**Proceedings of the 39th IEEE
Conference on Decision and Control
Sydney, Australia • December, 2000**

# Stability of Hop-by-Hop Congestion Control

Stephan Bohacek
Department of Mathematics
University of Southern California
1042 West 36th Place, DRB 155
Los Angeles, CA 90089-1113
bohacek@usc.edu

## Abstract

A hop-by-hop congestion control method is developed. Unlike other hop-by-hop schemes, this method does not require the router to keep track of per-virtual circuit information. Hence, this method puts little computational burden on the router. The method is hop-by-hop based, hence, it allows the flows to quickly adjust to changes in the available bandwidth. The network is modeled as an LPV system. However, standard LPV techniques prove too conservative and alternative methods are applied. It is shown that for certain feedback gains, the system is exponentially stable.

## 1 Introduction

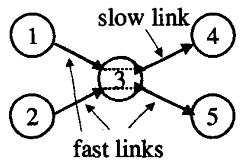
Congestion control in data networks has been the subject of extensive research since the Internet experienced congestion collapse in 1988 [5]. Currently, TCP [16] is the most popular algorithm for avoiding large scale congestion. According to the TCP protocol, the sender attempts to use any available bandwidth in the network by sensing if packets have been dropped. This sensing is accomplished by the receiver sending acknowledgments upon receiving a packet. If the sender receives an acknowledgment, it assumes that the network is not congested. Whereas, if the sender fails to receive an acknowledgment, then it assumes that the packet has been dropped indicating congestion. In this either case, the sender takes action in an attempt to utilize all available bandwidth and minimize congestion. This approach has been widely tested and preforms satisfactorily. Another congestion control method is rate based feedback for ATM networks [6]. These rate based methods make use of explicit "state" information from the routers internal to the network. Typically, the routers send congestion information to the receiver, which then passes this information on to the sender. Simulation has shown that these rate based methods perform well. Both TCP and rate based feedback are end-to-end methods. That is, the result of action taken by the sender is detected first by the routers. Either directly or indirectly, the routers convey the congestion level to the receiver. The receiver then relays this information back to the sender. Based on this information, the sender adjust its sending rate. One drawback of such and end-to-end approach is that there may be a large delay between when congestion occurs and when the sender reacts. Similarly, there may be a large delay between when the network load decreases and the sender makes use of the newly available bandwidth. This reaction time may be on the order of the lifetime of the connection had the sender been sending data at the maximum possible rate. For example, suppose that a 50kB file is to be sent from Los Angeles to New York. The time it takes a packet to travel from LA to NY and back, including propagation and queueing delay, is typically greater than 200ms. At 250kB/s, the entire file could be sent in 200ms. That is, by the time the sender is able to respond any change the available bandwidth, the transaction could have been completed. Of course, this assumes that 250kB/s of bandwidth is available. However, as bandwidth becomes cheaper (the price decreases by a factor of 2 every nine months), large amounts of available bandwidth may become common [12]. In this sense, end-to-end approaches may become inefficient.

Another approach is hop-by-hop congestion control [8], [10], [11], [13]. Here each router communicates with upstream routers to negotiate the rate at which data should be sent. Thus, each host must only communicate with a host one hop away. Typically, the delay over single hops is small compared to the round-trip delay. Hence, the sender and intermediate routers are able to quickly respond to changes in the available bandwidth. Such a scheme amounts to a series of feedback loops; one loop for each hop. It has been shown that this series of loops is stable [11]. However, most approaches require that the control be applied only to single connections, or virtual circuits (VCs). That is, if there are a million senders and receivers that communicate via a particular router, then this router must apply this control scheme to each of the million connections individually. At gigabit rates, the computational and memory burden on the router becomes unbearable. For this reason, hop-by-hop schemes were not chosen by the ATM steering group [6].

Mixed end-to-end/hop-by-hop approaches have also been suggested. For example, [1] discusses a split-TCP approach. Here a single connection is split into two consecutive TCP connections. This approach has the advantage that designers are familiar with TCP which is assumed to have good global stability properties. However, the split-TCP has a faster reaction time than standard TCP. One drawback is that the router which splits the connection must keep detailed information about the connection. This approach is well suited for wireless networks, where the packet loss between a fixed host and the base station is due to network congestion, whereas packet loss between the mobile host and base station is also due to transmission errors.

