

Control of Congestion in High-Speed Networks*

Orhan Ç. Imer and Tamer Başar†

Department of Electrical and Computer Engineering and Coordinated Science Laboratory, University of Illinois, 1308 West Main Street, Urbana, IL 61801-2307, USA

The problem of controlling congestion in high-speed communication networks is introduced. An easy-to-implement explicit rate congestion control algorithm is presented, and its stability properties are discussed. The algorithm is decentralized and is robust to network delays. Furthermore, it does not require per-flow information. It is shown that the network level implementation of this algorithm leads to a “hybrid” control system, whose analysis for stability presents challenges in a control context. A variant of the same algorithm is used in the paper to demonstrate the possibility of an Internet implementation using “marking” with the proper choice of a rate update function.

Keywords: Congestion control; High-speed communication networks; Hybrid systems; Lyapunov analysis; REM (Random Exponential Marking); Saturation non-linearities

1. Introduction

In the context of communication networks, the term “congestion control” is generally used to refer to the action of regulating various flows within a network. Broadly speaking, a need for such a control action arises due to scarcity of network resources, such as link capacities. There are two main objectives

of congestion control. First, the desire to achieve an efficient use of the network resources while keeping the loss rate, and average delay of all “connections” at some reasonable level. Second, it is desirable to maintain some notion of “fairness” between sources sharing a communication link. Several notions of fairness are formally defined in the next section. In this paper, we use the term connection loosely to mean any communication activity between a source–destination pair. Thus a connection could be a virtual circuit in the form of a data pipe, or a stream of data originating at one node and destined for another node.

In contrast to the traditional, low-bandwidth, voice-based telephone network, high-speed networks, such as B-ISDN (Broadband Integrated Service Digital Network), allow several types of network traffic (such as, voice and video) to coexist in the same transmission medium. The traffic in such networks can be broadly classified as *guaranteed-service* traffic and *best-effort service* traffic. Guaranteed service refers to a contract between the network service provider and the end user which requires the network to provide fixed QoS (Quality of Service) to the traffic. The QoS guarantees can be in the form of upper bounds on packet loss probability, delay, etc. In contrast, best-effort traffic is guaranteed very little *a priori*.

In the context of ATM (Asynchronous Transfer Mode) networks, the best effort traffic (in particular the ABR (Available Bit Rate) service) may be guaranteed a minimum rate, and a bound on the loss rate (ratio of the number of packets lost to the total number of packets transmitted). Instead of guaranteeing fixed QoS parameters, the idea is to fairly allocate network resources to competing users. In the Internet today,

*Paper to form the basis of a lecture to be delivered by T. Başar at the 2001 European Control Conference, Porto, Portugal, September 4–7, 2001.

†Research supported by NSF Grants ANI 98-13710 and CCR 00-85917 ITR and AFOSR MURI Grant AF DC 5-36128.

Tel: (217) 333-3607; Fax: (217) 244-1653;

Email: tbasar@decision.csl.uiuc.edu.

Correspondence and offprint requests to: T. Başar, Department of Electrical and Computer Engineering and Coordinated Science Laboratory, University of Illinois, 1308 West Main Street, Urbana, IL 61801-2307, USA. Email: tbasar@uiuc.edu.

Received 18 May 2001; Accepted 23 May 2001.

Recommended by Martins de Carvalho, J.M.G. Sá da Costa and António Dourado.

most users can be thought of generating best-effort type traffic, too. For instance, no traffic related service guarantees are made in advance when browsing a web page, or sending an email (unless the connection uses a leased line). Thus, in general, best-effort sources can adjust their rates to the level of available service, making it possible to control the congestion in the network.

For a best effort source to adapt its rate to changing network conditions, there must be a mechanism through which information about the state of the network is conveyed to the source. This information can be in the form of bandwidth availability, state of congestion, or impending congestion. In ATM networks, this is achieved via explicit rate control messages which are sent from intermediate switches to the sources using special packets called RM (Resource Management) cells. In the Internet, however, no explicit feedback from the routers is available. TCP/IP (the current Internet protocol) uses a window-based congestion control mechanism, where the sources dynamically adjust their window sizes using packet losses and timeouts as congestion indicators [11]. TCP is an end-to-end flow control protocol, in which the intermediate nodes (routers) in the network do not provide any congestion control information. Although TCP has several nice features, such as being self-clocking, easy-to-implement, etc., it does not perform well under various scenarios. For example, the performance of TCP over wireless links is quite poor, because the protocol cannot distinguish between losses due to congestion and losses due to the inherent nature of the wireless channel. For TCP a loss is a loss regardless of where and why it happens. Thus, with TCP it might happen that a source reduces its transmission rate thinking that a congestion loss occurred, while in reality the loss is due to the wireless link. This results in a low throughput, degrading the system performance. Recently, there has been a surge of interest in fixing this problem with so-called AQM (Active Queue Management) or “marking” schemes. The idea is to use routers to provide implicit feedback about the impending congestion by marking packets based on their queue length information. Several marking-based congestion control algorithms have been proposed since the introduction of this idea in [8]. Finding a good AQM scheme is still an active area of research.

There are several challenges to be met in designing congestion control algorithms both in explicit rate feedback networks and in the Internet. Any feedback from the intermediate routers in the network is subject to delay due to propagation, queueing, processing, etc. One challenge is to deal with these feedback delays, which may be unpredictable for a given connection. Another challenge is posed by the traffic characteristics of the

network. The term connection lifetime refers to the duration of a connection. Some connections might have a shorter connection lifetime than the smallest round-trip delay in the network. Controlling the rate of such connections is hard, if not impossible, because of their short duration as compared to the time constant of the feedback loop. Yet another challenge is to keep the final design as simple as possible, so that it can be implemented at a router with a few lines of code.

In this paper, we introduce a control-based mathematical model that helps us address these design issues. The network model assumes a set of communication links with finite capacities shared by a set of connections. Each source controls its own transmission rate using feedback messages from the links on its path to the destination. Each link has a finite-size buffer whose length is controlled by a switch. We assume that the switch can only measure the aggregate rate of arrivals, thus no per-flow information is required. In this setting, we propose an explicit rate congestion control algorithm, and discuss its stability, relegating details to [9]. The algorithm asymptotically achieves fairness among the sources as well as efficient use of the link capacity in the case of a single bottleneck link. Next, we extend the single link analysis to a network with multiple links, and show that the stability of the system can be cast in the framework of stability of hybrid systems. We provide a local stability result for this general case with multiple links, and a global stability result for the special case of two links. Finally, we establish a link between our explicit rate congestion control algorithm and a marking-based scheme, and discuss a possible Internet implementation.

Several other types of explicit rate congestion control designs have been considered before, particularly in the context of ATM networks. The simplest feedback control mechanism is called *rate matching*. In rate matching, the node measures the average rate available to the sources at periodic intervals and simply divides a fraction of this capacity equally among the various connections. This is the basic approach used in [12], although several modifications are used in the actual implementation. The main advantage of this scheme is its simplicity, but it is difficult to control queue length optimally to avoid buffer overflows. However, this scheme is stable (i.e., the queue length remains bounded in an appropriate stochastic sense [1]). The congestion control problem can also be viewed as a feedback control problem where queue length is used for explicit feedback. This approach is used in [5,6] to study the problem using classical control techniques or using a state-space approach. As in rate matching, the primary goal is not optimality, but simply queue length stability. In these approaches, the available

bandwidth is treated as an unmodeled disturbance. Thus, the algorithms in [5,6] ensure stability in the presence of this disturbance, but they require per-flow information. Alternatively, the explicit-rate congestion control problem can be formulated as a stochastic team decision problem with the sources viewed as members of a team [2]. In this formulation, the main objective is queue length and source rate tracking optimality, which is built into the model via a cost functional. Nevertheless, the final design needs to be modified in an ad hoc way to dynamically achieve fairness between the connections. An extensive simulation study of a nonlinear variant of the algorithm we present here, and the algorithms of [2] can be found in [10].

