Warp control: a dynamically stable congestion protocol and its analysis

Kihong Park
Computer Science Department
Boston University
Boston, MA 02215
park@cs.bu.edu

September 25, 1993

Abstract

This paper presents a distributed, end-to-end congestion control protocol for use in high-traffic, packet-switched networks. The network is represented as a stochastic single-server queue, with arrival rates being the control variables. A time-stamp based measure of network state called *warp* is defined, and it is shown to be an estimator of network utilization. Congestion is modeled explicitly using *unimodal load-throughput functions*, and its monotonicity property is exploited to yield characterizations of the interplay between optimality and stability. First, a protocol based on "perfect" information is analyzed, whose prowess is then shown to be emulated by one which only uses locally computable, delayed information. Both are subject to the unimodality assumption which has the effect of inducing a division of the phase space into stable and unstable regions, the optimal operating point lying at its boundary. Protocols are devised for dealing with each regime separately, *rate adjustment protocol* being the control that guides the system to the optimal operating point, *proactive* and *reactive rate protocols* dealing with the issue of maintaining near optimal throughput in the long-run. Protocols for handling fairness and structural perturbation augment the basic suite. The potential instability of the optimal operating point, even with control, carries implications for the different costs associated with proactive versus reactive congestion control mechanisms and their feasibility in high-speed networks. The analysis is supported by simulations showing the global dynamical properties of the system.

Keywords: Stability, Optimal Utilization, Unimodal Load-Throughput Function, Proactive vs. Reactive Congestion Control

1 Introduction

This paper presents a distributed end-to-end congestion protocol and its analysis of stability, optimality, and fairness. The present approach is closest in flavour to the rate-based access control schemes of [4, 8, 18, 22, 24, 29] where a variety of analytical methods are employed to determine the effects of various control laws under different network assumptions. This paper differs in two respects. First, we propose a time-stamp based measure of network state called *warp* and show that it estimates utilization for networks modeled as FCFS single-server queues. This leads to a natural control law which forms the basis of the warp protocol suite. Preliminary experiments on a network of seven workstations linked via Ethernet and running a communication-intensive application seem to support its usefulness even for small scale systems [13]. Second, we explicitly model a network's congestion mechanism via a unimodal curve relating effective throughput to *actual* load which we call the *load-throughput function*. Similar curves have been discussed in the literature that relate throughput to *offered* load [9, 10] which necessitates the inclusion of control laws in its formulation. In the present scheme, we view the unimodal shape of the load-throughput curve as an *inherent* property shared by a multitude of physical networks, in some sense, being "independent" of any particular control algorithm. To illustrate our claim, we compare a control having access to perfect information (the control of "Big Brother") against warp control and show that both suffer under potential instability in the long run. This brings forth a fundamental conflict in achieving optimal throughput in conjunction with stability, even when control is applied. The dynamics of protocols that incur a variance in their control variable due to stochastic effects or less-than-perfect information is explicated in this framework to yield qualitative insight into how difficult a problem congestion is, and what one may expect to get achieved.

Recently, the advent of broadband packet networks has raised a special set of issues induced by the large increase in the propagation delay–bandwidth product as well as the speed-mismatch between the increase in data transfer rate and limited processing power at routing stations [5, 6, 7, 11, 14, 31]. A general consensus, albeit tentative, has taken shape which recommends that congestion control protocols be simplified to minimize switching overhead with a resultant emphasis on end-to-end protocols, and proactive schemes are to be preferred over reactive ones due to the high cost associated with retransmission in case of packet loss. In the context of Broadband Integrated Services Digital Network (B-ISDN) with the Asynchronous Transfer Mode (ATM) transport mechanism, numerous congestion control protocols have been proposed that address the above problems [2, 3, 7, 10, 14, 21, 27, 30]. What is lacking in many of the approaches is a global viewpoint of the congestion problem as it pertains to large-scale, networked systems, where it is not clear how the system as a whole will behave given the complex interactions among circuit-switched/packet-switched traffic flows, sharing common network resources mediated by control protocols. In particular, we are interested in understanding the relationship between optimal throughput and asymptotic stability since it seems unlikely that the best of both worlds can be feasibly attained, even with sophisticated controls. One goal of this paper is to find a tractable, first approximation model which gives some insight into the global properties of large-scale, networked systems, and their dynamics. We will show that even in a simplified setting, achieving optimal throughput in a stable manner is an intrinsically difficult task, and some throughput may need to be sacrificed for the sake of maintaining stability. This parallels the unattainability of full utilization in M/M/1 queues, albeit for different reasons. Although the above does not imply that the full-fledged, physical system must suffer from the same consequences, it can be taken as evidence to mean that the more complex, nonlinear system may be at least as hard to control as the simplified one.

The second goal of this paper is to advance a "simple" solution to the congestion problem that effectively addresses the optimality/stability issues mentioned above. The protocol suite is based on a quantity called *warp* which for networks modeled as FCFS single-server queues can be shown to approximate network utilization. It will be shown how a congestion protocol based on *warp* having access only to local, delayed information can nevertheless emulate the power of Big Brother's control. The analysis of the protocol will bring forth the different costs associated with applying proactive versus reactive controls which has direct implications for feasible congestion control in high-speed networks. For a dynamically changing high-speed network environment, the modus operandi, in essence, will be a hybrid of the two methodologies. To provide long, stable, predictable, hence marketable network bandwidth, proactive congestion control is applied, which incurs the cost of operating at an unknown suboptimal throughput, the uncertainty of which grows with the duration of the interval of application. Let us call this the *proactive phase*. This is followed by a much shorter interval, the *reactive phase*, in which the system is elevated to a near-optimal, but "safe" throughput level. While doing so, there is a higher probability of unintentionally encountering the unstable region in phase space, necessitating a drastic, reactive, recovery mechanism. This latter "tuning" phase is indispensable in competitive B-ISDN environments where high network throughput will be of critical importance, assuming an infra-structure for delivering high-volume multi-media services is in place. A primitive solution based on rigorous underutilization will be not be a viable option.

This paper is organized as follows. First, we formalize the main variables of congestion control of interest to us. Second, we will analyze an idealized situation in which the control is centralized and has access to perfect information. Third, we will show how two agents interacting over a communication network, each with its own local time frame, can estimate a global property of the medium, its utilization ρ. Fourth, we construct a distributed congestion control algorithm and derive the equations governing its behavior. Fifth, we will analyze the equations with respect to stability, optimality, and fairness. Lastly, we report simulation results and conclude with refinements and extensions.

2 Network model and congestion control

2.1 Network model

Let N be a store-and-forward communication network of some fixed topology, consisting of a set of nodes $p_1, p_2, \ldots, p_n$. Each node p_i can be a source, sink, as well as a switching station for packets traversing the network. Each node is connected to N via a network interface I, and all packet submissions as well as receptions are cleared through I. The interface executes two primary protocols, one for sending data and one for receiving data. A packet m_{ij}, destined from node j to node i, is said to be *in* the network, if m_{ij} has passed through I_j's message generation protocol (MGP), but has not yet been processed by I_i's message reception protocol (MRP). The *load* of the network, $L(t)$, at time t, is defined to be the total number of packets in the network. The message generation rate $\lambda_i(t)$ of node p_i is defined as the number of packets sent out from I_i at time t. The total message generation rate is $\lambda(t) = \sum_{i=1}^{n} \lambda_i(t)$.

The service rate $\mu(t)$ of N is defined to be the total number of packets delivered over the entire network at time t. Although we will allow each node p_i to have its own local clock, assume the existence of an absolute time frame to which every local time frame can be mapped. The system is completely specified by indicating what the message generation and message reception protocols are, what we assume about network N including its service rate $\mu(t)$, and the message generation rate $\lambda_i(t)$ at each node's interface. The goal of our protocols MGP and MRP is to adjust $\lambda_i(t)$ as a function of $\mu(t)$ and $L(t)$ so as to maximize utilization $\rho(t) = \lambda(t)/\mu(t)$, while minimizing congestion which will be reflected in $\mu(t)$ as a function of $L(t)$. Thus $\mu(t)$ and $L(t)$ are the quantities to be controlled, and the control variables are the $\lambda_i(t)$'s. Mainly out of notational convenience, all quantities will be treated as continuous variables with suitable discretizations being implicit where applicable. Figure 1 shows a pictorial representation of the network.

In this highly abstracted viewpoint of the network, no reference to routing is made, and flow control is divorced from routing by formulating a theory that is independent of network topology. In particular, we will view the network N as a giant single-server queue, and derive our results from this assumption. This yields the first approximation model. The formulation leads to a more general characterization if we allow the arrival rate of each node to be further split into components $\lambda_j = \sum_{i \neq j} \lambda_{ij}$ where λ_{ij} is the individual message generation rate from j to i. N can be viewed as a coupled system of $n^2 - n$ single-server queues N_{ij}, each N_{ij} being a virtual channel linking j to i with service rate μ_{ij}. This allows node i to throttle its traffic to p_j upon detecting congestion

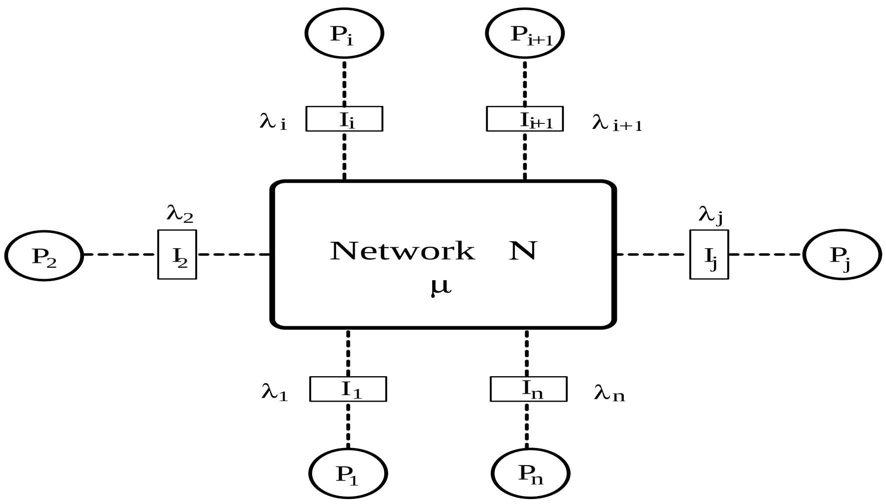


Figure 1: Pictorial representation of the network model.

on N_{ij}[1] while continuing with its current rate on other noncongested channels. A totally trivial coupling is realized if we let the N_{ij}'s be mutually independent. In general, they will be highly nonlinear due to the shared nature of the underlying network medium. In order to ignore the complex issue of what coupling to consider, we will assume N to be one single-server FCFS queue with service rate $\mu(t)$. That is, all interaction is affected through a single queue. It is an interesting question to consider how the interdependency can be modeled in a more realistic, yet tractable manner. Congestion control can always be modeled at the link level by unifying routing with flow control [17, 19].

