A Theoretical Analysis of Feedback Flow Control

Scott Shenker
Xerox PARC
3333 Coyote Hill Road
Palo Alto, CA 94304

Abstract

Congestion is a longstanding problem in datagram networks. One congestion avoidance technique is feedback flow control, in which sources adjust their transmission rate in response to congestion signals sent (implicitly or explicitly) by network gateways. The goal is to design flow control algorithms which provide time-scale invariant, fair, stable, and robust performance. In this paper we introduce a simple model of feedback flow control, in which sources make synchronous rate adjustments based on the congestion signals and other local information, and apply it to a network of Poisson sources and exponential servers. We investigate two different styles of feedback, aggregate and individual, and two different gateway service disciplines, FIFO and Fair Share. The purpose of this paper is to identify, in the context of our simple model, which flow control design choices allow us to achieve our performance goals.

Aggregate feedback flow control, in which congestion signals reflect only the aggregate congestion at the gateways, can provide time-scale invariant and stable performance, but not fair or robust performance. The properties of individual feedback flow control, in which the congestion signals reflect the congestion caused by the individual source, depend on the service discipline used in the gateways. Individual feedback with FIFO gateways can provide time-scale invariant, fair, and stable performance, but not robust performance. Individual feedback with Fair Share gateways can achieve all four performance goals. Furthermore, its stability properties are superior to those of the other two design choices. By making robust and more stable performance possible, gateway service disciplines play a crucial role in realizing effective flow control.

1. Introduction

As datagram computer networks have grown in both size and usage, network congestion has become an increasingly common problem. Fortunately, there have been some recent advances in datagram network congestion avoidance techniques. The algorithmic ideas of Jacobson, as implemented in the 4.3bsd TCP networking code [Jac88], have helped alleviate the previously severe congestion in the Internet. Similarly, the DECbit algorithm of Ramakrishnan, Jain, and Chiu [Jai87, Jai88, Ram87, Ram88, Chi89] appears, based on simulations, to greatly reduce congestion. Both of these approaches, as well as the earlier but less well studied source quench mechanism [Pos81, Pos87, Fin89], are examples of *feedback flow control*. In this paper we analyze a qualitative model of feedback flow control, in order to better understand its power and limitations.

Whereas originally flow control was seen solely as a mechanism to ensure adequate buffer resources at the receiving end of a connection, feedback flow control is designed to avoid network congestion. In feedback flow control, sources adjust their flow in response to feedback, or congestion signals, sent by network gateways. In the case of DECbit, the feedback signal is a special congestion bit contained in each packet header; gateways which are experiencing congestion set this bit in packets as they pass through the gateway, and this signal reaches the source in the returning ACK packets. The TCP feedback flow control algorithm of Jacobson, which is similar in spirit, uses packet drops as an implicit feedback signal (the dropping of a packet is not explicitly communicated to the source, but is detected by the source's flow control algorithm via timeouts or duplicate acknowledgements). Source quench uses explicit choke packets sent by gateways directly to the source.

These different implementations of congestion avoidance all share the two common characteristics of feedback flow control: gateways sending congestion signals (either implicitly or explicitly) and sources adjusting their transmission rate in response to these signals. Note that there is an important conceptual separation between these two aspects of feedback flow control. The congestion signalling mechanism is only a function of the congestion in the network, and is not based on how the source will respond to that signal. Similarly, the rate adjustment algorithm is solely a function of the congestion signals received, and the source's local state, and is conceptually independent of the congestion signalling mechanism. We study the properties of this general approach to congestion avoidance. To this end, we introduce a simple qualitative model of feedback flow control which, while admittedly unrealistic, captures some essential aspects of the feedback flow control approach. This model allows us to explore the implications of different design choices in feedback flow control.

This is a revised version of the preprint *Making Flow Control Work in Networks: A Control-Theoretic Analysis of Gateway Service Disciplines*

The first basic design choice is the nature of the congestion feedback, of which there are two basic styles. In the aggregate feedback approach, the feedback sent to a given source is a function of the aggregate congestion at the gateway. All sources receive the same congestion signal from the gateway, regardless of which sources are actually causing the congestion. This feedback style was used in the original version of the DECbit scheme [Jai88, Ram88, Chi89] and also is implicitly present in the Jacobson algorithm (here the feedback signal is a property of the gateway's packet dropping algorithm, which is usually just to drop the arriving packet when there are no free buffers, and is independent of the TCP code itself). Alternatively, the feedback to a source can be more individualized, reflecting the relative traffic load of that particular source. This individual feedback approach has been adopted in the latest version of DECbit [Ram87, Jai87].

The other basic design choice is the nature of the service discipline at the gateway. This is of no concern when using aggregate feedback, but is of crucial importance for individual feedback. Most congestion control schemes assume that gateways use the FIFO service discipline [Ram87, Jai87, Fin89]. We consider the effect of using a rather different service discipline called *Fair Share* (FS), first introduced in [She89].

In order to evaluate these different design choices, we identify a set of very general performance goals for flow control algorithms. These goals are to provide uncongested throughput allocations that are time-scale invariant, fair, and stable. In addition, we require that the performance remain adequate even when the sources are not all using the same flow control algorithm. We include this requirement, which we call robustness, because we believe future networks will likely have some degree of diversity.

The purpose of this paper is to determine, in a qualitative manner, which design choices make for effective feedback flow control. All of the design choices we consider can provide time-scale invariant performance. The issue then becomes which of the other performance goals can be met with these time-scale invariant algorithms. We find that no time-scale invariant aggregate feedback flow control can guarantee fair or robust performance. While every time-scale invariant individual feedback flow control with FIFO gateways guarantees fair performance, no such algorithm guarantees robust performance. The precise stability properties of these two design choices, aggregate feedback and individual feedback with FIFO gateways, remain undetermined, but we conjecture that there are such algorithms that can guarantee stable performance. In contrast, we find that every time-scale invariant individual feedback flow control with Fair Share gateways guarantees fair and robust performance. Furthermore, we can show that there are such algorithms that can guarantee stable performance.

In the next section we introduce our mathematical model, defining the network, the feedback flow control algorithm, and the goals of flow control. In Section 3 we analyze the performance of various feedback flow control schemes, examining their ability to achieve each of the four performance goals in turn. We then, in Section 4, briefly discuss the applicability of these results to real networks and conclude with a summary.