The rest of this paper is organized as follows. In the next section, we describe the general network model. A link level explicit-rate congestion control algorithm is presented in Section 3, along with a discussion of stability and an illustrative numerical example. A network level implementation of this algorithm is carried out in Section 4, along with results on its stability. Section 5 extends the analysis to marking based schemes, and the paper ends with the concluding remarks of Section 6, which identifies some challenges for future research.

2. The Network Model

Consider a graph of communication links shared by a number of connections (source–destination pairs). Let $\mathcal{L} = \{1, 2, \dots, L\}$ denote the set of links in the network, and let $\mathcal{S} = \{1, 2, \dots, S\}$ denote the set of connections using these links. For simplicity, we associate each connection with a flow rate, denoted by r_s . We use $\mathcal{L}_s \subset \mathcal{L}$ to designate the set of links used by connection s , and $\mathcal{S}_l \subset \mathcal{S}$ to designate the set of connections using link l .

Each link has a capacity denoted by C_l , and associated with each link there is a finite size buffer. We assume each connection is guaranteed a minimum rate, MR_s which could be zero. Let $x_s := r_s - \text{MR}_s$, which corresponds to the rate in excess of MR_s . Let F_l denote the aggregate flow on link l . Then,

$$F_l = \sum_{s \in \mathcal{S}_l} r_s = \sum_{s \in \mathcal{S}_l} x_s + G_l,$$

where $G_l := \sum_{s \in \mathcal{S}_l} \text{MR}_s$. We have the following constraints on the vector $x := \{x_s | s \in \mathcal{S}\}$ of flow rates:

$$\begin{aligned} x_s &\geq 0, \quad \forall s \in \mathcal{S}, \\ F_l &\leq C_l, \quad \forall l \in \mathcal{L}. \end{aligned}$$

Note that for a given vector $C := \{C_l | l \in \mathcal{L}\}$ of capacities, no vector x of rates might exist if $G_l > C_l$ for

any l . Therefore, we further assume that the vector C of capacities satisfies

$$C_l \geq G_l, \quad \forall l \in \mathcal{L}.$$

A pair of vectors (x, C) satisfying these constraints is said to be feasible.

One of the goals of congestion control is to treat all connections fairly. Several definitions of fairness have been considered in the literature [7,13]. The most widely accepted notion of fairness is *max–min fairness*. A vector of rates x is said to be max–min fair if it is feasible and for each $s \in \mathcal{S}$, x_s cannot be increased while maintaining feasibility without decreasing $x_{s'}$ for some connection s' for which $x_{s'} \leq x_s$. As pointed out in [13], the max–min fairness criterion gives an absolute priority to smaller flows, in the sense that if $x_{s'} < x_s$ then no increase in x_s , no matter how large, can compensate for any decrease in $x_{s'}$, no matter how small. *Proportional fairness* is proposed to remedy this problem by favoring the smaller flows less emphatically [13]. A vector of rates x is said to be proportionally fair if it is feasible and if for any other feasible vector x' , the aggregate of proportional changes is zero or negative:

$$\sum_{s \in \mathcal{S}} \frac{x'_s - x_s}{x_s} \leq 0.$$

Given a feasible rate vector x , we say that a link is a *bottleneck link* with respect to x for a connection s crossing l if $C_l = F_l$ and $x_s \geq x_{s'}$ for all connections s' crossing link l . It can be shown that a feasible rate vector x is max–min fair if and only if each connection has a bottleneck link with respect to x [7]. Hence, in a max–min fair allocation each connection necessarily has a bottleneck link. Now, consider a bottleneck link l in the network. By the definition of a bottleneck link we must have $F_l = C_l$. The set \mathcal{S}_l of connections crossing l , can be partitioned into two disjoint subsets, \mathcal{B}_l and \mathcal{B}_l^c , where c is the complement operation. Here \mathcal{B}_l denotes the set of connections that have l as a bottleneck link. By definition, for any two connections $p, q \in \mathcal{B}_l$, we have $x_p \geq x_s$ and $x_q \geq x_s$ for all $s \in \mathcal{S}_l$. In particular, $x_p \geq x_q$ and $x_q \geq x_p$, which implies that $x_p = x_q$. In other words, to have a max–min fair allocation, all connections bottlenecked on a particular link must have the same flow rate. This observation yields itself to an alternative definition of max–min fairness provided that the number of bottlenecked connections on a given link is known. Under this assumption, the max–min fair share of a connection bottlenecked on link l should be equal to

$$r_s = \text{MR}_s + \frac{C_l - (u_l + G_l)}{M_{l,l}(\infty)}, \quad \forall s \in \mathcal{B}_l, \quad (1)$$

where u_l is the sum of the rates of connections which are not bottlenecked on link l , and $M_{l,l}(\infty)$ is the number of connections bottlenecked on link l . (1) can be obtained by using the definition of a bottleneck link and our previous observation. We have

$$\begin{aligned} F_l = C_l &\Rightarrow \sum_{s \in \mathcal{S}_l} x_s + G_l = C_l \\ &\Rightarrow \sum_{s \in \mathcal{B}_l} x_s + \sum_{s \in \mathcal{B}_l^c} x_s = C_l - G_l. \end{aligned}$$

Since each connection $s \in \mathcal{B}_l$ has the same rate x_s , the last equality implies

$$\begin{aligned} M_{l,l}(\infty)x_s + \sum_{s \in \mathcal{B}_l^c} x_s &= C_l - G_l \\ \Rightarrow x_s &= \frac{C_l - \left(\sum_{s \in \mathcal{B}_l^c} x_s + G_l\right)}{M_{l,l}(\infty)}. \end{aligned}$$

Defining $u_l := \sum_{s \in \mathcal{B}_l^c} x_s$, and using the fact that $r_s = x_s + MR_s$, we arrive at (1).

Providing fairness between connections is not the only goal of congestion control. Equally important is to maintain a high utilization of the resources (link capacities), as well as a low loss rate for all connections. These objectives can be achieved by regulating the queue length at routers (switches) around a target level. Tracking such a nominal queue length (whose exact value is determined based on QoS requirements) is desirable in order to avoid losses due to overflow and waste of the link capacity due to underflow. Let q_l denote the size of the queue at the router controlling link l . Because of the physical constraints, q_l must satisfy

$$0 \leq q_l \leq Q_l, \quad (2)$$

where Q_l denotes the size of the link buffer. Also let q_l^* denote the desired value of q_l .

Now, given this network setting, the goal of a *congestion control algorithm* is to achieve some criterion of fairness and queue length regulation in a decentralized way. In other words, starting with arbitrary vectors $r = \{r_s | s \in \mathcal{S}\}$ of rates, and $q = \{q_l | l \in \mathcal{L}\}$ of queue sizes, we want the pair (r, q) to converge to (r^*, q^*) , where r^* is a vector of max-min (or proportional) fair rates, and q^* is the vector of desired queue lengths. Of course, this is a trivial task if carried out by a central entity which has access to all the information about the network, and at the same time has control over all connections. The challenge is to obtain a decentralized algorithm, where connection s updates its own rate r_s based on a limited amount of feedback from the network.

