



On the propagation of congestion waves in the Internet

József Stéger^{a,b,*}, Péter Vaderna^{a,c}, Gábor Vattay^{a,b}

^a*Department of Physics of Complex Systems, Eotvos University, Budapest, Hungary*

^b*Collegium Budapest, Institute for Advanced Study, Hungary*

^c*Traffic Analysis and Network Performance Laboratory, Ericsson Hungary Ltd., Hungary*

Received 10 January 2005

Available online 6 June 2005

Abstract

Traffic modeling of communication networks such as the Internet has become a very important field of research. A number of interesting phenomena are found in measurements and traffic simulations. One of them is the propagation of congestion waves opposite to the main packet flow direction. The purpose of this paper is to model and analyze packet congestion on a given route and to provide a possible explanation for this phenomenon.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Computer networks; TCP/IP protocol; Congestion propagation

1. Introduction

In the recent past, many aspects of computer data traffic on the Internet have been investigated. Signatures of long-range correlations [1], scaling [2], chaos [3] and phase transitions [4] have been found. In Ref. [5], $1/f^\alpha$ noise has been observed in the time series of round trip times similar to those observed in highway traffic

*Corresponding author. Department of Physics of Complex Systems, Eotvos University, Budapest, Hungary.

E-mail addresses: steger@complex.elte.hu (J. Stéger), Peter.Vaderna@ericsson.com (P. Vaderna), vattay@complex.elte.hu (G. Vattay).

measurements [6]. Computers connected to a network communicate via data flows consisting of discrete data packets. The flow of these packets along a network path is quite similar to a one-dimensional granular flow of matter through a pipe. The basic concepts of the kinetic model of this traffic has been proposed by Antoniou et al. [7]. Since the slowing down and acceleration of packet flow in computer networks are very similar to those of cellular automaton models of cars in highway traffic [8,16], it has been argued that a valid analogy exists between these subjects which later has been demonstrated quantitatively [9,10]. One of the most peculiar features of car traffic and flow of granular media is the propagation of density waves [11]. In car traffic stop-and-go-type congestion waves propagate against the direction of the flow. In Ref. [12], the authors made an attempt to observe the propagation of congestion in the routers of a real computer network. They studied the spatiotemporal correlations of the level of congestion in routers and showed that congestion can propagate from a heavily loaded router to one of its empty neighboring routers.

Present day computer networks can be studied with the help of network simulators, which have been developed by engineers. These tools enable us to assemble any computer network configuration, to use the most commonly used packet-sending mechanism TCP/IP protocol and to emulate its behavior without building the system from hardware components. These simulators are developed to reproduce the behavior of computers and routers accurately by implementing their real transport layer protocols in the simulators. Due to some minor differences between the real protocols and their implementation in the simulator, minor details of simulation results may differ from the corresponding outcome of real networks, which is well documented in the engineering literature [14]. One of the most popular tools is the Berkeley Network Simulator (ns-2.26 [13]) which is used in this paper.

In this paper we study the network traffic generated in a unidirectional ring of identical routers connected. Ring topology was chosen to investigate to have some analogy with other granular flow simulations. The ring geometry mimics periodic boundary conditions. This way we can study the propagation of congestion in an isolated, clean setup, where the effects of inhomogeneity and the complex topology of the real Internet do not interfere with the basic mechanism creating the congestion wave. The ring topology makes it possible to compare the results with those obtained in cellular automaton car traffic simulations. We show that this system drives itself in a self-organized way into a critical congested state, where the system is overloaded and packets are lost for a long period of time. Both the position of the congested router (where packets are dropped) and the rate of the packet-sending activity (packets sent by the router in unit time) at the sites propagate against the direction of the packet flow. The profile of the congestion wave can be reconstructed from the activities of the computers connected to the ring. The velocity of the congestion wave can be measured. Then a simple model of the system is introduced, which is able to explain the main features of the congestion wave such as shape and speed. A formula for the velocity of the congestion wave is obtained and tested.

2. The model

In our model system a ring is formed by N identical routers, which can forward packets in the clockwise direction as depicted in the left part of Fig. 1. To each router a terminal computer is attached to generate the traffic. Routers are connected with a line of capacity C (measured in data bits per second) with a constant forwarding delay τ (measured in seconds). Incoming data flow in a router, which is a mixture of packet flows injected by the terminal computer and the background traffic coming from the neighboring router, can temporarily exceed the capacity of the outgoing line. To avoid data loss in this situation, the router contains a buffer of size B (measured in data packets) where packets can be stored. Terminal computers are instructed to send data persistently to their anti-clockwise neighbors, so that the packets traverse the longest possible route in the ring. The traffic studied is “granular” as computers send in data packets of size P (measured in bits). The dynamics of the data traffic of computers is controlled by the TCP/IP protocol [15]. This protocol ensures that the data packet-sending rate is decreased whenever congestion occurs and that it is increased when there is an available unused capacity in the system. The control algorithm has become quite sophisticated over the years. Here we can only sketch the main properties of the most popular version used currently.