In many ways this work is inspired by the pioneering efforts of Jacobson, Jain, Ramakrishnan, and Chiu. They have identified the control-theoretic aspects of feedback flow control (this connection is particularly clear in [Chi89] and [Jac88]) and have devised algorithms that produce remarkable benefits in real networks. Furthermore, Ramakrishnan, Chiu, and Jain [Ram87, Jai87] have recognized the inherent unfairness of aggregate feedback flow control schemes, and have introduced an individual feedback DECbit algorithm which delivers better fairness than their previous algorithm. The service discipline in this algorithm is still FIFO, but the congestion signals are now sent to selected connections.

In contrast, the work presented here is not practical and we do not propose a particular feedback flow control scheme. Rather, we attempt to provide a more systematic and abstract exploration of the control-theoretic issues, focusing especially on the previously underappreciated role of gateway service disciplines in achieving satisfactory congestion avoidance. Nagle [Nag87] was perhaps the first to identify this role, and indeed the Fair Share service discipline studied here is modelled after his proposal of Fair Queueing (see also [Dem89] and [Zha89]). Morgan [Mor89] has also investigated this role in a slightly different context.

In addition to their practical proposals, Chiu and Jain [Chi89] investigated a control-theoretic model of feedback flow control similar to the aggregate feedback model considered here, except that their model has only binary feedback and they focused primarily on linear rate adjustment functions. They produced a rather complete taxonomy and characterization of such rate adjustment algorithms in networks with a single congested gateway. They investigated issues of fairness and stability, but did not examine robustness (time-scale invariance was inherent in the model they considered). Also, since the feedback in [Chi89] was aggregate, the issue of service disciplines was not addressed. Because of the limitation of binary feedback, the feedback flow control in [Chi89] was either increasing or decreasing at every point, and thus the system was never in a steady state. Our analysis here concentrates almost exclusively on the steady state behavior, and thus our results are rather different than those of [Chi89].

There have also been many other theoretical discussions of how to achieve stable and fair throughput allocations. Hayden [Hay81], Gafni [Gaf82,84] and Jaffe [Jaf80,81] introduced algorithms which, through synchronous source responses to network gateway signals, produced stable and fair throughput allocations. Mosely [Mos84] defined an algorithm that was able to do this asynchronously. This line of inquiry utilizes a more general sort of feedback in which the congestion signals directly dictate the resulting transmission rate of the sources. Thus, their results depend on the rate adjustment algorithms and the signalling mechanisms being closely coupled. This is in contrast to our model, in which there is a conceptual independence between the two different pieces of feedback flow control. Our model is similar in style, but obviously not in detail, to currently implemented feedback flow control algorithms.

2. Mathematical Model

In this section we first present our mathematical model, describing in turn the network, service disciplines, feedback flow control, and performance goals. We then discuss the limitations of this model.

2.1. Network

We consider networks of communication lines connected by gateways. For notational simplicity, at every physical gateway we associate with each outgoing line a separate logical gateway (i.e. a gateway that connects five incoming lines to three outgoing lines will be considered logically as three distinct gateways, one for each outgoing line), so there is a one-to-one correspondence between gateways and communication lines (traffic on a line goes in only one direction). The traffic on these networks is composed of a number of connections, which are source-destination pairs, exchanging packets. The sources are Poisson, with r_i denoting the sending rate of connection i. The gateways are exponential servers with infinite buffer space, with gateway α having service rate μ^α. In addition to the queueing delay at each gateway, there are also latencies l^α (traffic-independent propagation delays) associated with each communication line. The set of source-destination pairs and the routing pattern are both assumed to be static. Thus, each network and traffic topology is completely described, for our purposes, by the two sets $\gamma(i)$ and $I(\alpha)$. $\gamma(i)$ is the set of gateways through which the i'th connection flows, and $I(\alpha)$ is the set of connections flowing through the α'th gateway. Let N^α be the number of connections flowing through the α'th gateway.

In steady state, the i'th connection will have an average queue length $Q^\alpha{}_i(\mathbf{r})$ at the α'th gateway. This queue length is a function of the vector of sending rates of all of the other connections using this gateway (we will use **bold** characters to denote vectors). The function $\mathbf{Q}^\alpha(\mathbf{r})$ depends on the service discipline used at the gateway. We will focus on two specific service disciplines, the FIFO service discipline and the Fair Share service discipline, which are discussed below.

We make two additional modelling approximations. First, we assume that the queue lengths $Q^\alpha{}_i(\mathbf{r})$ always reflect the current instantaneous sending rates; we neglect the transient equilibration process. Essentially we are assuming, for the purpose of this model, that the time scale of the transient equilibration process is fast compared to that of the flow control rate adjustment process. Second, in order to consider networks of gateways, we assume that the flow of a connection's packets out of a gateway still constitute a Poisson stream, regardless of the service discipline used (this assumption, while true for FIFO service, is not true for Fair Share service).

2.2. Service Disciplines

The service discipline is represented by the function $\mathbf{Q}(\mathbf{r})$. Since gateways in datagram networks have no *a priori* knowledge about connections (which is not true in reservation-based communication networks), the function $\mathbf{Q}(\mathbf{r})$ must be symmetric in $\mathbf{r}$ (i.e., a permutation of the r_i's results in the same permutation of the Q_i's). Furthermore, we assume that the service disciplines are time-scale invariant, which means that, holding $\boldsymbol{\rho} \equiv \frac{\mathbf{r}}{\mu}$ constant, $\mathbf{Q}(\mu\boldsymbol{\rho})$ must be independent of the server rate μ. We also make, for technical reasons, two monotonicity assumptions: (1) $\frac{\partial Q_i}{\partial r_j} \geq 0$, and (2) $Q_i > Q_j \Leftrightarrow r_i > r_j$.

The function $\mathbf{Q}(\mathbf{r})$ cannot be arbitrary; those that can be realized by a nonstalling service discipline (one in which the server is idle only when the queue is empty), must satisfy the constraint $\sum_{i=1}^{N} Q_i = g(\sum_{i=1}^{N} r_i)$ where $g(x) \equiv \frac{x}{\mu - x}$ [Cof80, Reg86]. The function must satisfy the further constraint that, numbering the connections so that the $\frac{Q_i}{r_i}$ are in increasing order, the following inequalities hold for each integer $k \in [1, N-1]$: $\sum_{i=1}^{k} Q_i \geq g(\sum_{i=1}^{k} r_i)$.

