V. Misra, D. Towsley, and W. Gong, "A control theoretic analysis of RED," in *Proc. of IEEE INFOCOM*, 2001. To appear in.

[26] Y. Fan and N. D. Georganas, "On merging and splitting of self-similar traffic in highspeed networks," in *Proc. of ICCC'95*, (Seoul, Korea), pp. 8A.1.1 – 6, 1995.