After establishing a connection between two computers over the network TCP algorithm regulates the packet-sending rate. First a single packet is sent out. Upon receiving that packet, the receiver acknowledges the arrival of the packet by sending back a small size acknowledgement packet (ACK). The time elapsed between the sending out of a packet and receiving the corresponding ACK is called round trip time (RTT). The TCP maintains an internal variable, the Congestion Window (w), which is used to control the number of packets sent out when the ACK is received. It starts with the initial value $w = 1$ and then it is increased according to $w \mapsto w + 1/w$

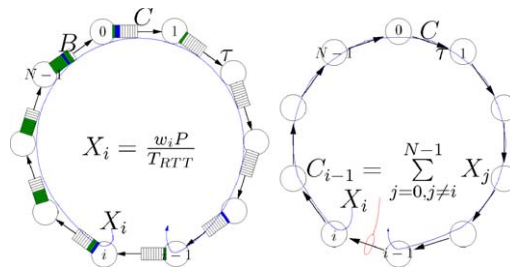


Fig. 1. The ring structure. Left, Network simulator: the TCP agent at site i , ($\forall i: 0 \leq i < N$) continuously transfers data through the ring of network routers connected with constant capacity C lines with time delay τ . The packets not yet processed by the routers are queued in buffers of size B . The stream of data can traverse in the clockwise direction only to arrive at the destination node terminal at site $i - 1$. In the simulations a set of parameters typical for the real Internet has been set ($C = 10^7$ bits/s, $\tau = .031$ s, $B = 300$ packets, $P = 4416$ bits, $N = 10$). Right, Continuous model: each TCP is modeled with its sending rate X_i and its traffic flow traverses the ring topology. On the link connecting sites $i - 1$ and i , all the sending rates of traversing TCP flows are summed up to yield the total link utilization.

each time an ACK is received. Two new packets are sent out if the congestion window crosses an integer value and only a single packet otherwise. This way the integer part of the window $[w]$ gives the number of sent but not yet acknowledged packets in the network. Assuming constant RTT during this process, the congestion window and the number of packets sent out in the network are increased linearly with time. This process lasts until a packet is lost somewhere in the network, indicating congestion. As a response the packet-sending rate should be decreased. So, the TCP reduces the value of the congestion window $w \mapsto \beta w$ ($\beta < 1$) and does not send out any new packets in response to ACKs until the number of still unacknowledged packets decreases to the integer part of the new (reduced) value of the congestion window. After this, the packet-sending algorithm returns to the original linear increase phase described above.

In case the congestion window variable is large, we can neglect its granularity and can treat it as a continuous variable. In the linear increase phase the window variable is increased by one in each RTT period and its dynamics can be well approximated with the differential equation

$$\frac{dw}{dt} = \frac{1}{T_{RTT}(t)}, \tag{1}$$

where $T_{RTT}(t)$ is the actual value of the RTT measured by the TCP. In case of packet loss the window is reduced.

$$w(t_+) = \beta w(t_-), \tag{2}$$

where $t_{\pm} = t \pm 0$ denotes the time right after and before the packet loss and $\beta \in (0; 1)$. The sending rate X (bit/s) in the continuous approximation can be estimated as the amount of data sent within an RTT, $X = Pw/T_{RTT}$. Neglecting the change of RTT on the scale of RTTs the sending rate satisfies the following pair of equations:

$$\frac{dX}{dt} = \frac{P}{T_{RTT}^2(t)}, \tag{3}$$

$$X(t_+) = \beta X(t_-) \quad \text{at packet loss,} \tag{4}$$

where in typical applications $\beta = \frac{1}{2}$.