We will be primarily focusing on two service disciplines, FIFO and Fair Share. The FIFO service discipline makes no distinction between connections; packets are just serviced in order of arrival. This gives rise to the well-known result for the average queue sizes: $Q_i(\mathbf{r}) = \frac{\rho_i}{1 - \rho_{tot}}$ where $\rho_i \equiv \frac{r_i}{\mu}$ and $\rho_{tot} \equiv \sum_{i=1}^{N} \rho_i$.

The Fair Share service discipline, introduced in [She89], is designed to guarantee that each connection receives fair treatment at the gateway. It embodies the same intuition of protecting sources from each other that led to our version of Fair Queueing [Dem89] which essentially approximates a Head-of-Line Processor Sharing algorithm without using time-slicing. The similarity between Fair Share and Fair Queueing is based only on their being derived from the same intuition; we make no claims about the two algorithms being mathematically related.

The Fair Share service discipline is a preemptive priority queueing discipline which is perhaps best explained through the following example. In this example, there are four connections, labelled so that the r_i are in increasing order. There are also four priority classes. We will separate each connection's Poisson stream into several substreams, one for each priority class, with the total rate for each connection summing to r_i. All of connection 1's packets are in the highest priority class, and all of the other connections get the same rate r_1 of packets in the highest priority class. Similarly, the rest of connection 2's packets are in the second highest priority class, and all of the other connections get the same rate r_2-r_1 of packets in the second highest priority class. The pattern repeats until all of the throughput is assigned a priority.

FS	Priority Level			
connection	A	B	C	D
1	r_1	-	-	-
2	r_1	r_2-r_1	-	-
3	r_1	r_2-r_1	r_3-r_2	-
4	r_1	r_2-r_1	r_3-r_2	r_4-r_3

Table 1: The Fair Share Service Discipline

The resulting function for $\mathbf{Q}(\mathbf{r})$ can be computed in a straightforward manner from well-known results on preemptive priority service disciplines. With the r_i labelled in increasing order, the average queues can be defined recursively from the following relation:

$$Q_i(\mathbf{r})=\frac{1}{N-i+1}\left\{g(\sum_{k=1}^{N} Min(r_i,r_k))-\sum_{k=1}^{i-1} Q_k(\mathbf{r})\right\}$$

recalling that $g(x)\equiv\frac{x}{\mu-x}$. Note that locally Q_i depends only on those r_j such that $r_j<r_i$. The resulting triangularity of the matrix $\frac{\partial Q_i}{\partial r_j}$ is crucial in deriving the properties of the Fair Share service discipline.

2.3. Feedback Flow Control

Feedback flow control consists of two parts; congestion signals from gateways and rate adjustment at sources.

2.3.1. Congestion Signals: Each gateway α sends a congestion signal b^α_i to each connection i flowing through it. These signals b^α_i are real numbers in [0,1], with a signal of 1 reflecting maximal congestion and a signal of 0 reflecting minimal congestion. For each connection i, the set of signals b^α_i from the individual gateways are combined into a single congestion signal b_i. Employing the philosophy of bottleneck flow control [Jaf81], in which each source responds only to the signal from the most congested gateway, we set $b_i\equiv Max(b^\alpha_i)$, where the maximum is taken over all $\alpha\in\gamma(i)$. Any gateway α for which $b^\alpha_i=b_i$ is deemed a bottleneck for connection i.

While, for the purposes of our analysis here, it is irrelevant how the signals reach the source, one can think of the signal b_i as being carried in a special field of each packet. The value of b_i is inserted into this field as the packet passes through the gateways and then returned back to the original source through the returning ACK packets. The combining procedure could be implemented by having, at each gateway α, b_i set according to $b_i\rightarrow Max(b_i,b^\alpha_i)$. Then, the congestion number b_i contained in returning ACK packets would be the maximum of all the congestion signals b^α_i over all gateways α along the path of the connection. This scenario is quite similar to the DECbit scheme, with the gateways entering their congestion information into a designated field on packets as they pass though, except that here the gateways set a real number rather than flip a single bit. This is an illustration of how feedback flow control might be implemented in practice; in our model, however, we ignore the feedback delay inherent in such a scheme.

We assume that the gateways do not exchange congestion information with each other, so gateways have no global information. Thus, the congestion signals must only reflect conditions local to the gateway. The only local congestion information available is the set of average queue lengths Q^α_i. The larger these values are, the more congested a gateway is. There are two basic kinds of congestion signals, aggregate and individual. In an aggregate feedback flow control scheme, all connections receive the same congestion signal which is a function of the aggregate congestion C^α: the signal is given by $b^\alpha_i=B(C^\alpha)$ for some function B. There are many ways to measure the aggregate congestion; here we take C^α to be just the total queue size $C^\alpha=Q^\alpha_{tot}\equiv\sum_{i\in I(\alpha)} Q^\alpha_i$. These signals b^α_i reflect the aggregate congestion at the gateway, independent of the amount each connection is contributing to the problem. The signals are totally insensitive to the the service discipline used (since the feasibility constraints require that Q_{tot} be conserved). We assume that the congestion signal function $B(C)$ is nowhere constant, $\frac{\partial B}{\partial C}>0$, and that $B(0)=0$ and $B(\infty)=1$.

In individual feedback flow control, the connections are treated separately. We want the congestion signal sent to the i'th connection, b^α_i, to reflect the i'th connection's contribution to the total congestion at gateway α. If we denote this contribution by C^α_i, then we want the congestion signal to be a function of this: $b^\alpha_i=B(C^\alpha_i)$. In designing the function C^α_i we have two requirements: (1) the individual feedback algorithm yields results that are consistent with the aggregate feedback, in that at steady state the total queue length Q^α_{tot} at each bottleneck gateway is the same under the two algorithms, and (2) the congestion signal sent to the i'th connection does not reflect the congestion due to queues larger than its own. These requirements lead to the individual congestion measure $C^\alpha_i\equiv\sum_{k\in I(\alpha)} Min(Q^\alpha_k,Q^\alpha_i)$. For the connection with the smallest Q^α_i, $C^\alpha_i=N^\alpha Q^\alpha_i$ and the individual congestion signal is the same as the aggregate feedback signal would be in the case where every connection has the same queue size. For the connection with the largest Q^α_i, $C^\alpha_i=Q^\alpha_{tot}$ and the connection must respond to the aggregate congestion. The performance of feedback flow control with individual congestion signals depends crucially on the service discipline employed.

