Stabilized Edge-to-Edge Aggregate Flow Control

Hyung-Keun Ryu, Jeong-Woo Cho, and Song Chong

Dept. EECS, KAIST, 373-1 Gusung-dong, Yusung-gu, Taejon, 305-701, Korea
{hkryu, ggumdol}@netsys.kaist.ac.kr, song@ee.kaist.ac.kr

Abstract. In this paper, we present a distributed flow control scheme which achieves weighted max-min fair bandwidth allocation among all source-destination pairs on a per-aggregate basis within its network. The motivation behind the scheme is the absence of per-aggregate flow control in the current Internet, resulting in inability to enforce a certain fairness on source-destination flows. In our scheme, the distributed algorithm to compute weighted max-min fair rates is based on PI control in feedback control theory. We mathematically prove the asymptotic stability of the algorithm in presence of aggregate flows with heterogeneous round-trip delays. Through simulations we demonstrate the effectiveness of the proposed scheme in controlling per-aggregate flows.

1 Introduction

The current Internet is a TCP-controlled network, hence, the Internet congestion control relies mostly on TCP, and TCP is an end-to-end protocol over which Internet service providers(ISPs) do not have any control. Consequently, ISPs are facing a big problem as they sell their bandwidth to customers but in the Internet core they have no tool to explicitly control or engineer bandwidth.

In order to tackle this problem, we present a distributed flow control scheme, in which traffic engineering is carried out by means of edge-to-edge aggregate flow control. More specifically, instead of relying on explicit admission control and/or explicit reservation [1,2], the edge-to-edge flow control on aggregate level plays a major role to ensure fair bandwidth sharing between aggregate flows. The proposed flow control scheme is hierarchical. In the upper layer, weighted max-min flow control is implemented and acting on a per-aggregate and edge-to-edge basis, and in the lower layer, TCP flows belonging to a source-destination flow share its per-aggregate bandwidth allocated by the upper layer in their normal way. Thus, the scheme does not require modification nor replacement of present TCP congestion control.

Several studies have been done in the area of aggregate flow control. [3] proposed an architecture to achieve a fair bandwidth allocation among individual flows without using per-flow state in the network core. However, [3] does not provide any minimum rate guarantees but also can cause a serious performance degradation of TCP, especially in a large delay-bandwidth network, because it achieves fair bandwidth allocation by probabilistically dropping packets in

N. Mitrou et al. (Eds.): NETWORKING 2004, LNCS 3042, pp. 574–587, 2004.

network core. [4] proposed an overlay congestion control architecture for edge-to-edge traffic control, but it does not support minimum rate guarantee and differentiated service on a per-aggregate level. Some other studies [5,6,7,8] have addressed aggregate flow control by employing an aggregate TCP connection which multiplexes local TCP connections into a single, persistent TCP connection and is operated by TCP or its modified congestion control algorithm. The problem with these studies is that they still depend upon packet loss within the network to detect congestion in the bottleneck link. Compared to the previous works, the proposed scheme has the following features: 1) It achieves the stable target queue lengths at the bottlenecks and maximizes the network utilization. 2) It achieves almost no packet losses inside core network. Instead, it distributes the interior network congestion to the network edges. 3) It also guarantees minimum rates and supports weighted max-min fairness on a per-aggregate basis. 4) It imposes minimal complexity in the network core and is highly scalable, in that no per-aggregate state management are necessary at the network core.

In this paper we propose a distributed algorithm which is based on the rate-based closed loop control for the aggregate flow control. It is highly responsive and adaptive to available network bandwidth change and network congestion. There are extensive prior works on the design of distributed algorithms for rate-based flow control. However most of those works is not completely satisfactory for their complexity or for the lack of analysis from the point of view of stability. [9] derived analytically a control-theoretic fair rate allocation algorithm which allows for arbitrary control of the closed-loop performance, but its practical use is limited by high degree of implementation complexity as the round-trip delay increases. The work by [9,10,11] proposed distributed algorithms which adapt quickly to congestion while achieving max-min fairness, either with or without minimum rate guarantee, among competing flows. The work by [12] addressed a weighted max-min fair bandwidth sharing with minimum rate guarantee, but employs per-flow state management to calculate fair rate for each flow.

Our main objective is to develop a distributed algorithm to compute common fair rate in a weighted max-min fair sense, which is based on proportional and integral(PI) control in feedback control theory [13,14]. The algorithm is *scalable* in that the computational complexity imposed on each link is $O(1)$, i.e., independent of number of aggregate flows travelling through the link. It is *stable* in that it converges asymptotically to the desired equilibrium, and has explicit *link buffer control* in that buffer occupancy of every bottlenecked link in a path asymptotically converges to the desired value. Another objective is to derive an explicit and usable, sufficient and necessary condition to ensure asymptotic stability of the network employing the proposed distributed algorithm even in presence of aggregate flows with heterogeneous round-trip delays.