2.2 Utilization versus congestion

There exist many characterizations of congestion phenomena and its causes, all more or less stating that overflooding a network results in a real decrease in its service rate. This may be due to buffer-overflow and time-out of packets, deadlock, and a host of other problems. Elegant accounts of the origins of congestion can be found in [25, 28]. Such considerations allow us to conclude that network congestion is primarily a function of the *actual* load $L(t)$ in the network. A commonly used scheme for depicting the relationship between "offered" load and effective throughput for controlled and uncontrolled networks can be found in [9, 10]. Figure 2 shows a curve relating actual load to throughput making explicit our assumption that congestion is synonymous to a drop in network service rate μ as a function of load L. This leads to the notion of an *inherent* optimal load L^* (in general, an interval $[L_l^*, L_u^*]$) of a network N, and its associated optimal service rate μ^*. It is not clear that this makes any sense since the load-throughput curve was defined independent of

[1] For simplicity, we shall assume symmetric channels throughout the paper.

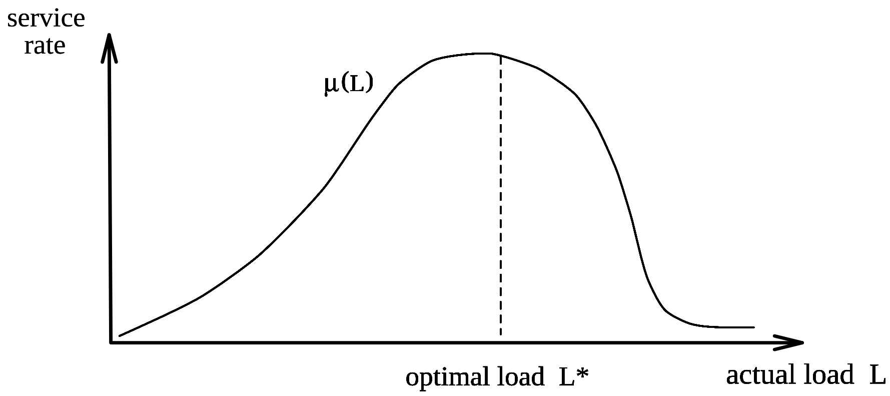


Figure 2: This figure shows the unimodal load-throughput function which relates actual load to service rate.

time and routing/flow control algorithms. Nevertheless, what we want to say is that congestion is an *inherent* property of resource-bounded traffic networks, and for a large class of "admissible" control laws, the family of associated load-throughput functions possesses an invariant property, its unimodal shape. We will refrain from giving formal justifications, except to say that this is purely an assumption for modeling congestion, whose validity is open to debate. Assuming this to be the case, can an optimal load be maintained? Given the conservation equality for $L(t)$

$$\frac{dL(t)}{dt} = \lambda(t) - \mu(t), \tag{1}$$

to achieve and maintain maximum throughput, $\lambda(t) \to \mu^*$, as $t \to \infty$. For an *uncontrolled* $M/M/1$ queue, we know that in steady state, the mean queue length (i.e., expected load) is $E(L) = \rho/(1-\rho)$, hence by $\rho^* = \lambda(t)/\mu^* = 1$, optimal utilization cannot be maintained. Since the expected delay of the system, $E(W)$, is given by Little's formula as $E(L) = \lambda E(W)$, this is often construed to mean that there is an explosion of delay near the optimal service rate. Trade-off measures between delay and throughput such as power, weighted objective functions, and other criteria have been proposed as candidates for defining the optimal "operating point" of a network [9, 15].

Note, the above only applies to uncontrolled systems. If a control law were able to regulate the load in the queue such that $E(L(t)) \to L^*$ and $\mathrm{var}(L(t))$ is kept small as $t \to \infty$ (analogously for $\lambda(t)$), then clearly there is no a priori reason for associating a prohibitive delay with the optimal operating point (μ^*, L^*). That is, it is perfectly possible to operate at maximum throughput without incurring a large delay penalty. This is easily seen in the case where we assume the existence of an all-powerful Orwellian "Big Brother" who knows the value of L^*, has access to instantaneous information about $L(t)$, and furthermore can instantaneously change the message generation rates $\lambda_1(t), \ldots, \lambda_n(t)$ at will. To make the problem interesting, we will let $\mu(t)$ be given as a stochastic process $\mu(t) = \bar{\mu} + \omega(t)$ over which Big Brother has no control. Here $\bar{\mu}$ is the mean service rate and $\omega(t)$ is white Gaussian noise. First, let us treat the simple case where $\bar{\mu}$ is constant. Let Big Brother's control law be given by

$$\frac{d\lambda(t)}{dt} = \epsilon(L^* - L(t)) - \frac{dL(t)}{dt}, \tag{2}$$

where $\epsilon > 0$. By $\mu(t) = \bar{\mu} + \omega(t)$, equations (1) and (2) form a system of stochastic differential equations. Let $\tilde{\lambda} = \lambda - \bar{\mu}$, and $\tilde{L} = L - L^*$. Then (1) and (2) can be written in matrix form as

$$\begin{bmatrix} d\tilde{\lambda}/dt \\ d\tilde{L}/dt \end{bmatrix} = \begin{bmatrix} -1 & -\epsilon \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \tilde{\lambda} \\ \tilde{L} \end{bmatrix} + \begin{bmatrix} \omega \\ \omega \end{bmatrix}.$$

We will denote this equation by $d\tilde{\mathbf{x}}/dt = A\tilde{\mathbf{x}} + \mathbf{w}$. The solution can be expressed using Itô's stochastic integral [1] as

$$\tilde{\mathbf{x}}(t) = e^{At}c + \int_0^t e^{A(t-s)} d\mathbf{W}(s) \tag{3}$$

where c is a constant that depends on the initial condition. The solution is asymptotically stable if and only if for all eigenvalues of A, the real parts are negative. It can be easily checked that this is the case for $\epsilon > 0$. Whether the solution is a damped oscillation or not depends on the exact value of ϵ, which occurs in the discriminant of the characteristic equation. In steady-state, $E(\tilde{\mathbf{x}}) = 0$, and $\mathrm{var}(\tilde{\mathbf{x}}) = O(\mathrm{var}(\omega))$. Thus $\lambda(t) \to \bar{\mu}$ and $L(t) \to L^*$. Moreover, if c is normally distributed or constant, then the solution $\tilde{\mathbf{x}}(t)$ is a Gaussian stochastic process. We conclude that Big Brother fares pretty well.

Let us consider what happens if we let $\bar{\mu} = \mathcal{G}(L)$ where $\mathcal{G}$ is strictly unimodal. That is, $d\mathcal{G}/dL > 0$ if $L < L^*$, $d\mathcal{G}/dL = 0$ if $L = L^*$, and $d\mathcal{G}/dL < 0$ for $L > L^*$. In general, $\mathcal{G}(L)$ may be highly nonlinear, hence we may not be able to find closed form solutions. Nevertheless, we can exploit the monotonicity property of $\mathcal{G}(L)$ by making piecewise linear approximations, and thus gain qualitative information regarding the stability of the more complicated system. Let I_l be a interval belonging to $[0, L^*)$, and let $I_r \subseteq (L^*, \infty)$. Define

$$\mathcal{G}(L) = \begin{cases} \alpha_l + \beta_l L < \mu^*, & \text{if } L \in I_l \\ \mu^*, & \text{if } L = L^* \\ \alpha_r - \beta_r L < \mu^*, & \text{if } L \in I_r \end{cases}$$

where α_l, β_l, and α_r, β_r are constants depending on I_l and I_r, respectively, with the restriction that $\beta_l, \beta_r > 0$. Simplifying the control law in equation (2) by dropping the damping term dL/dt, we obtain a new system of stochastic differential equations

$$\begin{bmatrix} \dot{\lambda} \\ \dot{L} \end{bmatrix} = \begin{bmatrix} 0 & -\epsilon \\ 1 & -\beta_i \end{bmatrix} \begin{bmatrix} \lambda \\ L \end{bmatrix} + \begin{bmatrix} \epsilon L^* \\ -\alpha_i \end{bmatrix} + \begin{bmatrix} \omega \\ \omega \end{bmatrix}.$$

Denoting the equation by $d\mathbf{x}/dt = A\mathbf{x} + \mathbf{b} + \mathbf{w}$, its solution can be expressed as

$$\mathbf{x}(t) = e^{At}c + \int_0^t e^{A(t-s)}\mathbf{b}ds + \int_0^t e^{A(t-s)}d\mathbf{W}(s). \tag{4}$$

Again, asymptotic stability depends on the eigenvalues of A. The solution to the characteristic equation is given by

$$r = -\frac{\beta_i}{2} \pm \frac{1}{2}\sqrt{\beta_i^2 - 4\epsilon}. \tag{5}$$

For $\beta_i > 0$ (i.e. $L \in [0, L^*)$), we have $Re(r) < 0$, hence the system is stable and converges to the optimal solution (μ^*, L^*). For $\beta_i < 0$, at least one eigenvalue will have a positive real part, hence the system will become unstable. Big Brother, by incorporating a damping factor $\gamma > 0$ via $\gamma dL(t)/dt$ in (2), may avoid, to a certain extent, instability, but only if $\mathcal{G}(L)$ is known. If this is not the case, or if $\mathcal{G}(L)$ has very sharp drops in the right interval (L^*, ∞), then even Big Brother, with all its might, stands to pay a nonnegligible price. We conclude that the left interval $[0, L^*)$ is stable whereas the right interval (L^*, ∞) is "potentially" unstable.

