The Chaotic Nature of TCP Congestion Control

András Veres
Ericsson, Traffic Analysis and Network Performance Laboratory
Budapest, Hungary
e-mail: Andras.Veres@lt.eth.ericsson.se

Miklós Boda
Ericsson Research
Stockholm, Sweden
e-mail: Miklos.Boda@era-t.ericsson.se

Abstract— **In this paper we demonstrate how TCP congestion control can show chaotic behavior. We demonstrate the major features of chaotic systems in TCP/IP networks with examples. These features include unpredictability, extreme sensitivity to initial conditions and odd periodicity. Previous work has shown the fractal nature of aggregate TCP/IP traffic and one explanation to this phenomenon was that traffic can be approximated by a large number of ON/OFF sources where the random ON and/or OFF periods are of length described by a heavy tailed distribution. In this paper we show that this argument is not necessary to explain self-similarity, neither randomness is required. Rather, TCP itself as a deterministic process creates chaos, which generates self-similarity. This property is inherent in todays TCP/IP networks and it is independent of higher layer applications or protocols. The two causes: heavy tailed ON/OFF and chaotic TCP together contribute to the phenomena, called fractal nature of Internet traffic.**

Keywords—**TCP congestion control, fractal traffic, chaotic models.**

I. Introduction

Traffic models used to model current Internet traffic can be categorized into two major groups: *link/source* and *network level models*. Link level models fit statistical models to measurements of traffic on network links or traffic sources for example a WWW server. Recently, the largest contribution in this area was the exploration of fractal and long-range dependent property of traffic, namely that the second order statistics of traffic volumes observed at different scales does not change. This resulted as a revolution in performance modeling and questioned previous models based on Markovian behavior (see Paxson and Floyd [12]). We mention two major publications in this area: Leland, Taqqu, Willinger and Wilson demonstrated through rigorous tests that Ethernet traffic is self-similar [8], Crovella and Bestavros proved how WWW as the major contributor to current Internet traffic can cause long-range dependence and self-similarity [4]. Chaotic-maps appeared as efficient and parsimonious methods to generate packet traffic on the link/source level, see for example the work by Erramilli and Singh [7] and the same authors with Pruthi [6].

The drawback of link/source level models is that they disregard one of the major properties of todays Internet, namely that the majority (80-90%) of traffic is generated and controlled by the TCP protocol, which is adaptive in nature. The consequence of adaptivity is that the source behavior cannot be disconnected from the network configuration (e.g., routing, scheduling, buffer management). Traffic statistics change if the network configuration changes, so a link/source model is valid only for the configuration (and all other circumstances) that were present at the time of model fitting. The recent paper of Arvidsson and Karlsson [1] demonstrates that the adaptive simulated traffic behaves significantly differently in the buffers as the link/source models.

This problem called for *network level models*, which try to form a unified model taking into account the cooperation of all source and network mechanisms. Due to the complexity of this problem the models published in this area are still in early phase. Mathis, Semske, Mahdavi and Ott [9] published an analytic model about macroscopic behavior of TCP, Padhye, Firoiu, Towsley and Kurose [13] model the impact of the time-out mechanism on TCP throughput. The drawback of these models is that they assume that TCP congestion control always behaves in a nice periodic and predictable fashion. This is in contradiction with the measurements and simulations of TCP traffic.

In this paper we bridge the two modeling approaches by modeling the network level behavior of aggregate TCP flows while reproducing the complexity found in link/source level models. The key finding in this paper is that cooperating TCP congestion control processes together form a deterministic chaotic system which is able to produce periodic and non-periodic, predictable and non-predictable, short-range dependent and also self-similar behavior. We are going to demonstrate some of the key properties of chaotic systems present in the Internet.

What is chaos? A system is called chaotic if it satisfies the following conditions [5]:

- nonlinearity;
- determinism;
- order in disorder;
- sensitivity to initial conditions or the ”butterfly effect”;
- unpredictability.

Nonlinearity means that the system is controlled through nonlinear functions. In case of TCP, nonlinear functions are used for round-trip time (RTT) measurements, slow start and congestion avoidance.

Determinism means that the system's future is fully described by the past. This is also true, as TCP works in a self-clocking way, no randomness is used, every event (e.g., sending of a packet or time-out) is completely determined by the past.

We are not going to discuss the above two properties because they are obviously characteristic of TCP. However, the further attributes need more insight and proof – these properties are discussed in detail in the paper.

The present *network level* models for TCP/IP traffic assume periodic and stable behavior. Such behavior is demonstrated by simulations in Section II, but in the next section we prove that this behavior is not universal by giving examples for more complex periodic and finally non-periodic patterns.

In Section III we introduce the notion of *attractors* – hidden multidimensional trajectories of the TCP process, and give a method to efficiently visualize them, thus making it possible to examine the *hidden order in an otherwise seemingly random process*.

A system satisfying the *butterfly effect* is extremely sensitive to small changes in the initial parameters or minute perturbations of the system. This property is the major landmark of chaotic systems. This property of TCP is demonstrated in Section V and we also quantify the sensitivity of the system by measuring the *Lyapunov exponent* of the system's trajectory.

Although deterministic, the chaotic nature of TCP has inhibited correct *network level* models in this area so far. This property results in a qualitative difference compared to previous macroscopic models. In Section VI we show that TCP can generate traffic with scaling behavior on large timescales. This finding throws new light on self-similar traffic modeling, explaining self-similarity with *deterministic chaotic mechanisms* and not with higher layer *stochastic mechanisms* [4].

II. Macroscopic models' assumptions: Periodicity and order

First we take a simple configuration containing just two greedy TCPs sharing a single link. This configuration helps us to explain the graphical methods used later in the paper. We used the *ns-2* [11] simulator for all experiments in this paper. The link parameters are: link rate $C = 0.2$ Mbps, delay $d = 10$ ms, buffer size $B = 20$ packets. The receiver window was set to a very large value so that the congestion window (*cwnd*) was the limitation. We used the Tahoe version of TCP.