2.3.2. Rate Adjustment: Rate adjustment is a synchronous process occurring at discrete time steps. At each such time step, every connection updates its sending rate from the old value r_i to a new value $\hat{r}_i$ according to the formula $\hat{r}_i=r_i+f$ (to prevent unrealistic transmission rates, if $\hat{r}_i<0$ we truncate it to 0). As with the gateways, we assume that the sources base their updating decisions only on local information. Each source has three pieces of local congestion information; the current sending rate r_i, the congestion signal b_i, and the average roundtrip delay of its packets d_i. The rate adjustment is then a function of these three quantities; $\hat{r}_i = r_i + f(r_i, b_i, d_i)$. The function $f(r_i, b_i, d_i)$ contains the essence of the flow control mechanism, as it determines how sources respond in the presence of congestion. We assume that this function is never insensitive to changes in the congestion signal, $\frac{\partial f}{\partial b} < 0$. The dynamics of this iterative procedure are captured in the notation $\hat{\mathbf{r}} = \mathbf{F}(\mathbf{r})$.

We hope that this iterative procedure, repeatedly applying $\mathbf{F}$ to the initial set of transmission rates $\mathbf{r}$, leads eventually to a steady state allocation of throughput. Steady state is a vector of sending rates $\mathbf{r}_{SS}$ such that $f(r_i, b_i, d_i) = 0$ for all i; equivalently, $\mathbf{r}_{SS} = \mathbf{F}(\mathbf{r}_{SS})$. In our analysis of flow control algorithms, we will primarily be concerned with the properties of the resulting steady state. These steady states need not be unique; flow control algorithms can have a multidimensional manifold of steady state solutions. Furthermore, the iterative procedure does not always lead to a steady state, but can lead to oscillatory or chaotic behavior.

2.4. Goals

There are four performance goals that we want to achieve with a flow control algorithm: time-scale invariance, fairness, stability, and robustness in the presence of heterogeneity. We discuss each of these goals in turn.

2.4.1. Time-Scale Invariance: The flow control algorithm should not have any intrinsic time scales, but rather should respond to the time scales defined by the server rates present in the network. This leads to two different conditions. First, we require that the steady state scale with the server rates. Denote by $\mathbf{r}_{SS}(\boldsymbol{\mu})$ the steady state solution of a network with a set of server rates $\boldsymbol{\mu}$. Time-scale invariance (TSI) requires that, for any positive scaling constant c, $\mathbf{r}_{SS}(c\boldsymbol{\mu}) = c\mathbf{r}_{SS}(\boldsymbol{\mu})$. Second, time-scale invariance requires that the steady state solution be independent of the latencies in the communication lines.

2.4.2. Fairness: The flow control algorithm should result in fair allocations of bandwidth. The particular notion of fairness we use here is borrowed from [Jaf80, Jaf81, Gaf82, Gaf84, Mos84, Ram87], but is modified for our specific application where connections always consume as much bandwidth as flow control allows. A steady state is then deemed fair if, at each bottleneck α for connection i, there are no sending rates greater than r_i: $b_i = b^{\alpha}{}_i \Rightarrow r_i \geq r_j$ *for all* $j \in I(\alpha)$. This condition essentially states that throughput is allocated evenly among those connections for whom the gateway is a bottleneck. A flow control algorithm is considered *guaranteed fair* if all its resulting steady state solutions (for all network configurations) are fair. However, steady state solutions are not always unique, so there is a weaker form of fairness that can be defined; we consider a flow control algorithm *potentially fair* if every network configuration has at least one fair steady state.

2.4.3. Stability: The previous two conditions refer only to the steady state allocation $\mathbf{r}_{SS}$ itself. However, it is important that the iterative procedure $\hat{\mathbf{r}} = \mathbf{F}(\mathbf{r})$ actually converges to steady state. Optimally, all initial configurations $\mathbf{r}$ should converge to the steady state. This condition is extremely hard to verify so, instead, we consider the weaker condition of linear stability. Linear stability requires that small deviations from the steady state dissipate under the iterative procedure $\hat{\mathbf{r}} = \mathbf{F}(\mathbf{r})$ (in the case where we have a manifold of steady state solutions, rather than a unique steady state solution, only those deviations perpendicular to the steady state manifold need dissipate under iterations). If all of the partial derivatives are continuous, one can define the stability matrix, $\mathbf{DF}$, via $DF_{i,j} = \frac{\partial F_i}{\partial r_j}$. A steady state is linearly stable if all eigenvalues of $\mathbf{DF}$ have magnitude less than one.

2.4.4. Robustness in the Presence of Heterogeneity: Most of the treatment here assumes that all sources use the same rate adjustment algorithm f. In practice, however, it is perhaps unreasonable to expect universal implementation of any one particular flow control algorithm. It is then important that the resulting steady state provide a certain guaranteed level of throughput, even when there is a heterogeneous set of rate adjustment algorithms f_i in use. We call a feedback flow control algorithm *robust* if, for all networks and all traffic patterns, each connection receives at least as much throughput as it would if it were the only connection in a network with reduced server rates $\bar{\mu}^{\alpha} = \frac{\mu^{\alpha}}{N^{\alpha}}$. One could, in a reservation-based network (as opposed to a datagram network) reserve for each connection an equal share of the server rate. This ensures adequate service in spite of heterogeneity in flow control algorithms, but at the cost of losing statistical multiplicity at the server. The robustness performance guarantee says that the datagram approach should never give worse service, in terms of throughput allocation, than the reservation-based approach.

2.5. Limitations of Model

Before exploring the possibility of achieving these goals, we first critically assess our model. This model makes several rash and uncontrolled deviations from the real world. There is the traditional, if unjustified, modelling assumption of Poisson sources and exponential servers. The iterative procedure is synchronous, with finite and fixed feedback delays, even though such a process in the real world is asynchronous, with much of the asynchrony arising from communication delays brought about by network congestion and differing roundtrip delays. We have also assumed instant equilibration of queues sizes (the queue lengths $Q^{\alpha}{}_i(\mathbf{r})$ reflect the current sending rates $\mathbf{r}$). By considering only file-transfer-like traffic (infinitely long-lived essentially one-way connections that send at the maximum rate allowed by flow control) we have neglected the effects of dynamic traffic patterns and different type-of-service requirements.

There are two reasons why, in spite of these unrealistic assumptions, we can still derive insight from this model. First, the assumptions tend to make congestion control easier, lending credence to our negative conclusions. Second, most of the conclusions in this paper are qualitative, and thus may be unaffected by particular modelling inaccuracies. However, the lack of asynchrony in our model certainly affects the stability results, and we are currently investigating the extent of this effect.

