On Dynamics of Network Transport Protocols

Nageswara S. V. Rao
Center for Engineering Science Advanced Research
Computer Science and Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831, USA
raons@ornl.gov

Leon O. Chua
Department of Electrical
Engineering and Computer Science
University of California
Berkeley, CA 94720 , U.S.A.
chua@fred.eecs.berkeley.edu

Abstract – *TCP is the most widely used protocol in computer networks and accounts for a large fraction of the Internet traffic. TCP deployed over wide-area networks together with the routers represents a distributed non-linear dynamical system with delayed feedback. We characterize the state space of TCP using congestion window size, end-to-end packet delay, and also the number of packet retransmissions and acknowledgments. We present an analytical model for the dynamics of a simplified version of TCP by a suitable composition of two unstable linear-like regimes. This model generates bounded trajectories with a complicated attractor. We show that its dynamics embed a tent-like map for the update of window sizes which generates chaotic dynamics. We present simulation results that indicate chaotic dynamic behavior of TCP in interacting with drop-tail routers with small buffers and background UDP traffic. We also present wireless and wireline Internet measurements that indicate chaos-like end-to-end delay response for fixed size messages sent at regular intervals.*

Keywords: computer networks, transport control protocol, chaotic dynamics, tent map.

1 Introduction

The Transport Control Protocol (TCP) is a connection oriented network mechanism that provides reliable data transport from a source to a destination over the IP (Internet Protocol) networks [13]. TCP is the most widely used protocol for computer networks, and accounts for a vast majority of the traffic over the wide-area networks, particularly the Internet. TCP is extremely effective in providing reliable transport of bulk data, but it has not been as effective in the control applications over long-haul network connections. For example, it is quite challenging to achieve a stable closed-loop control of a manipulator located on the west coast from a location on the east coast of United States. Often, the control of fast mechanical devices over long-haul Internet connections (using socket-based TCP data transfers) suffers from two problems: (a) lack of responsiveness of the device, and (b) presence of high frequency components that produce uncontrolled motions. The conventional controllers designed for electrical connections with very short delays are particularly vulnerable when implemented over the Internet. The difficulty is that TCP is known to exhibit complicated dynamics over various time scales. While such behavior does not directly affect the large bulk transfers, it can have serious negative effects on the controllability and stability of control loops implemented over wide-area networks.

The next generation of network applications, such as instrument grids, remotely deployed mobile robot teams and interactive simulations distributed on supercomputers, require stable control mechanisms over wide-area networks. Thus it is important to understand the dynamics of TCP at the time scales appropriate for the application at hand. TCP dynamics also play a crucial role in other scenarios that require high throughputs over large time scales. For example, the parallel-TCP utilizes the collective dynamics of TCP to achieve throughput rates that are significantly larger than a single TCP stream [17, 20]. Also, methods such as the dynamic right-sizing manipulate the TCP window dynamics to overcome the buffer size limitations [5]. In this paper, however, our main focus is the dynamics at the time scales of the order of congestion window-size updates. This time scale for wide-area networks is typically of the order of few milliseconds but can be as large as hundreds of milliseconds depending on the connection and traffic levels.

TCP dynamics have been studied under various formulations, and its complicated nature been observed in many cases. The recent work of Veres and Boda [23]

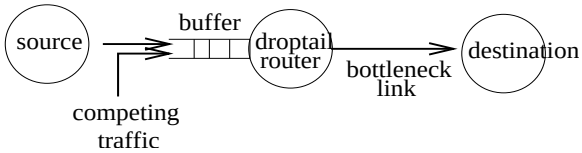

Figure 1: *TCP interacting with a drop-tail router.*

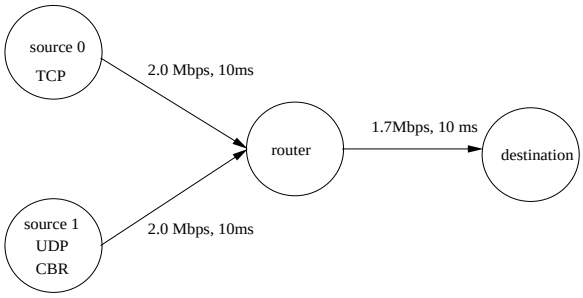

Figure 2: *Simulation setup.*

illustrated, using ns-2 simulations, the chaotic dynamics of two competing TCP streams interacting with a single router using simulation. They showed that the congestion window dynamics of both streams generate chaotic time series, but no formal explanations, such as routes to chaos, have been presented there. Ranjan and Abed [14] showed that TCP interacting with a Random Early Detection (RED) routers exhibits periodic doubling which eventually leads to chaos in the buffer occupancy rates. In this paper, we restrict our focus to the formulations related to the above two references, namely TCP interacting with a router in presence of indigenous traffic as shown in Figure 1. In particular for simulations, we consider a basic version of TCP interacting with drop-tail routers in presence of background UDP traffic.

From a dynamics view point, TCP deployed over wide-area networks together with the routers represents a non-linear dynamical system with delayed feedback. In general, such systems can be chaotic even under very simple formulations, and in fact it is less often that such system are non-chaotic [2]. Furthermore, the feedback delay plays a crucial role in nonlinear systems [9]: for ordinary differential equations, a 1-dimensional system with delayed feedback can be chaotic, whereas without feedback it takes at least the dimensionality of 3 to generate chaos [24].

TCP together with its interactions with the routers is a very complicated dynamical system, and it is extremely hard to capture all its details using concise models. In particular, a single differential equation that is amenable to the standard tools from chaos theory [12] is not obvious. Instead, we make several simplifications in TCP and traffic behavior to obtain a model that is similar to some of the known chaotic systems. Our model and analysis method is similar in spirit to that studied by Sparrow [21], where a system composed of two "glued" unstable linear systems exhibits chaos under certain conditions.

We characterize the state space of TCP using the congestion window size, end-to-end packet delay, the number of re-transmissions and acknowledgments. We model TCP by a suitable composition of two unstable linear-like regimes each of which generates bounded dynamics with a very complicated attractor. In regime one, the congestion window-size constantly grows in response to the successful packet delivery, while the packet delay is relatively stable. This regime is followed by the second regime, wherein the packet delay becomes unstable due to overflowing of router buffer and in response the window-size is drastically reduced. TCP dynamics move back and forth between these two regimes and are cyclic in the absence of delays and buffers at the routers. In particular, if the network delays are negligible, the second regime is extremely short-lived resulting in the familiar saw-tooth behavior of TCP for large flows. The existence of network delays prolongs the duration of regime two, which when coupled with the background traffic and small buffers generates very complicated dynamics under certain conditions.

We show that the dynamics of the window-size updates embed a map which is qualitatively similar to the well-known tent map [2] that generates chaotic trajectories. The former, however, is more complicated and has gaps in certain regions of the domain. Full analysis of this map will be presented in a forth-coming paper.

We adopt the following informal working definition of chaos (more formal treatments can be found for example in [24]): (a) time trajectories include non-periodic orbits; (b) trajectories are very sensitive to input conditions, namely, trajectories starting at nearby points move significantly farther apart in time, and (c) the attractor set is very complicated.

We present two types of experimental results to complement our analysis.

(a) Using ns-2 simulation we explicitly illustrate the chaotic behavior by computing the time series of window sizes and packet delays together with their Fourier spectra and the attractor sets in the Poincare plane. We consider two specific cases: (i) long message streams where the source has unlimited amount of data to send, and (ii) short bursts wherein messages are sent in an evenly spaced on/off streams. In both cases the simulation re-

sults show chaos-like behavior, more so in the latter as to be expected.

(b) We also present Internet measurements that indicate chaos-like dynamic behavior exhibited by TCP both using wireless and wireline connections. Here we send a fixed size message at regular intervals and measure the end-to-end delay for each message. These measurements indicate very complicated dynamic profiles for the end-to-end delay in response to a stream of evenly spaced messages of fixed size.

We wish to point out the obvious limitations of our results. First, we only discuss a model for a simplified TCP and highlight the conditions for chaotic dynamics. The experimental results thus may not exactly match the results predicted by the model. In particular it is not clear if the chaotic behavior will be dominant for TCP in real-life network conditions. Particularly for Internet measurements, the collected information is very limited due to the lack of access to the internal TCP parameters. Nevertheless, both analytical and experimental results point to rather complicated dynamics of TCP, particularly, at the time scales needed for control applications.

This paper in organized into two complementary parts. In Section 2, we describe the state-space of TCP, and its dynamics in dealing with a bottleneck link with a small buffer. In Section 3, we describe simulation results based on ns-2 followed by the measurements collected over Internet.