The two TCPs are started simultaneously. After a short transient the two TCPs settle down into a periodic pattern, one of them always a little ahead of the other, but both follow the same pattern of slow start, congestion avoidance, packet loss and backoff. This is clearly visible in Fig. 1. Another way of displaying the system evolution is to use a spatio-temporal graph where the window size of a TCP is displayed as a colored strip: the larger the window, the brighter the color (we borrowed the idea from [3]). This method is used in Fig. 2, the two TCPs are displayed on top of each other: the first TCP is displayed in the first row, the row below corresponds to the second TCP. The periodic pattern appears as synchronized ”waves”, the back-off times colored as black strips are always close to each other for the two TCPs.

One can observe the system's evolution not only as a function of time, but also by drawing the trajectory of the system as it moves in the phase space. The phase space is a multi-dimensional space where each dimension represents a system variable, thus each point in the phase space represents a unique state of the system. If we plot the evolution of the system in this space, then – as the system is completely deterministic – if the system gets back to a previous point, it will continue that path again, creating a closed loop. If the system is periodic, then the corresponding trajectory will be a loop and *vice versa*, if the system evolution can be represented by a loop in the phase space, the system is periodic. This method is thus very appealing to examine the periodicity of a multi-TCP system.

Even in this 2-TCP system the number of state variables that completely describe the system is very large (e.g., the whereabouts of previously sent packets, internal variables of the sending and receiving TCPs), it is not possible to draw them on a paper but it is possible to properly choose a section of the phase space. We chose the TCP congestion window (*cwnd*) size because it is in close relation with the sending rate of TCP and we have logged the *cwnd* values every 10ms for each TCP. Fig. 3 shows this plot for the previously examined simple configuration. The transient part is removed from the bottom graph. As it can be seen the process gets into a periodic loop, although the period is fairly large. Interestingly, this loop is very stable,

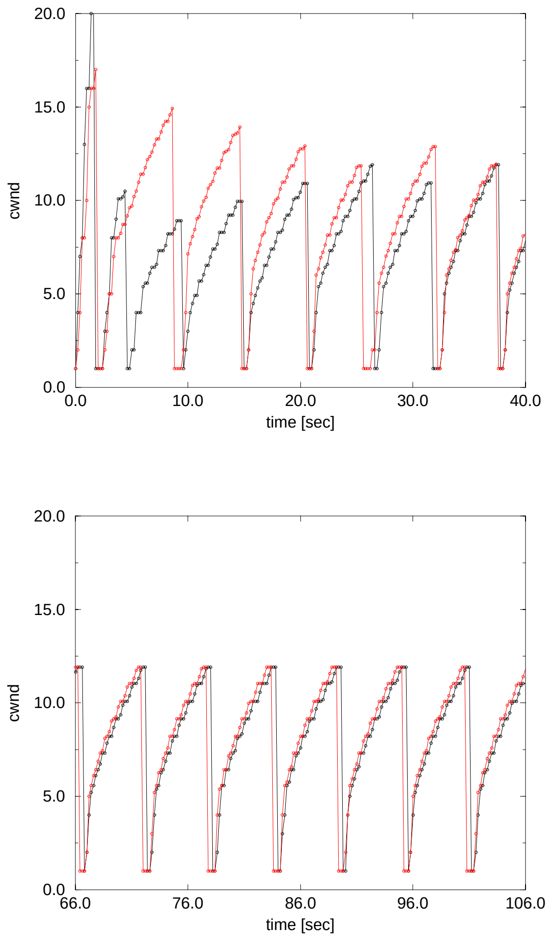


Fig. 1. The congestion window processes of two competing TCP sources. a) with transient part b) transient removed.

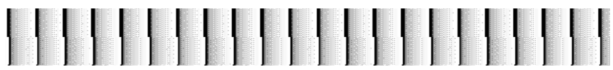


Fig. 2. Spatio-temporal evolution of the congestion window processes.

which means that it does not matter how we disturb the system (e.g., drop a packet randomly) or choose the initial conditions (e.g., we start the second TCP a few seconds later) eventually the system will return to the same pattern. The graph also reveals that the two TCPs are synchronized, they move along a "staircase": they increase their *cwnd* one after the other, finally at the top loss is detected and both TCPs decrease their *cwnd*s to 1 packet and the period starts over again.

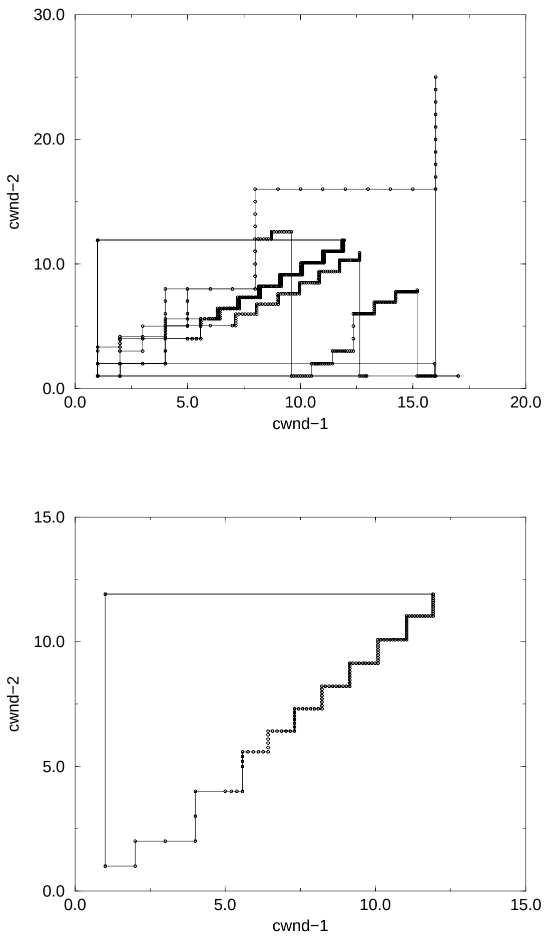


Fig. 3. The congestion window processes of two competing TCP sources. a) with transient part b) transient removed.