In addition to the above unrealistic modelling assumptions, the present treatment is limited in that it considers only a restricted class of feedback flow control algorithms. There are three main restrictions. First, we look only at steady state phenomena, so that oscillatory algorithms like those in [Chi89] are not considered. Second, in our model only the congestion at a connection's bottleneck matters. Third, the congestion signals b^{α}_i are time-scale invariant, in that they remain constant under a scaling of both the line speeds and source transmission rates. This restriction enforces the conceptual separation between the congestion signalling and the source response alluded to in the Introduction, and thus excludes algorithms such a those in [Hay81, Gaf82, Gaf84, Jaf80, Jaf81, Mos84]. It is important to note that the results in this paper apply only to the limited class of algorithms under consideration.

3. Performance Results

We now discuss the conditions under which the four goals of time-scale invariance, fairness, stability, and robustness can be met.

3.1. Time-Scale Invariance: Which design choices yield time-scale invariant steady states? It turns out that time-scale invariance is a property only of the rate adjustment algorithm.

Theorem 1:

A feedback flow control algorithm is time-scale invariant iff there is a unique value b_{SS} such that

(1) $f(r,b_{SS},d)=0$ for all r and d.

(2) $f(r,b,d)\neq 0$ for all r and d when $b\neq b_{SS}$.

Proof:

First, to prove sufficiency, assume that we have a rate adjustment algorithm that satisfies conditions (1) and (2). Then, steady state occurs only when $b_i=b_{SS}$ for each connection i. However, for both individual and aggregate feedback, the signals b^{α}_i depend only on the ratios $\frac{r_i}{\mu^{\alpha}}$, so that $\mathbf{r}_{SS}$ necessarily scales with the line speeds.

To prove the necessity of conditions (1) and (2), consider a single connection at a gateway with server speed μ and latency l in the communication line. There is no distinction between aggregate and individual feedback in this case. With $\rho\equiv\frac{r}{\mu}$, the roundtrip delay is given by $d=l+\frac{1}{\mu(1-\rho)}$. The congestion signal b is merely a function of the average queue length $Q(\mu\rho)=\frac{\rho}{1-\rho}$. Time-scale invariance requires that there be a steady state solution ρ_{SS} which satisfies $f(\mu\rho_{SS},B(\frac{\rho_{SS}}{1-\rho_{SS}}),l+\frac{1}{\mu(1-\rho_{SS})})=0$ for all μ and l. This shows that condition (1) must hold. Condition (2) actually follows from condition (1) and our condition that f be strictly monotonic in b.□

Thus, time-scale invariance depends only on the rate adjustment function f, and not on the signalling function B or service discipline $\mathbf{Q}(\mathbf{r})$. In the rest of the paper, we will restrict ourselves to TSI rate adjustment algorithms. For TSI rate adjustment algorithms, the steady state condition is merely that $b_i=b_{SS}$ for all i. For aggregate feedback flow control, this is merely a condition on $r^{\alpha}_{tot}\equiv\sum_{i\in I(\alpha)} r_i$ at each gateway, which can give rise to a multidimensional manifold of steady state solutions. In contrast, the steady state is always unique for individual feedback flow control. Furthermore, the steady state is independent of the service discipline. We will prove these statements in the next section after we develop a result on fairness.

3.2. Fairness: We now consider under what conditions TSI feedback flow control algorithms are fair. Our first result is the following theorem:

Theorem 2:

For aggregate feedback flow control,

(1) No TSI flow control is guaranteed fair.

(2) Every TSI flow control is potentially fair.

Proof:

(1) If we consider a network with a single gateway with N connections, the steady state condition is merely the condition $r_{tot}=A$ for some constant A. There is clearly an $N-1$ dimensional manifold of solutions $\mathbf{r}$ to the equation $\sum_{i=1}^{N} r_i=A$; only one of these solutions, the one where $r_i=\frac{A}{N}$, is fair. Thus, no TSI flow control can guarantee fairness.

(2) We demonstrate that among the various steady state solutions there is exactly one that is fair. The following construction is similar to the one in [Ram87]. Every TSI rate adjustment function is characterized by a steady state congestion signal b_{SS} as in Theorem 1. A signalling function $B(C)$ defines a steady state value ρ_{SS} where $B(\frac{\rho_{SS}}{1-\rho_{SS}})=b_{SS}$. Let β be the gateway with the smallest value of the quantity $\frac{\mu^{\alpha}}{N^{\alpha}}$ and, for each connection i in $I(\beta)$, set $r_i=\frac{\mu^{\beta}}{N^{\beta}}\rho_{SS}$. These sending rates are considered fixed and we remove them from the system. For each of these removed connections i, we decrement N^{α} by one and subtract the quantity $\frac{\mu^{\beta}}{N^{\beta}}$ from μ^{α} for each $\alpha\in\gamma(i)$. Continuing this iterative

procedure results in a fair steady state solution. Note that there can be no other fair allocations, since all fair allocations must satisfy the conditions used in this construction; this fair steady state is unique.□

Thus, TSI aggregate feedback flow control can never guarantee fairness. What happens with individual feedback flow control? Here, we have the opposite result:

Theorem 3:
Every TSI individual feedback flow control is guaranteed fair.

Proof:
Recall that for TSI flow control, at steady state we must have $b_i = b_{SS}$ for all i. Assume that we have a TSI individual feedback flow control algorithm which is not guaranteed fair. Then, there must be some network configuration with a steady state in which there exist connections j and k and a gateway α such that (1) $j,k \in I(\alpha)$, (2) $b^\alpha{}_j = b_j$, and (3) $r_k > r_j$. But,

$$r_k > r_j \Rightarrow Q^\alpha{}_k > Q^\alpha{}_j \Rightarrow C^\alpha{}_k > C^\alpha{}_j \Rightarrow b^\alpha{}_k > b^\alpha{}_j$$

This leads to the contradiction

$$b_{SS} = b_k \geq b^\alpha{}_k > b^\alpha{}_j = b_j = b_{SS}$$

Thus, every TSI individual feedback flow control is guaranteed fair.□

We can now easily establish the aforementioned fact that the steady state is unique and independent of the service discipline for TSI individual feedback flow control.

Corollary:
Every TSI individual feedback flow control has a unique steady state, which is independent of the service discipline.