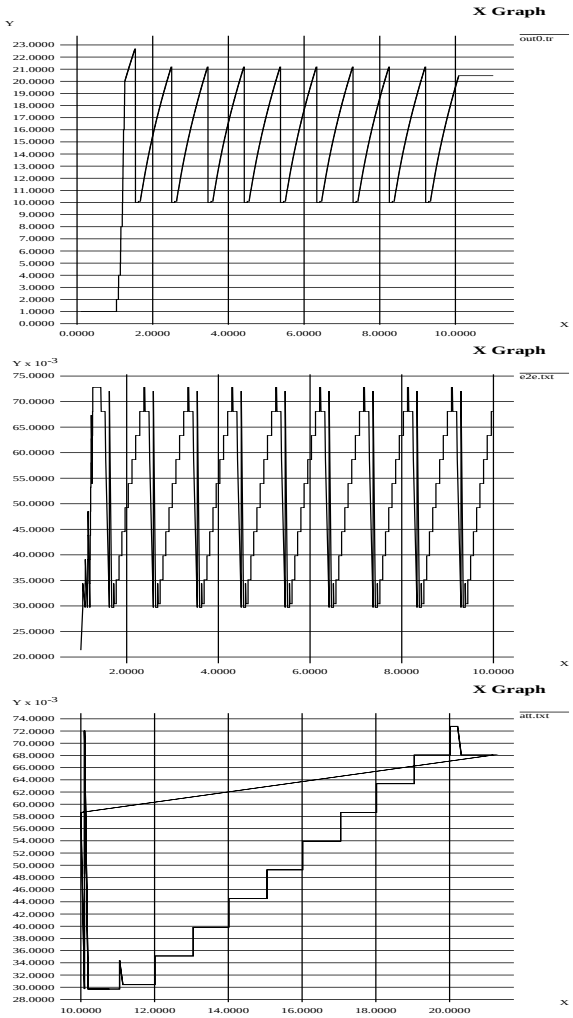

Figure 3: *Top and middle plots show $w(t)$ and $e(t)$ vs t, respectively, and bottom plot shows $w(t)$ vs $e(t)$.*

2 Dynamics of TCP

In this section we first describe a simplified model of TCP, followed by a description of its state space variables. Then we show the decomposition of TCP dynamics into two distinct unstable regimes, which gives rise to chaotic dynamics.

2.1 Simplified TCP Model

TCP provides a connection-based reliable transport mechanism from a source to a destination [13]. The sender interacts with the destination to ensure a reliable delivery of packets; each received packet is specifically acknowledged by the destination. At any given time, there are no more than a certain number of unacknowledged packets, given by the window size, at the source. These include the packets that are in flight, dropped at a router or destination, or whose acknowledgments have been delayed or lost. The individual packets and their acknowledgments can potentially travel different paths and delayed by different amounts (indefinitely if they have been dropped). The packets are put into the proper order at the destination. The receiver maintains a certain window, called the receive window or buffer, of packets which are not in the correct order. Packets arriving at the destination when the receive buffer is full are simply dropped. Source maintains the flow window which corresponds to the receive buffer. The packet flow rate is restricted by the flow window which it typically fixed at the time of initiation of the connection.

The source attempts to adjust its sending rate in response to the traffic condition on the connection, as per a process called the congestion control. There are a large number of variations of the basic TCP [6] in the manner the congestion control is implemented. We restrict our focus on a simplified model of the basic version that was originally described in [8]. TCP maintains the congestion window $w(t)$ which limits the

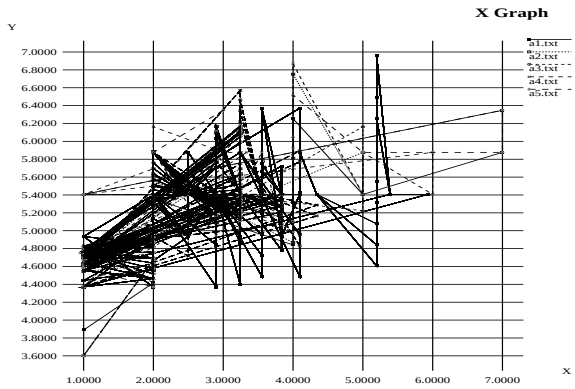

Figure 4: *Attractor in $w(t) - e(t)$ Poincare plane as UDP traffic is increased from 0 to 1.6 Mbps*

number of unacknowledged packets at the source (in addition to the flow window constraint). Since flow window is fixed in the basic version, the dynamics are due to the congestion window; note however, if methods such as dynamic right-sizing combined with parallel streams [16] are applied, the dynamics are not solely dependent on congestion window alone.

2.2 Dynamic Regimes of TCP

TCP responds to the network traffic by adjusting $w(t)$ which in turn controls its flow rate. TCP dynamics consist of two phases as shown in the top plot of Figure 3: (a) initial slow-start phase, and (b) the subsequent congestion control phase. During the slow start phase the window size grows fast until a loss is inferred or a set threshold W_t is reached. In congestion control phase, $w(t)$ is incremented with each acknowledgment until a loss occurs, and then it is reduced by half. The value of $w(t)$ has a significant effect on the packet end-to-end delay (defined formally in the next section). If $w(t)$ is small most messages will reach the destination and hence packet delay is fairly stable. At the other extreme, if $w(t)$ is large, losses and retransmissions occur which increase the end-to-end packet delay.

Consider that the delays are extremely small such that source knows immediately The congestion window size $w(t)$ is recomputed based on the response to the packets sent. After every acknowledgment of a new packet, $w(t)$ is recomputed as follows:

$$\text{if } w < W_t, \text{ then } w \leftarrow w + 1 \text{ during slow start}$$

$$\text{else } w \leftarrow w + 1/w \text{ during congestion control}$$

After a loss is inferred at the source, the window size is reduced by half such that $w \leftarrow w/2$. The above method is referred to as the Additive Increase and

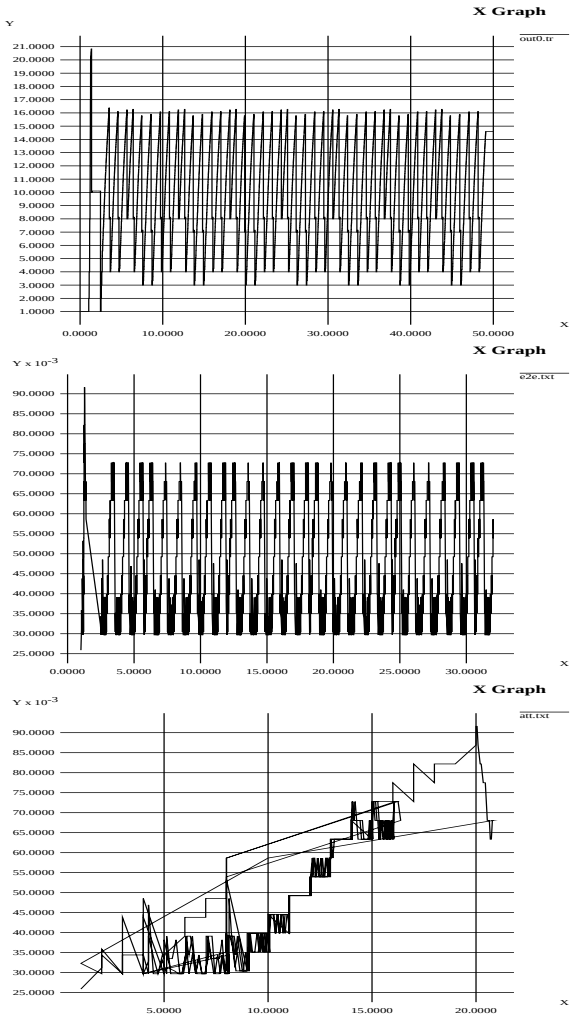

Figure 5: *Long TCP stream with UDP rate of 0.5Mbps.*

Multiplicative Decrease (AIMD) method for congestion control. We make the assumption that the loss is inferred at the source almost instantaneously and the acknowledgments are not lost. This is a simplified description of the congestion control but captures the essential components of TCP needed here.

2.3 Dynamic Regimes of TCP

In the congestion control phase, there are two distinct regimes denoted by $\mathcal{R}_1$ and $\mathcal{R}_2$ as typified in the top plot of Figure 3. In $\mathcal{R}_1$ TCP starts with a low value of $w(t)$ and keep incrementing as along as the packets are being acknowledged. When packet loss is inferred due to explicit notification or time-out, regime $\mathcal{R}_2$ is entered, wherein w is drastically reduced, particularly so in case of multiple losses.

2.4 State-Space Characterization of TCP

In addition to $w(t)$, we characterize the TCP dynamics using three other variables. Let the time instant t_1, called the *epoch*, correspond to the time a packet transmission started by TCP for first time for this packet. For epoch t_1, let $e(t_1)$ denote the *end-to-end delay* of the packet under the assumption that the acknowledgments are not lost. We extrapolate $e(t)$ for t in between two consecutive epochs t_1 ans t_2 by $e(t) = e(t_1)$. Let $r(t)$ and $a(t)$ denote the number of retransmissions and acknowledgments, respectively since the start; note that they both are non-decreasing functions of t. Usually, the sending rate of TCP is specified by $w(t)$ per round trip time; for simplicity of presentation we assume that $w(t)$ has been scaled for unit time. Thus the number of packets sent from the source during time interval $[T_1, T_2]$ is given by $\int_{T_1}^{T_2} w(t)dt$.