The foregoing analysis illustrates the basic feasibility of attaining the optimal operating point, but also the "inherent" difficulty of maintaining it, even with perfect information when faced with unimodal load-throughput functions. In the following sections, we will replace Big Brother by homogeneous, ignorant, but friendly group of "little folk," and show that the resulting distributed protocol which does not have access to global information such as L^*, and which cannot respond instantaneously to changes, can nevertheless closely emulate Big Brother's computational power.

3 Estimating utilization over a shared network

Let p_i and p_j be two nodes communicating over a network channel via packet switching or circuit switching. Let l_i and l_j be the local clocks of p_i and p_j, respectively. We allow l_i and l_j to run at different clock rates, but we require that the velocities be constant relative to an absolute time frame. In other words, there is a constant c_{ij} such that an event that has lasted for Δt_i in l_i and Δt_j in l_j will be related by $\Delta t_i = c_{ij}\Delta t_j$. If every node has access to a common global clock, $c_{ij} = 1$, $\forall i, j$. Let m_{ij} denote a packet addressed from p_j to p_i. m_{ij}, for our purposes, will contain three pieces of information. First, the actual data. Second, the source address i and destination address j. Third, the local time stamp of l_j at the time the packet was sent out at I_j, p_j's network interface unit. Thus the message encoding part of the message generation protocol at I_j for m_{ij} is as follows:

Message encoding protocol(MEP):

1. Create a header containing i, j, and time stamped

by its local clock l_j. (I.e., $m_{ij}.time\_stamp := l_j$.)

2. Attach the header to the data section of m_{ij} and submit to network N.

Of course, somewhere in the process there must be a routing protocol, but from our perspective, it will be transparent. The message decoding part of the message reception protocol, now at I_i receiving m_{ij}, will update a data structure maintained for node p_j called $hist_j$. $hist_j$ is a record containing two main components $hist_j.last\_in$, $hist_j.last\_out$, plus auxiliary log data for proactive congestion controls purposes. Into $hist_j.last\_in$, the local time at I_i, l_i, when m_{ij} arrived is recorded. Into $hist_j.last\_out$, the local time at p_j when m_{ij} was sent out, $m_{ij}.time\_stamp := l_j$, gets recorded. The decoding part of the message reception protocol at I_i on receiving m_{ij} works as follows:

Message decoding protocol(MDP):

1. $warp := \frac{1}{c_{ij}}\,(l_i - hist_j.last\_in)\,/\,(m_{ij}.time\_stamp - hist_j.last\_out)$.
2. $hist_j.last\_in := l_i$.
3. $hist_j.last\_out := m_{ij}.time\_stamp$.

With the basic protocol out of the way, let us see what properties the quantity $warp$ has under the assumption that network N is a single-server, multiple-client, FCFS queue.

Let $\mu(t)$ be a stochastic process given by $\mu(t) = \bar{\mu} + \omega(t)$ where $\bar{\mu}$ is a constant mean service rate and $\omega(t)$ is the white noise process. The state of N at time t, $N(t)$, is completely specified by $L(t)$, $\mu(t)$, $\{\lambda_1(t), \ldots, \lambda_n(t)\}$, and the packets in the queue. Let p_j send two messages, m^1_{ij} and m^2_{ij}, to p_i, one at global time T_1, and the other at global time T_2 ($T_2 > T_1$). Let t_j be the local time of p_j at which this event occurs relative to the global time, and let t_i be the local time of p_i at which the same event occurs. Thus we have lined up the two local time frames l_i, l_j for the first event with respect to the global reference frame. Let $t_j + \Delta t_j$ be the local time of p_j corresponding to T_2. For the subsequent treatment, it suffices to express all time-dependent quantities in terms of p_j's reference frame.

Assume that $\lambda(t) = \sum_{k=1}^{n} \lambda_k(t)$ is a very slowly varying function of time. In fact, in the stability analysis phase after specifying the protocol, we will see that our choice of parameters will dictate this to be so. The service rate $\mu(\Delta t)$ over a time interval Δt is just the Brownian motion of the underlying white noise process, hence $E(\mu(\Delta t)) = \bar{\mu}$ and its variance is proportional to Δt. If the initial load $L(t_j)$ at time t_j is not too large, then $d_1 = L(t_j)/\bar{\mu}$ is a good approximation for the time needed for m^1_{ij} to exit from the queue, i.e. get delivered to p_i. Thus the time stamp on m^1_{ij} will be t_j, and the local time at p_i when the packet arrives will be $a_1 = t_i + c_{ij}d_1$. What remains is to estimate the corresponding arrival time at p_i of the second packet in order to explicate $warp$. Let us compute the load at $t_j + \Delta t_j$, the time at which m^2_{ij} gets submitted to the queue. By the conservation equality (1), the load is given by

$$L(t_j + \Delta t_j) \;=\; L(t_j) + \int_{t_j}^{t_j+\Delta t_j} (\lambda(t) - \bar{\mu})\; dt \;\approx\; L(t_j) + \Delta t_j(\lambda(t_j) - \bar{\mu}). \tag{6}$$

The approximation stems from the fact that we have assumed $\lambda(t_j) \approx \lambda(t_j + \Delta t_j)$. Also, assume $L(t_j) + \int_{t_j}^{t_j+\Delta t_j}(\lambda(t) - \bar{\mu})\; dt > 0$ which will be the case for large-scale systems and Δt_j small. Using equation (6), the time needed for m_{ij}^2 to exit from the queue can be estimated as $d_2 = L(t_j + \Delta t_j)/\bar{\mu}$. Hence, the local time of p_i at which m_{ij}^2 arrives is given by

$$\begin{aligned} a_2 &= t_i + c_{ij}(\Delta t_j + d_2) \;=\; t_i + c_{ij}(\Delta t_j + \frac{L(t_j) + \Delta t_j(\lambda(t_j) - \bar{\mu})}{\bar{\mu}}) \\ &= t_i + c_{ij}(\Delta t_j + d_1 + \Delta t_j(\frac{\lambda(t_j)}{\bar{\mu}} - 1)). \end{aligned}$$

Finally, combining the previous results, the *warp* computed at p_i by MDP upon arrival of m_{ij}^2 is

$$\begin{aligned} warp &= \frac{1}{c_{ij}}\frac{l_i \;-\; hist_j.last\_in}{m_{ij}.time\_stamp - hist_j.last\_out} \;=\; \frac{1}{c_{ij}}\frac{a_2 \;-\; a_1}{(t_j + \Delta t_j) - t_j} \\ &= \frac{1}{c_{ij}\Delta t_j}\,[\,(t_i + c_{ij}(\Delta t_j + d_1 + \Delta t_j(\frac{\lambda(t_j)}{\bar{\mu}} - 1))) \;-\; (t_i + c_{ij}d_1)\,] \\ &= \frac{\lambda(t_j)}{\bar{\mu}} \quad\approx\quad \rho. \end{aligned} \tag{7}$$

Equation (7) shows that *warp*, in the case of FCFS single-server queues with the service rate perturbed by white noise, is an "estimator" of the utilization of the queue. A pictorial representation of commucation over a "warped" medium is shown in figure 3. *warp* has a time-lag associated with it, and its accuracy depends on several factors. First, the total delay, $\Delta t_j + d_2$ must not be too large. Since Δt_j is the inverse of the message generation rate, this implies that packets carrying the control information cannot be too far in between. Second, $\lambda(t)$ needs to be slowly time-varying for the estimates to be accurate. But λ *is* the control variable of the system, and as we shall see in the next section, for the system to be stable, one requirement is that $|d\lambda(t)/dt|$ be very small. Third, the steady-state load cannot be too large since this would make the steady-state delay, as reflected in d_1, become large as well. The use of time stamps to estimate network delay is a common technique and other treatments can be found in [10, 18].