In [7] and [14] a hop-by-hop scheme was developed that did not require routers to keep track of individual flows. The router simply examines its queue size, the size of its neighbors queues and the current rate at which data is traveling between routers. Based on this information, the router increases or decreases its out-going rates. In [7], it was shown that when applied globally, this approach leads to a stable network. However, this approach suffers from a problem known as *blocking*. Consider the topology depicted in the figure below. Node 1 is sending data to node 4 via node 3, while node 2 is sending data to node 5 via node 3. Suppose that links $(1,3)$, $(2,3)$ and $(3,5)$ are high bandwidth links, while link $(3,4)$ is low bandwidth. Since data is traveling from node 1 to 3 at a high rate and the link between node 2 and 4 is slow, the queue in node 3 fills. As this queue fills, the method developed in [7] dictates that *both* nodes 1 and 2 decrease their rate. The problem is that node 2 decreases its rate even though the data it is sending to node 3 is continuing on to node 5 across a high speed link. That is, node 2 is not responsible for the congestion in node 3, yet it decreases its rate anyway. This problem is addressed in this paper and is alleviated by designing a control law so that node 2 only decreases its rate when the traffic it is sending it responsible for the congestion.

The paper proceeds as follows. Section 2 develops a system of differential equations that represent a network. Section 3 presents the main results and shows that, for a particular set of feedback gains, the control law suggested results in a stable network.



Blocking. Although the data flowing over link $(2,3)$ proceeds to link $(3,5)$, the slow link, $(3,4)$ leads to the a slower rate over $(2,3)$.

## 1.1 Notation

| | |
|---|---|
| Link $(i,j)$: the link between node $i$ and node $j$. | |
| $q_{i,j}(t)$: the size of the out-going queue on link $(i,j)$. | |
| $r_{i,j}(t)$: the data rate from node $i$ to node $j$. | |
| $\alpha_{i,j,k}(t)$: | the portion of data that is traveling from node $i$ to $j$ that then travels to node $k$. |
| $I_i(t)$: the data input rate into node $i$. | |
| $F_i$: scalar, time-invariant feedback gains. | |
| $x^{(p)} = [x_1^p, \cdots, x_n^p]^T$, element-wise exponentiation. | |
| $\rho$: the no-loop parameter. See (12). | |
| $n$: the number of nodes in the network. | |
| $\bar{q}$: the maximum allowable size of queue. | |
| $\bar{r}$: the maximum rate on link. | |
| $\mathcal{L}_q$: the state limiter for the queues, see (2). | |
| $\mathcal{L}_r$: the state limiter for the rates, see (4). | |
| $\mathcal{L} = \begin{pmatrix} \mathcal{L}_q \\ \mathcal{L}_r \end{pmatrix}$. | |
| $\|x\|_\infty = \max \|x_i\|$. | |
| $\|x\|_p = (\sum_i \|x_i\|^p)^{1/p}$. | |
| $Co(x) = \{y : y = \sum \alpha_i x_i \text{ with } \sum \alpha_i = 1, \text{ and } \alpha \geq 0\}$. | |

## 2 Data Network Model

We use a fluid flow model of the network where $q_{i,j}$ is the size of the out-going queue from node $i$ to node $j$, $r_{i,j}$ is the rate that data is sent from node $i$ to node $j$, $I_i$ is the rate at which data enters the network at node $i$ and $\alpha_{i,j,k}$ is the portion of the data that is going from node $i$ to node $j$ that will proceed to node $k$. Furthermore, the term $\alpha_{o,i,j} I_i$ represents the amount of data that enters the network at node $i$ and enters the $i,j$ queue. Note that the dependence of $q$, $r$, $I$ and $\alpha$ on time is suppressed.