To describe the dynamics of the system, we adopt here a discrete-time model, where the discrete-time

unit corresponds to the update interval. For simplicity, we assume that the updates across the links are synchronized. Then, assuming a fluid model of network flows, the queue length at link l , $q_l(n)$, evolves according to

$$q_l(n+1) = \max\{0, \min\{Q_l, q_l(n) + F_l(n) - C_l\}\}, \quad (3)$$

where $F_l(n)$ is the aggregate flow on link l at the beginning of the update interval $[n, n+1)$. The rate $r_s(n)$ of connection s is an input to a number of queue lengths $q_l(n)$ for $l \in \mathcal{L}_s$. In determining the rate r_s of connection s , some feedback needs to be provided back to the source which controls the rate of the connection s . This feedback takes several forms depending on the particular network protocol and architecture.

One approach is to let routers (switches) provide explicit rate feedback messages to the connections crossing their links. An example network service that does this is the ABR category of the ATM networks [4]. Alternatively, routers can employ “marking” to signal congestion to the connections crossing their links. Marking can be done either by dropping packets, or flipping a congestion indication bit in the packet header with a certain probability. Early Congestion Notification (ECN) schemes, such as RED (Random Early Detection) [8], or REM (Random Exponential Marking) [3] are example marking schemes.

In explicit-rate congestion control, each source s periodically sends out a special packet (termed as a resource management cell in the ATM context) with an ER (explicit-rate) field, which travels along the same route, \mathcal{L}_s , as the data packets but is treated specially by the routers (switches) along the way. This packet is eventually turned around by the destination and sent back to the source. The source initially sets the ER field to the rate it would like to transmit. As this special packet passes through various switches on the way to the destination and back to the source, those that are congested may reduce ER. When the source receives the ER field back, it adjusts its rate accordingly. As the links along the way may only reduce the ER field, the explicit rate received by the source will be the minimum of the rates dictated by the links along its path. In other words, for the rate r_s of connection s we have

$$r_s(n) = MR_s + \min_{l \in \mathcal{L}_s} ER_l(n - d_{s,l}), \quad (4)$$

where $d_{s,l}$ represents the action delay, which is the time from the moment feedback information is sent by link l to source s , until an action is taken by s . As the link speeds continue to rise, the delay-bandwidth product (i.e., the product of the round-trip propagation delay and the link capacity) increases. An issue of importance that arises in this context is how to deal with

these action delays, as they may not be known accurately. With (3) and (4), the design of an explicit-rate congestion control algorithm boils down to finding a decentralized update scheme for $ER_l(n)$ with desired convergence properties. In addition to being decentralized, it is desirable to have an algorithm which does not require per-flow information. That is, the ER controller on link l has access to only the aggregate flow $F_l(n)$, and the queue length $q_l(n)$. In the next section, we describe an explicit-rate congestion control algorithm, which has all these desired properties, and is also easy-to-implement.

The main drawback of explicit-rate schemes is the need for special switches which are capable of distinguishing between explicit rate and data packets. Also, each link is associated with an ER controller, which might be difficult to build into the switch architecture. Marking-based schemes avoid this problem altogether by providing implicit feedback about the state of the network. In marking-based congestion control, the objective is to signal the congestion before it actually happens. All marking schemes use some variant (low-pass filtered, scaled, etc.) of the queue length as the congestion indicator. The idea is to adjust the marking rate in proportion to the queue length (not necessarily to its instantaneous value). Now, suppose on link l we mark packets with probability $1 - e^{-\lambda_l(n)}$. It is conceivable that source s measures the fraction $f_s(n)$ of unmarked packets in time slot n , which is given by

$$f_s(n) = e^{-\sum_{l \in \mathcal{L}_s} \lambda_l(n - d_{s,l})}.$$

Thus, source s can estimate $\sum_{l \in \mathcal{L}_s} \lambda_l(n - d_{s,l})$ as

$$\sum_{l \in \mathcal{L}_s} \lambda_l(n - d_{s,l}) = -\ln f_s(n).$$

The rate r_s of source s can then be updated as

$$r_s(n) = MR_s + g\left(\sum_{l \in \mathcal{L}_s} \lambda_l(n - d_{s,l})\right)$$

where $g(\cdot): \mathcal{R} \rightarrow \mathcal{R}$ is a monotonically decreasing function. In Section 5, we consider one possible choice for the function $g(\cdot)$, which leads to the notion of proportional fairness between the connections. We close our account on this section with the following two definitions which we shall need in the sequel.

Definition 1. A communication network $\mathcal{N}(\mathcal{L}, \mathcal{S})$ with a set of links \mathcal{L} , and a set of connections \mathcal{S} is said to have a *max–min (proportionally) fair equilibrium point*, if there exists a feasible triplet of vectors (x^*, C^*, q^*) such that the pair (x^*, C^*) is max–min (proportionally) fair, and q^* is the vector of desired queue lengths.

Definition 2. A congestion control algorithm for the network $\mathcal{N}(\mathcal{L}, \mathcal{S})$ is said to be *globally convergent* if for any initial set of vectors $(r(0), C^*, q(0))$, the triplet $(r(n), C^*, q(n))$ of vectors converge to the max–min (proportionally) fair equilibrium point of the network.

3. A Link Level Explicit-Rate Congestion Control Algorithm

In this section, we present an easy-to-implement globally convergent explicit-rate congestion control algorithm under some simplifying assumptions. Before attempting to find an algorithm which works for the entire network, we first focus on a single bottleneck link l in the network. Recall that the queue length $q_l(n)$ evolves according to (3) which we repeat here for convenience:

$$q_l(n+1) = \max\{0, \min\{Q_l, q_l(n) + F_l(n) - C_l\}\}.$$

Let M_l denote the total number of connections crossing l . Then, $M_l = \text{card}(\mathcal{S}_l)$, where $\text{card}(\mathcal{A})$ represents the cardinality (number of elements) of set \mathcal{A} . We assume that M_l does not change with time. As indicated in the preceding section, the set \mathcal{S}_l of connections crossing l can be partitioned into two disjoint subsets, $\mathcal{B}_l(n)$ and $\mathcal{B}_l^c(n)$. In general, these sets are time-varying, as the number of connections bottlenecked on link l at time n may become bottlenecked on some other link l' at a later time. Let us assume that the network has a max-min fair equilibrium point. If this equilibrium is at least locally stable, then if we are close enough to the equilibrium, we expect the sets $\mathcal{B}_l(n)$ and $\mathcal{B}_l^c(n)$ to be time-invariant. In other words, assume that there exists a time-invariant set $\mathcal{B}_l(\infty)$ such that

$$\lim_{n \rightarrow \infty} \mathcal{B}_l(n) = \mathcal{B}_l(\infty), \quad \forall l \in \mathcal{L}. \quad (5)$$

We will investigate the time-varying nature of $\mathcal{B}_l(n)$ in Section 4. Now, let $M_{l,l}(n) = \text{card}(\mathcal{B}_l(n))$ denote the number of bottlenecked connections at time n . Equation (5) implies that there exists an $M_{l,l}(\infty)$ such that

$$\lim_{n \rightarrow \infty} M_{l,l}(n) = M_{l,l}(\infty).$$

By the definition of a bottleneck link, we have the following relation

$$\sum_{s \in \mathcal{B}_l(n)} x_s(n) + u_l(n) = C_l - G_l,$$

where $x_s(n) := r_s(n) - MR_s$, and $u_l(n) := \sum_{s \in \mathcal{B}_l^c(n)} x_s(n)$. From (4), we have

$$x_s(n) = ER_l(n - d_{s,l}), \quad \forall s \in \mathcal{B}_l(n), \quad (6)$$

where $ER_l(n)$ denotes the action of the switch (ER controller) at time n , and $d_{s,l}$ stands for the action delay of connection s from link l . Without any loss of generality, we assume that the $d_{s,l}$'s are ordered such that

$$0 \leq d_{1,l} \leq \dots \leq d_{M_{l,l}(n)}, l \leq b_l,$$

where b_l corresponds to the maximum network delay from link l .