4 The warp protocol

4.1 Protocol for rate matching

The previous section has shown that two agents interacting over a shared medium can estimate a global quantity, in our case, the network utilization ρ, making use of only "locally" available

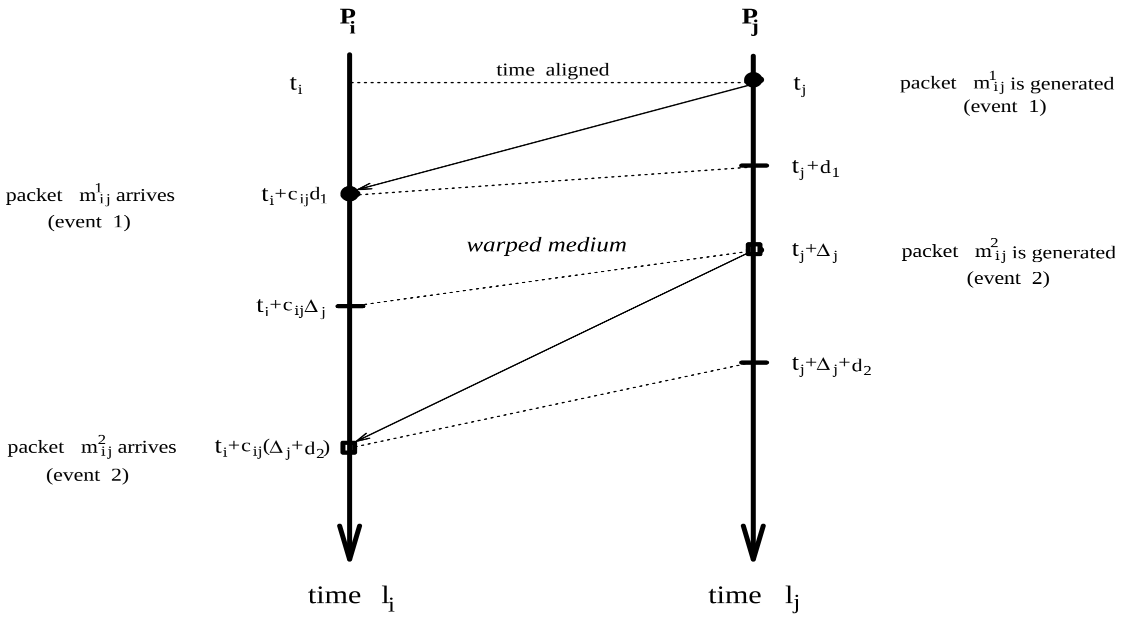


Figure 3: Communicating over a warped medium.

information. Having established local observability of a global quantity, the next step is to show the local controllability of this global quantity. The main goal of the first control protocol is to achieve *rate matching* between $\lambda(t)$ and $\mu(t)$ where $\mu(t)$ is an unknown quantity. That is, $E(\rho) = 1$, and $E((\rho - 1)^2) < \delta$, for δ small. Consider the case where $\mu(t) = \bar{\mu} + \omega(t)$, and $\bar{\mu}$ is fixed. By equation (7) of the previous section, *warp* approximates network utilization $\rho(t)$. This suggests using the following control law:

$$\frac{d\lambda(t)}{dt} \;=\; \epsilon(1 - warp) \;\approx\; \epsilon(1 - \frac{\lambda(t-\tau)}{\bar{\mu}}) \tag{8}$$

where τ is a time-lag introduced by network delay. Note that we have suppressed the noise term $\omega(t)$ which would have occurred in the denominator of the right-hand-side of (8). This is technically problematic. Even for a simpler equation, $dx(t) = x(t-r)dW(t)$, where the noise term $W(t)$ (1-D Brownian motion) occurs multiplicatively, a closed-form solution does not seem to be known [23]. For our purposes, it suffices to extract features that describe the qualitative behavior of the system (mainly its stability), and we shall proceed with this in mind. When considering the deterministic delay-differential equation (8), it suffices to look at the homogeneous equation

$$\frac{d\lambda(t)}{dt} = -\frac{\epsilon}{\bar{\mu}}\lambda(t-\tau) \tag{9}$$

whose solution can be expressed in exponential form $e^{\gamma t}$. This involves finding the complex roots of the characteristic equation $\gamma + \frac{\epsilon}{\bar{\mu}}e^{-\tau\gamma} = 0$. The solution is a sinusoidal oscillation, and it is

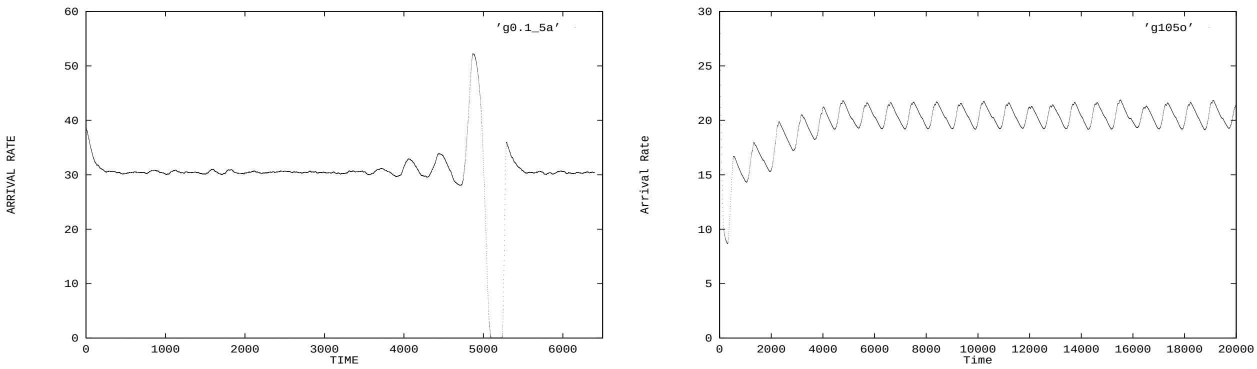


Figure 4: Simulation run of two systems with different ϵ values: (left) unbounded oscillation; (right) bounded oscillation.

asymptotically stable if for all roots of the characteristic equation, the real parts lie in the negative half plane. For $\tau = 1$, it can be shown [12] that this is the case when $0 < \frac{\epsilon}{\bar{\mu}} < \pi/2$. To see the effect of any fixed $\tau > 0$, we will transform equation (9) into its "normal" form (i.e. $\tau = 1$) by a change of variables. Let $t' = t/\tau$. Then,

$$\frac{d\lambda(\tau t')}{dt'} = \frac{\tau\epsilon}{\bar{\mu}}\lambda(\tau(t'-1)).$$

Setting $\tilde{\lambda}(t') = \lambda(\tau t')$, we get the normalized equation

$$\frac{d\lambda(t)}{dt} = -\tau\frac{\epsilon}{\bar{\mu}}\lambda(t-1) \tag{10}$$

where we have suppressed the *tilde* for notational clarity. Finally, we can state the condition for asymptotic stability of the deterministic delay differential equation as

$$0 < \tau\frac{\epsilon}{\bar{\mu}} < \frac{\pi}{2}. \tag{11}$$

Thus the greater the network delay, the smaller ϵ needs to be in order to keep the system stable. This agrees well with our previous requirement for making $\lambda(t)$ a slowly varying function of time.

The effect of the noise term for μ stochastic can be gleamed from (11). If the variance of the Gaussian noise is large, then for ϵ not sufficiently small, the stability condition may be frequently violated leading to growing fluctuation. Figure 4 shows the simulation run of two 10 node systems, each exhibiting a different asymptotic behavior. The left figure shows a system whose ϵ value was too large, violating the stability condition (11), thus leading to unbounded oscillation resulting in a crash. The right figure shows a system whose ϵ value is small, leading to bounded oscillation, but not small enough to achieve damping. Figure 5 (upper left) shows a system which achieves damping with a sufficiently small ϵ.

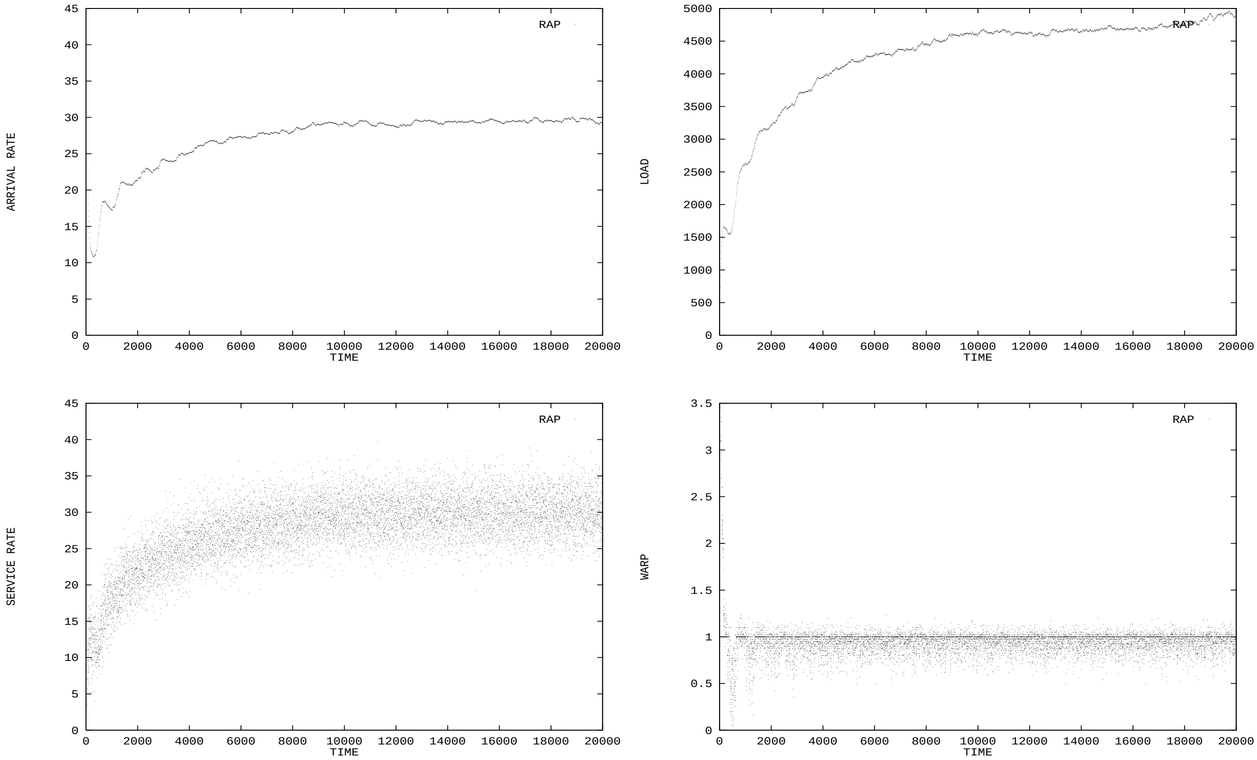


Figure 5: Simulation of a 10 node system using a unimodal load-throughput function running RAP.