2 A Distributed Flow Control Scheme

The Internet can be thought of as a concatenation of heterogeneous network clouds. Our scheme is applied to a network cloud which consists of edge nodes

at the network boundary and core nodes at the network interior. In each ingress edge node, incoming TCP flows having the same ingress-egress edge pair are classified and multiplexed into a single aggregate flow in a per-aggregate queue.

In the network, the proposed scheme runs a distributed and asynchronous algorithm to share available network bandwidth among competing aggregate flows. The distributed algorithm consists of two components, a link algorithm and a source algorithm.[1] The link algorithm, implemented at each outgoing link of edge nodes and core nodes, computes locally the *common fair rate* using the occupancy information of the link buffer. The common fair rate is same as the excess bandwidth normalized by the sum of pre-assigned weights of the aggregate flows sharing the link. The source algorithm, implemented in each edge node, computes the allowed source rate using feedback rate in its path.

For communication between sources and links, each source generates and inserts a control packet each time N_b bytes of data is transmitted. The control packet carries current source rate, pre-assigned weight, and minimum rate. The control packet is transmitted into the network together with data packets and travel along the forward path down to egress edge node, then returned to the source along the backward path, which may not be identical with the forward path in the current IP network. The control packets travelling in the forward and backward paths are called forward control packet(FCP) and backward control packet(BCP), respectively.

Each outgoing link along the forward path intercepts every FCP arriving at the link and updates the FCP's feedback rate field with the same result of the common fair rate computation no matter which aggregate flow it belongs to.

2.1 Link Algorithm

The proposed common fair rate computation at each outgoing link is as follows. The common fair rate is calculated periodically with an update interval T by

$$f[k] = -C_P(q[k] - q_T) - C_I \sum_{n=0}^{n=k} (q[n] - q_T) \tag{1}$$

where $C_P > 0$ and $C_I > 0$ are the PI controller gains, $q[k]$ is the queue length at the link buffer, and q_T is the target queue length. The choice of C_P and C_I determines the convergence rate of the iteration as well as the stability of the distributed algorithm. The notable feature of this algorithm is that the common fair rate computation is virtually independent of the number of aggregate flows travelling through the link and thus highly scalable. Moreover, the proposed algorithm jointly controls rate allocation and link buffer control, meaning that as the iteration proceeds, it makes the link buffer occupancy converge to the target value, i.e., $\lim_{t\to\infty} q(t) = q_T$, while finding the weighted max-min fair rate as is proved in Section 3.

[1] In the proposed scheme, the edge nodes function as the effective sources and destinations of aggregate flows. More specifically, each ingress edge node has per-aggregate sources(say, virtual sources), each of which performs per-aggregate queueing and rate adaptation for an aggregate flow.

At each outgoing link, common fair rate allocation per aggregate flow is performed aperiodically upon arrival of the corresponding FCPs in forward path. That is, upon arrival of a FCP at time t, the common fair rate $f(t)$ computed locally by the link is compared with the feedback rate being carried by the FCP's feedback rate field, and the smaller value is written onto the field and delivered to the source. Note that $f(t)$ is the present value of $f[k]$.

2.2 Source Algorithm

Each aggregate flow has its own class of service, each of which is characterized by a 3-tuple consisting of weight w_i, minimum rate m_i, and peak rate p_i. Let f_i be the common fair rate notified by a BCP from the link to the source. Using the common fair rate f_i, each source calculates the *weighted fair rate* which supports minimum rate and excess bottleneck bandwidth proportional to a weight. Then it calculates allowed source rate from the minimum of the weighted fair rate and peak rate constraint.

Upon receipt of a BCP, allowed source rate a_i for aggregate flow i is computed as follows.

$$a_i = \min[w_i \cdot f_i + m_i,\ p_i] \ . \tag{2}$$

Now, the weighted max-min fair bandwidth allocation among competing aggregate flows is simply obtained by regulating the transmission rate of each aggregate flow using its allowed source rate.

3 Modelling and Analysis

3.1 Network Model

First, consider a network model in Fig. 1 where we model a single link explicitly and the other links implicitly to simplify the analysis. The link is an outgoing link with a FIFO queue and has N aggregate flows passing through it. We use a continuous-time fluid flow approximation to model the system.

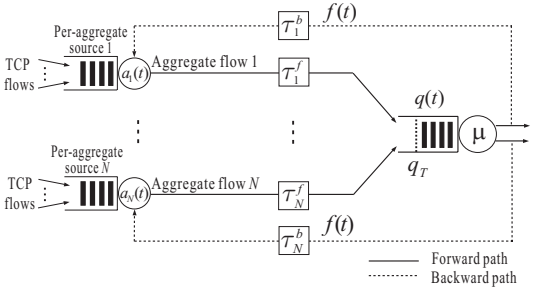

Fig. 1. Network model with a link of interest

Let AF_i denote an aggregate flow i and τ_i, τ_i^f, and τ_i^b denote the round-trip delay, forward-path delay, and backward-path delay of AF_i, respectively. We

assume that $\tau_i = \tau_i^f + \tau_i^b$ is constant and the sources are *persistent* until the system reaches steady state. We also assume that the available bandwidth μ at the link is constant until the system reaches steady state and the buffer size at the bottleneck link is finite and larger than the target queue length.