The action delay, $d_{s,l}$, for connection s , consists of several components, such as the round-trip propagation delay, the queuing and processing delays, etc. Since queuing and processing delays are subject to variation, it is impossible for a link level controller to predict the exact value of $d_{s,l}$. Further, as evident from (1), any calculation of the fair share at the switch requires the knowledge of the number of bottlenecked connections, $M_{l,l}(\infty)$, which is not known to the switch either. Motivated by these observations, we want to develop a robust explicit rate congestion control algorithm, which does not require $M_{l,l}(\infty)$, or the exact value of the delays $d_{s,l}$.

Note that (3) can be written in the following form:

$$q_l(n+1) = \max \left\{ 0, \min \left\{ Q_l, q_l(n) + \sum_{k=0}^{b_l} m_{k,l}(n) ER_l(n-k) + u_l(n) - (C_l - G_l) \right\} \right\}, \quad (7)$$

where $m_{k,l}(n)$ denotes the number of bottlenecked connections on link l having k units of action delay at time n , and by (5) we must have

$$\lim_{n \rightarrow \infty} m_{k,l}(n) = m_{k,l}(\infty),$$

where $m_{k,l}(\infty)$'s satisfy

$$\sum_{k=0}^{b_l} m_{k,l}(\infty) = M_{l,l}(\infty).$$

In general, the aggregate rate of connections which are not bottlenecked on link l , $u_l(n)$, varies with time, and moreover it is affected by the current value of $q_l(n)$, and $ER_l(n)$. However, under assumption (5), there exists a limiting value $u_l(\infty)$ of $u_l(n)$. In order to simplify the final design, we also want to ease the effect of multiple time delays on the system dynamics. One way for the switch to do this is to wait long enough after issuing the explicit rate, $ER_l(n)$, so that all connections s in $\mathcal{B}_l(n)$ adjust their rates to $ER_l(n)$. As the switch has an estimate of the round-trip delay of each connection on its link,

putting an upper bound, b_l , on the maximum round-trip delay is feasible. Thus, if link l updates $ER_l(n)$ every $(b_l + 1)$ time units, all connections s in $\mathcal{B}_l(n)$ will have enough time to modify their transmission rates according to the explicit rate fed back at the previous update interval.

For ease of notation, let us introduce the following subsequences:

$$\begin{aligned} q_l^d(n) &:= q_l(n(b_l + 1)), \\ ER_l^d(n) &:= ER_l(n(b_l + 1)), \\ F_l^d(n) &:= F_l(n(b_l + 1)), \end{aligned}$$

which are obtained by down-sampling the original sequences $q_l(n)$, $ER_l(n)$, and $F_l(n)$. Note that $ER_l(n)$ is kept at the same value for an interval of length $(b_l + 1)$. That is,

$$\begin{aligned} ER_l(n(b_l + 1)) &= ER_l(n(b_l + 1) + 1) = \dots \\ &= ER_l((n + 1)(b_l + 1) - 1). \end{aligned}$$

To achieve the dual goal of max-min fairness and queue length stability, $ER_l^d(n)$ will be updated according to

$$ER_l^d(n) = \max \{ 0, \min \{ C_l, ER_l^d(n-1) - \alpha_l (F_l^d(n) - C_l) - \beta_l (q_l^d(n) - Q_l^*) \} \}, \quad (8)$$

$$ER_l^d(0) = C_l, \quad (9)$$

where Q_l^* is the target queue length, and α_l and β_l are parameters to be selected to meet various design criteria. Here the max function is introduced to ensure that the switch asks the sources to transmit at a positive rate in excess of their minimum rates, as required by the QoS specifications, and the min function puts an upper bound on the maximum allowed rate for the sources. In (8), the term $-\beta_l(q_l^d(n) - Q_l^*)$ is introduced to drive the queue length to the desired set point by providing negative feedback in the closed-loop system dynamics. Note that, this algorithm does not require any per-flow information, as it only uses the aggregate flow $F_l(n)$ on link l .

Now, if assumption (5) holds, the network has a max-min fair equilibrium point, and the discrete-time system described by (7)–(8) is stable, then $q_l^d(n)$ converges to Q_l^* , and $ER_l^d(n)$ converges to

$$ER_l^d(\infty) = \frac{C_l - (u_l(\infty) + G_l)}{M_{l,l}(\infty)}.$$

If we can show the stability of the system (7)–(8), then by (4), the rate r_s of connection s in $\mathcal{B}_l(\infty)$

asymptotically achieves

$$r_s(\infty) = \text{MR}_s + \text{ER}_s^d(\infty) = \text{MR}_s + \frac{C_l - (u_l(\infty) + G_l)}{M_{l,l}(\infty)},$$

which is the minimum rate plus max–min fair share of the capacity, see (1). Hence, we can conclude that the explicit-rate congestion control algorithm is globally convergent. It can be shown that if the controller gains (α_l, β_l) are picked properly, and if the size of the buffer, Q_b , is large enough, this algorithm is indeed globally asymptotically stable [9]. The proof in [9] is based on a Lyapunov analysis. First, note that the system dynamics (7)–(8) can be written in the form

$$x(n+1) = \text{sat}(Ax(n)),$$

where the saturation function enables us to write the min and max nonlinearities in (7)–(8) in a compact way. Thus, the stability of the algorithm can be analyzed in the framework of linear systems with saturation nonlinearities. Details can be found in [9].

To illustrate how our explicit rate congestion control algorithm performs, we have simulated it on a single bottleneck link, with capacity C . In simulations, a fixed rate of uncontrolled connections, u , and a slowly varying number of bottlenecked connections, $M(n)$, are used. For the simulation example, we assume that all bottlenecked connections have the same minimum rate (MR). The following parameter values are used in the simulation:

$$\begin{aligned} \text{MR}_s &= 100, \quad b = 5, \quad u = 400, \quad Q^* = 1500, \\ Q &= 3000, \quad C = 1500. \end{aligned}$$

The controller gains are picked as $(\alpha, \beta) = (0.1, 0.01)$ satisfying the theoretical bounds for stability. Note that the rate of the bottlenecked connections is unknown to the link. In order to investigate the ability of our algorithm to track variations in the number of bottlenecked connections, we let $M(n)$ be a random walk with boundaries at 1 and M_{\max} . Here, M_{\max} corresponds to the maximum number of connections a link can accommodate. We initially start with $M(0) = 3$ connections, and update this figure every $T(\eta)$ time units, where $T(\eta)$ is an exponential random variable with mean $1/\eta$. We take $\eta = 0.005$. A typical sample-path of $M(n)$ is depicted in Fig. 1. Each user comes with a network delay, $d_s \leq b$, which we assume to be unknown to the switch. We would like to demonstrate that our algorithm provides a max–min fair allocation of the link capacity with a stable queue length despite the uncertainties and changing network conditions.