Distributed control is made possible via $\lambda(t) = \sum_{k=1}^{n} \lambda_k(t)$, where at each node, the control part of the message reception protocol executes the same control function given in (8). This we call the *rate adjustment protocol* (RAP). At each p_i:

Rate adjustment protocol (RAP):

1. $\lambda_i(t) = \lambda_i(t-1) + \epsilon_i(1 - warp(t))$.
2. If $\lambda_i(t) < 0$, then $\lambda_i(t) = 0$.

There are several points worth noting here. First, ideally, $\lambda_i(t) = \lambda(t)/n$, $i = 1, 2, \ldots, n$. As $t \to \infty$, $E_i(\lambda_i(t)) = \lambda(t)/n$, but $E_i((\lambda_i(t) - \lambda(t)/n)^2)$ will drift in the absence of some preventive mechanism (the expectation is taken over i). One consequence is the breakdown of *fairness*. By fairness, we mean the equal allocation of message generation rates over all constituents. Second, note that only one control variable, λ_i, is kept at each node. As we have indicated in section 2, a more general scheme is to split up λ_j at each p_j such that $\lambda_j = \sum_{i \neq j} \lambda_{ij}$. This allows for individualized communication rate control over different pairs of nodes, and is useful for the case where the interconnection network is not "uniform." A typical simulation run of a 10 node system using an actual unimodal load-throughput function (note, the previous analysis was for $\bar{\mu}$ fixed) is shown in figure 5. The optimal load interval was set at $[4700, 5000]$, and $\mathcal{G}(L)$ was a straight line with positive and negative slopes on the left and right segments, respectively. The optimal service rate was 30 packets per unit time, and the service rate was subject to a white noise process with variance 9. The graphs show the stable rate matching properties of the protocol, even under nonnegligible noise conditions.

4.2 Protocol for load matching

In the previous section we have shown that RAP is able to maintain full utilization for the case when $\bar{\mu}$ is fixed. In this section, we will investigate the general case where the system is characterized by a unimodal load-throughput function. Basically, we will show that there is a stable region and an unstable region, and optimal utilization cannot be maintained in the long run by the rate-matching protocol alone. Consider our system of equations (1) and (8), and simplify it by ignoring the noise term $\omega(t)$ as well as letting $\lambda(t) = \lambda(t-\tau)$. The latter is partly justified by our requirement for setting ϵ very small. We get a system of nonlinear differential equations coupled via $\mu = \mathcal{G}(L)$,

$$\begin{bmatrix} d\lambda/dt \\ dL/dt \end{bmatrix} = \begin{bmatrix} \epsilon - \epsilon\lambda(t)/\mathcal{G}(L) \\ \lambda(t) - \mathcal{G}(L) \end{bmatrix}. \tag{12}$$

Let us denote it by $d(\lambda, L)/dt = F(\lambda, L)$. To understand its dynamics, we will first linearize it, then analyze the stability of the linearized system. The gradient in the first-order term is given by

$$\frac{\partial F(\lambda, L)}{\partial(\lambda, L)} = \begin{bmatrix} -\epsilon/\mathcal{G}(L) & \epsilon\lambda(t)\mathcal{G}_L(L)/\mathcal{G}(L)^2 \\ 1 & -\mathcal{G}_L(L) \end{bmatrix}$$

where $\mathcal{G}_L(L) \equiv d\mathcal{G}(L)/dL$. Note, all (λ, L) such that $\lambda = \mathcal{G}(L)$ are solutions to (12). Denote this set by $\mathcal{S}$. For any $(\bar{\lambda}, \bar{L}) \in \mathcal{S}$, the Taylor expansion around $(\bar{\lambda}, \bar{L})$ is given by

$$\begin{bmatrix} d\lambda/dt \\ dL/dt \end{bmatrix} = \begin{bmatrix} -\epsilon/\mathcal{G}(\bar{L}) & \epsilon\mathcal{G}_L(\bar{L})/\mathcal{G}(\bar{L}) \\ 1 & -\mathcal{G}_L(\bar{L}) \end{bmatrix} \begin{bmatrix} \lambda - \bar{\lambda} \\ L - \bar{L} \end{bmatrix} + \text{Higher Order Terms}$$

where $F(\bar{\lambda}, \bar{L}) = 0$. To determine stability, we need to solve for the eigenvalues of $\partial F(\lambda, L)/\partial(\lambda, L)$ at $(\bar{\lambda}, \bar{L})$. This yields one eigenvalue of $r = 0$ (the matrix is singular), and another given by

$$r = -\epsilon/\mathcal{G}(\bar{L}) - \mathcal{G}_L(\bar{L}). \tag{13}$$

Clearly, for $\bar{L} < L^*$, $\mathcal{G}_L(\bar{L}) > 0$, hence $r < 0$. That is, the system is stable[2]. The same holds for $\bar{L} = L^*$. For $\bar{L} > L^*$, $\mathcal{G}_L(\bar{L}) < 0$, and depending on the magnitude of $\mathcal{G}_L(\bar{L})$, it is possible that

[2]For a more complete analysis, the higher order terms need to be examined since the solutions are degenerate. This entails taking into consideration centre manifolds which will be omitted here.

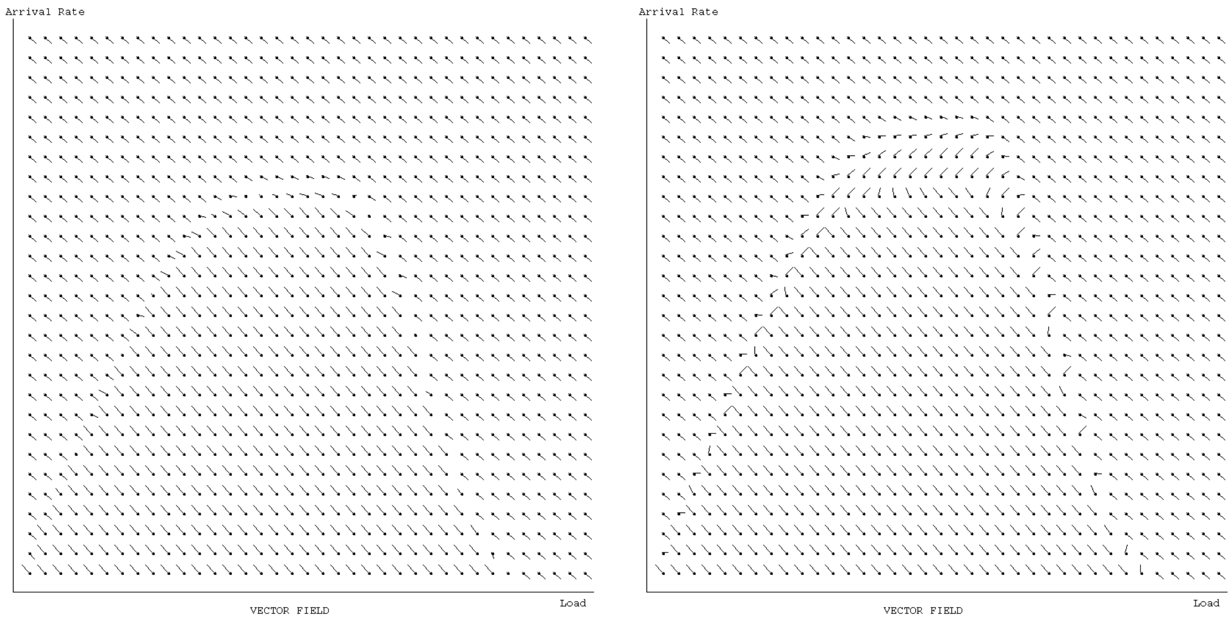


Figure 6: Vector field for system with (right), and without bias control (left).

$r > 0$, leading to instability. This is analogous to the situation that Big Brother encountered when faced with a unimodal load-throughput function (section 2.2). Figure 6 (left) shows the vector field for the above system where the phase portrait can be glimpsed. The lines with filled circles at one end denote arrows where the circle corresponds to the tail of the vector. As is evident, for solutions lying in the left half interval ($L < L^*$), flow is convergent whereas for solutions lying on the other side, flow is divergent.