We consider that TCP is responding to a simple network scenario as in Figure 1 with a single bottleneck link with maximum packet delivery rate of w_b. The packet transmission through this link is controlled by by a drop-tail router with buffer size B_τ. There is an underlying traffic that arrives at the router which competes for the link bandwidth and buffer space. Let $B(t)$ denote the number of elements in the buffer at time t with $B_\tau - B(t)$ denoting the free buffer space. We assume that there is a processing delay of a packet at the router that increases slowly at a rate μ with the sending rate at the source.

Consider that there is no competing traffic at the router and source rate is fixed (by some non-TCP mechanism) such that $w(t) = w_f \leq w_b$. Then no packets will be dropped and dynamics of $w(t)$ are constant at w_b. In general w_b is not known and varies with the competing traffic. Roughly speaking, TCP attempts to estimate w_b, and to keep $w(t)$ lower than the estimated value to avoid packet losses and retransmissions.

Consider that the delays are extremely small such that source knows immediately after a packet is dropped and also that $B_\tau = 0$. In such case, dynamics of $w(t)$ for TCP are periodic: in regime $\mathcal{R}_1$, $w(t)$ starts around $w_b/2$ and increases with the acknowledgments until it exceeds w_b; then it infers a single packet loss and enters regime $\mathcal{R}_2$, where $w(t)$ is reduced by half. Here regime $\mathcal{R}_2$ is very short lived since it is the result of a single instantaneous loss. Now consider that the delays are non-zero and it takes T_1 units of time before the loss is inferred since the source rate exceeds the available bandwidth. Consider that $w(t_1) = w_b$ such that at time t_1 the sending rates becomes equal to the bottleneck bandwidth. TCP sends

$$n_{[t_1, t_1+T_1]} = \int_{t_1}^{t_1+T_1} w(t)dt$$

packets in the interim period $[t_1, t_1 + T_1]$ of which

$$n_d = \int_{t_1}^{t_1+T_1} (w(t) - w_b)dt = [n_{[t_1,t_1+T_1]} - w_b T_1]_+$$

will be dropped, where $[x]_+$ is x if $x > 0$ and is 0 otherwise. If buffer size is non-zero, then no packets will be dropped until $t = t_1 + t_{B_\tau}$ such that

$$B_\tau = \int_{t_1}^{t_1+t_{B_\tau}} [w(t) - w_b]_+ dt$$

if the entire buffer B_τ is available. Notice that the availability of buffer delays the time the packet is dropped by t_{B_τ} in this case. Thus the number of packets dropped will be

$$n_{B_\tau} = \int_{t_1+t_{B_{tau}}}^{t_1+T_1} [w(t) - w_b]_+ dt \qquad (2.1)$$

for $t_{B_\tau} < T_1$. If the entire buffer is not available, the computation of the number of dropped packets is more involved. Let t_r denote the time the packet that left the source at time t_1 reaches the buffer. Then let $t_b \geq t_1$ be the earliest time such that the packet sent from source at t_b arrives at full buffer at time t_f such that $B(t_f) = B_\tau$, i.e. earliest time that the packet leaving the source faces a full buffer. Let $w_S(t)$ denote the rate measured at the source with which the packets will eventually pass through the bottleneck link. Note that if there is no competing traffic we have the simple relation

$$w_S(t) = \begin{cases} w(t) & \text{if } w(t) \leq w_b \\ w_b & \text{if } w(t) > w_b \end{cases}$$

Then the number of dropped packets is given by

$$n_b = \int_{t_b}^{t_b+T_1} (w(t) - w_S(t))dt.$$

Again note that the available buffer space delays the packet loss by tme $t_b < T_{B_\tau}$, and since $w_S(t) \leq w_b$ we have $n_b \geq n_{B_\tau}$. Then after all the n_b dropped packets are accounted for, the resultant window-size is set to $w(t_1)/2^{n_b}$ for the regime $\mathcal{R}_2$.

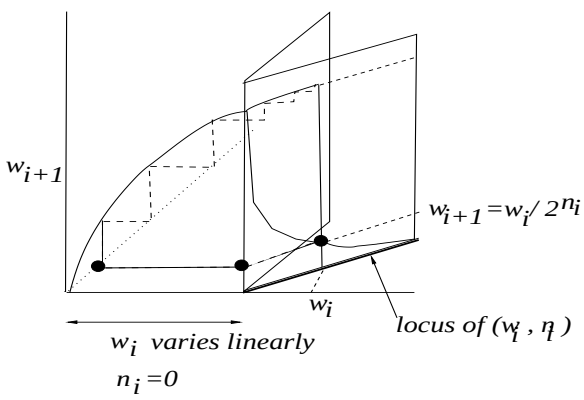

Figure 6: *Examples of map M for zero delays and zero buffer size for various B_τ.*

2.5 Dynamic Regimes of TCP

Consider that the source is sending an infinitely long message. In case of zero delays, TCP dynamics in congestion control mode constitute a stable cycle with period $T_{\mathcal{R}_1} + T_{\mathcal{R}_2}$, where $T_{\mathcal{R}_i}$ is the duration of regime $\mathcal{R}_i$. This stable cyclic behavior is well-known in TCP literature as shown in the top plot of Figure 3 which corresponds to the ns-2 simulation with no background UDP traffic; despite its ubiquity in literature, we observed such nice periodic behavior only under very special conditions in simulation. The corresponding packet end-to-end delay is shown in the middle plot of Figure 3 where in $e(t)$ increases slowly during $T_{\mathcal{R}_1}$ but jumps in large steps briefly due to retransmissions; nevertheless, it is still cyclic. In the bottom plot of Figure 3, we show the periodic sampling of $e(t)$ and $w(t)$ corresponding to the Poincare map in $w(t)-e(t)$ plane. After an initial transient period, the trajectories of $w(t)$ and $e(t)$ settle into periodic motions, and the corresponding attractor in $w(t)-e(t)$ plane is represented by a closed curve.

We can conceptualize the dynamic behavior of TCP, in term of the alternating regimes $\mathcal{R}_1$ and $\mathcal{R}_2$, which are of varying duration in general. For t corresponding to $\mathcal{R}_1$, the dynamics can be approximated locally as follows

$$\begin{bmatrix} \frac{dw}{dt} \\ \frac{de}{dt} \\ \frac{da}{dt} \end{bmatrix} = \begin{bmatrix} 1/w\frac{da}{dt} \\ \mu\frac{da}{dt} \\ \frac{da}{dt} \end{bmatrix}$$

Note that $\frac{dr}{dt}(t) = 0$ during this period, and the acknowledgments arrive at a non-zero rate. Computation of Eigenvalues of the Jacobian matrix shows that at least one of them will have positive real part; hence the system is unstable. One can intuitively draw such conclusion since the arrival of acknowledgments constantly increases $w(t)$ at a local rate of $1/w$ and also increases $e(t)$ at a slow rate of μ.

Now consider the regime $\mathcal{R}_2$ wherein the source infers a packet loss due to the buffer overflow. The packets sent during (t_1, t_1+T_2) will result in overflow where $w(t_1) = w_b$, for t_1 in $\mathcal{R}_1$; the exact number dropped packets depends on the occupancy of the buffer. The behavior can be approximated locally by

$$\begin{bmatrix} \frac{dw}{dt} \\ \frac{de}{dt} \\ \frac{dr}{dt} \end{bmatrix} = \begin{bmatrix} \frac{w}{2^{-\frac{dr}{dt}}} \\ \eta\frac{dr}{dt} \\ \frac{dr}{dt} \end{bmatrix}$$

for some η that depends on the delays. In this regions $\frac{da}{dt}(t) = 0$ and $\frac{dr}{dt} > 0$. Intuitively, this is an unstable regime since the packets are dropped which increases $e(t)$ and drastically reduces $w(t)$. This can also be verified by explicitly computing the Eigenvalues of the Jacobian matrix.