As can be seen from Fig. 2, the queue length is regulated around the desired value of $Q^* = 1500$. The

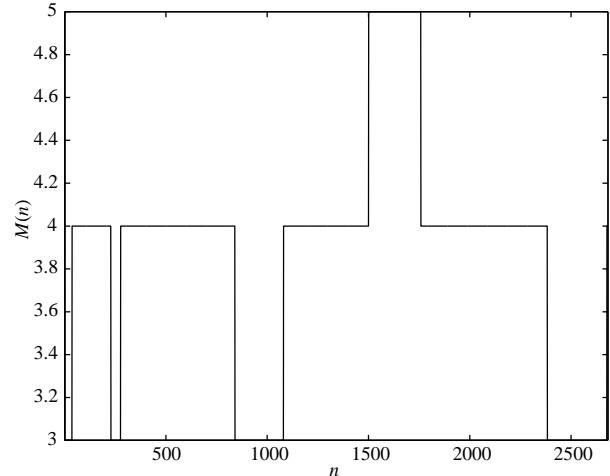


Fig. 1. A typical sample-path behavior of the number of bottlenecked connections.

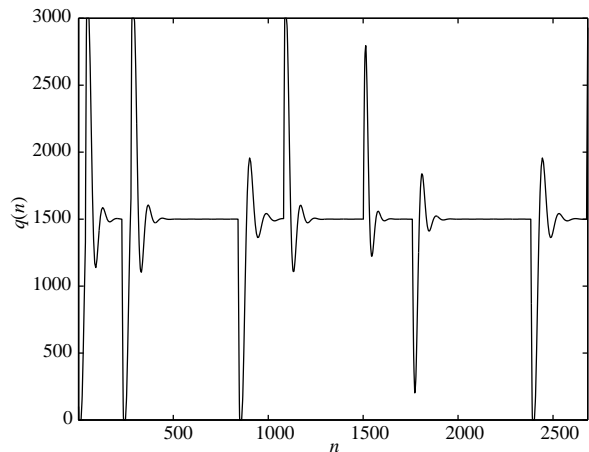


Fig. 2. Typical sample-path behavior of the queue length.

overshoots in Fig. 2 occur when the value of $M(n)$ changes. They are inevitable when the amount of change in the number of bottlenecked connections is comparable to the maximum number of connections, which is the case in this simulation example. As the number of connections increases, the transitions in the queue dynamics become smoother. In order to see that our design achieves max–min fairness as well, we first calculate the max–min fair share of each connection according to (1). As the number of bottlenecked connections varies over time, this share varies, too. We want the output of the explicit rate controller, $\text{ER}(n)$, to continuously track this share. In Fig. 3, we plot $\text{ER}(n)$. The flat portions of this graph corresponds to the max–min fair share of the link capacity, as can be verified by direct calculation.

Finally, we note that for a fixed value of α , the design parameter β can be chosen to tradeoff between

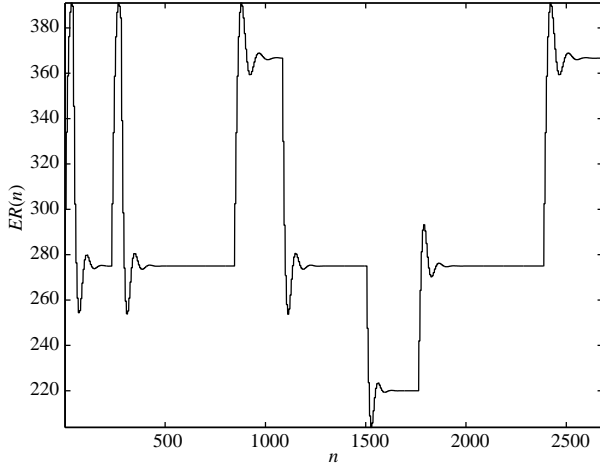


Fig. 3. Typical sample-path behavior of the explicit rate controller.

the rate of convergence and the magnitude of overshoots. A smaller value of β results in a smaller overshoot, but a larger settling time.

4. Network Level Implementation of the Algorithm

We start our analysis at the network level by taking a closer look at the assumption (5):

$$\lim_{n \rightarrow \infty} \mathcal{B}_l(n) = \mathcal{B}_l(\infty), \quad \forall l \in \mathcal{L},$$

which essentially states that in steady state the set of links bottlenecked on link l does not change. The convergence analysis of the link level algorithm makes use of this assumption [9]. In this section, we want to justify (5) by fixing the explicit-rate update scheme of all links $l \in \mathcal{L}$ to the form (8)–(9) with possibly different gain vectors (α_l, β_l) . For tractability, let us first make some simplifying assumptions. First, we assume that there is no delay in the network. Even though the ensuing analysis can be equally applied to a system with delays, we choose not to include any time delays in the system dynamics, mainly because we want to concentrate on coupling effects between the links. We also ignore the saturation nonlinearities in the queue dynamics (7) and the update scheme (8)–(9), which are not activated for small deviations around the equilibrium. Let us further assume that all sources have zero minimum rate (MR) requirements. The case where MRs are nonzero and different for each connection can be easily incorporated into the analysis, but is left out here for the sake of clarity in presentation of the main message of this section. Finally, we assume that the total number of sources

M_l using a particular link l and the topology of the network do not change with time. Under these assumptions, for each link $l \in \mathcal{L}$ we have:

$$q_l(n+1) = q_l(n) + \sum_{s \in \mathcal{S}_l} r_s(n) - C_l, \quad (10)$$

$$\begin{aligned} \text{ER}_l(n+1) &= \text{ER}_l(n) - \beta_l(q_l(n) - Q_l^*) \\ &\quad - \alpha_l \left(\sum_{s \in \mathcal{S}_l} r_s(n) - C_l \right). \end{aligned} \quad (11)$$

Recall from (4) (with $\text{MR}_s = 0$ and $d_{s,l} = 0$) that for each source $s \in \mathcal{S}$, the rate $r_s(n)$ is given by

$$r_s(n) = \min_{l \in \mathcal{L}_s} \text{ER}_l(n).$$

Now, the aggregate flow $F_l(n) = \sum_{s \in \mathcal{S}_l} r_s(n)$ on link l can be written as

$$\begin{aligned} F_l(n) &= \sum_{s \in \mathcal{S}_l} r_s(n) = \sum_{s \in \mathcal{B}_l(n)} r_s(n) + \sum_{s \in \mathcal{B}_l^*(n)} r_s(n) \\ &= M_{l,l}(n) \text{ER}_l(n) + \sum_{s \in \mathcal{S}_l \setminus \mathcal{B}_l(n)} \min_{l \in \mathcal{L}_s} \text{ER}_l(n). \end{aligned} \quad (12)$$