Two issues need to be dealt with at this point. First, since all points $(\bar{\lambda}, \bar{L}) \in \mathcal{S}$ with $\bar{L} \leq L^*$ are stable solutions, if the system starts off in a state where $L < L^*$, L will settle into an equilibrium below L^*. An extra thrust, or bias, needs to be applied to give the process an upward drift. Consider the modified rate adjustment control

$$\frac{d\lambda}{dt} = \epsilon(1 + \theta(t) - warp) \tag{14}$$

where $\theta(t)$ is a variable that controls the drift. Setting $0 < \theta(0) = \theta_a \ll 1$ has the effect of destabilizing the left half interval solution points, which in turn imparts an upward motion towards (μ^*, L^*). The vector field corresponding to the biased equation (14) is shown in figure 6 (right) where the upward flow is discernible. The second issue is, in part, caused by addressing the first issue using the bias parameter $\theta(t)$. That is, although $\theta(t)$ allows the system to climb up the continuous "equilibrium curve" $\mathcal{G}(L)$ towards (μ^*, L^*), unless the process is terminated at some point near L^*, instability will ensue. To avoid "crossing over the line," one, it needs to be detected that $L(t) \approx L^*$, and two, $\theta(t)$ must then be turned off. If we assume that for most networks the

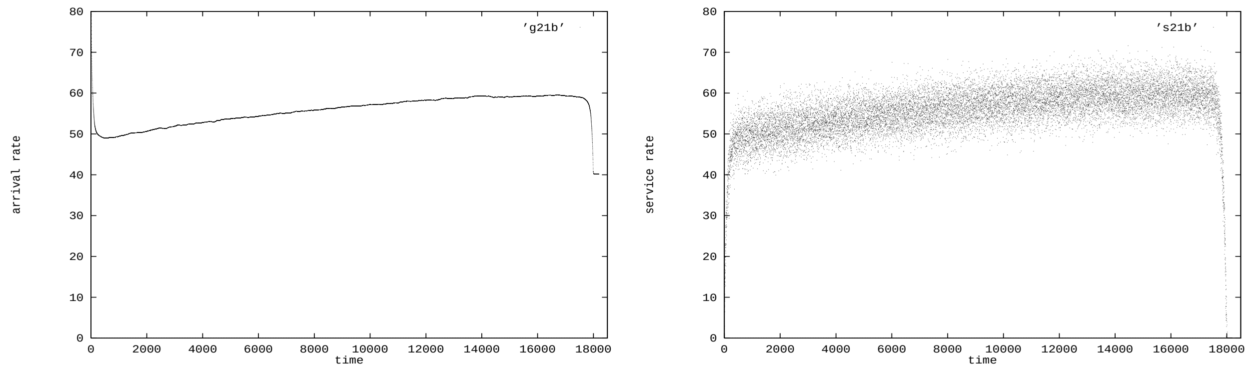


Figure 7: Simulation of a 10 node system and its crash in the unstable region.

left neighborhood of L^* is relatively flat, then we can exploit this property by noting that for small $\theta(t)$, the history of $\lambda(t)$ will reflect the slope of $\mathcal{G}(L)$. This leads to the first solution method which is of a proactive nature. In our case, regression analysis is used to estimate the slope of the message generation rate history. If flatness is detected, $\theta(t)$ is shut off. More sophisticated controls can be designed by making $\theta(t)$ a continuous function of the estimated slope. Let $rate\_hist_i$ be an array of elements of past λ-values at p_i.

Proactive rate protocol (PRP):

1. $slope = regress(rate\_hist_i)$.
2. If $|slope| < slope\_threshold$, then $\theta(t) = 0$.
3. If $t \equiv 0 \pmod{slope\_interval}$, then do

   Discard oldest element of $rate\_hist_i$.

   Record $\lambda_i(t)$ into $rate\_hist_i$.

$Regress()$ is a function that performs linear regression on $rate\_hist_i$, $slope\_threshold$ is a parameter that tests whether the slope is 0 or not, and $slope\_interval$ represents the time interval between successive stored values since keeping a large history is expensive. The longer the optimum load interval $[L_l^*, L_u^*)$, the more effective the proactive scheme will be since there is more room to detect the "plateau." The dynamics of a system using PRP is very similar to one that uses only RAP before crashing (figure 5).

An altogether different approach is the reactive scheme, which upon detection of instability, performs an "exponential back-off" type operation. For a sudden drop in service rate due to congestion or structural effects, in the absence of a rapid recovery mechanism the system will crash ($\mu \approx 0$) as seen in the previous analysis. This "congestion collapse" [16] can be observed in figure 7 for a 10 node system which has an optimum service rate of 60 packets per unit time. The system was simulated running only RAP. To minimize the disruptive effects of a total crash, a reactive

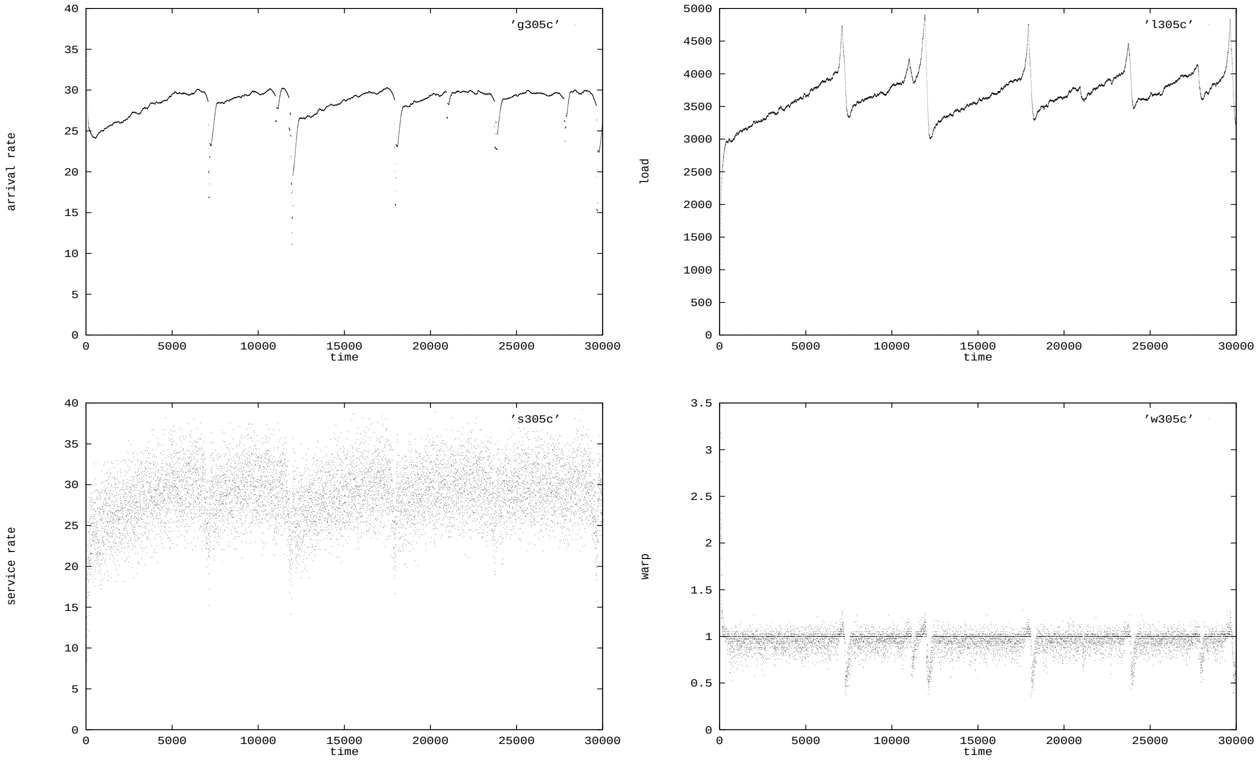


Figure 8: Simulation of a 10 node system running RAP and RRP.

protocol is constructed that detects a sharp increase in $warp$, then goes into sleep for $lock$ number of steps after setting the message generation rate to a lower value. At each p_i,

Reactive rate protocol (RRP):

1. If $warp_i > 1 + warp\_threshold$ then do

    Set $new\,\lambda_i = old\,\lambda_i * (2 - warp_i)$.

    Set $lock_i = lock\_constant * (warp_i - 1)$.

2. Go into sleep for $lock_i$ steps.

$Warp\_threshold$ is a parameter for detecting a critical change in $warp$, and $lock\_constant$ determines the sleep period. It may very well be possible to incorporate the exponential back-off scheme in the control equation itself. Figure 8 shows a 10 node system with optimum service rate of 30 and optimal load interval $[3700, 4000]$. $Warp\_threshold$ was set at 0.3 and $lock\_constant$ was 60. The system was run for 30000 time steps under the RAP + RRP protocol. Note, the warp and arrival rates shown are averaged over all nodes. Each node executes its protocol independently, and it is not the case that all nodes get a $warp$ value exceeding 1.3 all at the same time. If the system is at a critical level, most nodes will make a similar decision within a short interval of each other. Even

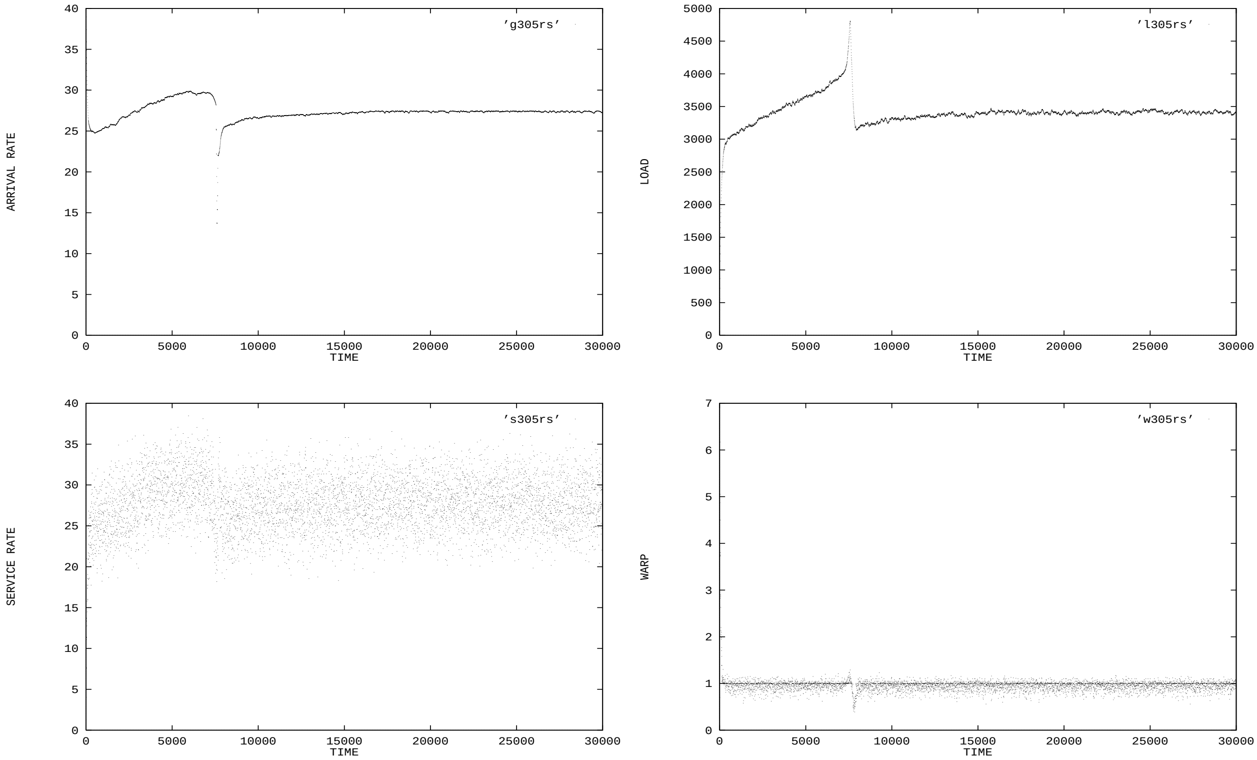


Figure 9: Simulation of a 10 node system running RAP and PRP.

if the system is not critical, a processor may decide to "hybernate" due to noise.