Using this notation, the dynamics of the queue are

$$\dot{q}_{i,j} = \mathcal{L}_{q_{i,j}} \left( -r_{i,j} + \sum_{h=1}^{n} \alpha_{h,i,j} r_{h,i} + \alpha_{o,i,j} I_i \right), \quad (1)$$

where the nonlinear operator $\mathcal{L}$ accounts for queue saturation, i.e.

$$\mathcal{L}_{q_{i,j}}(v) := \begin{cases} v \text{ if } q_{i,j} \neq 0, \bar{q} \\ v \text{ if } q_{i,j} = 0 \text{ and } v > 0 \\ v \text{ if } q_{i,j} = \bar{q} \text{ and } v < 0 \\ 0 \text{ otherwise} \end{cases} \quad (2)$$

The control law will determine the out-going rates $r_{i,j}$. In particular, we consider controllers of the form

$$\dot{r}_{i,j} = \mathcal{L}_{r_{i,j}} \left( F_1 q_{i,j} + F_2 \dot{q}_{i,j} - F_3 \sum_{k=1}^{n} \alpha_{i,j,k} q_{j,k} \right. \quad (3)$$
$$\left. - F_4 \sum_{k=1}^{n} \alpha_{i,j,k} \dot{q}_{j,k} + \sum_{h=1}^{n} \alpha_{h,i,j} \left( F_5 q_{i,j} + F_6 \dot{q}_{i,j} \right) \right),$$

where

$$\mathcal{L}_{r_{i,j}}(v) := \begin{cases} v \text{ if } r_{i,j} \neq 0, \bar{r} \\ v \text{ if } r_{i,j} = 0 \text{ and } v > 0 \\ v \text{ if } r_{i,j} = \bar{r} \text{ and } v < 0 \\ 0 \text{ otherwise.} \end{cases} \tag{4}$$

and $F_m \geq 0$ are time-invariant scalar gains. This controller structure can be intuitively explained as follows. The term $F_1 q_{i,j} + F_2 \dot{q}_{i,j}$ leads to the rate increasing if $q_{i,j}$ is large or increasing. Thus the node tries to empty its queue. The term $-F_3 \sum_{k=1}^{n} \alpha_{i,j,k} q_{j,k} - F_4 \sum_{k=1}^{n} \alpha_{i,j,k} \dot{q}_{j,k}$ is known as back pressure and leads to the rate decreasing if the next hop is congested or is becoming congested. However, $r_{i,j}$ decreases in relation to how much it is responsible for the congestion. For example, if all the data flowing from node $i$ to node $j$ will next go to node $h$, then $-F_3 \sum_{k=1}^{n} \alpha_{i,j,k} q_{j,k} - F_4 \sum_{k=1}^{n} \alpha_{i,j,k} \dot{q}_{j,k} = -F_3 q_{j,h} - F_4 \dot{q}_{j,h}$; hence $r_{i,j}$ is independent of the congestion in $q_{j,k}$ with $k \neq h$. This feature alleviates the blocking problem as discussed in the Introduction. The term $F_5 \sum_{h=1}^{n} \alpha_{h,i,j} q_{i,j} + F_6 \sum_{h=1}^{n} \alpha_{h,i,j} \dot{q}_{i,j}$ is referred to as forward pressure and forces the downstream rates to increase in the event of upstream congestion. Again, this control action takes into account the degree that this rate could effect the upstream congestion. This forward pressure is a unique aspect of this hop-by-hop scheme.

In order to elucidate the structure of the system, consider the simple network shown in Figure 1. The dynamics for the queues are

$$\begin{aligned} \dot{q}_{1,2} &= \mathcal{L}_{q_{1,2}}\left(-r_{1,2} + \alpha_{3,1,2} r_{3,1} + \alpha_{o,1,2} I_1\right), \\ \dot{q}_{1,3} &= \mathcal{L}_{q_{1,3}}\left(-r_{1,3} + \alpha_{2,1,3} r_{2,1} + \alpha_{o,1,3} I_1\right), \\ &\vdots \\ \dot{q}_{4,3} &= \mathcal{L}_{q_{3,1}}\left(-r_{4,3} + I_4\right), \\ \dot{q}_{i,OUT} &= 0, \end{aligned}$$