Let $a_i(t)$ denote the rate at which the source i transmits data at the source time t, and denote p_i the peak rate constraint of AF_i. Next, let $f_i(t)$ denote the common fair rate allocated to AF_i by the link of interest and $b_i(t)$ be the latest minimum value of the common fair rates allocated to AF_i by other links along the AF_i's path. Moreover, let $f_i^w(t)$ and $b_i^w(t)$ denote the weighted fair rates of AF_i which are computed by source i at the source time t as follows:

$$f_i^w(t) = w_i f_i(t) + m_i, \;\; f_i(t) = f(t - \tau_i^b), \quad \forall\, i \in N \tag{3}$$

and

$$b_i^w(t) = w_i b_i(t) + m_i, \quad \forall\, i \in N \tag{4}$$

where w_i and m_i denote a weight value and the minimum rate which the link is required to guarantee during the entire holding time of AF_i, respectively. We assume that $m_i \leq p_i$, $\forall\, i \in N$ and there exists an admission control which guarantees $\sum_{i \in N} m_i < \mu$.

The source behavior of AF_i can be modeled by

$$a_i(t) = \min[\; f_i^w(t),\; b_i^w(t),\; p_i \;], \quad \forall\, i \in N \tag{5}$$

where N denotes the set of all the aggregate flows whose route includes the bottleneck node of interest. This model implies that a source transmits data at the smallest value among the weighted fair rates allocated by the nodes along the route and the peak rate constraint of the aggregate flow.

By neglecting the buffer floor, the dynamics of the link buffer of interest is modelled in continuous time by

$$\dot{q}(t) = \sum_{i \in N} a_i(t - \tau_i^f) - \mu \;. \tag{6}$$

The common fair computation in equation (1) can be rewritten in continuous time by

$$f(t) = -C_P\{q(t) - q_T\} - C_I \int_0^t \{q(t) - q_T\} dt \;. \tag{7}$$

Note that $f(t)$ is the common part of per-aggregate weighted fair rate allocations, $f_i^w(t)$, $\forall\, i$, which implies that all the sources bottlenecked at the link are fed with the same feedback rate. Thus no per-aggregate computation is required.

Let Q denote the set of locally-bottlenecked aggregate flows, at a link, containing all those aggregate flows having common fair rate determined at the link in the steady state for a given network loading. In the same way, let $N - Q$ denote the set of remotely-bottlenecked aggregate flows, at a link, containing all those aggregate flows having common fair rate determined at some other link in the path, or having data transfer rate limited by their peak rate constraint. Let

$a_{is} = \lim_{t\to\infty} a_i(t)$, $f_{is}^w = \lim_{t\to\infty} f_i^w(t)$, and $b_{is}^w = \lim_{t\to\infty} b_i^w(t)$. Then Q at the link of interest is given by $Q = \{i | i \in N \text{ and } a_{is} = f_{is}^w\}$ and $N - Q$ at the link of interest is given by $N - Q = \{i | i \in N \text{ and } a_{is} = \min[b_{is}^w, p_i]\}$.

3.2 Steady State and Fairness

Suppose that the closed-loop dynamics have an equilibrium point at which the derivatives of the system variables are zero. Let $f_s = \lim_{t\to\infty} f(t) > 0$. Then, from (3), (5) and (7), we have

$$a_{is} = \min[f_{is}^w,\ b_{is}^w,\ p_i], \quad f_{is}^w = w_i f_s + m_i, \qquad \forall\ i \in N \tag{8}$$

and $q_s = q_T$ where $q_s = \lim_{t\to\infty} q(t)$. Since $q_s = q_T > 0$, the buffer equation (6) implies that

$$\sum_{i\in N} a_{is} = \mu\ . \tag{9}$$

By combining the equations (8), (9), and the definitions of Q and $N - Q$, we obtain

$$f_s = \frac{\mu - \sum_{i\in N-Q} \min[b_{is}^w, p_i] - \sum_{i\in Q} m_i}{|Q|_w} \tag{10}$$

where $|Q|_w$ denote the weighted cardinality of Q, which is the weighted number of locally-bottlenecked aggregate flows, i.e., $|Q|_w = \sum_{i\in Q} w_i$. In addition, let $|Q|$ denote the cardinality of Q, which is the number of locally-bottlenecked aggregate flows. The following theorem summarizes the result.