The simulation run of a hybrid method which combines proactive and reactive controls as indicated in section 1 is shown in figure 9. In the reactive phase which includes the initial transient period, the biased equation is used to seek out the optimal operating point. Without the regression-based PRP active (e.g. slope estimation may not be very accurate), upon entering the unstable region, RRP kicks in. Before switching to the proactive phase, this particular protocol whose behavior is shown in figure 9 remembers the message generation rate immediately before RRP got activated, multiplies it by 0.9, and applies the resultant quantity as a ceiling on the arrival rate during the following proactive phase. This has the effect of preventing an accidental drift into the unstable region during the lengthy proactive phase. Of course, it does not prevent an accidental drift in the other direction, although the same method could be applied to yield a lower bound. As is evident, there is a large degree of freedom in designing proactive/reactive congestion controls, the before mentioned protocols being just a representative subset.

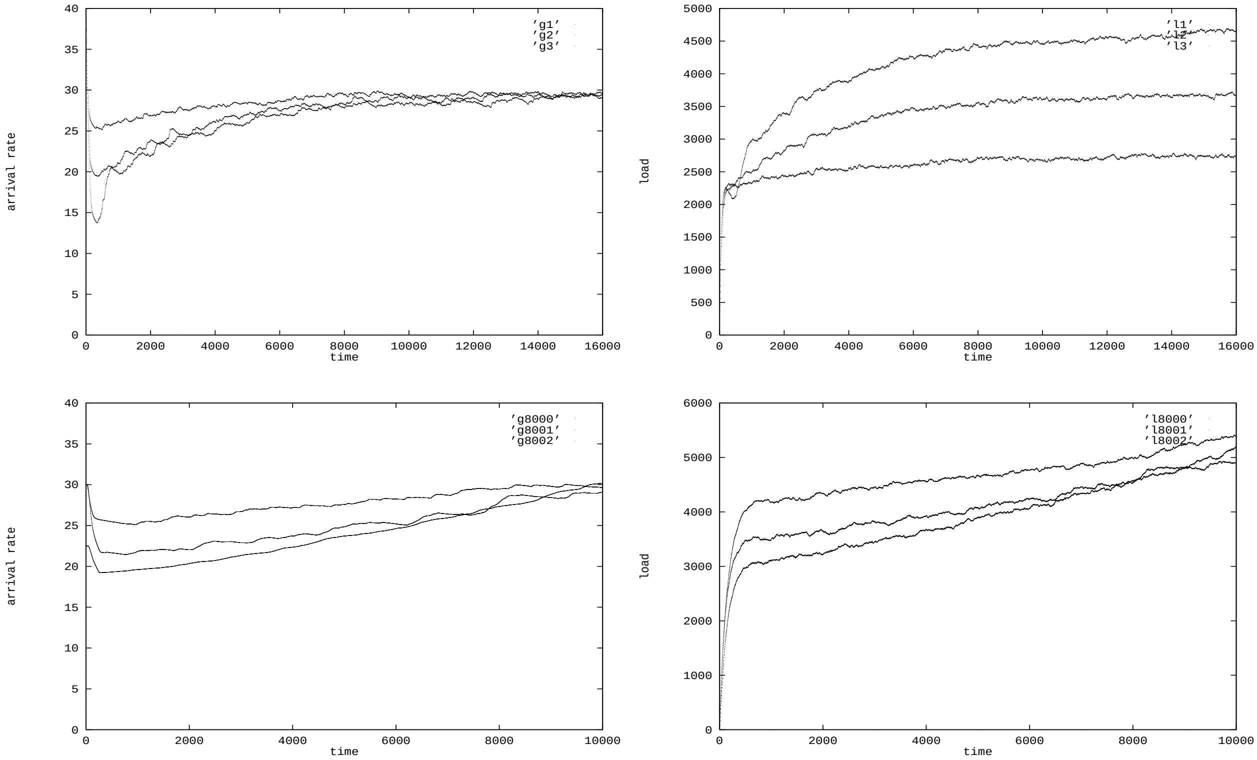


Figure 10: Top: 10 node system under three different network conditions. Bottom: 30, 60, and 90 node system under same network conditions.

5 Simulation and extensions

In the following sections, we will show simulation results of typical system behavior under different system configurations, structural perturbation, and implementation of a fairness protocol. All simulations were run under the following procedure: at every global time step t, dequeue $\mathcal{G}(L(t)) + \omega(t)$ messages and deliver them to their destinations. Each receiving node executes its message reception protocol which consists of MDP, RAP, and other applicable control protocols. For each "able" (e.g. not locked) node p_i, generate $\lambda_i(t)$ messages with destination addresses formed from a uniform distribution. This entails executing MEP, and enqueueing messages into the queue. Increment global time, and repeat. Various details have been omitted for brevity, but essentially, it is just a straightforward simulation of a multi-node system sharing a common queue. It should be mentioned that more than one message is allowed to be generated per unit time, and λ is kept as a real number with the fractional part being transferred over to the next round.

5.1 Robustness

The top part of figure 10 shows the behavior of a 10 node system with optimum service rate 30, variance 9, running only RAP, under three different load-throughput functions given by $[4700, 5000]$, $[3700, 4000]$, and $[2700, 3000]$, respectively. The results demonstrate the stable rate matching properties of RAP as well as convergence to the optimal load interval which agrees well with our previous analysis. All three runs were performed with the same parameters, partially showing the robustness of the protocol. Simulation runs for a 30, 60, and 90 node system is shown in the bottom part of figure 10. They all shared a common optimal load interval $[5000, 6000]$ and an optimal service rate of 30.

5.2 Structural stability

The previous analysis was for the case when the system was fixed, i.e. constant number of nodes, fixed variance, and invariant load-throughput function. Structural variability was to some extent

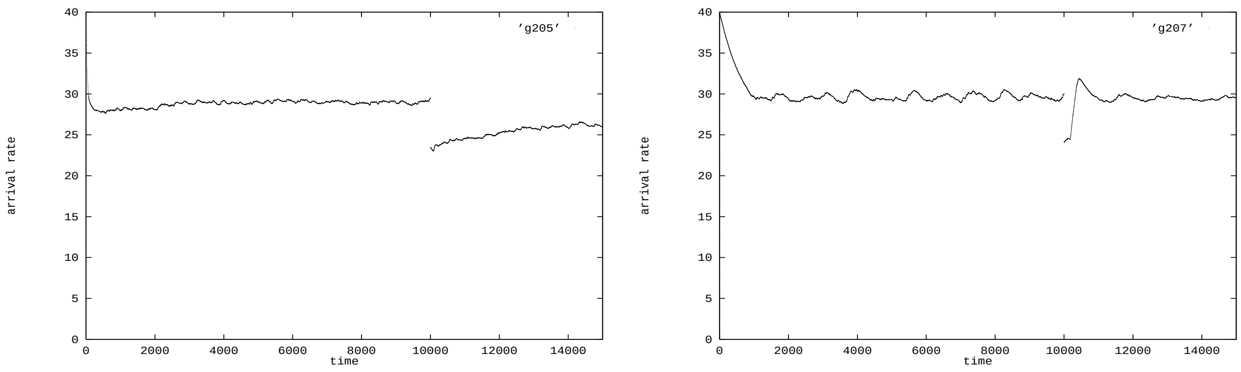


Figure 11: Simulation of a 10 node system with structural (node number) perturbation. Left: slow recovery without BRP. Right: responsive recovery with BRP.

incorporated by the variance of the service rate, but clearly this is not enough. Figure 11 (left) shows the evolution of a 10 node system with an optimal load at 2000, and optimal service rate 30, which is structurally perturbed at time 10000 by taking away 20% of its nodes, in our case 2. There is an immediate drop in the arrival rate due to the loss, which is steadily, but slowly made up. This is due to the small rate adjustment constant ϵ. Is it possible to have a speedier recovery? The gap in the message generation rate suggests using λ as an indicator of sudden underutilization. There is a more uniform approach based on *warp* which in this case works opposite to what was done in RRP. Instead of watching for a sharp increase in *warp*, each node upon seeing a sharp decrease will give its generation rate a temporary boost, thereby quickening the response.

Boost rate protocol (BRP):

1. If $warp_i < 1 - warp\_threshold$ then do

    Set $new\,\lambda_i = old\,\lambda_i * (1 + warp)$.