Proof:
From Theorem 3 we know that all steady states of a TSI individual feedback flow control algorithm are fair. Also, given our form of individual feedback where $C^\alpha{}_i = Q^\alpha{}_{tot}$ for the maximal r_i in $I(\alpha)$, these fair steady states are identical to the unique fair steady state associated with the aggregate version of the flow control algorithm, as constructed in the proof of Theorem 2, which are independent of the service discipline.□

Note that the converse to Theorem 3 is not true; there are guaranteed fair individual feedback flow control algorithms that are not TSI. Similarly, there are guaranteed fair aggregate feedback flow control algorithms that are not TSI. The steady states resulting from guaranteed fair feedback flow control must be independent of latencies, but they need not scale with the server rates. Consider, for example, the rate adjustment algorithm $f = (1-b)\eta - \beta b r$. At steady state, $r_i = \frac{\eta(1-b_i)}{\beta b_i}$. Every connection sharing the same bottleneck has the same sending rate, so the rate adjustment algorithm is guaranteed fair. However, the steady state result does not scale linearly with line speeds, so it is not TSI.

3.3. Stability: Given a particular network and traffic pattern, it is possible to devise a stable feedback flow control algorithm using either aggregate or individual feedback. However, one would like to find an algorithm that is stable for all networks and traffic patterns. Barring that, one would like to be able to guarantee that the system is stable without having to investigate the entire stability matrix **DF**. In a large network, it will be hard to determine **DF** in practice. However, the responses of individual connections, that is the matrix elements $DF_{i,i}$, are easy to determine (connections can make small variations in r_i and easily measure their own response; measuring the off-diagonal terms requires synchronization among connections). Thus, we are interested in the relationship between stability of the system as a whole and stability of individual connections. Recall that a sufficient condition for systemic stability is that all of the eigenvalues of **DF** must have magnitude less than one. This condition is somewhat inadequate because not all of the partial derivatives are continuous at the steady state solutions, which is a result of the *MAX* and *MIN* functions in the definitions of b_i and $C^\alpha{}_i$. In our description that follows, we do not introduce the cumbersome notation needed to express these discontinuities. While the resulting theorems remain true in spite of these discontinuities, some of the following text avoids the complications associated with the discontinuities.

We will call a system unilaterally stable if, at the steady state, when we hold all other r_j fixed and vary only r_i, any small initial deviation from steady state will disappear exponentially under the iterative procedure $\hat{r}_i = r_i + f(r_i, b_i, d_i)$. When the derivatives are continuous, this is equivalent to requiring $|DF_{i,i}| < 1$ for all i. In this section we focus on the relationship between this notion of unilateral stability and that of systemic stability. There are two different stability questions that emerge from this focus.

First, under what circumstances does unilateral stability imply systemic stability? For aggregate feedback flow control unilaterally stable systems need not be stable. This can be seen in the following example. Consider a single gateway with $\mu = 1$ shared by N connections. Let the feedback flow control have $B(C) = \frac{C}{C+1}$ and $f(r,b,d) = \eta(\beta - b)$ with β, η positive constants. The fair steady state is $r_i = \frac{\beta}{N}$. For aggregate feedback flow control, $DF_{i,j} = \delta(i,j) - \eta$. When $\eta < 2$ this is unilaterally stable. However, the leading eigenvalue of **DF** is $1 - \eta N$ which, for large enough N, is unstable.

As an aside, we note that when the steady state is unstable the iterative process $\hat{\mathbf{r}} = \mathbf{F}(\mathbf{r})$ can lead to oscillatory and chaotic behavior. If we change the signalling function to be $B(C) = \frac{C^2}{(1+C)^2}$ and start in a symmetric initial condition, then the updating process reduces to $\hat{r}_{tot} = r_{tot} + \eta N(\beta - r_{tot}^2)$. For small enough η, as one increases N this recursion relation proceeds from stable behavior, to oscillatory behavior, to chaotic behavior (see [Col80] for a review of such iterative dynamics).

One can give similar examples showing that for individual feedback flow control with FIFO service, unilaterally stable systems need not be stable. However, for this case, the discontinuity in the partial derivatives render the matrix **DF** inappropriate and necessitates the direct investigation of the dynamics of the original recursion relation. We will not do that here.

In contrast to the examples above, unilaterally stable systems are always systemically stable when employing individual feedback flow control with Fair Share service.

Theorem 4:

For TSI individual feedback flow control with Fair Share service, unilaterally stable systems are always systemically stable.

Proof:

First, assuming continuity of the matrix $DF_{i,j}$, we argue that the eigenvalues of the matrix are just the diagonal elements $DF_{i,i}$. The matrix $DF_{i,j}$, at steady state is given in general by $DF_{i,j} = \delta(i,j) + \frac{\partial f}{\partial b_i}\frac{\partial B}{\partial C_i}\frac{\partial C_i}{\partial r_j}$.

The simultaneous triangularity of $\frac{\partial C_i}{\partial Q_j}$ and of $\frac{\partial Q_i}{\partial r_j}$ implies the triangularity of $DF_{i,j}$, which then makes the eigenvalues just the diagonal elements $DF_{i,i}$. Since unilateral stability requires that $|DF_{i,i}|<1$ for all i, this implies systemic stability. The above argument assumes the continuity of derivatives. While the partial derivatives can be discontinuous, the triangularity of the matrix always holds with Fair Share service. It is straightforward to show that in the presence of discontinuities, this triangularity and the condition of unilateral stability are sufficient to guarantee systemic stability. Thus, the conclusion of the theorem holds despite the discontinuities.□

Thus, individual feedback flow control with Fair Share service is the only design combination we have discussed which allows us to guarantee systemic stability merely through checking unilateral stability. This guarantee of systemic stability might be achieved through a more general condition on the unilateral stability properties. We will call a feedback flow control algorithm guaranteed unilaterally stable if at steady state, for all network configurations, it is always unilaterally stable. An example of such a system is either aggregate or individual feedback flow control with $B(C) = \frac{C}{C+1}$ and $f(r,b,d) = \eta r(\beta - b)$ with $\eta < 2$. Our second stability question is: does guaranteed unilateral stability imply systemic stability? The answer to this question is clearly *yes* when we use individual feedback with the Fair Share service discipline. For other design choices we have no firm answers to this question, but we do have the following conjecture:

Conjecture:

A guaranteed unilaterally stable TSI feedback flow control algorithm, with either aggregate or individual feedback, will always be systemically stable.