There is no surprise in this behavior – actually this behavior is assumed widely when calculating macroscopic performance values. Unfortunately this nice behavior is not universal as it will be shown later.

III. Complex periods and attractors

Can this simple system produce different behavior? Surprisingly, yes! If we change the system parameters: link rate $C = 0.5$ Mbps, delay $d = 10$ ms, buffer size $B = 4$ packets, we get a period again, but it is much more complex, see Fig. 4 and Fig. 5. There is still an underlying regular beat, but on a larger timescale there is an alternating pattern: one gains speed over the other for a given time and then the other takes over.

Changing the parameters further we can make this simple system change from the simple regular beat to a very complex

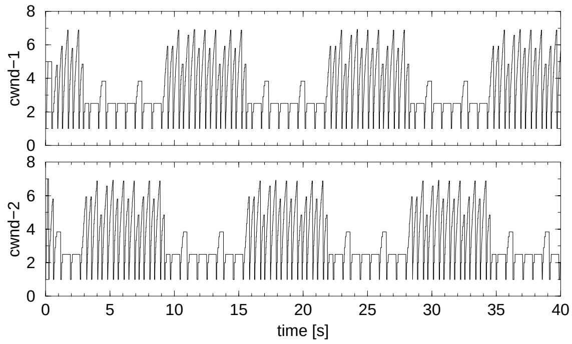


Fig. 4. The congestion window processes of two competing TCP sources. $C = 0.5$ Mbps, $d = 10$ ms, $B = 4$ packets.

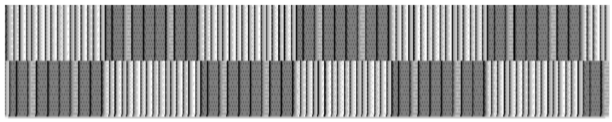


Fig. 5. Spatio-temporal graph: $C = 0.5$ Mbps, $d = 10$ ms, $B = 4$ packets (the same configuration was used as for Fig 4).

pattern, and here our displaying method reaches its limit. The problem is that the chosen set of system variables i.e., the *cwnd*s is just a projection of the complete set of system variables and the value of *cwnd* is not continuous. This means that it is not possible to efficiently visualize periods for complex behavior, because only a limited number of points will be touched, thus meeting and then diverging trajectories are indistinguishable. Another problem is that the congestion window process at a certain time instant does not reveal the underlying state of the system in all details.

It was proposed in [15] to use the time shifted past values $[x_t, x_{t-\delta t}, x_{t-2\delta t}, ...]$ of an easily measurable quantity for complex systems and it was shown that it can be used equivalently to reconstruct the underlying multidimensional trajectories if there is no access to the state variables. The choice of δt can be nearly arbitrary in a wide range. The result is a multidimensional vector that is projected to the 2D plane simply by averaging the values $\hat{X} = 1/n(x_t + x_{t-\delta t} + ...)$. The above described method is used for the *cwnd* values:

$$x[i] = \frac{1}{n}\sum_{j=1}^{n} \mathrm{cwnd}_x[i-j] \tag{1}$$

$$y[i] = \frac{1}{n}\sum_{j=1}^{n} \mathrm{cwnd}_y[i-j] \tag{2}$$

Here x and y denote the two TCPs. n controls the scale over which the congestion windows are averaged, the larger the value is, the more hidden dimensions can be reconstructed. The method has two further benefits:

- The number of possible points on the (x, y) plane is increased from W^2 where W is the number of possible *cwnd* values to $(nW - n)^2$ (if *cwnd* is counted in packets).
- Consecutive results $x[i]$ and $x[i+1]$ are placed close to each other, actually no further than $(2 * W - 2)/n$. Thus using this construction the produced graph can be made as smooth as required.

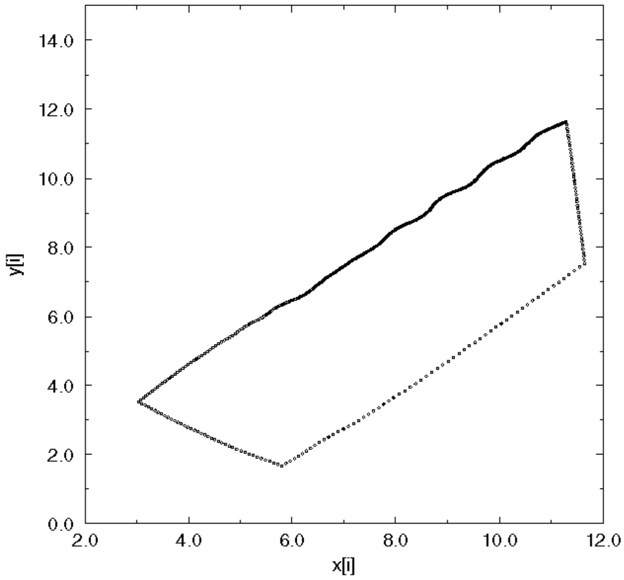
A nice property of the graph is that it preserves the periodicity property: periodic trajectories are displayed as closed loops in (x, y).

Fig. 6 shows the periodic trajectories of the simple (staircase) and the more complex (alternating) periodic systems discussed so far. The alternating behavior can be easily observed on the bottom graph. Both systems are represented as closed loops, which is a sign of periodicity, but the complexity of the two loops is significantly different, the "staircase" system has a nice simple loop, while the "alternating" system has a more complex but more or less symmetric trajectory. Although being very complex, both trajectories prove to be very stable: disturbing the system (e.g., changing the relative starting times of TCPs thus giving gain to one of them or perturbing the congestion control by artificially changing the size of the *cwnd*) does not destroy the trajectory, after a short detour both systems get back to the same regular pattern. Such trajectories are called *attractors*. An attractor is a set of points to which nearby trajectories are attracted to. A more formal definition of an attractor can be found in [14].