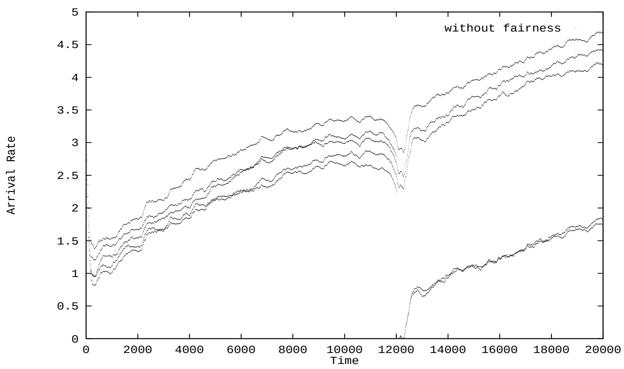
    Set $lock_i = lock\_constant * (1 - warp_i)$.

2. Turn off BRP and RRP for $lock_i$ steps.

Figure 11 (right) shows the effect of running RAP in conjunction with BRP for a 10 node system where 20% node perturbation is applied at time 10000. The arrival rate again experiences a sharp drop, but in contrast to figure 11 (left), the arrival rate immediately jumps back and settles to a rate close to its pre-perturbation level. It is important to shut down BRP and RRP for a short moment following the boost in order to avoid positive feedback. Structural perturbations involving the service rate are dealt with in a similar manner using RRP.

5.3 Fairness

Due to accumulation of random noise and increase in variance, $E_i((\lambda_i(t) - \lambda(t)/n)^2)$ will grow as $t \to \infty$. Figure 12 (left) shows the deviation in bandwidth allocation for a 10 node system with optimal service rate 30. The time evolution of the arrival rates for the first 5 nodes are shown. The drift is clearly visible. At time 12000, the RRP protocol performs a recovery routine, and two nodes "luck out" by performing a back-off operation that is costlier than the others. This further increases the variation. Figure 12 (right) shows a simulation run of the previous system under the exact same conditions except that a fairness protocol was active. The fairness protocol works by



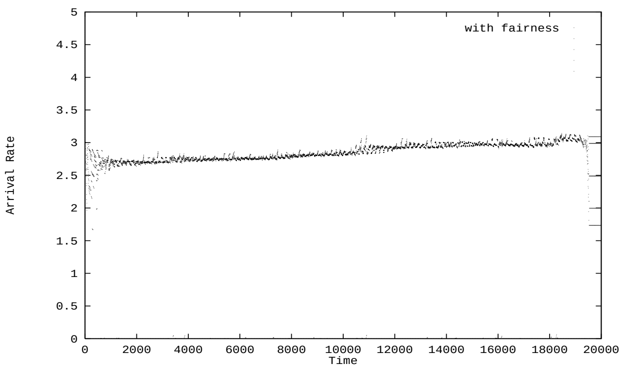


Figure 12: Simulation of a 10 node system with (right), and without (left), fairness protocol.

including λ in the output packet at regular time intervals, which then at the receiver's end is used to perform incremental averaging of the message generation rate. The scheme used in the above simulation is a nonincremental version whereby the receiver's λ value is replaced with the average of its sampled neighbors', every 40 time units. One interesting side effect of averaging is to speed up the initial convergence rate as can be seen in figure 12.

6 Conclusion

A characterization of communication networks modeled as stochastic single-server queues was presented, and its global dynamics analyzed with respect to stability, optimality, and fairness. Based upon the notion of an *inherent* load-throughput function and its unimodality assumption, stable and unstable regions in phase space were identified, and protocols were proposed for maintaining stable, near-optimal utilization. The basis of the protocol suite, rate adjustment protocol, was facilitated by a quantity called *warp* which was shown to estimate network utilization. The most interesting aspect of this protocol is its tendency to converge and "linger" at the optimal operating point, which then can be exploited to achieve load matching by not crossing over into the unstable region.

Several issues remain to be resolved. First, perform extensive simulations on actual network topologies with specific link rates, buffer sizes, switching speeds, and routing algorithms. In tandem, extend the model by incorporating more complicated coupling mechanisms. Second, explicitly model the burstiness of traffic sources using the recently advanced self-similar characterization of Ethernet traffic [20]. Third, consider alternative methods for modeling congestion based on functional relationships between service rate and effective load, and the necessary machinery to achieve tractable analysis.

Finally, we remark that in large-scale, high-speed networks, it is not enough to find analytical models of individual components, study their behavior, and conclude stability or optimality for the system as a whole. Unless the control protocol is extremely conservative in its resource usage (therefore probably highly suboptimal with respect to throughput), due to the nonlinear nature of the interaction among traffic flows sharing the communication network, the global dynamics of the system may be quite different from what was expected.

Acknowledgements

The author would like to thank Abdelsalam Heddaya, Eugene Pinsky, and Himanshu Sinha for many helpful discussions. The author was supported in part by NSF grant CCR-9204284. A short version of this paper appears in [26].

References

[1] Ludwig Arnold. *Stochastic Differential Equations: Theory and Applications.* John Wiley & Sons, 1974.

[2] K. Bala, I. Cidon, and K. Sohraby. Congestion control for high speed packet switched networks. In *Proc. IEEE INFOCOM '90*, pages 520–526, 1990.

[3] Ernst Biersack. Performance evaluation of forward error correction in ATM networks. In *Proc. ACM SIGCOMM '92*, pages 248–257, 1992.

[4] Jean-Chrysostome Bolot and A. Udaya Shankar. Analysis of a fluid approximation to flow control dynamics. In *Proc. IEEE INFOCOM '92*, pages 2398–2407, 1992.

[5] Imrich Chlamtac and William R. Franta. Rationale, directions, and issues surrounding high speed networks. *Proc. IEEE*, 78(1):94–120, 1990.

[6] R. Dighe, C. J. May, and G. Ramamurthy. Congestion avoidance strategies in broadband packet networks. In *Proc. IEEE INFOCOM '91*, pages 295–303, 1991.

[7] A. E. Eckberg. B-ISDN/ATM traffic and congestion control. *IEEE Network*, pages 28–37, September 1992.

[8] K. Fendick, M. Rodrigues, and A. Weiss. Analysis of a rate-based control strategy with delayed feedback. In *Proc. ACM SIGCOMM '92*, pages 136–148, 1992.

[9] M. Gerla and L. Kleinrock. Flow control: a comparative survey. *IEEE Trans. Commun.*, COM-28:553–574, 1980.

[10] Z. Haas and J. Winters. Congestion control by adaptive admission. In *Proc. IEEE INFOCOM '91*, pages 560–569, 1991.

[11] Zygmunt Haas. A communication architecture for high-speed networking. In *Proc. IEEE INFOCOM '90*, pages 433–441, 1990.

[12] Jack K. Hale. *Functional Differential Equations.* Springer-Verlag, 1971.

[13] A. Heddaya, K. Park, and H. Sinha. Using warp to control network contention in mermera. *To appear in HICSS-27*, 1994.

[14] Duke Hong and Tatsuya Suda. Congestion control and prevention in ATM networks. *IEEE Network Magazine*, pages 10–16, July 1991.

[15] A. Hordijk and F. Spieksma. Constrained admission control to a queueing system. *Adv. Appl. Prob.*, 21:409–431, 1989.

[16] Van Jacobson. Congestion avoidance and control. In *Proc. ACM SIGCOMM '88*, pages 314–329, 1988.

[17] F. P. Kelly. The optimization of queueing and loss networks. In O. J. Boxma and R. Syski, editors, *Queueing Theory and its Applications*. North-Holland, 1988.

[18] S. Keshav. A control-theoretic approach to flow control. In *Proc. ACM SIGCOMM '91*, pages 3–15, 1991.

[19] H. Kobayashi. Optimal routing in closed queueing networks. *ACM Trans. Comp. Sys.*, 1(4):294–310, 1983.

[20] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of ethernet traffic. In *Proc. ACM SIGCOMM '93*, pages 183–193, 1993.

[21] N. Maxemchuk and M. Zarki. Routing and flow control in high-speed wide-area networks. *Proc. IEEE*, 78(1):204–221, 1990.

[22] D. Mitra and J. Seery. Dynamic adaptive windows for high speed data networks: theory and simulations. In *Proc. ACM SIGCOMM '90*, pages 30–37, 1990.

[23] S-E A. Mohammed. *Stochastic functional differential equations*, volume 99 of *Research Notes in Mathematics*. Pitman Advanced Publishing Program, 1984.

[24] A. Mukherjee and J. Strikwerda. Analysis of dynamic congestion control protocols - a Fokker-Planck approximation. In *Proc. ACM SIGCOMM '91*, pages 159–169, 1991.

[25] G. F. Newell. *Applications of Queueing Theory*. Chapman and Hall, 2nd edition, 1982.

[26] Kihong Park. Warp control: a dynamically stable congestion protocol and its analysis. In *Proc. ACM SIGCOMM '93*, pages 137–147, 1993.

[27] G. Ramamurthy and R. S. Dighe. Distributed source control: a network access approach to integrated broadband packet networks. In *Proc. IEEE INFOCOM '90*, pages 896–907, 1990.

[28] J. Robinson, D. Friedman, and M. Steenstrup. Congestion control in BBN packet-switched networks. *Computer Communication Review*, 20(1):76–90, 1990.

[29] Scott Shenker. A theoretical analysis of feedback flow control. In *Proc. ACM SIGCOMM '90*, pages 156–165, 1990.

[30] M. Sidi, W. Liu, I. Cidon, and I. Gopal. Congestion control through input rate regulation. *IEEE Trans. Communications*, 41(3):471–477, 1993.

[31] Y. T. Wang and B. Sengupta. Performance analysis of a feedback congestion control policy under non-negligible propagation delay. In *Proc. ACM SIGCOMM '91*, pages 149–157, 1991.