If this conjecture is true then, by choosing guaranteed unilaterally stable TSI feedback flow control algorithms, we can guarantee systemic stability with either aggregate or individual feedback.

3.4. Robustness in the Presence of Heterogeneity: Since aggregate feedback flow control is not guaranteed fair even for networks with a single gateway, it clearly cannot be robust. However, its performance in the presence of heterogeneity of rate adjustment algorithms f_i is appallingly bad. Consider a single gateway with two connections having TSI aggregate feedback flow control with steady state congestion signals b_{SS}^1 and b_{SS}^2 respectively, with $b_{SS}^1 > b_{SS}^2$. Under the iterative procedure $\hat{\mathbf{r}} = \mathbf{F}(\mathbf{r})$, this system drives $r_2 \to 0$ and $r_1 \to r_{SS}$ where $B(\frac{r_{SS}}{\mu - r_{SS}}) = b_{SS}^1$. This is a steady state despite the fact that $f_2(r_2, b_2, d_2) < 0$ since we truncate the $\hat{r}_i$'s to be nonnegative. Thus, any connections sharing a bottleneck gateway with a connection which has a larger b_{SS} will eventually be completely shut down.

Whether or not TSI individual feedback flow control is robust depends on the service discipline.

Theorem 5:

TSI individual feedback flow control is robust if and only if the service discipline satisfies $Q_i(\mathbf{r}) \le \frac{r_i}{\mu - N r_i}$ for all $\mathbf{r}$, where N is the number of connections at the gateway.

Proof:

Consider a TSI individual feedback flow control algorithm and let C_{SS} be the value such that $B(C_{SS}) = b_{SS}$, and define $\rho_{SS} = \frac{C_{SS}}{1 + C_{SS}}$. If connection i were the only connection at a gateway with service rate $\bar{\mu}$, its steady state sending rate $\bar{r}_i$ is given by $\bar{r}_i = \bar{\mu}\rho_{SS}$. To see the necessity of the condition in Theorem 5, consider a single gateway with N connections, and let r_i be the minimal steady state sending rate, so that $C_i = NQ_i$. Then, from robustness, we must have $\bar{r}_i \le r_i$ so, holding all the other r_j fixed and dropping them from the notation, $Q_i(\bar{r}_i) \le Q_i(r_i) = \frac{C_{SS}}{N} = \frac{\bar{r}_i}{\mu - N\bar{r}_i}$. Since this must hold for all b_{SS}, the necessity is established.

To see sufficiency, assume that the above condition holds. Assume that the robustness criterion is violated, so that, for some connection i, $r_i < \bar{r}_i$ where, with b_{SS} and ρ_{SS} as defined above for connection i, $\bar{r}_i \equiv \frac{\mu^\beta}{N^\beta}\rho_{SS}$ where β minimizes $\frac{\mu^\alpha}{N^\alpha}$ over all $\alpha \in \gamma(i)$. Again using a notation where only the functional dependence on r_i is shown, at every bottleneck gateway α,

$$C_{SS} = C^\alpha{}_i(r_i) < C^\alpha{}_i(\bar{r}_i) \le N^\alpha Q^\alpha{}_i(\bar{r}_i) \le \frac{N^\alpha \bar{r}_i}{\mu^\alpha - N^\alpha \bar{r}_i} \le C_{SS}$$

which is a contradiction.□

The FIFO service discipline does not satisfy this condition. Thus, TSI individual feedback flow control with FIFO service at the gateways is not robust. However, it is not as bad as TSI aggregate feedback flow control in this regard, in that in general all connections get a nonzero amount of throughput. The Fair Share service discipline does satisfy the condition in Theorem 5. This service discipline is able to provide robust service in the presence of heterogeneous rate adjustment algorithms.

A robust flow control algorithm never gives worse service, in terms of throughput allocation, than the reservation-based approach. In addition, the queueing delays arising from robust TSI individual feedback flow control algorithms are lower, compared to those of a reservation-based approach, by at least a factor of N^{α} at each gateway.

4. Relevance to Real Flow Control Algorithms

It may not be appropriate to analyze currently implemented feedback flow control algorithms, with their attention to the real-life issues of averaging, transients, dropped packets, and dynamic traffic patterns, within the oversimplified framework presented here. However, if we examine only the underlying design principles of these real algorithms, there may be some insight to be gained.

The original DECbit algorithm [Jai88, Ram88, Chi89] uses a linear-increase multiplicative-decrease window adjustment algorithm (Jacobson [Jac88] uses something similar). This can be modelled as $f=(1-b)\frac{\eta}{d}-\beta br$, with b heuristically interpreted as the probability that the DECbit is set. This is neither TSI nor guaranteed or potentially fair. The lack of fairness is due to the latency sensitivity of d; connections with longer roundtrip times get less throughput. This is easily corrected by interpreting the algorithm as a rate, not window, adjustment algorithm where the form becomes $f=(1-b)\eta-\beta br$ which is guaranteed fair but still not TSI. The analysis leading to the adoption of linear-increase multiplicative-decrease as the optimal choice for rate adjustment [Chi89] assumed only a single congested gateway and a binary aggregate congestion signal so that the asymptotic behavior is not a steady state but rather a periodic oscillation. In this setting, the linear-increase multiplicative-decrease algorithm is yields long-term averages that are both TSI and guaranteed fair. However, the period of oscillation grows linearly with the server rate.

It is hard to analyze the stability of these algorithms, given the complicated nature of their averaging processes. However, Zhang [Zha89] and Hashem [Has89] have observed pronounced oscillatory behavior in Jacobson's algorithm, indicating that there may be some stability problems. These authors have also observed the lack of robustness of aggregate feedback flow control by comparing the performance with two different kinds of flow control present (see also [Dem89]).

In [Dem89], following the insights of Nagle [Nag87], we analyzed the Fair Queueing service discipline, which can be considered a realistic version of Fair Share (see also [Zha89]). Simulation results show that, in the context of real flow control algorithms (DECbit, Jacobson's algorithm, and generic sliding window flow control), Fair Queueing provides better fairness and robustness than FIFO service. We did not investigate issues of stability or time-scale invariance, nor did we explore other flow control algorithms.

5. Discussion

We began with four goals: time-scale invariance, fairness, stability, and robustness. Time-scale invariance is perhaps the most important goal. Networks will soon have line speeds ranging from 1.2 kbits/sec phone lines to gigabit/sec optical fibers. Similarly, there will be latencies ranging from half of a second (satellite links) to microseconds. With both line speeds and latencies spanning six orders of magnitude, any algorithm that has an intrinsic time scale is likely to seriously malfunction.