IV. STRANGE ATTRACTORS

For certain parameter sets (i.e., the number of competing TCPs, service rates, buffer sizes or transmission delays) the system exhibits simple, for other sets very complex behavior. It is not difficult to find parameters, where the system seems to never repeat itself. Such a parameter set is $C = 0.1$ Mbps, $d = 10$ ms, $B = 4$ packets. Of course as the system is finite dimensional and the dimensions are discrete numbers, the trajectories will always be periodic, but the size of the period is extremely large. The attractor of this system is displayed in Fig. 7. The structure of the trajectory is very fine and the simulation time used to obtain it was more than 4 hours.

A very interesting experiment can be made to show the fine structure of the attractors presented before, namely we show that the projection of the attractor has fractal dimension. A usual drawing on a 2-dimensional plane (e.g., a loop) can be easily measured to get its length. This is not the case with fractal structures that have length depending on the size of the unit we use for the measurement [14]. The dimension of such an object is represented by a non-integer value. We can, for example, measure the attractor's box-counting dimension D, which is done in the following way: choose a grid on the plane of size s, then count the number of boxes which have a part of the object in it. Denote this number as $N(s)$. Then change s to cover several scales e.g., half it each step. Finally plot the values $log(N(s))$ versus $log(1/s)$, if the object is fractal, then $log(N(s)) \sim Dlog(1/s)$, where D is a non-integer value. In practice one should fit a least-squares regression to the logarithmically spaced sample points s_i and calculate the slope. As the graph is made up of a finite number of points, the final dimension is 0 so the last few values are dropped, and also the first few values of s_i are omitted because they are in the scale of the object itself. In between lies the area where the fractal property holds. Fig. 8 shows the calculated fractal dimension of the "staircase" and the "nonperiodic" trajectories. The fitted regression follows the plot through 4-5 magnitudes, which supports the significance of the measurements.



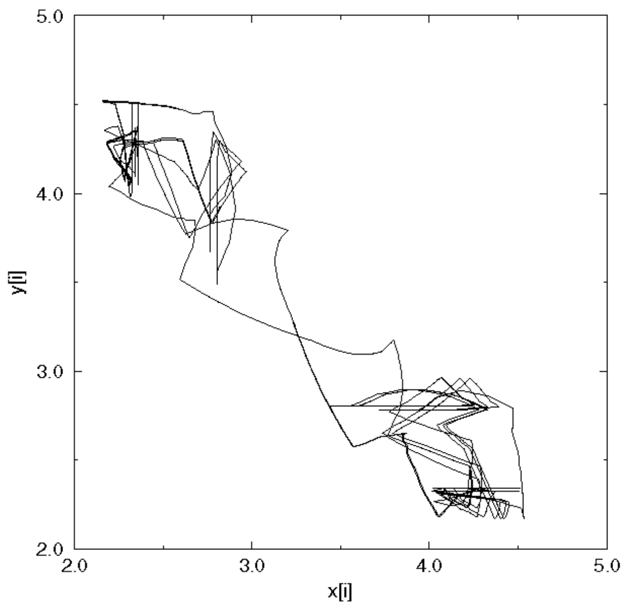


Fig. 6. Periodic attractors of two competing TCP sources: a) $C = 0.2$ Mbps, $d = 10$ ms, $B = 20$ packets, time shifts $n = 100$. b) $C = 0.5$ Mbps, $d = 10$ ms, $B = 4$ packets, time shifts $n = 90$.

The simple periodic "staircase" system has an attractor with dimension $D \approx 1$ which equals a simple 1 dimensional line and so it is not fractal. In the case of the "nonperiodic" system we can calculate $D \approx 1.61$, which is significantly different from 1, but below 2 - the attractor is a fractal. Attractors with fractal properties are called *strange attractors*. Furthermore, if a system shows *sensitivity to initial conditions*, then the corresponding attractor is called a *strange chaotic attractor*.

Note: there exist chaotic systems with non-fractal attractors, and strange attractors of non-chaotic systems.

V. SENSITIVITY TO INITIAL CONDITIONS

In this section we demonstrate that in certain cases TCP congestion control is prone to large sensitivity to initial conditions, which means that very small perturbations in the system may