Theorem 1. *For $\sum_{i\in N} m_i < \mu$ and $\min[b_{is}^w, p_i] \geq m_i$, there exists a unique equilibrium point at which* (i) *the buffer occupancy is equal to the target value ($q_s = q_T$),* (ii) *the capacity at the link is fully utilized ($\sum_{i\in N} a_{is} = \mu$), and* (iii) *individual minimum rates are guaranteed at the link and the unreserved portion of capacity, $\mu - \sum_{i\in N} m_i$, is allocated in the weighted max-min fair sense to the aggregate flows travelling through the link. That is,*

$$a_{is} = \begin{cases} \frac{w_i(\mu - \sum_{i\in N-Q} \min[b_{is}^w, p_i] - \sum_{i\in Q} m_i)}{|Q|_w} + m_i, & i \in Q \\ \min[b_{is}^w, p_i], & i \in N - Q\ . \end{cases} \tag{11}$$

3.3 Asymptotic Stability and Optimal Controller Gain

In this subsection, we study the local stability of the closed-loop system in the neighborhood of the equilibrium point in which the dynamics of the other links are in steady state, i.e., $b_i^w(t) = b_{is}^w,\ \forall\ i \in N$. Thus, (5) can be rewritten by

$$a_i(t) = \begin{cases} f_i^w(t), & i \in Q \\ \min[b_{is}^w, p_i], & i \in N - Q\ . \end{cases} \tag{12}$$

In a link, by combining (3), (6), and (12), we obtain

$$\dot{q}(t) = \sum_{i\in Q} w_i f(t - \tau_i) + \underbrace{\sum_{i\in Q} m_i + \sum_{i\in N-Q} \min[b_{is}^w, p_i] - \mu}_{constant}\ . \tag{13}$$

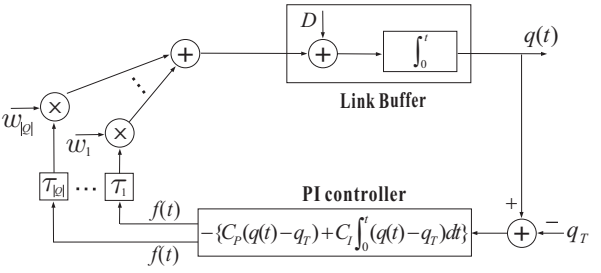

Fig. 2. The closed-loop system model

The constant part in (13) can be considered as an external disturbance, which is denoted by D. By substituting (7) for $f(t-\tau_i)$ in (13), we obtain the following closed-loop equation of the system.

$$\dot{q}(t) = D - \sum_{i \in Q} w_i \left[C_P\{q(t-\tau_i) - q_T\} + C_I \int_0^{t-\tau_i} \{q(t) - q_T\} dt \right] . \tag{14}$$

Fig. 2 depicts this closed-loop system model.

We define the controller gains to be $(C_P, C_I) = (A/|Q|_w, B/|Q|_w)$, where A and B are some positive constants. The open-loop transfer function of the closed-loop system in Fig. 2 is then given by

$$F(s) = \left(\frac{A}{s} + \frac{B}{s^2} \right) \sum_{i \in Q} \rho_i e^{-\tau_i s} \tag{15}$$

where $\rho_i = \frac{w_i}{|Q|_w} \geq 0$, $\forall i \in Q$ and $\sum_{i \in Q} \rho_i \leq 1$.

Now, the sufficient and necessary condition for the asymptotic stability of the closed-loop system is found in an usable form. For the page limitation, the detailed derivations of the theorems and corollary below are given in [16].

First, we consider a single source case, i.e., $|Q| = 1$ with round-trip delay τ and $\rho_1 = 1$ and $\rho_i = 0$, $\forall i > 1$. Note that this case is equivalent to the multiple source case with homogeneous delays τ. By letting $s = j\omega$, the open-loop transfer function becomes

$$F(j\omega) = \left(-\frac{B}{\omega^2} - j\frac{A}{\omega} \right) e^{-j\omega\tau} . \tag{16}$$

Then, by appealing to the Nyquist stability criterion[15], we can find the stability condition for the single source system which is stated in the following theorem.

Theorem 2. *The closed-loop system with a single delay $\tau \geq 0$ is asymptotically stable if and only if the delay is bounded by*

$$0 \leq \tau < \frac{\arccos\left(\frac{B}{\bar{\omega}^2}\right)}{\bar{\omega}} \triangleq \tau^u \tag{17}$$

where $\bar{\omega}$ is a unique $\omega > 0$ such that $F(j\omega) = 1$.

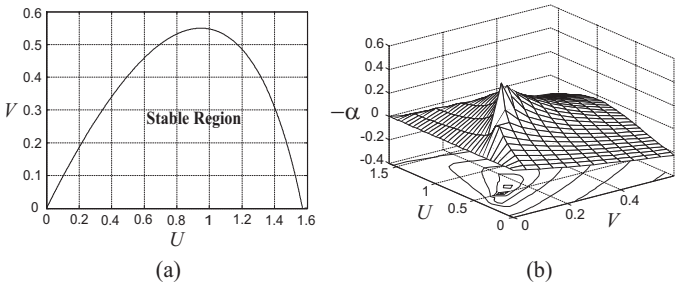

Fig. 3. (a) Stable region (b) Asymptotic decay rate $-\alpha$ as a function of U and V

We have found the upper bound of the round-trip delay for the single source system to be asymptotically stable. It is, however, difficult to apply the stability condition itself (17) to the design of a controller. We modify the condition into an usable form in the following corollary.