Since time-scale invariance merely puts a condition on the rate adjustment algorithm, both individual and aggregate feedback algorithms are able to provide time-scale invariant performance. In the rest of our analysis, we considered only TSI rate adjustment algorithms, and investigated the extent to which individual and aggregate feedback algorithms could achieve the other three goals.

TSI aggregate feedback flow control algorithms are inherently potentially fair, in that there is always one fair steady state, but are never guaranteed fair, in that there are always some networks which necessarily have other unfair steady state solutions. Aggregate feedback is not robust in the presence of heterogeneity; in fact, the less greedy connections receive no throughput at all. We conjecture that TSI aggregate feedback that is guaranteed unilaterally stable will always result in systemic stability. However, unilateral stability does not always result in systemic stability.

In contrast, TSI individual feedback flow control is guaranteed fair, with a unique steady state. These conclusions apply no matter which service discipline is used in the gateways. When individual feedback is combined with FIFO service at the gateways, the performance is not robust against heterogeneity in rate adjustment algorithms, but the inequities are not as severe as for aggregate feedback. The stability provided by this combination is similar to that provided by aggregate feedback.

When we combine TSI individual feedback flow control with Fair Share service, we get robust performance. Furthermore, unilateral stability does guarantee systemic stability. At least in this simplified model, it is clear that TSI individual feedback flow control with Fair Share service provides the best performance among all of the options we have considered.

The question of implementation remains largely unaddressed. It is clear that the implementation issues become more vexing as we proceed from (1) aggregate feedback flow control to (2) individual feedback flow control with FIFO service to (3) individual feedback flow control with

Fair Share service. The goal of this paper is to examine, at least in this limited theoretical context, to what extent the quality of the flow control improves as one makes this progression.

6. Acknowledgements

I would like to thank the following for helpful and stimulating discussions: Dave Clark, Lixia Zhang, Chuck Davin, Andrew Heybey, Van Jacobson, K. K. Ramakrishnan, Srinivasan Keshav, Abel Weinrib, and the members of the End-to-End Services Research Group.

7. References

[Chi89] D.-M. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks", Computer Networks and ISDN Systems, Volume 17, pp 1-14, 1989.

[Col80] P. Collet and J.-P. Eckmann, "Iterated Maps on the Interval as Dynamical Systems", Birkhauser, Boston 1980.

[Dem89] A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of a Fair Queueing Algorithm", Proceedings of ACM Sigcomm, Computer Communications Review, Volume 19, Number 4, pp 1-12, 1989.

[Fin89] G. Finn, "A Connectionless Congestion Control Algorithm", Computer Communications Review, Volume 19, Number 5, pp 12-31, 1989.

[Gaf82] E. Gafni, "The Integration of Routing and Flow Control for Voice and Data in a Computer Communication Network", MIT Report, LIDS-TH-1239, 1982.

[Gaf84] E. Gafni and D. Bertsekas, "Dynamic Control of Session Input Rates in Communication Networks", IEEE Transactions on Automatic Control, Volume 29, No. 9, pp 1009-1016, 1984.

[Has89] E. Hashem, "Analysis of Random Drop for Gateway Congestion Control", MIT-LCS thesis, 1989.

[Hay81] H. Hayden, "Voice Flow Control in Integrated Voice and Data Communication Network", MIT Report, LIDS-TH-1115, 1981.

[Jac88] V. Jacobson, "Congestion Avoidance and Control", Proceedings of ACM Sigcomm, Computer Communications Review, Volume 18, Number 4, pp 314-329, 1988.

[Jaf80] J. Jaffe, "A Decentralized, 'Optimal', Multiple User, Flow Control Algorithm". Proceedings of the Fifth International Conference on Computer Communications, pp. 839-844, October 1980.

[Jaf81] J. Jaffe, "Bottleneck Flow Control". IEEE Transactions on Communications, Vol 29, No. 7, pp. 954-962, July 1981.

[Jai87] R. Jain, K. K. Ramakrishnan, and D.-M. Chiu, "Congestion Avoidance in Computer Networks with a Connectionless Network Layer", DEC Technical Report TR-506, Digital Equipment Corporation, 1987. (also reprinted in "Innovations in Networking", edited by Craig Partridge, Artech House, Boston, 1988)

[Jai88] R. Jain and K. K. Ramakrishnan, "Congestion Avoidance in Computer Networks with a Connectionless Network Layer: Concepts, Goals, and Methodology", Proceedings of Computer Networking Symposium, pp 134-143, 1988.

[Mor89] S. Morgan, "Queueing Disciplines and Passive Congestion Control in Byte-Stream Networks", IEEE INFOCOM '89 Proceedings, pp. 711-720, 1989.

[Mos84] J. Mosely, "Asynchronous Distributed Flow Control Algorithms", MIT LIDS thesis, 1984.

[Nag87] J. Nagle, "On Packet Switches with Infinite Storage", IEEE Transactions on Communications, Volume 35, pp 435-438, 1987.

[Pos81] J. Postel, "Internet Control Message Protocol (ICMP)", RFC-792, Network Information Center, SRI International, 1981.

[Pos87] J. Postel, "Something a Host Could Do with Source Quench: The Source Quench Introduced Delay (SQuID)", RFC-1016, Network Information Center, SRI International, 1987.

[Ram87] K. K. Ramakrishnan, D.-M. Chiu, and R. Jain "Congestion Avoidance in Computer Networks with a Connectionless Network Layer, Part IV-A Selective Binary Feedback Scheme for General Topologies", DEC Technical Report TR-510, Digital Equipment Corporation, November 1987.

[Ram88] K. K. Ramakrishnan and R. Jain, "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks with a Connectionless Network Layer", Proceedings of ACM Sigcomm '88, Computer Communications Review, Volume 18, Number 4, pp 303-313, 1988.

[Reg86] J. Regnier, "Priority Assignment in Integrated Services Networks", Report LIDS-TH-1565, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, Massachusetts, December, 1986.

[She89] S. Shenker, "Making Greed Work in Networks: A Game-Theoretic Analysis of Gateway Service Disciplines", preprint, 1989.

[Zha89] L. Zhang, "A New Architecture for Packet Switching Network Protocols", MIT-LCS thesis, 1989.