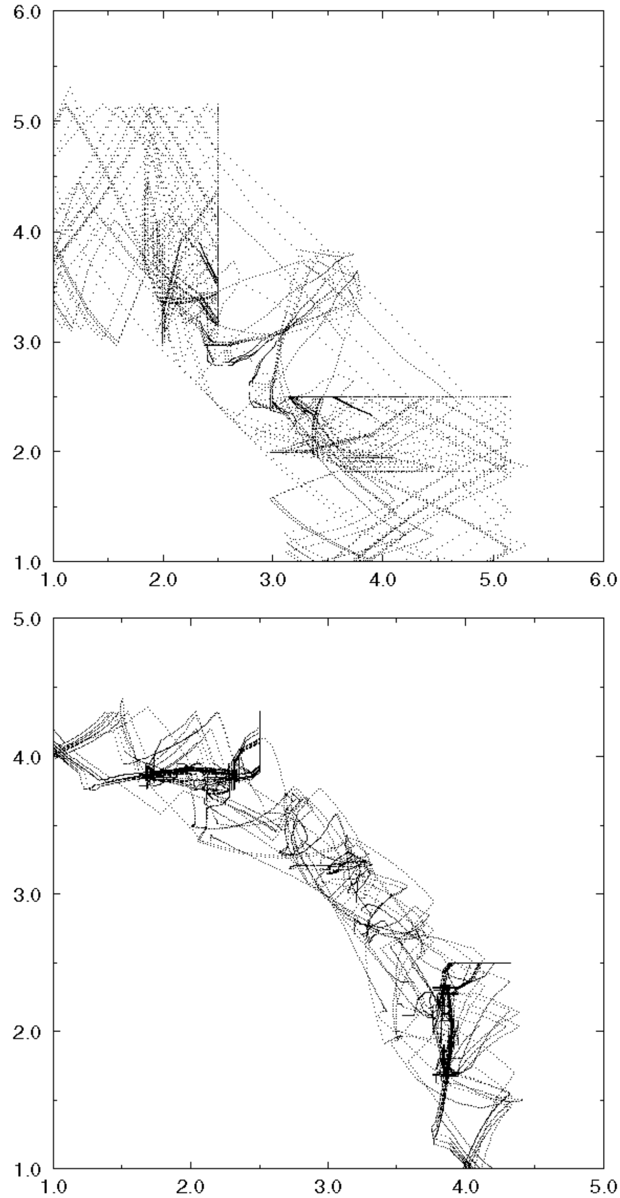


Fig. 7. Strange attractor. $C = 0.1$ Mbps, $d = 10$ ms, $B = 4$ packets. a) time shifts $n = 100$, b) time shifts $n = 300$.

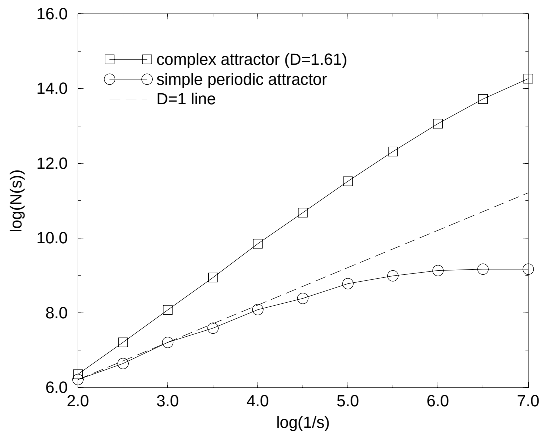


Fig. 8. Box-counting dimension of the simple periodic "staircase" and "nonperiodic" trajectories (time shifts $n = 300$ was used for the "nonperiodic" and $n = 100$ for the "staircase" trajectory).

lead to large differences compared to an unperturbed system in a very short time. The distance grows to such an extent that it gets into the range of the signal itself. This is one of the major properties of chaotic systems.

In the following experiment we increased the number of simultanious TCP sessions to 30. ($C = 1$ Mbps, $d = 15$ ms, $B = 60$ packets). First we let the system evolve for a while, then at $t = 50s$ we artificially increased the congestion window of one of the TCPs with one packet. Then we plotted the spatio-temporal graph of both systems, see the original in Fig. 9 (top) and the perturbed system in Fig. 9 (middle). The length of the plot is 100s so the perturbation was done right at the middle of the plot. If we compare the two systems, first the differences are invisible, but a few seconds later the two systems look completely different. To make this more visible, we plotted the difference of the two systems in a way that each dot was colored according to the distance defined as $d(i,t) = |w^{orig}(i,t) - w^{pert}(i,t)|$, where i and t is the id. of the TCP and the time respectively, $w^{orig}(i,t)$ is the *cwnd* of ith TCP in the original system at time t and $w^{pert}(i,t)$ is the same for the perturbed system. See Fig. 9 (bottom). The first part is white, which means that the two systems are identical, then a few dim dots appear and a few seconds later the difference looks like the original plots themselves.

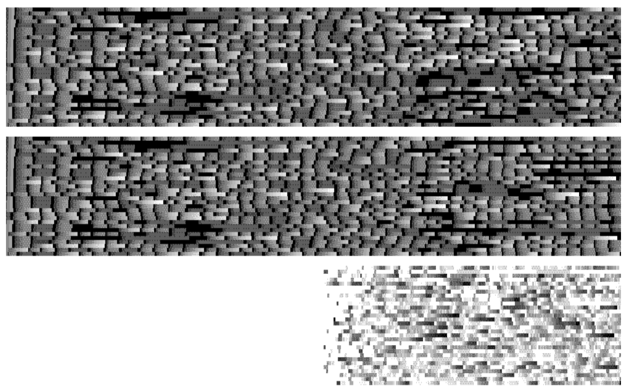


Fig. 9. Spatio-temporal graph of the original system (top). Spatio-temporal graph of the perturbed system (middle). Difference between the two systems (bottom).

To quantify how fast this divergence happens, we define the distance between the two systems at time t as the Eucledian distance in the *cwnd* space: $E(t) = \sqrt{\sum_{i=1}^{N}(w^{orig}(i,t) - w^{pert}(i,t))^2}$. See Fig. 10.

The rate at which the systems diverge after a small perturbation of the ith TCP ϵ_i at time t_0 can be described by the so called *Lyapunov exponent,* which we approximate by measuring the time Δt it takes for the two systems to reach a given distance $E(t_0 + \Delta t) > \hat{E}$, then:

$$\lambda(t_0, i) \approx \frac{1}{\Delta t} \ln |\frac{E(t_0 + \Delta t)}{\epsilon_i}| \qquad (3)$$

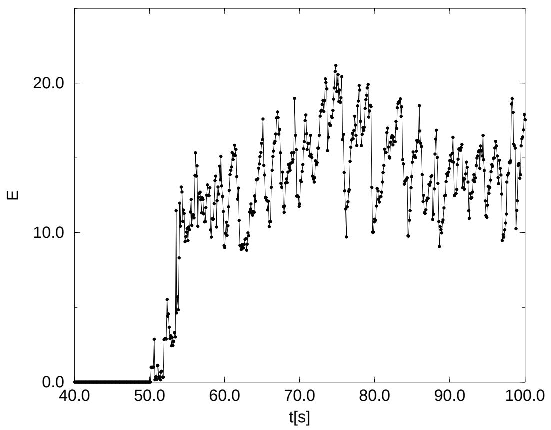


Fig. 10. Divergence of the original and the perturbed systems.