Corollary 1. *Let $U = A\tau$ and $V = B\tau^2$. Then the closed-loop system is asymptotically stable if and only if*

$$0 < U < \frac{\pi}{2} \text{ and } 0 < V < \omega_1^2 \cos\omega_1 \tag{18}$$

where ω_1 is the unique solution of $U = \omega \sin\omega$ for $0 < \omega < \pi/2$.

We provide the stable region of U and V in Fig. 3(a). Now, we derived that the stable gain for the case of multiple sources with heterogeneous round-trip delays can be easily found from (18) by applying the theorem below.

Theorem 3. *The closed-loop system with heterogeneous delays is asymptotically stable for all $0 \le \tau_i \le \bar{\tau}$ and for all ρ_i satisfying $\sum_{i\in Q} \rho_i \le 1$ if and only if the closed-loop system of the single-delay case with delay $\bar{\tau}$ is asymptotically stable.*

Consequently, once the upper bound of all the round-trip delays is known, the stable gain for the multiple source system can be obtained from $A = U/\bar{\tau}$ and $B = V/\bar{\tau}^2$ where U and V satisfies (18). In [16], we found the asymptotic decay rate or convergence speed of the closed-loop system numerically. Fig. 3(b) is the result of our numerical approach. The asymptotic decay rate is maximized approximately at $(U, V) = (0.5, 0.1)$. Hence, we can find a stable and optimal controller gain from $(A, B) = (0.5/\bar{\tau}, 0.1/\bar{\tau}^2)$. $\bar{\tau}$ is a possible maximum round-trip delay which can be obtained by off-line measurement in the network domain.

3.4 $|Q|_w$ Estimation

Based on the pair (A,B) found in the above subsection, we can find the controller gain as $(C_P, C_I) = (A/|Q|_w, B/|Q|_w)$ where $|Q|_w$ is obtained through the estimation of weighted number of locally-bottlenecked aggregate flows, $|\hat{Q}|$. We estimate $|Q|_w$ without doing per-aggregate accounting as follows. Suppose that the jth FCP arrives at a link at the link time t^j. If the jth FCP happens to

be a control packet of AF_i, it carries the value $a_i(t^j\text{-}\tau_i^f)$, m_i, and w_i. The link monitors the FCP arrivals in a synchronous fashion over fixed-length intervals of W seconds. For the lth interval, the weighted number of locally-bottlenecked aggregate flows can be estimated by

$$|Q|_w^l = \sum_{t^j \in ((l-1)W, lW]} \frac{N_b + S_{CP}}{W \cdot a_i(t^j - \tau_i^f)} \cdot w_i \cdot 1\{a_i(t^j - \tau_i^f) - m_i \geq \delta \cdot w_i \cdot f(t^j)\}, 0 < \delta < 1 \quad (19)$$

where $1\{\cdot\}$ is the indicator function, S_{CP} is the byte size of a control packet, and $f(t^j)$ is the latest value of the common fair rate at time t^j. Note that all the rates have the values represented in terms of bytes per sec. Here δ is the margin to avoid the underestimation. Based on this estimate for each interval, the recursive estimate is computed at the end of every interval as follows.

$$|\hat{Q}|_w = \text{sat}_1^{|N|_w}[\lambda |\hat{Q}|_w((l-1)W) + (1-\lambda)|Q|_w^l], \quad 0 < \lambda < 1 \quad (20)$$

where λ is an averaging factor and the saturation function ensures that $1 \leq |\hat{Q}|_w(t) \leq |N|_w$ for all t. We choose large λ at a value close to 1 in the hope that the averaging operation in (20) will effectively filter out the variability of $|Q|_w^l$.

In the implementation of $|Q|_w$ estimation, we introduce *virtual packet* concept to remove the impact of different packet size among aggregate flows. We first define a virtual packet with a fixed byte size, then a stream of variable-sized packets can be regarded as a stream of virtual packets. Now, the choice of N_b is obtained as $N_b = N_{CP} \cdot S_{VP}$, where N_{CP} is the number of virtual packets transmitted between two adjacent FCPs and S_{VP} is a virtual packet size.

Table 1. Recommended parameter values in the distributed flow control scheme($\bar{\tau}$=max{τ_i, $i \in N$}, Δ=one virtual packet transmission time)

Common fair rate computation			$\lvert Q\rvert_w$-Estimation				
A	B	T	W	δ	λ	N_{CP}	S_{VP}
$0.5/\bar{\tau}$	$0.1/\bar{\tau}^2$	30Δ	300Δ	0.9	0.98	30	1Kbytes

Table 2. The aggregate flow models and theoretical fair rates in the single bottleneck link configuration

AF#	m_i (Mbps)	w_i	Arrival (sec)	Departure (sec)	Fair rate(Mbps) 0∼25	25∼50	50∼75	75∼∞
AF1	0	1	0	∞	10.63	8.5	6.36	10.63
AF2	5	1.5	0	∞	20.94	17.75	14.54	20.94
AF3	0	2	25	75		17	12.73	
AF4	10	2.5	0	∞	36.56	31.25	25.91	36.56
AF5	0	3	0	∞	31.88	25.5	19.1	31.88
AF6	15	1	50	75			21.36	