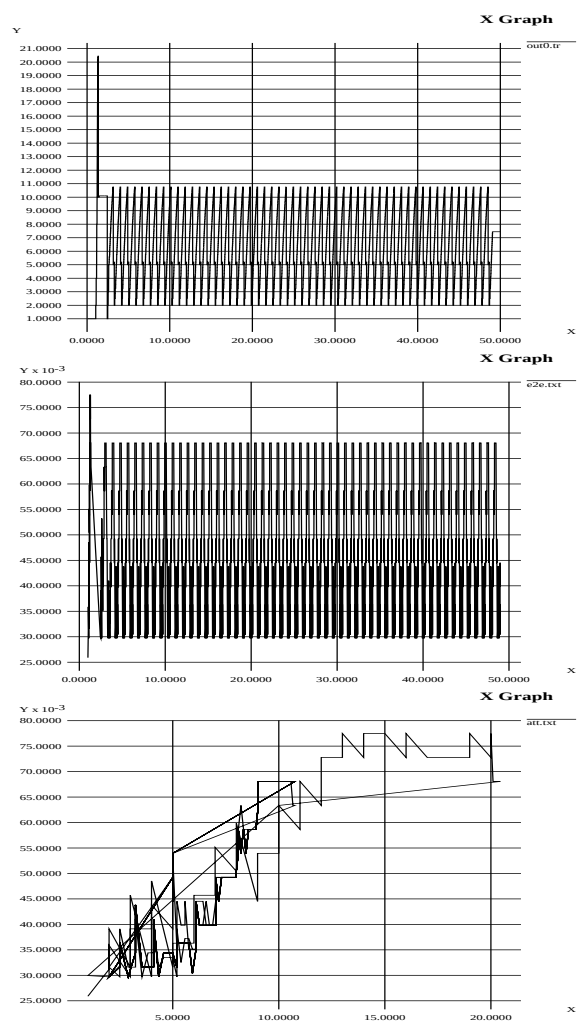

Figure 7: *Long TCP stream with UDP rate of 1.0Mbps.*

The TCP dynamics are due to the "gluing" together

of the regimes $\mathcal{R}_1$ and $\mathcal{R}_2$. The above equations in region $\mathcal{R}_1$ can be derived by ignoring the control terms in the fluid models of [11, 7] which are known in the literature. The dynamics in region $\mathcal{R}_2$ are a direct result of using $e(t)$ as a state variable, which plays a very critical role in our analysis; we are unaware of earlier works using this variable and highlighting its role. Note that both regimes are unstable in that there is no stationary point in either. Thus the trajectories in one regime will enter the other and vice versa thereby generating bounded trajectories. The period behavior is one of the way such trajectories can manifest but background traffic coupled with small buffer sizes result in more complicated trajectories. Such behavior appears not only in the time series of $w(t)$ and $e(t)$ but also in the attractor and the Fourier spectra of $e(t)$ as will be discussed in the simulation results in the next section. Note that as the background traffic is varied the attractor, which is a simple curve in Figure 3 becomes much more complicated as in Figure 4. The boundary between the two regimes is defined by the transition between the increasing and decreasing values for $w(t)$. We identify a particular subset of the state space where $w(t)$ reaches a value that just causes buffer overflow in regime $\mathcal{R}_1$. Under no delays and buffers this condition is met when $w(t) = w_b$, and in general is given by $n_d = 0$ in Eq (2.1), which is not easily visualized. As $w(t)$ is increased beyond this value in $\mathcal{R}_1$, TCP will infer loses and transit to regime $\mathcal{R}_2$. The resultant $w(t)$ value in $\mathcal{R}_2$ depends on n_d given by $w \leftarrow w/2^{n_d}$, which in turn depends on $B(t)$ during the appropriate period.

To understand the time evolution of $w(t)$ we define *w-update* map $M : [1, W_{\max}] \mapsto [1, W_{\max}]$ such that $M(w_i) = w_{i+1}$ gives the earliest changed value of $w(t)$ since the time the condition $w(t) = w_i$ is met. Let $w_i \in \mathcal{R}_j$ for $j = 1, 2$ denote that when $w(t) = w_i$, TCP is in regime $\mathcal{R}_j$. Then this map is specified as follows:

$$M(w_i) = \begin{cases} w_i + 1/w_i & \text{if } w_i \in \mathcal{R}_1, w_{i+1} \in \mathcal{R}_1 \\ w_i/2^{n_i} & \text{if } w_i \in \mathcal{R}_1, w_{i+1} \in \mathcal{R}_2 \\ w_i/2^{n_i} & \text{if } w_i \in \mathcal{R}_2, w_{i+1} \in \mathcal{R}_2 \\ w_i + 1/w_i & \text{if } w_i \in \mathcal{R}_2, w_{i+1} \in \mathcal{R}_1 \end{cases}$$

where n_i is the number of packet losses inferred during the period. In $\mathcal{R}_1$ we have $n_i = 0$ hence it can be represented as a simple map. In $\mathcal{R}_2$ it is more complicated and can be visualized as in Figure 6. For $w(t) > w_b$, there could be packet losses and but $w(t)$ attains a value higher than w_b. This process can be imagined in terms of a set L along which n_i increases together with $w(t)$. At any point on this set L, corresponding to (w_i, n_i) the map M is specified by $M(w_i) = w_i/2^{n_i}$. The plot of M can be imagined along the set $L \times [1, N_{\max}]$ so that L forms the x-axis

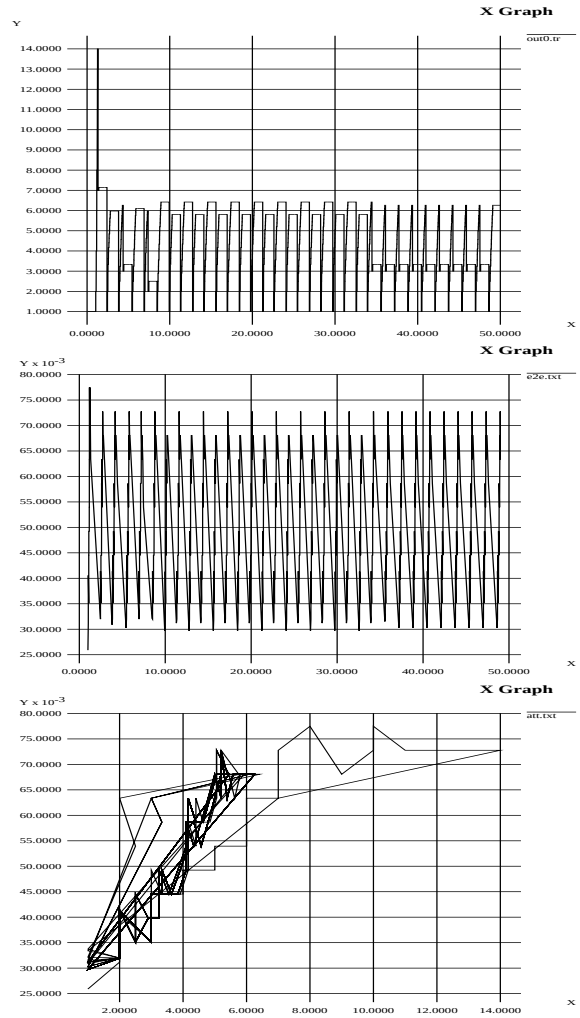

Figure 8: *Long TCP stream with UDP rate of 1.5Mbps.*

and $M(w_{i+1})$ is along the y-axis. Empirical computation of M is shown in Figure 10 as a function of w_i alone based on the simulation results in Section 3.1; note however that this map reflects only the limited number of simulated trajectories.

Periodic trajectories can be easily generated as illustrated in the Figure 6 in which the trajectory stays in $n_i = 0$ plane for part of the time and on the set L for the rest of the time in a cycle. This map shares some of the basic properties of the tent map [2] in that it has a monotonically increasing part in $n_i = 0$ plane and a decreasing part in the half plane $n_i > 0$. The map M is more complicated due to the presence of n_i which makes it two-dimensional in the half plane $n_i > 0$. It is instructive to visualize the map M in terms of a family of maps such that $M_b : [0, W_{\max}] \times [0, N_{\max}] \to [0, W_{\max}] \times [0, N_{\max}]$, is the two-dimensional map obtained in case of no back-

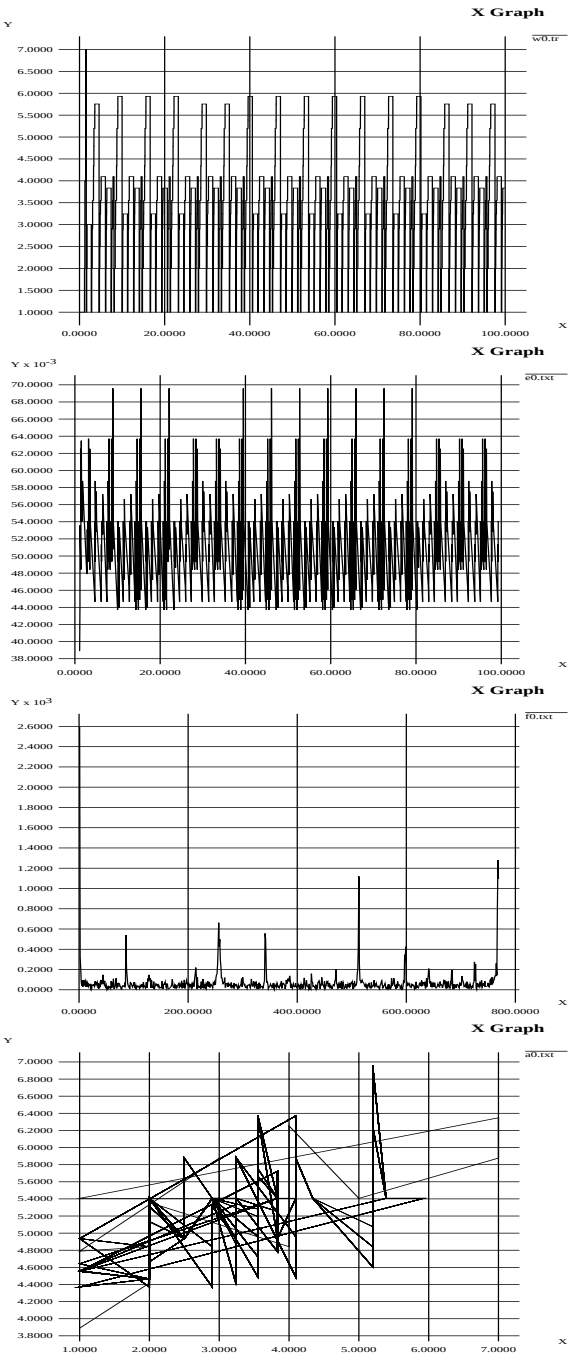

Figure 9: *Bursty TCP stream with 0.15 sec square wave with UDP rate of 1.6 Mbps.*