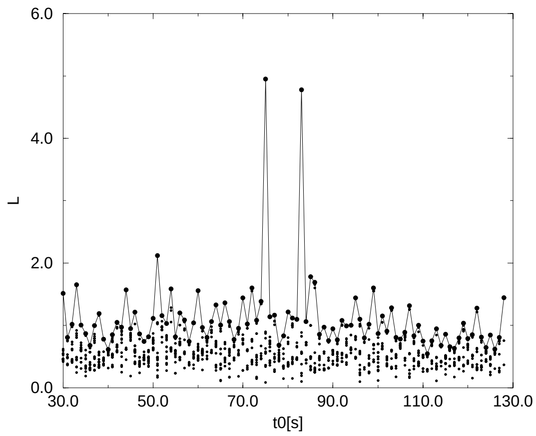


Fig. 11. Lyapunov exponents at different points of the trajectory and for all 30 TCPs, maximum exponents along the trajectory are connected with a line (average maximum $\lambda \approx 1.11$).

For the experiment we chose $\hat{E} = 10$ and $\epsilon_i = 1$. The motivation to calculate an *exponent* is that trajectories diverge at an exponential rate. However as the two systems cannot get arbitrarily far from each other (as the phase space is limited) only the increasing part of Fig. 10 should be considered, this explains the choice of $\hat{E} = 10$.

The Lyapunov exponent is the rate at which the two systems diverge from each other every time unit. This value of course depends on *which* TCP we perturb and also *when* the interference was done, in other words: to which direction in phase space we push the system and at what part of the phase space the system is at the time of perturbation. There are cases when even an otherwise sensitive system is not effected by a small interference. This case can be characterized by a negative exponent. (Of course our approximation can be used for positive exponents only.) There is no generally accepted definition when an attractor is called chaotic, but we can say that if sensitive points ($\lambda > 0$) are dense on the trajectory, then the attractor is chaotic.

To arrive at a more general numerical result for a given system we calculated the exponent at many different points of the trajectory (t_0) and for all 30 TCPs. Then for each t_0 we chose the most sensitive direction where the largest λ was measured and averaged these values over time to get the average maximum exponent of the trajectory:

$$\lambda = \mathbf{E}\left[\max_{\mathrm{i}} \lambda(\mathrm{t}_0, \mathrm{i})\right] \qquad (4)$$

See Fig. 11. In the experiment we got $\lambda \approx 1.11$, which means that after a perturbation the difference between the two systems increase at an average rate of $e^{\lambda} \approx 3.03$ every second.

VI. Testing for self-similarity of the time series

In this section we show that a system of competing TCPs with certain parameters generates second-order self-similar traffic over several timescales ranging from a few round-trip times to hundreds of seconds. A number of statistical tests were performed to search for scaling behavior: absolute values method, wavelet analysis, periodogram and R/S method. [17]

Not all configurations produce self-similarity, we have observed that it is mainly the B/N ratio that controls the behavior of the system. As the ratio decreases (the ratio of the ‘pipe’ for one TCP flow is getting smaller), the system goes through a phase transition from periodic to chaotic behavior, and for certain parameters it produces self-similar time-series.

The simulation setup was the following: $C = 1$Mbps, $d = 15$ms, $B = 20$ packets and $N = 40$ TCPs. Note that the buffer can store less packets than the number of active flows. This results in a packet loss ratio of around 16%, creating a strong bottleneck on the path. It was argued in [10] that such small pipes contribute substantially to the performance of the current Internet.

During the simulation the amount of sent bytes by all TCPs were logged individually every 0.1s. The packet trace was a few hours long and consisted of about 1.6 million data packets to be able to perform tests on sufficiently large timescales. Fig. 12 shows the trace of a one-TCP microflow filtered out from the trace.

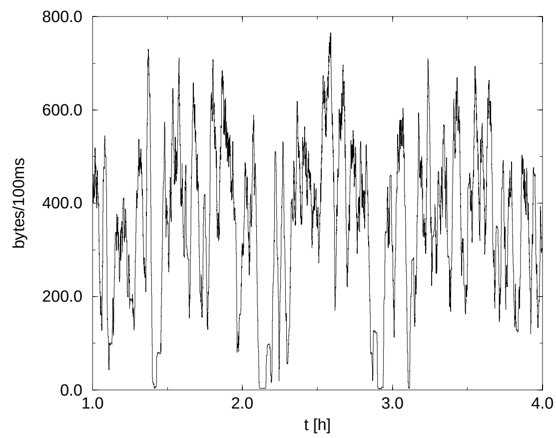
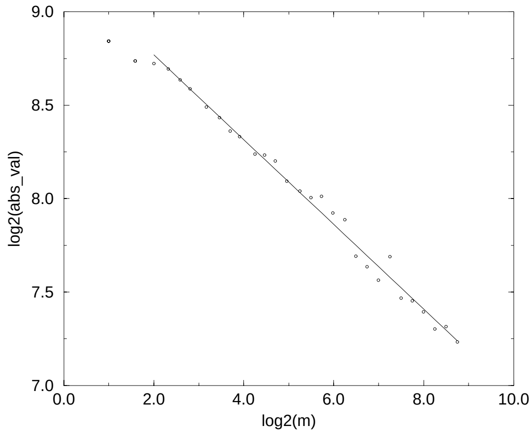
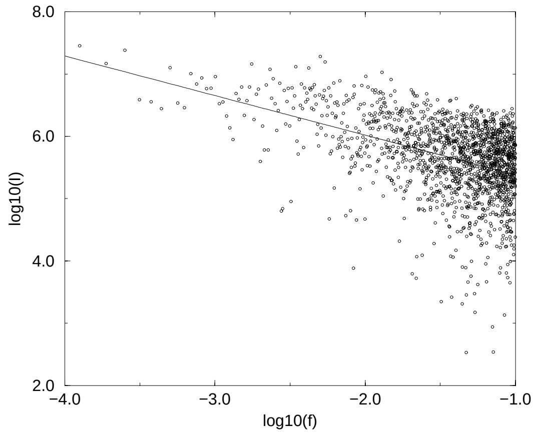