We present the results of our simulation study carried out with the network simulator. The geometry and parameters of the setup are shown on the left part of Fig. 1, which are essentially the same as we could have observed in real hardware realization. Fig. 2 shows the spatiotemporal diagram of the congestion wave occurring in the network simulator. One can see that after a short initial transient (up to 500 s), the pattern remains stable and propagates in the anti-clockwise direction. In this respect, it resembles the congestion propagation in car traffic. The speed of the congestion wave pattern is almost constant. Its average can be determined by measuring the average speed of the center of mass of the pattern. This should be carefully defined in the present situation as the system is spatially periodic. Mapping the vertices upon each of the N th roots of unity in

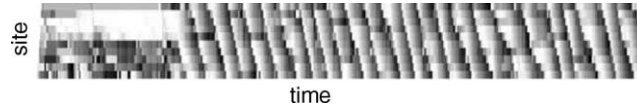


Fig. 2. Spatiotemporal diagram of congestion propagation. The horizontal axis is the time (covering 2000 s) and the site index (i) is on the vertical axis. In this simulation the number of sites was $N = 10$. Notice that the sites $i = 0$ and $i = N - 1$ are neighbors in the ring topology. The shade of the figure represents the sending rate X_i . Light patches indicate a very low sending rate due to high congestion. It can be seen that the most congested site propagates in the anti-clockwise direction with almost constant speed, while the packet traffic itself is clockwise directed.

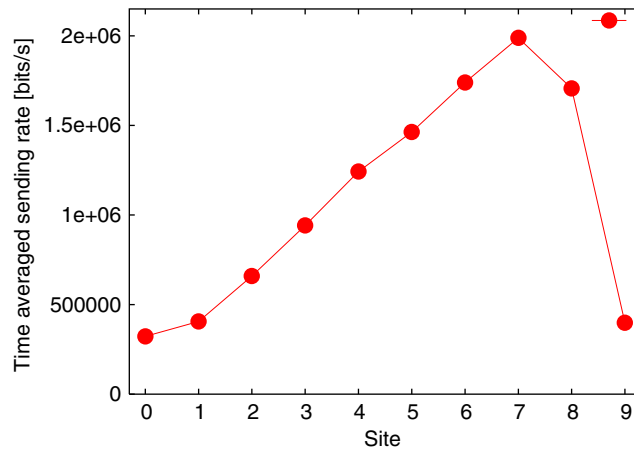


Fig. 3. The shape of the traveling wave profile. It has been determined by averaging the time series in co-moving spatial coordinates.

the complex plane weighed by the sending rates gives the complex number indicating the center of mass. The center of mass can thus be obtained by $\langle i \rangle(t) = (N/2\pi) \arg(\sum_{j=0}^{N-1} X_j(t) e^{i(2\pi/N)j})$. The speed of the pattern is the time derivative of this quantity.

Once the speed of the pattern is determined we can analyze the shape of the profile. Since the congestion waves have various shapes in different time instants, it is necessary to take the average of the sending rates at the wavefront. Representing the sending rates $X_{i+\lfloor i \rfloor}(t)$ in co-moving coordinates i' relative to the center of mass, we recover the shape of the traveling wave pattern. Averaging the new series in time the profile of the front emerges as in Fig. 3.

While the continuous equations constitute a gross simplification of the original TCP dynamics, the main properties of the traveling wave can be recovered from them with some additional assumption made on the packet loss process as we show next.

The bandwidth $C_{i-1}(t)$ utilized on the link connecting nodes $i - 1$ and i is the sum of sending rates of TCPs whose traffic flows through that link. In our case, the flow of all TCPs traverse that link except the one starting at node i and ending at node $i - 1$:

$$C_{i-1}(t) = \sum_{j=0, j \neq i}^{N-1} X_j(t) = \sum_{j=0}^{N-1} X_j(t) - X_i(t), \tag{5}$$

where site $i = N$ is identified with site $i = 0$ due to periodicity. The traffic of ACK packets emanating in $i - 1$ and absorbed in i is low due to their small size and their contribution to the traffic can be neglected (see also the right part of Fig. 1). Congestion and packet loss occur in the system whenever one of the bandwidths $C_i(t)$ reaches the link capacity C . According to (5), the largest link utilization $C_i(t)$ is at site $i = i^* - 1$, where i^* is the site where the sending rate $X_{i^*}(t)$ is the lowest. We then have to investigate which TCP flow will lose packet on link $i^* - 1$. In principle, all the TCP flows traversing the congested link can lose packets, so only the TCP at site i^* is immune. Our observation is that the TCP flow starting at the actual congested link (with sending rate X_{i^*-1}) experiences the packet loss almost surely. This is due to the fact that the processing of the small acknowledgement packets is faster than that of data packets and they tend to queue up and arrive in batches. As a response, TCPs send out data packets also in batches. Then obviously the TCP that ejects this burst of data packets directly into an almost saturated buffer, will lose packets in the process.