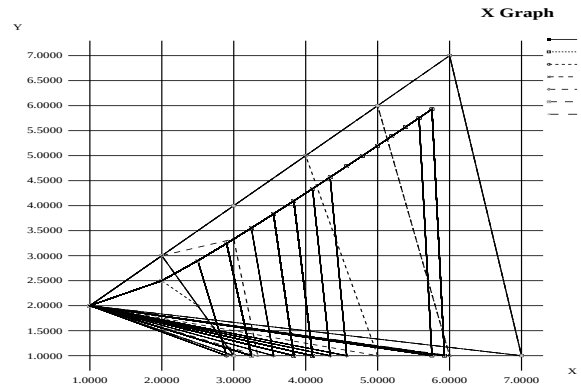

Figure 10: *Empirical map M plotted along w_i-axis.*

ground traffic and buffer size b, $0 \leq b \leq B$, i.e. entire buffer is available for this TCP stream as shown in Figure 11. Due to non-zero delay, $w(t)$ exceeds w_b at time t_1 for a time period T which will result in the over flow of

$$n_0(T) = \int_{t_1}^{t_1+T} (w(t) - w_b)dt$$

packets. Then $M_0(w(t_1 + T)) = w_i/2^{n_0}$ whose value depends on T. This map can be imagined along a locus L_0 of all points given by $\{(w(t_1 + t), n_0(t)) : t \in [0, T]\}$. The map M_0 is defined at any point on L_0 as $M_0(w(t_1 + t)) = w(t_1 + t)/2^{n_0(t)}$. We similarly define the locus L_b when the buffer size is b which is entirely available. Let t_b denote the period during which $w(t_1)$ increases before first packet is dropped such that

$$b = \int_{t_1}^{t_1+t_b} (w(t) - w_d)dt.$$

Then the number of overflown packets is given by

$$n_b(T) = \int_{t_1+t_b}^{t_1+t_b+T} (w(t) - w_b)_+ dt.$$

Then the M_b is defined as $M_b(w(t_1 + t_b + T)) = 2^{n_b(T)}$ whose value depends on T. Then we define the original map M by using M_b with the appropriately available buffer size b. The Poincare iterates $\{w_i\}$ under M depend on the available buffer space when $w(t)$ is in the regions $\mathcal{R}_2$ and is given by $M_{B_\tau - B(t)}$. Any trajectory of $w(t)$ stays in in the plane $n_i = 0$ while incurring loses and transits to a locus that lies in between L_0 and L_{B_τ} for a duration determined by $B(t)$ and the delay to the buffer. Then it reduces $w(t)$ in response to the resultant loses and thus jumps back onto the plane $n_i = 0$, and this process repeats.

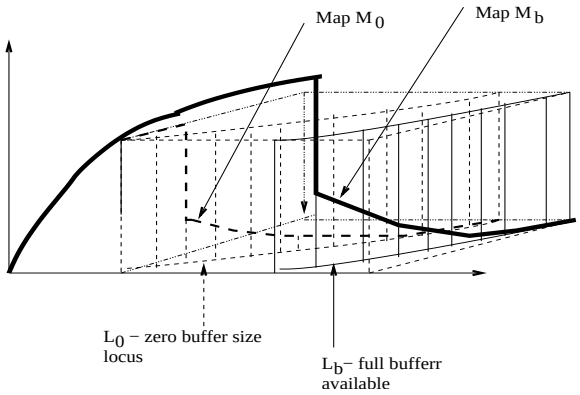

Figure 11: *Illustration of map M in terms of family of two-dimensional maps $M_b(t)$, $0 \leq b \leq B$.*

Unlike the tent map which has period 3 orbits, M starts at a much higher periodicity, can be shown to have any periodicity as well as aperiodic orbits using standard arguments such as Sharkovskii's Theorem [2]. The detailed study of the dynamics produced by this map are under investigation, but its general nature indicates that it generates behavior qualitatively similar to that produced by the tent map. Maps that are qualitatively similar to M have also been presented in [21] to show the chaotic behavior.

3 Experimental Results

In this section, we describe simulation and Internet measurements to illustrate the complicated dynamics of TCP. In simulations, we plot various TCP parameters, and in network measurements we plot the end-to-end delays of fixed size messages.

3.1 TCP competing with UDP

Figure 2 shows the network topology used in ns-2 simulations to perform our experiments. Source nodes 0 and 1 generate TCP and UDP traffic, respectively, that goes to a drop-tail router over 2.0 Mbps links and then to a bottleneck link of 1.7 Mbps before reaching the destination. Source node 1 generates UDP traffic at a constant bit rate (CBR) and serves as the background traffic that the TCP connection must adapt to. To simulate different levels of congestion, the traffic-generation rate is varied from 0 to 1.5 Mbps in steps of 0.5 Mbps. This set-up provides an easy way to increase the traffic on the bottleneck link so that different TCP behaviors can be observed.

3.2 Long Streams

The source node is attached to an FTP client that acts as an infinite source of TCP traffic. The results

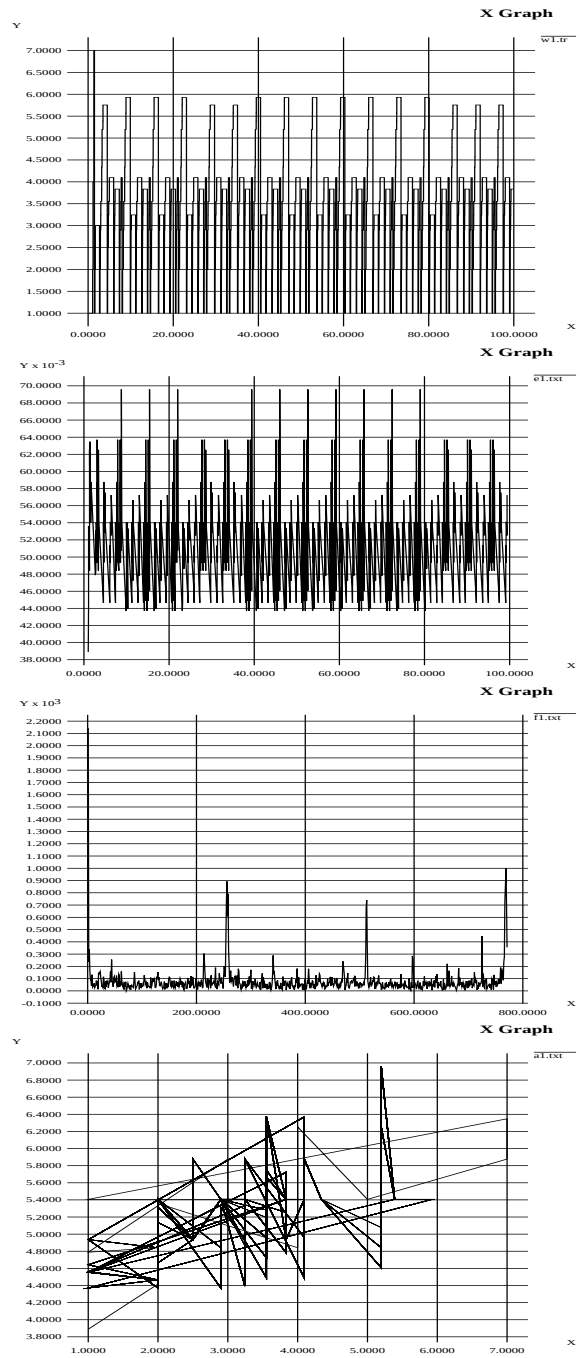

Figure 12: *Bursty TCP stream with 0.15 sec square wave with UDP rate of 1.6 Mbps with starting point delayed by 0.1 sec compared to 9.*

with no UDP traffic are shown in Figure 3; as discussed before both $w(t)$ and $e(t)$ are periodic with the attractor described by a simple curve. For other UDP rates, both $w(t)$ and $e(t)$ did not show repeated patterns during the observation period, and the attractor became significantly more complicated. The composite of attractors at all the UDP loads is shown in Figure 4. When the UDP rate is 0.5Mbps, the dynamics are more complicated as shown in Figure 5. While the trajectories remain bounded in the $w(t)-e(t)$ plane, the shapes of the attractors at various UDP rates appeared quite varied and did not follow any particular evolutionary pattern. At all UDP loads both $w(t)$ and $e(t)$ had significant periodic components but were not actually period – the digression from the periodic nature increased as the UDP rate is increased. For example the $e(t)$ has less periodic component in Figure 8 for a UDP rate of 1.5 Mbps compared to Figure 7 for a UDP rate of 1.0 Mbps. We only showed some typical plots in Figures 3-8 but the general nature of the $w(t)$ and $e(t)$ remained more or less the same. Since the observation time is limited, it is quite possible that these plots are small parts of periodic trajectories. But the radical difference in the attractor shape together with the map M discussed in the last section provide a strong indication that TCP can generates aperiodic orbits for $w(t)$ and $e(t)$.