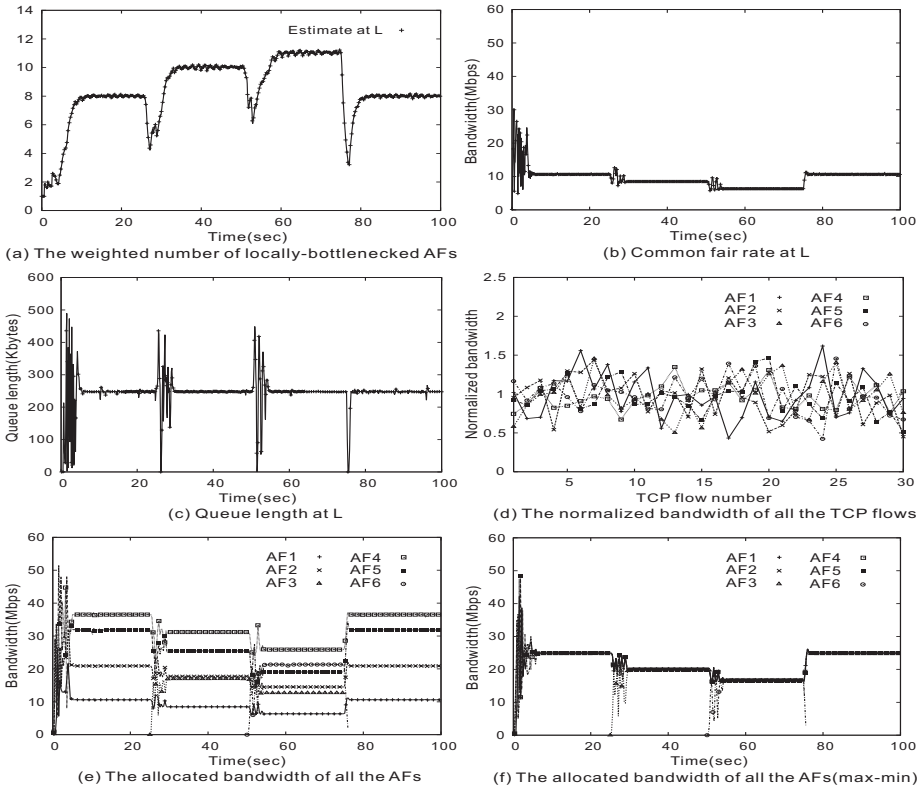

Fig. 4. Results in the single bottleneck link configuration

4 Simulation Results

In this section, we present simulation data to verify and demonstrate the performance of our scheme as described in the previous sections. All the simulation are performed in the *ns*-2[17] environment. We consider the single bottleneck link configuration, where the bottleneck is shared by six ingress-egress pairs and each ingress-egress pair establishes only one virtual path for an aggregate flow. The six aggregate flows have heterogeneous round-trip delays ranging from 20ms to 70ms. Thus the maximum round-trip delay($\bar{\tau}$) between ingress nodes and the egress nodes is about 70 ms. Each aggregate flow consists of 30 persistent TCP flows. In this section, we use simulations to verify and demonstrate the performance of our scheme as described in the previous sections. The simulations are performed in the *ns*-2[17] environment. We consider a typical configuration with a single bottleneck link, termed L, where the bottleneck link is shared by six ingress-egress pairs and each ingress-egress pair has an aggregate flow which consists of 30 persistent TCP flows. The six aggregate flows have heterogeneous round-trip delays ranging from 20ms to 70ms. Thus the maximum round-trip delay($\bar{\tau}$) between ingress nodes and the egress nodes is about 70 ms. In the con-

figuration, the capacity of each outgoing link is equally set to 100Mbps, the propagation delay of the bottleneck link L is set to 10ms, the target queue length(i.e., q_T) of each outgoing link buffer is set to 256Kbytes, and the maximum buffer size of the per-aggregate queue in each edge node is set to 256 Kbytes. All the TCP sources use TCP Reno algorithm and their data packet size is 1Kbytes. In Table 1, we summarize recommended values for simulation parameters in the proposed scheme. The aggregate flow models used in this simulation are summarized in Table 2 and an aggregate flow i is denoted by AFi. The simulation results for weighted max-min fair bandwidth allocation are shown in Fig. 4. For comparison purpose, we have computed the theoretical fair rates for the given simulation scenario based on Proposition 1, and include the results in Table 2. The transmission rate of each aggregate flow in Fig. 4(e) exactly follows the theoretical fair rates given in Table 2 although there is a transient period whenever an aggregate flow arrives or leaves. Fig. 4(b) shows the common fair rate computed by the bottleneck link L. Observe from Fig. 4(e) that the transmission rate of AF1 is equal to the common fair rate since its minimum rate is 0Mbps and its weight is 1. Fig. 4(c) shows that the queue length at the link L always converges to the target value 256Kbytes in steady state. The arrivals of AF3 at 25 sec and AF6 at 50 sec result in the surge of the queue length and the departures of AF3 and AF6 at 75 sec result in the sudden drop of the queue length. The flow control algorithm, however, rapidly recovers the queue length to the target value and restabilizes it at the value. Fig. 4(a) shows the estimate of the weighted number of locally bottlenecked aggregate flows, $|\hat{Q}|_w\ (t)$, at the link L. Fig. 4(d) shows the normalized average throughputs(over a 20sec interval) of all the TCP flows belonging to each aggregate flow. The TCP flows track and share the bandwidth allocated to the aggregate flow in their normal way.