The mechanism described above is responsible for the emergence of the congestion wave in the system. The TCP at site $i^* - 1$ suffers packet losses repeatedly until its sending rate X_{i^*-1} becomes smaller than X_{i^*} . From then on X_{i^*-1} will be the lowest in the system, link utilization C_{i^*-2} will be the highest and TCP at site $i^* - 2$ suffers the packet losses. This way congestion propagates site by site anticlockwise in the system. After several rounds of congestion propagation the propagating front of Fig. 3 emerges.

The shape of the front is linear with a sharp drop connecting its ends. We can determine the parameters of the linear front in our model. Let the nodes forming the linear part of the front range from 0 to $N - 1$ and let the minimum sending rate first be at the 0th node

$$X_i(0) = a + bi, \tag{6}$$

as shown in Fig. 4 that illustrates the evolution of a system containing $N = 5$ nodes. The packets start to be dropped at site $N - 1$. We start our description at the first packet drop, which occurs when $C_{N-1}(0) = \sum_{i=0}^{N-1} X_i(0) - X_0(0) = C$ holds. This initial condition gives the first condition for the initial shape of the front

$$C = (N - 1) \left(a + \frac{N}{2} b \right). \tag{7}$$

Immediately after the packet drop, the sending rate at site $N - 1$ decreases to $X_{N-1}(0+) = \beta X_{N-1}(0-)$ while the rest of the sending rates remain unchanged.

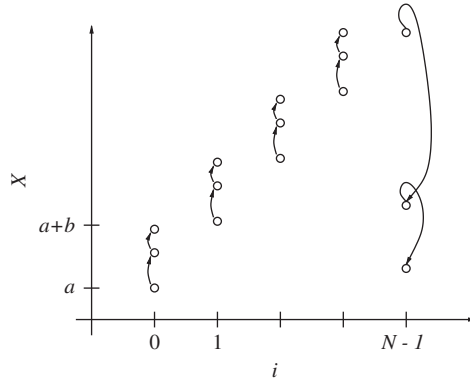


Fig. 4. Sending rate evolution in time. $N = 5$, $l = 2$; X_4 decreases until it reaches the value of X_0 .

Then the sending rates increase in a uniform manner with an amount $X'_i = X_i + (1 - \beta)X_{N-1}/(N - 1)$ until the next packet loss occurs. In particular, from before the first packet loss until the second packet loss the sending rate at site $N - 1$ changes to $X'_{N-1} = qX_{N-1}(0)$, where $q = \beta + (1 - \beta)/N - 1$. This process is then repeated l times until the sending rate $q^l X_{N-1}$ becomes lower than the actual value of X_0 . This way the wave moves one site to the left, while its linear shape is preserved as the sending rates of all nodes, except $N - 1$, increase with b . From Eq. (3), one can calculate the time needed for this $T_p = bT_{RTT}^2/P$. Accordingly, the speed of the congestion wave is $v = 1/T_p$ (measured in site/s).

The formula derived for the speed of the congestion waves can be tested against the data produced by the network simulator. In the simulation one can measure the average RTT observed by the TCPs, the mean slope of the linear part of the front b and the speed of the front. In the main part of Fig. 5, one can see that the measured values satisfy the relation $vbT_{RTT}^2/P = 1$ very well.

With the help of (7) and (6) the slope of the front in the model can also be directly expressed as

$$b = \frac{2C(1 - q^l)}{(N - 1)(N - 2)(q^l + 1)}, \quad (8)$$

where l should be determined independently. Our mathematical model allows several positive integer values of l with an upper bound due to the positiveness of b . We found numerically that only the largest possible l value is stable against small random perturbations of the wave front. Systems started at lower l values always shift towards a greater value of l . In the network simulator we always observed the realization of the most stable (the highest possible l) solutions of the model. In the inset of Fig. 5 we compare the measured values of b with (8). We again find good agreement.

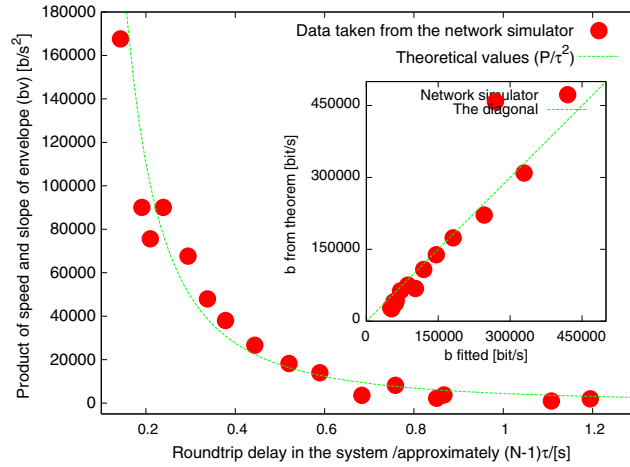


Fig. 5. Testing the relation $vbT_{RTT}^2/P = 1$ for various system sizes $N = 5-20$ in the network simulator. After the end of transients the values of b , T_{RTT} and v have been obtained by averaging over samples of 5000 s. Inset: Testing of Eq. (8), in the network simulator. Average values of b are plotted against the results of (8) with q^i obtained from the simulation.