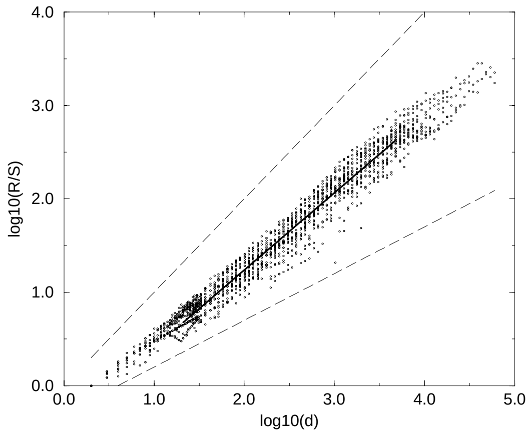
Fig. 12. Time series of sent bytes by a one-TCP microflow (running average of $\tau = 100$s.)



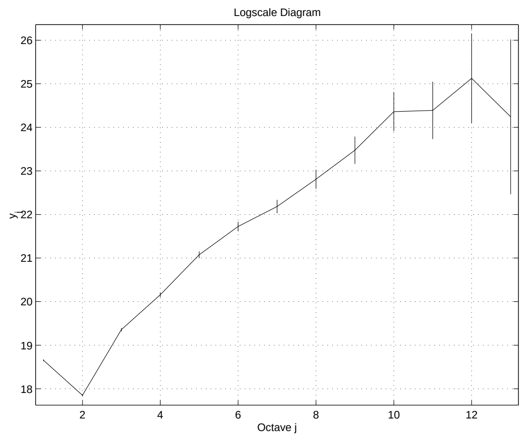
(a) Absolute values method $H = 0.79$.



(b) Periodogram method $H = 0.815$.



(c) R/S method $H = 0.813$.



(d) Wavelet method $H = 0.787$, $(0.754, 0.819)$.

Fig. 13. LRD tests for one-TCP microflow.

Consider a weakly stationary stochastic process X, with constant mean, finite variance and autocorrelation function $\rho(k)$. Let $X^{(m)}(k) = 1/m \sum_{i=(k-1)m+1}^{km} X(i)$ denote the m aggregated series of X. The process X is called *exactly self-similar* if for all m it satisfies $X =_d m^{1-H} X^{(m)}$. X is said to be *asymptotically self-similar* if this property holds as $m \to \infty$. Furthermore, X is *second-order self-similar* if $m^{1-H} X^{(m)}$ has the same variance and autocorrelation as X. If this holds asymptotically, then X is *asymptotically second-order self-similar*. [8]

The tests were made on the time series of the amount of sent bytes by a one-TCP microflow and the aggregate quantity of all TCPs as well. The results support that traffic of a one-TCP microflow is consistent with asymptotic second-order self-similarity with $H > 0.5$. On the other hand, aggregate traffic of all TCPs is short-range dependent with $H \approx 0.5$.

The first test is based on the behavior of the expectation of the absolute values of the series $X^{(m)}$ [18], [17]

$$\mu^{(m)} = \frac{1}{N/m} \sum_{k=1}^{N/m} \Big| X^{(m)}(k) - \frac{1}{N} \sum_{i=1}^{N} X(i) \Big|. \tag{5}$$

If X is self-similar, then on a log-log plot the absolute values increase linearly with m, with a slope of $H - 1$. Fig. 13(a) shows the result for the absolute values method ($H = 0.79$).

The periodogram method approximates the spectral density of the process with the expression

$$I(\lambda) = \frac{1}{2\pi N} \Big| \sum_{j=1}^{N} X_j e^{ij\lambda} \Big|^2$$

For a LRD series the spectrum is $I(\lambda) \sim |\lambda|^{1-2H}$ at the origin. Fig. 13(b) shows the result of the periodogram method for a one-TCP microflow, resulting in an approximation of $H = 0.815$.

The rescaled adjusted range statistics (R/S) [17] and the wavelet analysis method [2] are not described here, only the results are depicted in Fig. 13(c) (R/S: $H = 0.813$) and Fig. 13(d) (Wavelet: $H = 0.787$, with 95% quantiles at $[0.754, 0.819]$).

All methods resulted in a Hurst exponent of $H \approx 0.8$.

Aggregate traffic entering the bottleneck buffer shows a significantly different behavior. See Fig. 14 for the result of the R/S method performed on aggregate traffic. The results of other methods are not shown here, but all tests support that the Hurst exponent of aggregate traffic is around $H \approx 0.5$ i.e., it is short range dependent.

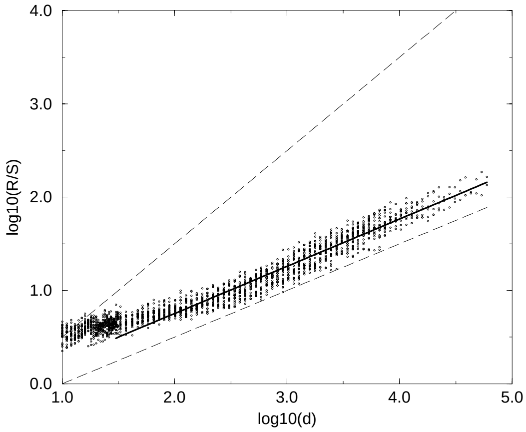


Fig. 14. R/S method for aggregate traffic. $H = 0.51$

The cause of short range dependence of aggregate traffic is the way TCP's congestion control works: it tries to smooth out the rate of traffic in the bottleneck buffer by keeping the bottleneck buffer occupancy close to the maximum. The outcome is that overshoots are followed by fast backoffs and then temporary decreases in traffic volume are fast corrected by increasing the transmission rate. This is especially true if there are a large number of TCP flows sharing a link. The mechanism smooths the aggregate but keeps the individual TCP flows long-range dependent as shown by the Hurst parameter of individual TCP flows.