Next, we show that max-min fairness is achieved if all aggregate flows have same weight. In this simulation, each aggregate flow has the weight value of 1 without minimum rate guarantee. Observe from Fig. 4(f) that the bottleneck link capacity is divided equally among the aggregate flows.

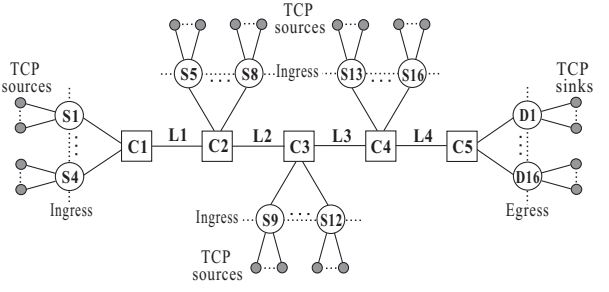

Fig. 5. Multiple bottleneck link configuration

Finally, we study the multiple bottleneck link configuration, shown in Fig. 5. In this simulation, we consider peak rate constraint. Each aggregate flow

Table 3. The aggregate flow models and the theoretical fair rates in the multiple bottleneck link configuration.

AF#	p_i (Mbps)	m_i (Mbps)	w_i	Fair rate (Mbps)	Bottlneck
AF1,AF5,AF9	100	0	1	5	L3
AF13	100	0	1	36.67	L4
AF2,AF6,AF10	100	0	2	10	L3
AF14	100	0	2	73.33	L4
AF3,AF7,AF11	20	10	1	15	L3
AF15	20	10	1	20	p_i
AF4,AF8,AF12,AF16	20	10	3	20	p_i

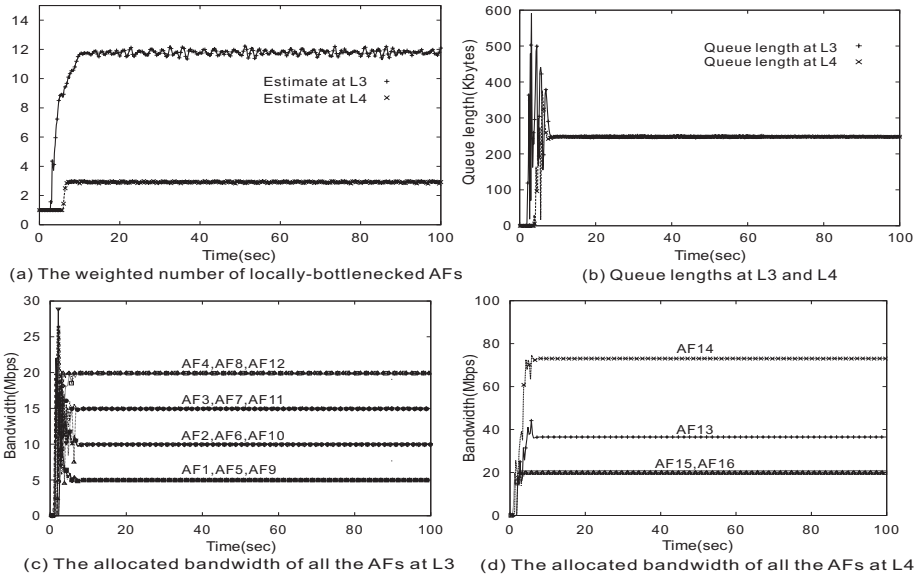

Fig. 6. Results in the multiple bottleneck link configuration

contains 50 TCP flows. 16 aggregate flows with different edge node locations are contained and the capacities of the links between core nodes are set to 300 Mb/s, except that the link between C3 and C4 is 150 Mb/s. The link delays between core nodes are all 10ms and the other link delays are 1ms. All other parameters have the same values used in the previous experiment. The aggregate flow models used in this simulation configuration are summarized in Table 3.