3.3 Short Bursty Streams

In applications involving large transfers, such as in data archival and high-performance storage tasks, the transfer times not particularly sensitive to the dynamics at the finer time scales. But the effects of TCP dynamics are more pronounced in transfers of small messages such as control signals. To study such effects, we now consider simple on-off ftp source simulating the square-wave function for the message transfers. Such short message are typical in applications involving the instrument control over wide-area networks; for such applications the end-to-end dynamics at the finer time scales are very critical, since jittery dynamics can destabilize the control loops. The profiles of $w(t)$ and $e(t)$ are significantly more complicated as shown in Figures 9-16. In Figures 13-16, we also show the Fourier spectra of $e(t)$ to illustrate the non-periodic nature.

A typical case is shown in Figure 9 where the duration of each ftp session is 0.15 sec with 0.15sec duration between the consecutive sessions. The UDP CBR rate is 1.6Mbps. The top plot shows $w(t)$ which is aperiodic during the observation interval. The plot below it shows $e(t)$ which is also aperiodic but shows much more variation compared to the similar plots in case of long streams; the plot below it shows its magnitude plot of the Fourier coefficients, which shows significant

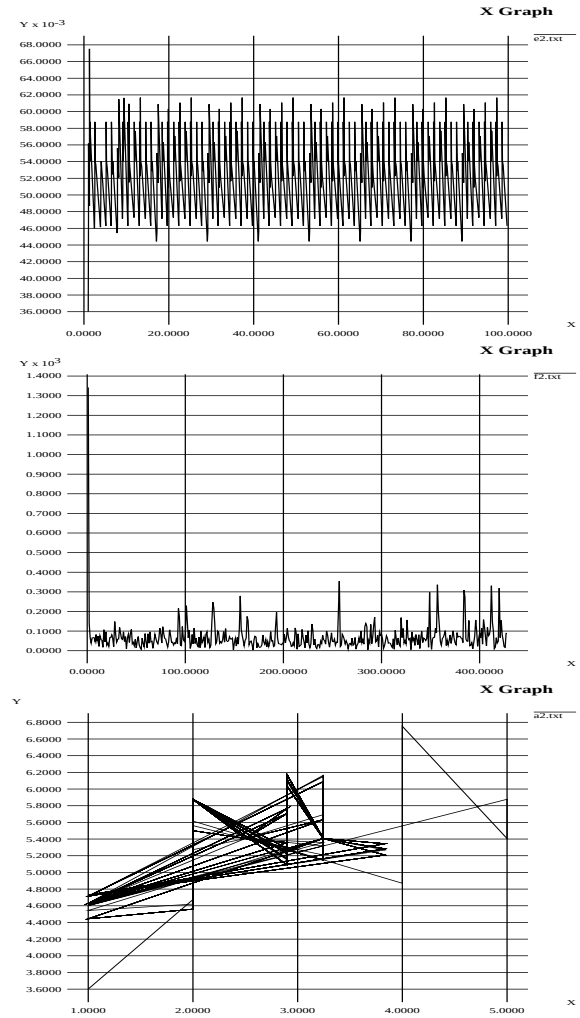

Figure 13: *Bursty TCP stream with 0.16 sec square wave with UDP rate of 1.65 Mbps with starting point same as 12.*

non-periodic parts. The bottom most plot shows the attractor which is also more complicated than in the long stream case.

To show the sensitivity of $e(t)$ with respect to the starting time, we advanced the start tome of the ftp by 0.1 sec, which resulted in significant changes as reflected in the Fourier spectrum shown in Figure 12. The $w(t)$ and $e(t)$ are also quite different and in particular they are not simply time shifted versions of those in Figure 7. However, it is interesting to note that the attractor is the same as before.

For each of Figures 13 - 16 the top plots show $e(t)$, the middle plots show the Fourier coefficients and the bottom plots show the attractor in $w(t)$–$e(t)$ plane. To show the sensitivity with respect to the UDP rate we increased it to 1.65Mbps and the results are shown in

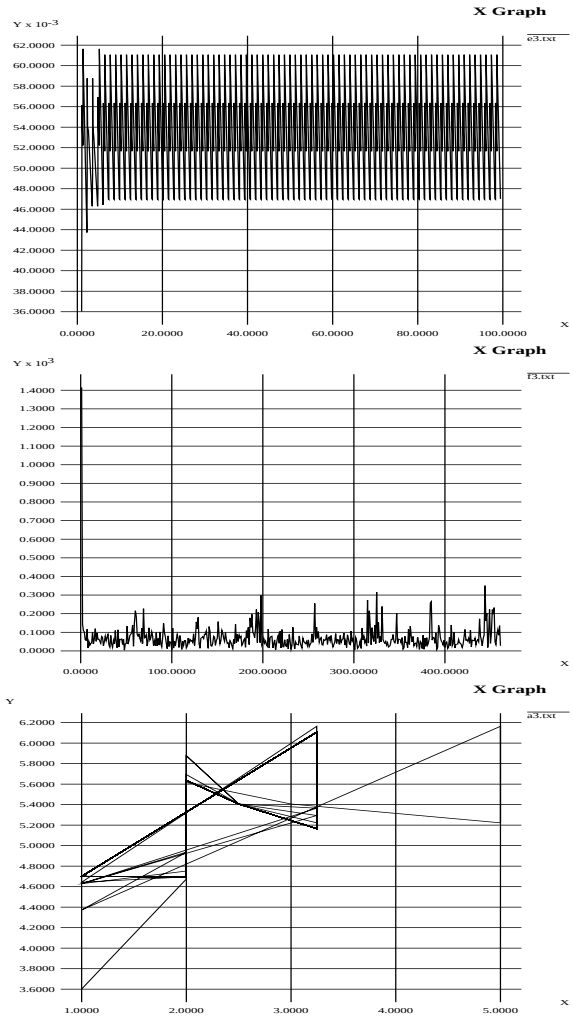

Figure 14: *Bursty TCP stream with 0.14 sec square wave with UDP rate of 1.65 Mbps with starting point same as 12.*

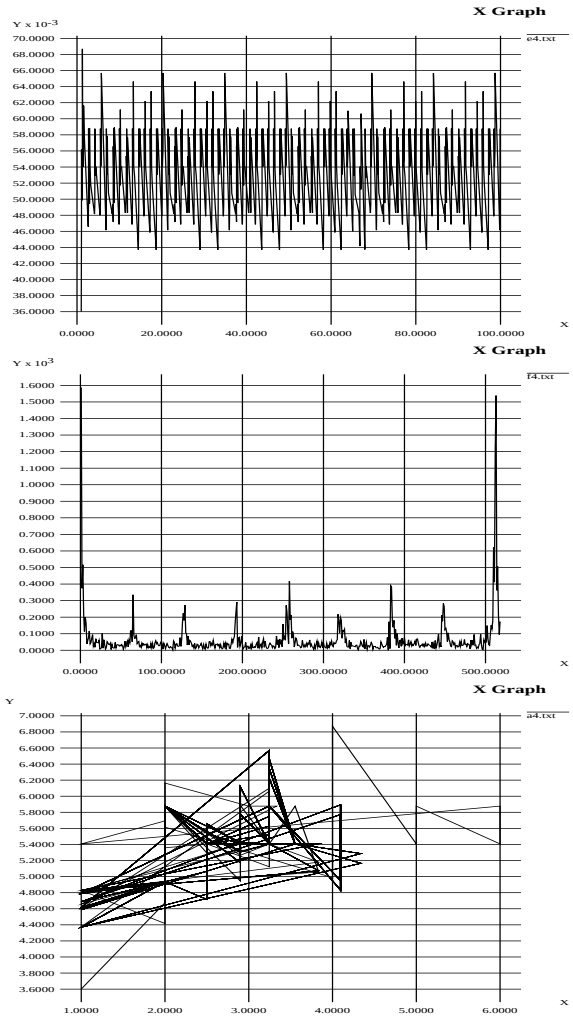

Figure 15: *Bursty TCP stream with 0.14 sec square wave with UDP rate of 1.65 Mbps with starting point same as 12.*

Figure 13. Note that $e(t)$ is still chaotic in appearance although it is quite different from the previous case. Also the attractor is quite different from the previous case. This illustrates the sensitivity of the trajectories of $e(t)$ and $w(t)$ to the background traffic.

Then we changed the ftp duration to 0.16s which is slightly different from the previous case. Interesting enough, $e(t)$ became quite periodic as shown by the Fourier spectrum as in Figure 14. Also the attractor now has a very simple appearance. When the ftp duration is changed to 0.14s the behavior appeared close to the quasi periodic as shown in the Fourier plot in Figure 15. Finally we changed the ftp duration to 0.13sec, which resulted in apparently periodic trajectories with a very simple attractor.

In summary, for this case we have seen evidence for all three components of the chaotic behavior. The trajectories are periodic as well as aperiodic with broad Fourier spectra. They showed extreme sensitivity to various parameters and their attractor is complicated. While these empirical results do not by themselves establish that TCP exhibits chaotic dynamics, they should be viewed in combination of the analytical results presented in the last section.