How can then one measure $H > 1/2$ or long-range dependence for aggregate network traffic? There are two possible explanations:

- The effect of TCP congestion control is mixed by higher layer protocols with heavy tailed properties (WWW file sizes, waiting times). [4]
- Individual long-range dependent TCP flows exit the bottleneck buffer and enter other non-bottleneck buffers. There they mix with other flows coming from other buffers.

Assume that we have a number of LRD processes X_is with autocovariance function given in the form

$$\gamma_i(t) = a_i t^{-\delta_i} \tag{6}$$

Then if $\delta_i = 2 - 2H \leq 1$ (or $H \geq 1/2$), then this process is long-range dependent (its autocorrelation function is non-summable) and if $\delta_i > 1$ then it is short-range dependent. Let us assume that if we multiplex such streams coming from different bottleneck buffers in a non-bottleneck buffer then they remain independent. The autocovariance function of multiplexed traffic is then

$$\gamma(t) = \sum_i \gamma_i(t) = \sum_i a_i t^{-\delta_i} \sim t^{-\min \delta_i} \quad \text{as} \quad t \to \infty \tag{7}$$

in other words long-range dependent traffic remains long-range dependent in a non-bottleneck buffer, and it is characterized by the largest exponent. This way TCP aggregates can show long-range dependence. This effect does not happen in the bottleneck buffer because there the assumption of independence does not hold.

VII. Conclusions

We have demonstrated how TCP can produce for certain parameters simple and for others very complex behavior. We have proved that for certain parameters TCP behaves *chaotically*. We have shown the main properties of chaos present in TCP. Finally, we have proved that TCP congestion control creates self-similar traffic with Hurst parameters showing both short-range and long-range dependence depending on system parameters. This property is more fundamental than the second order self similarity property reported before – because it is the property of a low level deterministic system (TCP) itself, regardless of the applications running on top of TCP.

The conventional paradigm that traffic sources are treated separately from the network has major drawbacks, it is not possible to disregard the network status when we create traffic models – this was clearly shown in our simulations. We have also shown that a chaotic and non-predictable system can be turned into a regular easy-to-handle system by choosing different system parameters.

Our future focus will be to investigate the possibility of controlling the system by cleverly chosen minute influences in the buffer management or the TCP congestion control to improve throughput or fairness and to arrive at more accurate *network level* macroscopic TCP models.

Acknowledgment

The authors would like to thank the help of Ádám Magi, András Oláh, Zoltán R. Turányi, András Valkó, Sándor Molnár and Gábor Vattay. We would like to thank Darryl Veitch for making the wavelet analysis code available on the net.

References

[1] A. Arvidsson and P. Karlsson, "On Traffic Models for TCP/IP", *ITC-16*, Edinburgh, UK, June 1999

[2] P. Abry and D. Veitch, "Wavelet Analysis of Long-Range-Dependent Traffic", *IEEE/ACM Transactions on Networking*, vol. 44, No.1., January 1998.

[3] Y. Braiman, J. F. Lindner and W. L. Ditto, "Taming Spatiotemporal Chaos with Disorder", *Nature*, Vol 378, 30. Nov. 1995

[4] M. E. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes", *in Proc. ACM SIGMETRICS*, pp. 160-169, May 1996.

[5] W. Ditto and T. Munakata, "Principles and Applications of Chaotic Systems",*Comm. of the ACM*, Nov. 1995/Vol. 38, No.11

[6] A. Erramilli, E. P. Singh and P. Pruthi, "Chaotic Maps As Models of Packet Traffic", *Proc. of the 14th ITC*, 329-338, June, 1994

[7] A. Erramilli and R. P. Singh, "Application of Deterministic Chaotic Maps to Model Packet Traffic in Broadband Networks", *Proc. 7th ITC Specialist Seminar*, Morristown, NJ, 8.1.1-8.1.3, 1990

[8] W. E. Leland, M. S. Taqqu, W. Willinger and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic",*ACM SIGComm '93*, San Francisco, CA, USA, Sep. 1993

[9] M. Mathis, J. Semske, J. Mahdavi and T. Ott, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm", *Computer Communication Review*, 27(3), July 1997

[10] R. Morris, "TCP Behavior woth Many Flows", *IEEE International Conference on Network Protocols*, October 1997.

[11] NS software and documentation is available at the following site: http://www-mash.CS.Berkeley.EDU/ns

[12] 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.

[13] J. Padhye, V. Firoiu, D. Towsley and J. Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation", *Proccedings of SIGCOMM'98*, Vancouver, CA, September 1998

[14] H. O. Peitgen, H. Jurgens and D. Saupe, "Chaos and Fractals - New Frontiers of Science", *Springer-Verlag*, NY, 1992

[15] N. H. Packard, J. P. Crutchfield, J. D. Farmer and R. S. Shaw, "Geometry from a Time Series", *Phys. Rev. Lett.*, 45 (1980) 712-716

[16] M. S. Taqqu, W. Willinger and R. Sherman, "Proof of a Fundamental Result in Self-Similar Traffic Modeling", *Computer Communication Review*, 27 (1997) 5-23.

[17] M. S. Taqqu, V. Teverovsky and W. Willinger, "Estimators for long-range dependence: an empirical study", *Fractals*, Vol 3, No. 4. pp. 785-788, 1995.

[18] M. S. Taqqu, V. Teverovsky and W. Willinger, "Is Network Traffic Self-Similar or Multifractal?", *Fractals*, 5 (1997) 63-73