Define
$q := \begin{bmatrix} q_{1,2} & q_{1,3} & q_{2,1} & q_{2,3} & q_{3,1} & q_{3,2} & q_{3,4} & q_{4,3} \end{bmatrix}$
and $r$ similarly. Define $A :=$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & \alpha_{3,1,2} & 0 & 0 & 0 \\ 0 & 0 & \alpha_{2,1,3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha_{3,2,1} & 0 & 0 \\ \alpha_{1,2,3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_{2,3,1} & 0 & 0 & 0 & 0 \\ 0 & \alpha_{1,3,2} & 0 & 0 & 0 & 0 & 0 & \alpha_{4,3,2} \\ 0 & \alpha_{1,3,4} & 0 & \alpha_{2,3,4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5}$$

then

$$\dot{q} = \mathcal{L}_q\left(\left(\mathcal{A} - I\right) r\right),$$

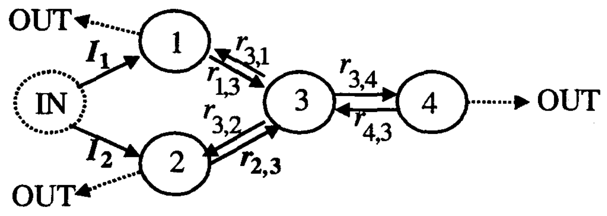


**Figure 1:** A simple topology with inputs and outputs.

where $\mathcal{L}_q$ accounts for the queue saturation. Similarly,

$$\begin{bmatrix} \sum_{k=1}^{n} \alpha_{1,2,k} q_{2,k} \\ \vdots \\ \sum_{k=1}^{n} \alpha_{4,3,k} q_{3,k} \end{bmatrix} = \mathcal{A}^T q.$$

Hence the dynamics of the closed-loop network given by (1) and (3) can be written

$$\begin{bmatrix} \dot{q} \\ \dot{r} \end{bmatrix} = \mathcal{L}\left(\begin{bmatrix} 0 & \mathcal{A} - I \\ F_1 I - F_3 \mathcal{A}^T + F_5 \mathcal{A} & F_2(\mathcal{A} - I) - F_4 \mathcal{A}^T(\mathcal{A} - I) + F_6 \mathcal{A}(\mathcal{A} - I) \end{bmatrix} \begin{bmatrix} q \\ r \end{bmatrix}\right); \tag{6}$$

where $\mathcal{L}$ accounts for the state saturation of the queues and rates.

Before discussing performance issues of this control method, the matrix $\mathcal{A}$ must be examined. Since $\alpha_{i,j,k}$ is the portion of data that is flowing from across link $(i,j)$ that will flow across link $(j,k)$, $\sum_{h=1}^{n} \alpha_{h,i,j} r_{h,i}$, is the amount of data flowing from any other node, into the queue $(i,j)$. Similarly, $\sum_{k=1}^{n} \alpha_{i,j,k}$, is the portion of data going from node $i$ to node $j$ that will go to any other node. Since some data may leave the network, not all of the data that travels from node $i$ to node $j$ will travel to another node; hence $\sum_{k=1}^{n} \alpha_{i,j,k} \leq 1$. Furthermore, if some of the data that enters node $i$ exits the network, then $\sum_{k=1}^{n} \alpha_{i,j,k} < 1$. It is assumed that the data is able to exit as fast as necessary. For example, if node $i$ is a end host, then $\sum_{j=1} \alpha_{h,i,j} = 0$. That is, any data that enters this node goes to the application and not into any out-going queue. Similarly, suppose that the network under control connects with a network without this hop-by-hop flow control. Since this extraneous network does not send control information, it is modeled as a node with a queue that never fills. That is, $\sum_{j=1}^{n} \alpha_{h,i,j} = 0$. In Figure 1, note that the nodes *IN* and *OUT* are fictitious in that they represent *all* the inputs and *all* the outputs. Furthermore, note that $\sum_k \alpha_{3,4,k} = 0$, that is all the data that enters link $(3,4)$ leaves the network.