3.4 Wireline measurements

We collected Internet measurements between Oak Ridge National Laboratory (ORNL) and University of Oklahoma (OU) by sending streams each consisting of messages of fixed size 10K bytes at the regular intervals of 10 seconds. Each message is received at the destination and sent back to the source. The round-

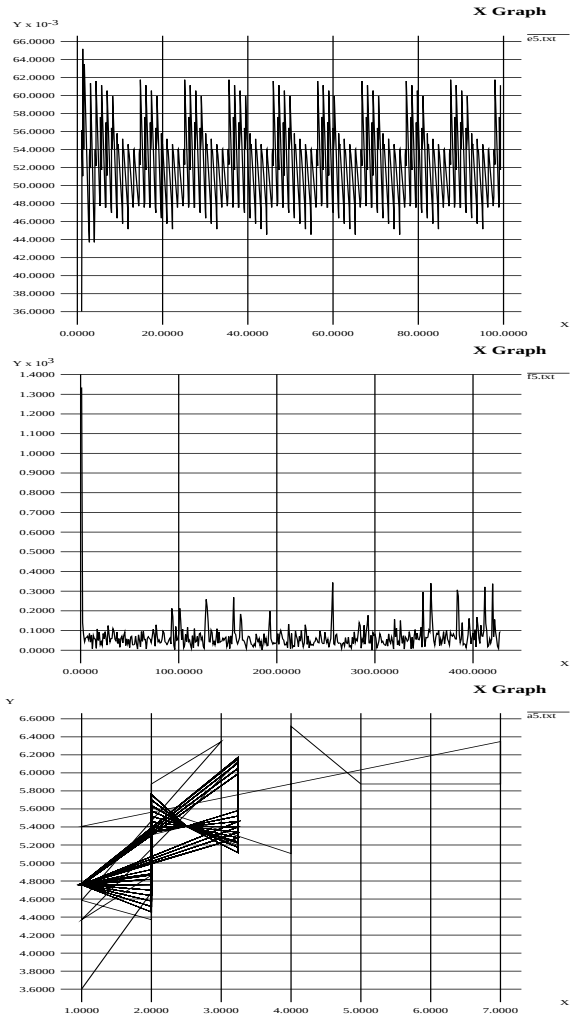

Figure 16: *Bursty TCP stream with 0.13 sec square wave with UDP rate of 1.65 Mbps with starting point same as 12.*

trip time is computed at the source and divided by two to estimate one way end-to-end delay for the message; this is only an estimate since the forward and return paths of a TCP stream could be different, and even in each way different packets may travel different physical paths. Due to lack of access to the TCP internal variables, we only consider the end-to-end delays of the messages rather than the packet-level delays discussed in the last section. But these two quantities are closely related: the end-to-end delay for the message is the difference between the time last packet of the message is received at the destination and the time first packet left the source. Thus the dynamics of the one will be reflected in the other. Each plot of Figure 17 corresponds to three streams each with 20 messages. The individual streams are separated by an hour. The

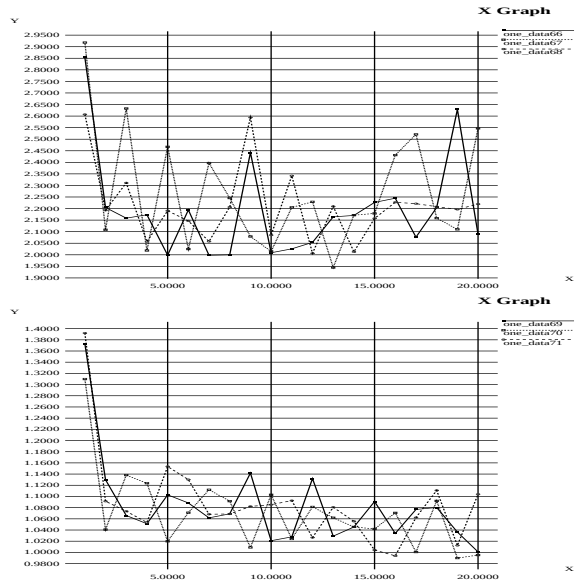

Figure 17: *Wire-line network measurements.*

top and bottom plots in Figure 17 are separated by a day. Notice that there is a considerable variation in the end-to-end delay even for the messages send only a second apart. Such variation is present in measurements collected within a day as well as those collected on different days. These dynamic variations are very important because they can play crucial role in the stability of the control loops between the source and destination.

There are some important differences in these measurements and simulation results of last section. First, the Internet paths from ORNL and OU consists of the parts of ESnet and Abiline networks, each of which in turn consists of a number of routers and perhaps more than one bottleneck link. Also the competing traffic is not entirely UDP. Nevertheless, the dynamic variations in the end-to-end delays are indicative of the qualitative behavior of the dynamics.

3.5 Wireless End Subnetwork

We now describe the measurements collected from a laptop connected to the Internet over a wireless local-area network using the software used in the previous section. The source node is located at the exhibit hall of Supercomputing 2001 conference in Denver and the destination node is at ORNL. Here, each stream consisted of 25 messages each separated by a minute in time. Such measurements are repeated hourly 13 times during the same day and all results are shown in Figures 18 and 19. Here the source node connected to the wireless network at the exhibit hall which had a large variation in the background traffic throughout

the day. The time scales of variation are very sensitive to the hour of the day, ranging from hundreds of milliseconds to hundreds of seconds. In the top plot of Figure 18 measurements of all 13 hourly streams are shown. In each plot below the hourly measurements that contained the highest delay are removed, and same procedure is repeated for all plots of Figure 19. The three orders of magnitude difference in the time-scales of various plots is mainly due to the competing traffic, which was low in the morning and at night but was much higher during the day. It is very interesting to note that given time scale, the end-to-end delays are very dynamic at that scale, and such behavior persisted at all the time scales. Thus for control application in such scenarios, both the time-scales as well as the dynamic variations in the end-to-end message delays are very important. Simply knowing the traffic levels (and hence the hourly time scales) is not sufficient to predict the end-to-end delays, which showed quite a variability at every time scale.

4 Conclusions

We presented analytical and experimental results that indicate the chaotic behavior of TCP at certain time scales. We presented a state-space description of TCP using the window size and end-to-end packet delays. By composing two almost linear but unstable models, we presented a model for TCP dynamics. This model shows that TCP generates bounded but highly complicated dynamics when it interacts with routers with small buffers. We discussed ways in which chaotic trajectories can be generated by TCP, by identifying a tent-like map embedded in the dynamics of TCP. But we consider this work to be a small step toward understanding the complicated nature of TCP dynamics over wide area networks. While such dynamics do not significantly impact large data transfer applications, they could be very important in closed-loop control applications implemented over wide-area networks.

There are several open questions in the area of understanding TCP dynamics. It would be interesting to see the differences in the dynamics generated by various versions of TCP. Our initial ns simulations of Vegas and Reno versions indicate a behavior qualitatively similar to the results presented here with Tahoe version, but the precise parameter values corresponding to various behaviors are quite different. Also we assumed here that $w(t)$ can take and can be initialized with arbitrary real numbers, and it would be interesting to see the effects of finite precision implementation on the TCP dynamics. Note however that a simple linear feedback system can exhibit chaotic behavior when implemented with finite precision [22], and hence it is non trivial to predict the effects of finite precision on TCP dynamics. It would be particularly interesting to see the relevance of such effects in ns-2 simulations, which are finite precision implementations.

Another important issue for further investigation is the practical implications of the chaotic behavior of TCP on real network applications. It is clear that dynamics are most important for control applications. In some cases, multiple path methods can be used to increase the effective bandwidth [15] which can then be traded-off with the jitter by using simple end-filtering. While some chaotic dynamics can be accounted for using such methods, the area of providing stable control loops over wide area networks has received very little attention. It would be interesting to see if there are methods that drive TCP dynamics away from the chaotic regions or control it if chaos become inevitable.

In terms of analysis, a detailed and explicit analysis of the proposed map M will provide more insights into the TCP dynamics. Also, it would be interesting to see if the TCP dynamics can be shown to be amenable to standard methods [12] such as snapback repellers [18, 10], Shilnikov Theorem [19] or delayed feedback effects [4, 3]. It would be of future interest to apply methods such as computation of various Lyapunov exponents on the time series [1] data of window sizes and end-to-end delays, particularly for Internet measurements.

Acknowledgments

This research is sponsored by the High Performance Networking Program of the Office of Science, U.S. Department of Energy, under Contract No. DE-AC05-00OR22725 with UT-Battelle, LLC, the Defense Advanced Projects Research Agency under MIPR No. K153 and by the Laboratory Director's Research Project at the Oak Ridge National Laboratory.