For comparison purpose, we also computed the theoretical fair rates satisfying the weighted max-min fairness with minimum rate guarantee for the given simulation scenario. We also include the theoretical bottleneck location of each aggregate flow in the table, signifying the location at which each fair rate is determined. Fig. 6 shows the simulation results. The actual transmission rate of each aggregate flow in Fig. 6(c) and 6(d) exactly follows the theoretical fair

rates given in Table 3, irrespective of their round-trip delays and the bottleneck locations. Thus weighted max-min fair bandwidth allocation among aggregate flows are achieved in multiple bottleneck links. In the given scenario, there are two congested links, L3 and L4. As expected, the queue length at these congested nodes converges to the target value, 256 Kbytes, which is shown in Fig. 6(b). Fig. 6(a) shows the estimate of the weighted number of locally bottlenecked aggregate flows, $|\hat{Q}|_w\ (t)$, at L3 and L4, respectively. We see that in the steady state the estimate stays around 12 and 3 at L3 and L4, respectively, which agrees with the data in Table 3.

5 Conclusions

In this paper, we propose a distributed flow control scheme for aggregate flows mainly concerning fair bandwidth allocation on a per-aggregate basis. The proposed scheme is simple and highly scalable because its common fair rate computation algorithm does not require any per-aggregate flow state management and operation in the network core.

Mathematical analysis of the proposed scheme concluded that it asymptotically converges to the equilibrium point at which the minimum plus weighted max-min fairness and the convergence of the link buffer occupancy to a target value at every bottlenecked link are achieved which subsequently means that it accomplishes full link utilization and no packet loss at steady state. In addition, we found the asymptotic stability condition of the controller gain in an usable form and the optimal controller gain satisfying the stability condition. Through simulations we verify that our scheme can perform the excellent bandwidth allocation for the aggregate flows based on weighted max-min fairness. We believe that the proposed scheme not only improves the Internet capacity significantly but also gives ISPs an effective tool to engineer traffic inside the network

In ongoing work, we are exploring the quantized common fair rate delivery using standard Explicit Congestion Notification(ECN) framework to remove control packet overhead in our scheme and thus making our algorithm readily implementable in current network status, hence, enabling easy deployment of our scheme.

References

1. Blake, S., Blake, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An Architecute for Differentiated Services. IETF RFC 2475 (Dec. 1997)
2. Braden, R., Zhang, L., Berson, S., Herzog, S., Jamin, S.: Resource Reservation Protocol (RSVP). IETF RFC 2205 (Sept. 1997)
3. Stoica, I., Shenker, S., Zhang, H.: Core-Stateless Fair Queueing: Achieving Approximately Fair Bandwidth Allocations in High-Speed Networks. Proc. ACM SIGCOMM'98 Conference (Sep. 1998) 118-130
4. Harrison, D., Kalyanaraman, S.: Edge-To-Edge Traffic Control for the Internet. Technical Report ECSE-NET-2000-I, RPI ECSE Networks Laboratory (Jan. 2000)

5. Chapman, A., Kung, H.T.: Traffic Management for Aggregate IP Streams. Proc. CCBR'99 (Nov. 1999) 1-9
6. Lee, B.P., Balan, R.K., Jacob, L., Seah, W.K.G., Ananda, A.L.: TCP Tunnels: Avoiding Congestion Collapse. Proc. IEEE LCN'00 (Nov. 2000) 408-417
7. Nandy, B., Ethridge, J., Lakas, A., Chapman, A.: Aggregate Flow Control: Improving Assurances for Differentiated Services Network. Proc. IEEE INFOCOM'01 (Apr. 2001) 1340-1349
8. Pradhan, P., Chiueh, T., Neogi, A.: Aggregate TCP Congestion Control Using Multiple Network Probing. Proc. ICDCS'00 (Apr. 2000) 30-37
9. Benmohamed, L., Meerkov, S.M.: Feedback Control of Congestion in Packet Switching Networks: The Case of Single Congested Node. IEEE/ACM Trans. on Networking, Vol. 1. (Dec. 1993) 693-708
10. Charny, A., Clark, D., Jain, R.: Congestion Control with Explicit Rate Indication. Proc. IEEE ICC'95 (June 1995) 1954-1963
11. Kolarov, A., Ramamurthy, G.: A Control Theoretic Approach to The Design of Closed Loop Rate Based Flow Control for High Speed ATM Networks. IEEE INFOCOM'97 (Apr. 1997) 293-301
12. Hou, Y.T., Tzeng, H., Panwar S.S., Kumar, V.P.: A Generic Weight-Proportional Bandwidth Sharing Policy for ATM ABR Service. Performance Evaluation, Vol. 38. (Sep. 1999) 21-44
13. Aström, K.J., Wittenmark, B.: Computer Controlled Systems: Theory and Design. Prentice-Hall, Englewood Cliffs, New Jersey (1984)
14. Barmish, B.R.: New Tools for Robustness of Linear Systems. MacMillan (1994)
15. Franklin, G.F., Powell, J.D., Workman, M.L.: Digital Control Systems. Addison-Wesley, Reading, MA, New York (1990)
16. Ryu, H.K., Cho, J.W., Chong, S.: Stabilized Edge-to-Edge Aggregate Flow Control (extended version). http://netsys.kaist.ac.kr/ ∼hkryu/networking2004.pdf
17. ns-2, Network Simulator (ver-2), Available at http://www.isi.edu/nsnam/ns/