In all cases there must be some data outlets. Thus,

$$\sum_j \alpha_{h,i,j} < 1 \text{ for some } h, i. \tag{7}$$

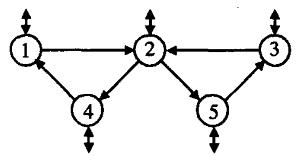


**Figure 2:** A toplogy susceptible to loops

Furthermore, for a properly working network, packets should not travel in loops. This property of networks will be referred to as the *no-loop* condition. For example, in Figure 2 we see that if $\alpha_{1,2,4}\alpha_{2,4,1}\alpha_{4,1,2} = 1$, $\alpha_{2,5,3}\alpha_{5,3,2}\alpha_{3,2,5} = 1$, or $\alpha_{2,4,1}\alpha_{4,1,2}(\alpha_{1,2,4}+\alpha_{1,2,5})\alpha_{2,5,3}\alpha_{5,3,2}(\alpha_{3,2,5}+\alpha_{3,2,4}) = 1$, then all the data that enters this network remains in the network. Clearly this is an anomaly and results in the queue overflowing. Let $LL$ be a set of links such that if $(i,j) \in LL$, then $(j,k) \in LL$ for some $k$ and $(h,i) \in LL$ for some $h$. That is, $LL$ is a set of links that makes up a loop, or a set of loops. The no-loop condition implies that any such set $LL$, we have

$$\prod_{\{(i,j)\in LL\}} \sum_{\{k:(j,k)\in LL\}} \alpha_{i,j,k} < 1 \tag{8}$$

Note that this idea is closely related to the idea of circuit product [2]. The difference is that the circuit product does not include the summation.

Define a mapping $\phi : I \times I \to I^2$ so $\phi(i,j)$ is some unique number (i.e. a mapping from link between $i$ and $j$ to link number $\phi(i,j)$) and define $\mathcal{A}_{\phi(j,k),\phi(i,j)} = \alpha_{i,j,k}$ as in (5). Using this notation we see that row sums of $\mathcal{A}^T$ are less than one, i.e.

$$\sum_a \mathcal{A}_{a,b} = \sum_{k=1}^{n} \mathcal{A}_{\phi(j,k),\phi(i,j)} \leq 1. \tag{9}$$

Hence, $\mathcal{A}^T$ is a *substochastic* matrix, where a *stochastic* matrix $\mathcal{P}$ is defined by Markov transition probabilities, where $\mathcal{P}_{a,b}$ is the probability of making a transition from state $a$ to state $b$. For stochastic matrices the rows sums are one, for substochastic matrices, the row sums are at most one, and for *strictly substochastic* matrices, the row sums are strictly less than one.

The condition (8) is equivalent to

$$\left(\mathcal{A}^k\right)_{a,b} \to 0 \tag{10}$$

for all $a, b$ as $k \to \infty$. Note that this condition is equivalent to

$$|eig(\mathcal{A})| < 1. \tag{11}$$

To see the that the no-loop condition implies (10), augment $\mathcal{A}$ as follows. Define $g_a = 1 - \sum_a \mathcal{A}_{a,b}$ and define

$$P := \begin{pmatrix} 1 & g \\ \vec{0} & \mathcal{A} \end{pmatrix},$$

where $\vec{0}$ is a vector of all zeros. In this case, $P^T$ is a stochastic matrix, and the no-loop condition implies that the state 1 is absorbing. Hence

$$\left(P^T\right)^k \to \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

Thus, $\left(\left(\mathcal{A}^T\right)^k\right)_{a,b} \to 0$ and $\left(\mathcal{A}^k\right)_{a,b} \to 0$. Inequality (11) easily follows. Note that Mason [9] has a similar characterization of loop-free routing. However, Mason focuses on a the static case, whereas here we focus on the case where $\alpha$ varies with time.

We quantify the no-loop condition by requiring that there exists a fixed $1 \geq \rho > 0$ such that any set of loops $LL$, we have

$$\prod_{\{(i,j)\in LL\}} \sum_{\{k:(j,k)\in LL\}} \alpha_{i,j,k} < 1-\rho. \tag{12}$$

The matrix $\mathcal{A}$ has many other properties. Since $\alpha_{i,j,k} = \mathcal{A}_{\phi(i,j),\phi(j,k)}$, we must have $\mathcal{A}_{\phi(k,j),\phi(i,j)} = 0$. In general,

$$\mathcal{A}_{a,b}\mathcal{A}_{b,a} = 0 \text{ and } \mathcal{A}_{a,a} = 0. \tag{13}$$

We are interested in dynamical systems involving $(\mathcal{A} - I)$ or $\mathcal{A}^T - I$.