References

[1] *H. D. I. Abarbanel.* Analysis of Observed Chaotic Data. *Springer-Verlag New York, Inc., 1996.*

[2] *K. T. Alligood, T. D. Sauer, and J. A. York.* Chaos: An Introduction to Dynamical Systems. *Springer-Verlag New York, Inc., 1997.*

[3] *U. an der Heiden and H. O. Walther. Existence of chaos in control systems with delayed feedback.* Journal of Differential Equations, *47:273–295, 1983.*

[4] *O. Diekmann, S. A. van Gils, S. M. Verduyn Lunel, and H. O. Walther.* Delay Equations: Functional-, Complex- and Nonlinear Analysis. *Springer-Verlag New York, Inc., 1995.*

[5] *W. Feng, M. Fisk, M. Gardner, and E. Weigle. Dynamic right-sizing: An automated, lightweight, and scalable technique for enhancing grid performance. In* Lecture Notes of Computer Science. *2002. preliminary*

version appeared in IEEE International Workshop on Protocols for High-Speed Networking, April 2002.

[6] *S. Floyd. A report on some recent developments in TCP congestion control.* IEEE Communcations Magazine, *April 2001.*

[7] *C. V. Hallot, V. Misra, and D. Towsley. Analysis and design of controllers for AQM routers supporting TCP flows.* IEEE Trans. on Automatic Control, *2002.*

[8] *V. Jacobson. Congestion avoidance and control. In* Proc. of SIGCOMM, *pages 314–329. 1988.*

[9] *Y. Kuang.* Delay Differential Equations With Applications in Population Dynamics. *Academic Press, Inc, 1993.*

[10] *F. R. Marotto. Snap-back repeller imply chaos in r^n.* Journal of Mathematical Analysis and Applications, *63:199–223, 1978.*

[11] *V. Misra, W. B. Gong, and D. Towsley. Fluid-based analysis of a network of AQM routers suporting TCP flows with an application to RED. In* Proc. of SIGCOMM. *2000.*

[12] *T. S. Parker and L. O. Chua. Chaos: A tutorial for engineers.* Proceedings of the IEEE, *75(8):982–1008, 1987.*

[13] *L. L. Peterson and B. S. Davie.* Computer Networks. *Morgan Kaufman Pub., 2000. Second Edition.*

[14] *P. Rajan and E. H. Abed. Nonlinear instabiliities in TCP-RED. Technical Report TR 2001-37, Institute for Systems Research, 2001. www.isr.umd.edu.*

[15] *N. S. V. Rao. Grid network instruments for end-to-end delay performance, 2002. submitted.*

[16] *N. S. V. Rao and W. Feng. Performance trade-offs of tcp adaptation methods. In* International Conference on Networking, *2002. submitted.*

[17] *D. Salamoni and S. Luitz. High-performance throughput tuning/measurements. In* PPDG Collaboration Meeting. *2000.*

[18] *K. Shiraiwa and M. Kurata. A generalization of a theorem of Marotto.* Nagoya Mathematics Journal, *82:83–97, 191.*

[19] *C. P. Silva. Shilnikov theorem – A tutorial.* IEEE Transactions on Circuits and Systems -I: Fundamental Theory and Applications, *40(10):675–682, 1993.*

[20] *H. Sivakumar, S. Bailey, and R. L. Gorssman. Psockets: The case for applicaiton-level network striping for data intensive applications using high speed wide area networks. In* Proc. of SC2000. *2000.*

[21] *C. Sparrow. Chaos in a three-dimensional single loop feedback system with a piecewise linear feedback function.* Journal of Mathematical Analysis and Applications, *83:275–291, 1981.*

[22] *T. Ushio and C. S. Hsu. Simple example of digital control systems with chaotic rouding errors.* International Journal of Control, *45(1):17–31, 1987.*

[23] *A. Veres and M. Boda. The chaotic nature of tcp congestion control. 2000.*

[24] *S. Wiggins.* Introduction to Applied Nonlinear Dynamical Systems and Chaos. *Springer-Verlag New York, Inc., 1990.*

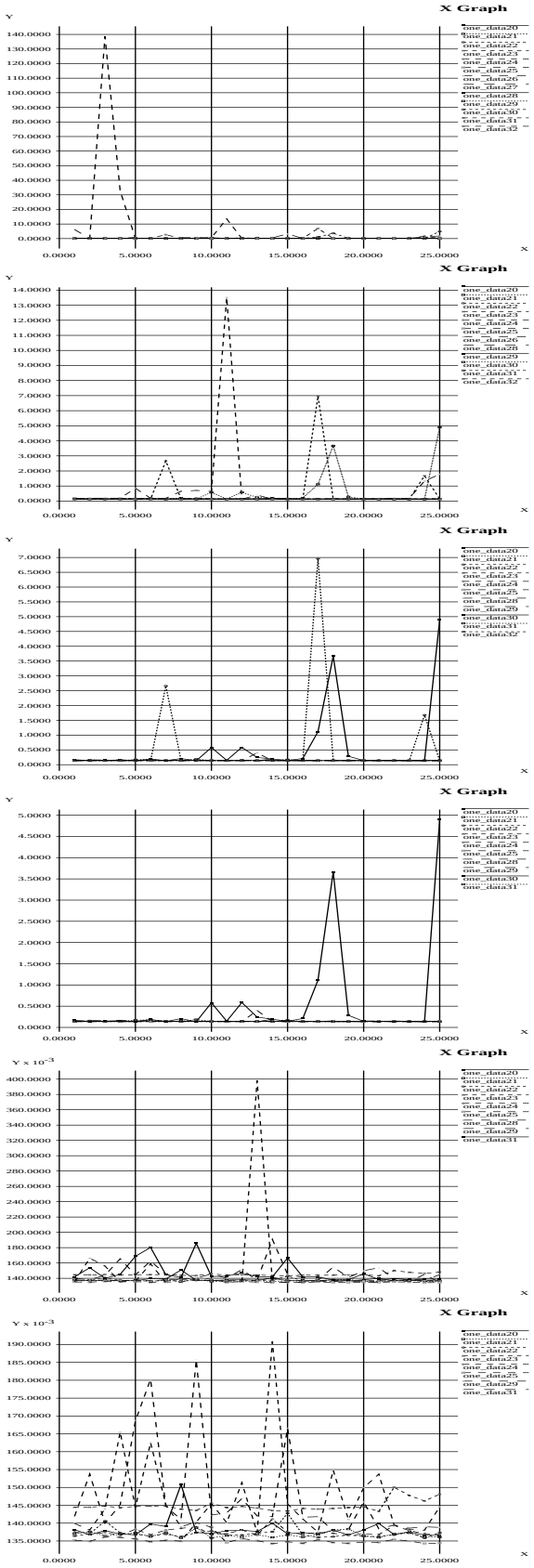

Figure 18: *Wireline network measurements.*

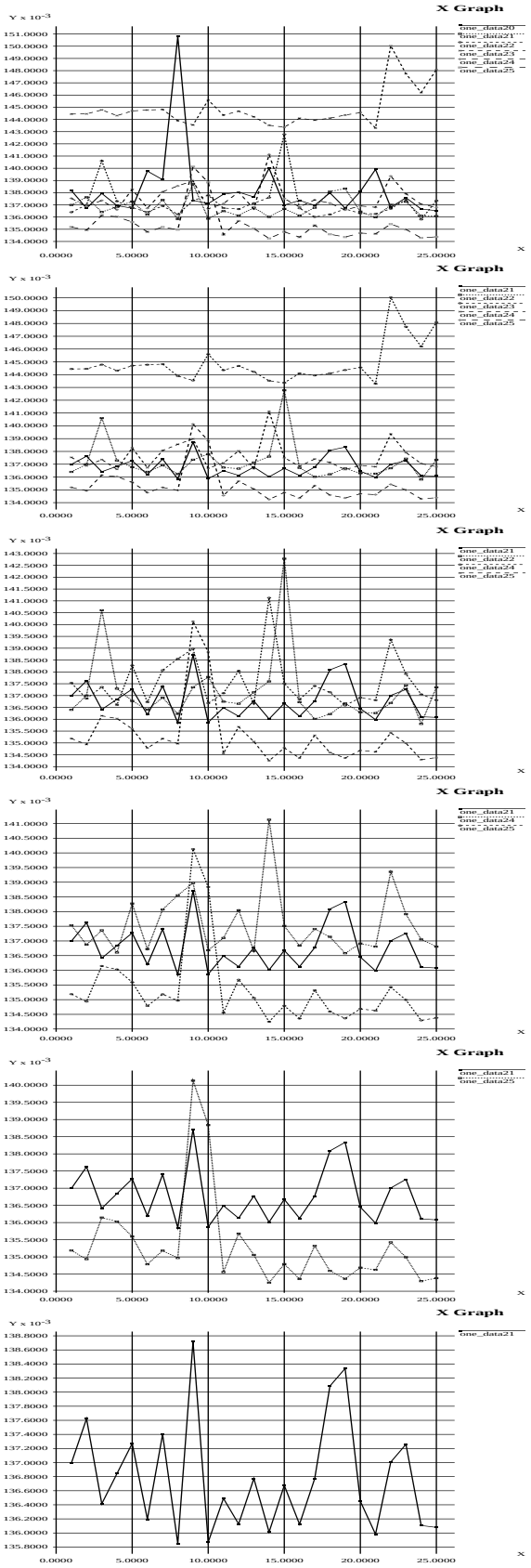

Figure 19: *Wireline network measurements.*