For a given ordering o of the explicit rate vector $\text{ER} := \{\text{ER}_l | l \in \mathcal{L}\}$, the minimization in (12) can be trivially carried out. There are $L!$ different orderings of the vector ER , each one leading to a different expression for $F_l(n)$. These expressions, when substituted back into the state equations (10)–(11), result in different linear systems each one valid for a particular ordering of the vector of rates ER . To investigate the structure of these linear systems more closely, let us consider a network where the connections have at most two hops. In other words, we assume that $\text{card}(\mathcal{L}_s) \leq 2$ for all $s \in \mathcal{S}$. Let $M_{l,k}$ denote the number of sources using links l and k . Observe that by definition we have $M_{l,k} = M_{k,l}$. Since the topology of the network does not change, $M_{l,k}$'s are actually constants. The total number of connections crossing link l can be calculated as

$$M_l = \sum_{k=1}^L M_{l,k}. \quad (13)$$

Recall that M_l also equals the cardinality of the set \mathcal{S}_l . Since each connection crosses at most two links, the minimization

$$\min_{l \in \mathcal{L}_s} \text{ER}_l(n),$$

is equivalent to either finding the unique link source s crosses, or a binary decision between the

two links s uses. Using this fact, for link l , $F_l(n)$ can be written as

$$F_l(n) = \sum_{s \in \mathcal{S}_l} r_s(n) = \sum_{k=1}^L M_{l,k}(n) \text{ER}_k(n),$$

where $M_{l,k}(n)$ denotes the number of connections using links k and l and bottlenecked on link k . $M_{l,k}(n)$'s can be expressed as a function of $\text{ER}_k(n)$'s in the following way:

$$M_{l,k}(n) = \begin{cases} M_{l,k} & \text{if } \text{ER}_k(n) \leq \text{ER}_l(n) \\ 0 & \text{if } \text{ER}_k(n) \geq \text{ER}_l(n) \end{cases} \quad (14)$$

for all $k \in \mathcal{L}$ such that $k \neq l$, and using (13) $M_{l,i}(n)$ can be obtained as

$$M_{l,i}(n) = M_l - \sum_{k=1, k \neq l}^L M_{l,k}(n). \quad (15)$$

Hence, given an arbitrary ordering of the explicit rate vector ER , the discrete-time system (10)–(11) takes the form of one of the possible $L!$ linear systems, each one described by a matrix A_p for $p = 1, \dots, L!$. Note that (10)–(11) can be rewritten as

$$q_l(n+1) = q_l(n) + \sum_{k=1}^L M_{l,k}(n) \text{ER}_k(n) - C_l, \quad (16)$$

$$\begin{aligned} \text{ER}_l(n+1) &= \text{ER}_l(n) - \beta_l(q_l(n) - Q_l^*) \\ &\quad - \alpha_l \left(\sum_{k=1}^L M_{l,k}(n) \text{ER}_k(n) - C_l \right). \end{aligned} \quad (17)$$

We introduce the states $y_0(n) := [q_1(n) \dots q_L(n)]^T$, $y_1(n) := [\text{ER}_1(n) \dots \text{ER}_L(n)]^T$, the vectors $C := [C_1 \dots C_L]^T$, $Q^* := [Q_1^* \dots Q_L^*]^T$, and the diagonal matrices, $D_\alpha := \text{diag}(\alpha_1, \dots, \alpha_L)$, and $D_\beta := \text{diag}(\beta_1, \dots, \beta_L)$. Now, a given ordering p of the explicit rates $\text{ER}_k(n)$ at time n , induces a matrix Π_p of $M_{l,k}$'s obtained by using (14)–(15). Thus, for the ordering p at time n , the system (16)–(17) can be written in the state-space form as follows:

$$\begin{aligned} y_0(n+1) &= y_0(n) + \Pi_p y_1(n) - C, \\ y_1(n+1) &= y_1(n) - D_\beta(y_0(n) - Q^*) - D_\alpha(\Pi_p y_1(n) - C). \end{aligned}$$

Letting $y(n) := [y_0(n) \ y_1(n)]^T$, we have

$$\begin{aligned} y(n+1) &= \begin{bmatrix} I & \Pi_p \\ -D_\beta & I - D_\alpha \Pi_p \end{bmatrix} y(n) \\ &\quad + \begin{bmatrix} -I \\ D_\alpha \end{bmatrix} C + \begin{bmatrix} 0 \\ D_\beta \end{bmatrix} Q^*, \end{aligned} \quad (18)$$

where we identify A_p as

$$A_p = \begin{bmatrix} I & \Pi_p \\ -D_\beta & I - D_\alpha \Pi_p \end{bmatrix}.$$

Defining

$$B := \begin{bmatrix} -I \\ D_\alpha \end{bmatrix} C + \begin{bmatrix} 0 \\ D_\beta \end{bmatrix} Q^*,$$

the system dynamics can be written in the form of an $L!$ -dimensional hybrid system

$$y(n+1) = A_p y(n) + B, \quad \text{if } y_1(n) \in \mathcal{D}_p, p = 1, \dots, L!, \quad (19)$$

where \mathcal{D}_p is the set of vectors ER_l in \mathcal{R}^L satisfying the ordering p . Hence, to give an affirmative answer to the question of whether $\lim_{n \rightarrow \infty} \mathcal{B}_l(n) = \mathcal{B}_l(\infty)$, one has to show that the hybrid system (19) is globally asymptotically stable. In [9], we have established the local stability of (19) under certain conditions on the controller gains (α_l, β_l) . We state this result here without a formal proof; we simply sketch the main ideas used in the proof.

Theorem 1. The simplified network level algorithm (16)–(17) is locally asymptotically stable if the link gains (α_l, β_l) satisfy

$$0 < \alpha_l < \frac{2}{M_l}, \quad (20)$$

$$0 < \beta_l < \alpha_l \quad (21)$$

for all $l \in \mathcal{L}$.

Sketch of the Proof. The proof of Theorem 1 uses the fact that if we are close enough to the max–min fair equilibrium point of the system (19), then the collection of sets $\mathcal{B}(n) := \{\mathcal{B}_l(n) | l \in \mathcal{L}\}$ must be time-invariant. Hence, the set of connections bottlenecked on link l is $\mathcal{B}_l(\infty)$. Now, reordering the links in such a way that $M_{l,k}(\infty) > 0$, $l < k$, $\forall l, k \in \mathcal{L}$, we obtain an upper-triangular matrix $M(\infty) = [M_{l,k}(\infty)]$ with nonzero diagonal entries. This leads to the following linear system around the equilibrium point:

$$y(n+1) = A(\infty)y(n) + B,$$

where

$$A(\infty) = \begin{bmatrix} I & M(\infty) \\ -D_\beta & I - D_\alpha M(\infty) \end{bmatrix}.$$

The local stability of (19) can be investigated by checking if the roots of the characteristic polynomial of $A(\infty)$ lie inside the unit circle. The conditions under which these are true are given in Theorem 1. \square

Now note that the controller gains (α_l, β_l) at each link can be chosen independent of the rest of the network. In other words, in order to decide on its own set of parameter values, the switch controlling link l does not need to know how the other switches in the network pick their gains. This feature makes it possible to implement the algorithm in a decentralized fashion.

Next, to shed some light on the dynamical behavior of the hybrid system (19), we analyze the special case of a two link communication network as depicted in Fig. 4 below.