**Lemma 1** *Let*

$$\dot{x}(t) = \left(\mathcal{A}^T(t) - I\right)x(t), \tag{14}$$

*where $\mathcal{A}^T(t)$ is a time-varying substochastic matrix. Then $\frac{d}{dt}\|x(t)\|_\infty \leq 0$.*

**Proof:** Since $\mathcal{A}^T$ is substochastic $\mathcal{A}^T(t)x_i(t) \in Co(x(t))$. Hence, if $x_{i^*} = \|x(t)\|_\infty$, then $\dot{x}_{i^*}(t) \leq 0$. ■

It is not hard to show that the set of substochastic matrices satisfying the no-loop condition describe above is convex. Since the parameters $\alpha_{i,j,k}$ can vary arbitrarily fast, the matrix $\mathcal{A}^T$ can vary arbitrarily fast. Thus, (14) is linear parametrically varying (LPV) system. LPV systems have been extensively studied [3]. These methods dictate that the stability of $\mathcal{A}^T$ can be proved by searching for a single matrix $P > 0$ such that $P\left(\mathcal{A}^T - I\right) + (\mathcal{A} - I)P < 0$ for all matrices that conform to the no-loop conditions specified above. However, this approach is conservative and, in this case, is certain to fail. For example, consider the strictly substochastic matrices with $\sum_b \mathcal{A}^T_{a,b} \leq 1-\rho$, for all $a$, and temporarily neglect the conditions (13). In particular, suppose that $\mathcal{A}$ is restrict to the convex polytope with extreme points

$$\mathcal{A}_1^T := \begin{pmatrix} 1-\rho & 0 \\ 1-\rho & 0 \end{pmatrix}, \qquad \mathcal{A}_2^T := \begin{pmatrix} 1-\rho & 0 \\ 0 & 0 \end{pmatrix},$$
$$\mathcal{A}_3^T := \begin{pmatrix} 0 & 1-\rho \\ 0 & 1-\rho \end{pmatrix}.$$

Defining $P = \begin{pmatrix} P_1 & P_2 \\ P_2 & P_3 \end{pmatrix}$, we search for $P > 0$ with $P\left(\mathcal{A}_i^T - I\right) + \left(\mathcal{A}_i - I\right)P < 0$ for $i = 1,2,3$. This reduces to

$$\begin{aligned} -2P_3P_2\left(1-\rho\right)^2 - P_3^2\left(1-6\rho+\rho^2\right) - P_2^2\left(1+\rho\right)^2 &> 0, \\ 4P_3^2\rho - P_2^2\left(1+\rho\right)^2 &> 0. \end{aligned}$$

It is not hard to check that for $\rho < 0.07$, there is no solution with $P_3 > 0$. However, (14) with $\mathcal{A}^T(t) \in Co\left(\mathcal{A}_1^T, \mathcal{A}_2^T, \mathcal{A}_3^T\right)$, is a *strict* contraction under the sup norm. To see this note that if $x_i(t) = \|x(t)\|_\infty > 0$, then $\left(\mathcal{A}^T(t)\,x(t)\right)_i < x_i(t)$, hence, $\frac{d}{dt}\|x(t)\|_\infty \le 0$ with equality only when $x(t) = 0$.

**Remark 1** *While it is true that the set of matrices $Co(\mathcal{A}_i : i = 1,2,3)$ does not comply with (13), it is not difficult to construct higher dimensional examples that satisfy all the conditions and reduce down to the this set.*

In light of the above, it is clear that a less conservative approach for assessing stability must be used to prove the stability of (14). [4] and [15] developed less conservative LPV methods. A variation of [4] will be used. One can think of the sup norm as the following limit $\|x\|_\infty = \lim_{q\to\infty}\|x\|_q = \lim_{q\to\infty}\left(\sum_i x_i^q\right)^{1/q}$. Inspired by Lemma 1, [4] and [15], we search for a Lyapunov function that is not quadratic.