3. Conclusion

As a summary, we show that congestion waves are formed naturally in the data traffic of computer networks. The mechanism behind the wave formation is that packet losses occur most likely in computers nearest to the site of the actual congestion and other computers sharing the congested link increase their sending rates, moving the site of the congestion one site downstream. This basic mechanism is quite general and can create congestion moving against the direction of the data traffic in more complicated geometries. A formula for the speed of the congestion wave has been derived in a simple ring topology and network simulations have confirmed it. Such formulas can be developed for more complicated geometries, which is our next research goal.

Acknowledgements

The authors are grateful for the partial support of the National Science Fund, Hungary (OTKA T37903), the National Office for Research and Technology (NKFP 02_032_04) and the EU IST FET COMPLEXITY EVERGROW Integrated Project.

References

- [1] W.E. Leland, M.S. Taqq, W. Willinger, D.V. Wilson, in: D.P. Sindhu (Ed.), Proceedings of ACM SIGCOMM, San Francisco, California, 1993, pp. 183–193;
K. Fukuda, H. Takayasu, M. Takayasu, *Physica A* 287 (2000) 289–301.
- [2] S. Valverde, R.V. Sole, *Physica A* 312 (2002) 636–648;
J. Yuan, Y. Ren, X. Shan, *Phys. Rev. E* 61 (2000) 1067.
- [3] A. Veres, M. Boda, Proceedings of IEEE INFOCOM Tel Aviv, March 2000, pp. 1715–1723,
<http://www.ieee-infocom.org/2000/papers/74.pdf>.
- [4] M. Takayasu, H. Takayasu, K. Fukuda, *Physica A* 277 (2000) 248–260.
- [5] I. Csabai, *J. Phys. A* 27 (1994) L417.
- [6] T. Musha, H. Higuchi, *Jpn. J. Appl. Phys.* 15 (1976) 1271.
- [7] I. Antoniou, V.V. Ivanov, Yu.L. Kalinovsky, *Physica A* 308 (2002) 533.
- [8] D. Chowdhury, L. Santen, A. Schadschneider, Statistical physics of vehicular traffic and some related systems, *Phys. Rep.* 329 (4–6) (2000) 199–329;
B.S. Kerner, P. Konhuser, Structure and parameters of clusters in traffic flow, *Phys. Rev. E* 50 (1) (1994) 54;
K. Nagel, M. Paczuski, Emergent traffic jams, *Phys. Rev. E* 51 (1995) 2909;
O. Biham, A. Middleton, D. Levine, Self-organization and a dynamical transition in traffic-flow models, *Phys. Rev. A* 46 (1992) R6124.
- [9] G. Simon, I. Csabai, The analogies of highway and computer network traffic, *Physica A* 307 (2002) 516–526.
- [10] T. Huisinga, R. Barlovic, W. Knospe, A. Schadschneider, M. Schreckenberg, *Physica A* 294 (2001) 249.
- [11] K. Nagel, M. Schreckenberg, *J. Phys. I* 2 (1992) 2221;
K. Nagel, H.J. Hemann, *Physica A* 199 (1993) 254;
M. Leibig, *Phys. Rev. E* 49 (1994) 184.
- [12] K. Fukuda, H. Takayasu, M. Takayasu, *Fractals* 7 (1) (1999) 23–31.
- [13] S. McCanne, S. Floyd. Ns (network simulator), 1995, available via <http://www-nrg.ee.lbl.gov/ns>.
- [14] S. Floyd, V. Paxson, Difficulties in simulating the Internet, *IEEE/ACM Trans. Networking* 9 (4) (2001) 392–403.
- [15] V. Jacobson, Congestion avoidance and control, Proceedings of SIGCOMM '88, August 1988, pp. 314–329.
- [16] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, Y. Sugiyama, Dynamical model of traffic congestion and numerical simulation, *Phys. Rev. E* 51 (2) (1995) 1035;
A. Schadschneider, Statistical physics of traffic flow, *Physica A* 285 (2000) 101.