Since $\mathcal{L} = \{1, 2\}$, any given connection may use only link 1 (T1), only link 2 (T2), or both links 1 and 2 (T). Using (14)–(15), $M_{1,1}(n)$ and $M_{2,2}(n)$ can be calculated as

$$M_{1,1}(n) = \begin{cases} M_1 - M_{1,2} & \text{if } ER_1(n) \geq ER_2(n) \\ M_1 & \text{if } ER_1(n) \leq ER_2(n). \end{cases}$$

Similarly,

$$M_{2,2}(n) = \begin{cases} M_2 & \text{if } ER_1(n) \geq ER_2(n) \\ M_2 - M_{1,2} & \text{if } ER_1(n) \leq ER_2(n). \end{cases}$$

Now, also using (13) we can write the system dynamics (16)–(17) as

$$y(n+1) = \begin{cases} A_1 y(n) + B & \text{if } ER_1(n) \geq ER_2(n) \\ A_2 y(n) + B & \text{if } ER_1(n) \leq ER_2(n), \end{cases} \quad (22)$$

where $y(n) = [q_1(n) \ q_2(n) \ ER_1(n) \ ER_2(n)]^T$, $B = [-C_1 \ -C_2 \ \alpha_1 C_1 + \beta_1 Q_1^* \ \alpha_2 C_2 + \beta_2 Q_2^*]^T$, and A_1 and A_2 are given by

$$A_1 = \begin{bmatrix} 1 & 0 & M_1 - M_{1,2} & M_{1,2} \\ 0 & 1 & 0 & M_2 \\ -\beta_1 & 0 & 1 - \alpha_1(M_1 - M_{1,2}) & -\alpha_1 M_{1,2} \\ 0 & -\beta_2 & 0 & 1 - \alpha_2 M_2 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 1 & 0 & M_1 & 0 \\ 0 & 1 & M_{1,2} & M_2 - M_{1,2} \\ -\beta_1 & 0 & 1 - \alpha_1 M_1 & 0 \\ 0 & -\beta_2 & -\alpha_2 M_{1,2} & 1 - \alpha_2(M_2 - M_{1,2}) \end{bmatrix}.$$

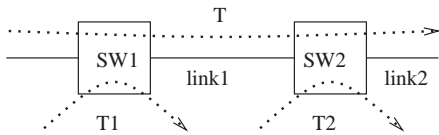


Fig. 4. Two link case.

For the purpose of analysis we assume without any loss of generality that¹

$$\frac{C_1}{M_1} \geq \frac{C_2}{M_2},$$

which essentially means that the capacity per connection of link 1 is larger than or equal to that of link 2. Thus, in steady-state, we would expect the connections using both links to be bottlenecked on link 2. Under this assumption, the hybrid system (22) has a unique max–min fair equilibrium point at

$$q_{1e} = Q_1^*, \quad q_{2e} = Q_2^*,$$

$$ER_{1e} = \frac{C_1 - C_2(M_{1,2}/M_2)}{M_1 - M_{1,2}},$$

$$ER_{2e} = \frac{C_2}{M_2},$$

and it can be shown that this equilibrium point is globally asymptotically stable if the controller gains (α_l, β_l) are picked appropriately [9]. In proving this result, we first bring the system (22) into the form

$$z(n+1) = \tilde{A}_1 z(n) - \tilde{B} \Phi(\tilde{C}z(n) - \theta),$$

where $\Phi(z) : \mathcal{R} \rightarrow \mathcal{R}$ is a function defined by

$$\Phi(x) = \begin{cases} 0 & x \leq 0 \\ x & x \geq 0. \end{cases}$$

Formulated as above, the stability of (22) can be investigated within the framework of absolute stability of sampled-data systems [16]. Extending the stability analysis in [9] to an arbitrary network \mathcal{N} is still an open problem. Hence, the link level explicit rate congestion control algorithm we derived in Section 3 may not be globally convergent for $L > 2$, since a proof for

$$\lim_{n \rightarrow \infty} \mathcal{B}_l(n) = \mathcal{B}_l(\infty), \quad \forall l \in \mathcal{L}$$

for networks with more than two links is not yet available.

5. An Extension to a Marking Based Scheme

In its current form, the congestion control algorithm (8)–(9) cannot be used in the Internet, as it requires the

¹If this inequality is reversed, the same analysis holds with indices 1 and 2 interchanged.

routers to provide the connections with explicit-rate feedback messages. In this section, we show that the same algorithm can be used to update the marking rate on link l if we interpret $ER_l(n)$ as an indicator of congestion. More precisely, suppose on link l we mark packets with probability $1 - e^{-\lambda_l(n)}$, where $\lambda_l(n)$ is generated by

$$\lambda_l(n+1) = \max\{0, \min\{C_l, \lambda_l(n) - \alpha_l(F_l(n) - C_l) - \beta_l(q_l(n) - Q_l^*)\}\}, \quad (23)$$

similar to (8). The idea of marking packets with a probability of the form $1 - e^{-\lambda_l(n)}$ has been previously suggested in [3]. In [3], an update scheme for $\lambda_l(n)$ in the following form is derived as a result of a utility-based optimization problem:

$$\lambda_l(n+1) = \max\{0, \min\{C_l, \lambda_l(n) - \gamma_l(F_l(n) - C_l + a_l(q_l(n) - Q_l^*))\}\}, \quad (24)$$

where γ_l is the step size, and a_l is a small constant. Algorithm (24) is commonly referred to as REM (Random Exponential Marking). As we mentioned before, it is conceivable that source s measures the fraction $f_s(n)$ of unmarked packets in time slot n , which is given by²

$$f_s(n) = e^{-\sum_{l \in \mathcal{L}_s} \lambda_l(n)},$$

where we have ignored the effect of action delay for the clarity of presentation. Hence, source s can estimate $\sum_{l \in \mathcal{L}_s} \lambda_l(n)$ as

$$\sum_{l \in \mathcal{L}_s} \lambda_l(n) = -\ln f_s(n).$$

Suppose source s uses this information, and updates its rate r_s according to

$$r_s(n) = MR_s + g\left(\sum_{l \in \mathcal{L}_s} \lambda_l(n)\right), \quad (25)$$

where $g(\cdot) : \mathcal{R} \rightarrow \mathcal{R}$ is a monotonically decreasing function to ensure that increasing $\sum_{l \in \mathcal{L}_s} \lambda_l(n)$ results in a lower value of the rate $r_s(n)$. Then, the aggregate flow $F_l(n)$ on link l becomes

$$F_l(n) = \sum_{s \in \mathcal{S}_l} g\left(\sum_{k \in \mathcal{L}_s} \lambda_k(n)\right) + G_l.$$

Substituting this expression for $F_l(n)$ into (3) and (23) yields

$$q_l(n+1) = \max\left\{0, \min\left\{Q_l, q_l(n) + \sum_{s \in \mathcal{S}_l} g\left(\sum_{k \in \mathcal{L}_s} \lambda_k(n)\right) - (C_l - G_l)\right\}\right\},$$

$$\lambda_l(n+1) = \max\left\{0, \min\left\{C_l, \lambda_l(n) - \alpha_l\left(\sum_{s \in \mathcal{S}_l} g\left(\sum_{k \in \mathcal{L}_s} \lambda_k(n)\right) - (C_l - G_l)\right) - \beta_l(q_l(n) - Q_l^*)\right\}\right\}.$$

If this algorithm converges, it must converge to its unique equilibrium point $(q_l(\infty), \lambda_l(\infty))$, where $q_l(\infty) = Q_l^*$, and $\lambda_l(\infty)$ is obtained from the solution of

$$\sum_{s \in \mathcal{S}_l} g\left(\sum_{k \in \mathcal{L}_s} \lambda_k(\infty)\right) = C_l - G_l, \quad \forall l \in \mathcal{L}. \quad (26)$$