**Theorem 2** *Let $\dot{x}(t) = (\mathcal{A}(t) - I)\,x(t)$ where $\mathcal{A} \in \mathbb{R}^{n\times n}$ varies arbitrarily over the set of substochastic matrices described above. Define the Lyapunov function $V(x) = \left(x^T\right)^{(p-1)}x = \sum_i x_i^p$.with $p > 1$ and $\varepsilon > 0$ such that $\frac{1}{p} + \frac{1}{q} = 1$ and $q > 0$ and $\varepsilon$ satisfy*

$$\left(\frac{q-1}{q}\frac{1}{\left((1-\varepsilon) - \frac{n}{q}\left(\frac{q-1}{(1+\varepsilon)q}\right)^{q-1}\right)}\right)^{q-1}\frac{1}{q} - \rho + \varepsilon < 0. \tag{15}$$

*Then $\frac{d}{dt}V(x(t)) < -\varepsilon p\,\|x(t)\|_p^p$.*

**Remark 2** *The parameters p and $\varepsilon$ can be found directly by finding p and $\varepsilon$ such that*

$$\begin{aligned} 0 \ge &-(1-\varepsilon)\,x_1^p - (1-\varepsilon)(n-2)\,x_i^p - 1 \\ &+ (n-2)\,x_1^{p-1}x_i + (1-\rho)\,x_i^{p-1} + x_1, \end{aligned}$$

*where*
$(n-2)\,x^{p-1} + (1-\rho)\,px_i^{p-2} - p(1-\varepsilon)(n-2)\,x_i^{p-1} = 0$
*and*

$$x_i = \frac{p(1-\varepsilon)\,x^{p-1} - 1}{(p-1)(n-2)\,x^{p-2}}.$$

*Figure 3 shows a plot of a suitable p and $\varepsilon$ versus n, the number of nodes in the system.*

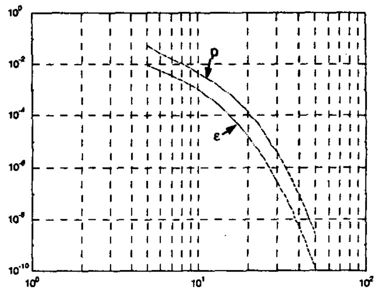


**Figure 3:** The figures above show values of $\varepsilon$ and $p$ that result in $\max_x\left(\dot{V}(x) - \varepsilon\|x\|_{p+1}^{p+1}\right) \le 0$. In this example $\rho = 0.1$.

## 3 Stability of Hop-By-Hop Congestion Control

Next it will be shown that the for certain values of the feedback gains, the network is stable.

### 3.1 Simple Case

For a special choice of feedback gains, it is easy to show that the system is stable.

**Theorem 3** *Suppose that $F_a := F_1 = F_3$, $F_b := F_2 = F_4$, $F_5 = 0$ and $-F_b\rho^2 + F_6\sqrt{n}\left(\sqrt{n} - 1\right) < 0$. In this case, the system is asymptotically stable.*

### 3.2 General Case

While stability is easily proved in the previous subsection, it is not clear how well such a system would preform. In particular, since the system is merely asymptotically stable, it may be unstable for small variations in the gains $F_i$. Exponentially stable systems are more robust. Unfortunately, it is far more difficult to prove that (6) is exponentially stable.

**Theorem 4** *System (6) is exponentially stable if there exists a $P_{11}, P_{12}, P_{22} \in \mathbb{R}$, a p and an $\varepsilon$ given by Corollary 2 or Remark 2 and the following conditions hold,*

$$\begin{aligned} \begin{pmatrix} P_{11} & P_{12} \\ P_{12} & P_{22} \end{pmatrix} &> 0, \\ 2P_{11} + P_{12}F_2 &= 0, \\ F_1 &> F_3\sqrt{n}, \\ F_1 &> 0, \\ P_{22}F_3 &> -P_{12}F_6, \end{aligned}$$

$$F_2 > \frac{\bar{r}}{\bar{q}},$$

$$P_{22} > -P_{12}\frac{\bar{q}_1}{p\bar{r}_1^{p-1}},$$

$$\begin{aligned} pP_{22}F_2\varepsilon \quad > \quad & -P_{12}\left(1+\frac{G}{4}F_4^2n^2\right) \\ & +\frac{G}{4}\frac{P_{22}^2p^2}{-P_{12}}\left(F_1+F_5\sqrt{n}\right)^2 n \\ & +pP_{22}\left(F_4\sqrt{n}+F_6\sqrt{n}\left(\sqrt{n}+1\right)\right) \\ & +F_4nP_{22}p\left(F_1+F_5\sqrt{n}\right), \end{aligned}$$

*where*

$$G := \left(F_1 - \left(F_3+F_5\right)\sqrt{n}\right)^{-1}$$

*and $\bar{r}$ and $\bar{q}$ are the maximum values of the rates and flows respectively.*

**Remark 3** *If $F_2$ is taken to be large, then it is not difficult to satisfy the above constraints.*

**Remark 4** *Since n, the number of nodes in the network, appears in the above constraints, it seems that the above result does not scale very well. However, the proof shows that n could possibility be replaced by the maximum fan-in of any node. The fan-in is the number of inputs to a node. This possibility will be investigated in future work.*

**Remark 5** *Note that delay was not taken into account when assessing the stability of the network. This can be justified by the fact that single hops usually have small delay, and inter-router communications do not experience queuing delays.*

## 4 Conclusion

A hop-by-hop scheme that does not use VC information has been developed. By design, this method does not suffer from the blocking problem. The resulting system can be modeled as an LPV system. However, standard LPV stability techniques are inadequate and alternative approaches have been applied. It has been shown that with properly chosen feedback gains, the system is stable. Future work will focus on determining feedback gains that are not only stabilizing, but meet other performance objectives such as maximizing good-put and fairness. For example, while blocking is clearly unfair, there are situations where the method presented here is unfair. Extensions to this method could alleviate this problem. For example, in this paper, only information from neighboring routers is utilized. With information from more distant routers, more sophisticated control algorithms can be considered. However, as more information is utilized, delay and complexity become a concern. Indeed, in the limit, the control scheme would resemble previously proposed methods that utilize VC information.

## References

[1] H. Balakrishnan, V. N. Padmanabhan, Srinivasan, and R. H. Katz. A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM Transactions on Networking*, 5(6):756–769, 1997.

[2] R. B. Bapat and T. E. S. Raghavan. *Nonnegative Matrices and Applications*. Cambridge University Press, 1997.

[3] G. Becker, A. Packard, D. Philbrick, and G. Balas. Control of parametrically-dependent linear systems: A single quadratic lyapunov approach. In *American Control Conference*, 1993.

[4] S. Bohacek and E. Jonckheere. Linear set valued dynamically varying (LSVDV) systems. In *American Control Conference*, 2000.

[5] V. Jacobson. Congestion avoidance and control. In *Proceedings SIGCOMM '88*, pages 314–329, 1988.

[6] R. Jain. Congestion control and traffic management in atm networks recent advances and a survey. *Computer Networks and ISDN Systems*, 28:1723–1738, 1996.

[7] V. Kalharni, S. Bohacek, and M. Safonov. Stability issues in hop-by-hop based congestion control. In *The 36th Allerton Conference on Communication, Control and Computing*, 1998.

[8] W. K. Lai, C.-C. Hwang, and W. J. Hsiao. A proportional feedback scheme for atm networks. *Information Sciences*, 110:237–253, 1998.

[9] L. G. Mason. Equilibrium flows, routing patterns and algorithms for store-and -forward networks. *Large Scale Systems*, 8:187–209, 1985.

[10] P. P. Mishra and H. Kanakia. A hop-by-hop rate-based congestion control scheme. In *Conference Proceedings on Communications Architeures and Protocols*, pages 112–123, Baltimore, MD, August 17-20 1992.

[11] P. P. Mishra, H. Kanakia, and S. K. Tripathi. On hop-by-hop rate-based congestion control. *IEEE/ACM Transaction on Networking*, 4(2):224–239, 1996.

[12] A. M. Odlyzko. The current state a likely evolution of the Internet. In *Proceedings of Globecom '99*, pages 1869–1875. IEEE, 1999.

[13] C. Ozveren, R. Simcoe, and G. Varghese. Reliable and efficient hop-by-hop flow control. In *Proceedings of the Conference on Communication Architectures, Protocols and Applications*, pages 89–100, London, England, August 31 - September 2 1994.

[14] C. M. Pazos and M. Gerla. ATM virtual private networks for internet data traffic. In *Proc. Of MMNS '97*, Montreal, Canada, 1997.

[15] J. S. Shamma and D. Xiong. Set-valued methods for linear parameter varying systems. *Automatica*, 35:1081–1089, 1999.

[16] W. Stallings. *Data and Computer Communications*. Prentice-Hall, 1997.