Now, suppose we use the function $g(z) = 1/z$, $z > 0$, as the rate update function in (25). Clearly, $g(z)$ is monotonically decreasing, since $g'(z) = -1/z^2 < 0$, $\forall z > 0$. With this choice of $g(\cdot)$, (26) becomes

$$\sum_{s \in \mathcal{S}_l} \frac{1}{\sum_{k \in \mathcal{L}_s} \lambda_k(\infty)} = C_l - G_l, \quad \forall l \in \mathcal{L}.$$

As a result, the rate $r_s(n)$ of source s converges to

$$r_s(\infty) = MR_s + \frac{1}{\sum_{k \in \mathcal{L}_s} \lambda_k(\infty)}$$

It can be shown that the vector of rates $x(\infty) = \{x_s(\infty) | s \in \mathcal{S}\}$ in excess of the minimum rates MR_s , is proportionally fair [13]. Hence, if the above update scheme is globally convergent, then the pair of vectors (r, q) asymptotically achieve proportional fairness, as well as queue length stability. The foregoing analysis can actually be linked to a utility maximization problem [3]. Suppose our objective is to choose source rates x so as to:

$$\max_{x_s \geq MR_s} \sum_{s \in \mathcal{S}} U_s(x_s), \quad (27)$$

$$\text{subject to } \sum_{s \in \mathcal{S}_l} x_s \leq C_l, \quad \forall l \in \mathcal{L} \quad (28)$$

where $U_s(x_s)$ is the utility level source s attains when its rate is x_s . The utility function $U_s(x_s)$ can be thought of

²This argument assumes that links mark packets independently.

as the enjoyment source s gets when it transmits at rate x_s . We assume that U_s are strictly concave increasing and twice continuously differentiable. This flow control problem is posed in [13], and solved in [14] using a penalty function approach. In [15] a gradient projection algorithm is suggested to solve the dual problem. This algorithm was originally given by

$$\lambda_l(n+1) = \max\{0, \lambda_l(n) + \gamma_l(F_l(n) - C_l)\} \quad (29)$$

$$x_s(n) = g_s \left(\sum_{l \in \mathcal{S}_l} \lambda_l(n) \right), \quad \forall s \in \mathcal{S}, \quad (30)$$

where $\lambda_l(n)$ is the dual variable of the optimization problem (27)–(28), and

$$g_s(z) = \max\{MR_s, U_s'^{-1}(z)\}.$$

Here $U_s'^{-1}$ denotes the inverse function of the marginal utility. It is proved in [15] that under the algorithm (29)–(30), the source rates x converge to the unique optimal solution of the optimization problem (27)–(28), provided that the step size $\gamma > 0$ is sufficiently small.

The drawback of this original algorithm is the fact that it can lead to large backlog (queue length) and delay. To see this recall that the queue length at link l evolves according to

$$q_l(n+1) = \max\{0, \min\{Q_l, q_l(n) + F_l(n) - C_l\}\}.$$

Comparing with (29), it can be seen that $\lambda_l(n)$ is related to $q_l(n)$ by $q_l(n) = 1/\gamma_l \lambda_l(n)$ when $q_l(n) \leq Q_l$. Thus $q_l(n)$ can be large (or it may saturate) when $\gamma > 0$ is small, which is undesirable. To regulate $q_l(n)$ around a target value Q_l^* , in [3] the original algorithm (29)–(30) has been modified to (24). Convergence of REM for the single bottleneck link case has been recently proven in [17]; however, the proof for the general case of multiple links is still an open problem.

6. Conclusions and Future Work

In this paper, we have described a framework for dealing with congestion control problems that arise in communication networks. We have shown that many results from control theory can be applied to provide solutions to these problems. In particular, we have presented a link level decentralized explicit rate congestion control algorithm, and have shown that it performs well under various criteria, such as max-min fairness, and minimum rate constraints. The convergence proof of this algorithm involves stability analysis of linear systems with saturation type nonlinearities. In addition, at network level, we have

shown that the stability of this algorithm can be studied in the framework of hybrid systems. Finally, we have provided a link between our explicit rate congestion control algorithm and some of the marking based early congestion notification schemes, such as REM.

Several directions of future research are possible. The discrete-time model of Section 3 assumes that updates at the sources and the links are synchronized to occur at times $n = 1, 2, \dots$. However, in reality, these updates are usually asynchronous due to many reasons, such as variations in feedback delays. Hence, it is desirable to have an asynchronous update algorithm which is known to be asymptotically stable. Finding such an algorithm, or showing if the explicit rate congestion control algorithm of Section 3 has this property is an open problem. Another direction of future research is to investigate the stability of the hybrid system described in Section 4. Finally, further study needs to be done in simulating the algorithm in a real network environment to see how well it reacts to network latencies, as well as to evaluate its performance under various scenarios.

References

1. Ait-Hellal O, Altman E, Başar T. Rate-based flow control with bandwidth information. *European T Telecommunications* 1997; 8(1): 55–65
2. Altman E, Başar T, Srikant R. Congestion control as a stochastic control problem with action delays. *Automatica* 1999; 35: 1937–1950
3. Athuraliya W, Low S. Optimization flow control-II: Random Exponential Marking 2000 (submitted for publication)
4. ATM Forum, Technical Committee. Traffic management specification, Version 4.1, af-tm-0121.000. March 1999, pp 43–55
5. Benmohamed L, Meerkov SM. Feedback control of congestion in packet switching networks: The case of a single congested node. *IEEE/ACM T Networking* 1993; 1(6): 693–707
6. Benmohamed L, Wang YT. A control-theoretic ABR explicit rate algorithm for ATM switches with per-VC queueing. In: *P IEEE INFOCOM* 1998
7. Bertsekas DP, Gallager R. *Data networks*. Prentice-Hall, Englewood Cliffs NJ 1987
8. Floyd S, Jacobson V. Random Early Detection gateways for congestion avoidance. *IEEE/ACM T Networking* 1997; 1(4): 397–413
9. Imer OÇ, Başar T, Srikant R. A distributed globally convergent algorithm for fair, queue-length-based congestion control. In: *40th IEEE conference on decision and control* 2001 (submitted) (Also submitted to *IEEE T Automatic Control* 2001)
10. Imer OÇ, Compans S, Başar T, Srikant R. ABR congestion control in ATM networks. *IEEE Control Systems Magazine* 2001; 21(1): 38–56
11. Jacobson V. Congestion avoidance and control. In: *P SIGCOMM* 1988

12. Kalyanaraman S, Jain R, Fahmy S, Goyal R, Vandalore B. The ERICA switch algorithm for ABR traffic management in ATM networks. *IEEE/ACM T Networking* 2000; 8(1): 87–98
13. Kelly FP. Charging and rate control for elastic traffic. *European T Telecommunications* 1997; 8: 33–37
14. Kelly FP, Maulloo A, Tan D. Rate control for communication networks: Shadow prices, proportional fairness and stability. *J Operations Research Society* 1998; 49(3): 237–252
15. Low SH, Lapsley DE. Optimization flow control-I: Basic algorithm and convergence. *IEEE/ACM T Networking* 1999; 7(6): 861–874
16. Szego GP, Pearson JB. On the absolute stability of sampled-data systems: the indirect control case. *IEEE T Automatic Control* 1964; 9: 160–163
17. Yin Q, Low SH. Convergence of REM flow control at a single link. *IEEE Communications Lett* 2001 (to appear)