Edge-to-Edge Aggregate Flow Control for the Internet

Hyung-Keun Ryu
Dept. EECS, KAIST
hkryu@netsys.kaist.ac.kr

Jeong-woo Cho
Dept. EECS, KAIST
ggumdol@netsys.kaist.ac.kr

Song Chong
Dept. EECS, KAIST
song@ee.kaist.ac.kr

ABSTRACT

We present an edge-to-edge flow control scheme which enables an ISP to achieve 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. The proposed flow control scheme is hierarchical in that 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 requires neither modification nor replacement of TCP congestion control. The distributed algorithm to compute weighted max-min fair rates is based on PI control. It is *scalable* in that the computational complexity imposed on each link is $O(1)$, i.e., independent of number of aggregate flows travelling through it, *stable* in that it converges asymptotically to the desired equilibrium satisfying the minimum plus weighted max-min fairness, and has explicit *link buffer control* in that the buffer occupancy of every bottlenecked link in the network asymptotically converges to the pre-defined value. By appealing to the Nyquist stability criterion, we mathematically prove the asymptotic stability of the algorithm in presence of aggregate flows with different round-trip delays. Through extensive simulations we demonstrate the effectiveness of the proposed scheme in controlling per-aggregate flows.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Data communications*; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Distributed networks, Network communications*

General Terms

Algorithms, Design, Performance

Keywords

Edge-to-edge mechanism, weighted max-min fairness, aggregate flow control, link buffer control

1. INTRODUCTION

As the current Internet is becoming omnipresent infrastructure with the wide spread of both business and residential Internet access and experiencing a commercially rapid growth despite its simple best-effort service, Internet service providers(ISPs) feel strong needs for Internet qualities of service(QoS) to provide their customers with more enhanced and differentiated services than the conventional best-effort service without any sort of guarantees. Accordingly, providing QoS, especially service differentiation, for the Internet has been an active research theme in recent years.

In this paper, we propose a new edge-to-edge aggregate flow control scheme in order to support different levels of service as a basis of the Intenet QoS. Basically, It provides relative QoS for aggregate traffic streams or aggregate flows rather than individual flows. An aggregate flow consists of many individual flows initiated by customers' hosts and servers. Fundamentally, it is an aggregation of IP traffic streams that are grouped together for the same routing and service differentiation between two edge nodes in a network. For example, It is the entire traffic sent or received by all users of a customer network, such as at a campus network or corporate network. Once the aggregation is done, the proposed scheme treats the aggregate flow as a single flow within the network. It is important for ISPs to provide different levels of service and provide contracted QoS for aggregates, as their QoS commitments to customers are likely to be at the aggregate level rather than for individual flows. To support such differentiated services, the proposed scheme can handle multiple aggregates through flow classification, per-aggregate queueing, and per-aggregate rate control at the network edges.

In the proposed scheme, only edge nodes operate state management and flow handling at aggregate level, but the network core doesn't any of them. In other words, our scheme implements stateless-core approach, like [16, 14], in that no per-aggregate flow state is maintained at the network core. Besides, it places only simple functionality within the network core, with more complex operations being implemented at the edge of the network. It also pushes the interior network congestion out to the network edges, and then QoS issues such as queueing, bandwidth sharing, packet loss, packet delay are all carried to the edge, allowing the network core to operate using stateless mechanisms. It reduces the number of unnecessary retransmission in the user TCP flows and avoid congestion collapse in the network.

In this paper, we define the problem of aggregate flow control as how to allocate available bottleneck bandwidth shared by multiple aggregate flows to each of them based on the fairness scheme. Naturally, a hierarchical flow control scheme is achieved because the aggregate flow control determines the bandwidth allocated to an aggregate, and then end-to-end user flows running congestion control algorithm(e.g., TCP flows) at the end systems are assigned their fair bandwidth shares within the given link bandwidth. The aggregate flow control scheme can be effectively applied to ISPs' network and corporate network at the first stage of QoS provision.

The proposed scheme adopts the rate-based closed loop control approach for the aggregate flow control. Edge-to-edge closed loops are established between ingress edges and egress edges in a network, and then *virtual links* (VLs) are set up by the closed loops. A VL is regarded as a dedicated circuit or path such as an ATM virtual channel(VC), a Frame Relay permanent virtual circuit(PVC), a MPLS label switched path(LSP), etc. It can also be a routing path determined by the routing algorithm between pair-edges in that it is not likely to change frequently. A VL transports data packets at a rate dynamically determined by the flow control algorithm. Certainly, it is possible to set up multiple VLs within an edge, and those VLs can be associated with different egress edges. The VL is elastic in the sense that the data transfer rate is adjusted depending on the available bandwidth at the network. The rate-based closed-loop control uses feedback information from the network to control the transmission rate of each VL. The feedback information from the network is an explicit rate carried by special control packets.

As the core part of the proposed scheme, we present a simple, scalable, and stable explicit rate based flow control algorithm for the weighted max-min flow control of elastic traffic services with minimum rate guarantee. The proposed algorithm is simple in that the number of operations required to compute ER algorithm at a node is minimized, scalable in that per-aggregate operations including per-aggregate queueing, per-aggregate accounting, and per-aggregate state management are virtually removed, and stable in that by employing it, the transmission rates of VLs and network queues are asymptotically stabilized at a unique equilibrium point at which weighted max-min fairness with minimum rate guarantee and target queue lengths are achieved.

The rest of this paper is organized as follows. Section 2 presents the overall scheme for aggregate flow control. Section 3 describes the explicit rate flow control algorithm. A simulation study and analysis of the proposed scheme are presented in Section 4. The deploy issues and further work are discussed in Section 5. Finally, we conclude the paper with section 6.

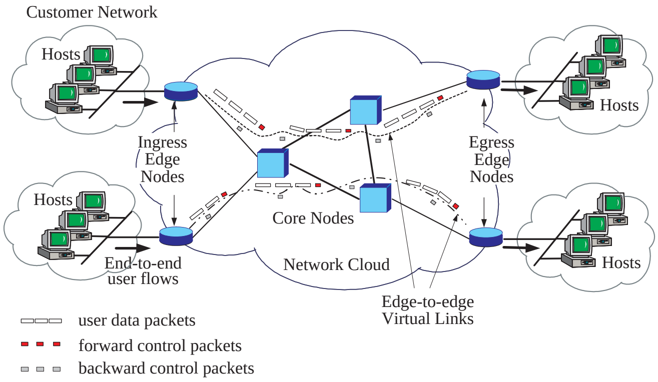


Figure 1: Overview of edge-to-edge aggregate flow control scheme.

2. OVERALL FLOW CONTROL SCHEME

In this section, we describe overall edge-to-edge aggregate flow control scheme in detail. The Internet can be thought of as a concatenation of heterogeneous network clouds. We assume our scheme is applied to a network cloud which consists of ingress and egress edge nodes at the boundary of the network and core nodes at the network interior. For example, as illustrated in Figure 1, our scheme is deployed to the ISP's network which provides different levels of service to customer networks, which include many hosts such as servers and clients for file transfer, HTTP, and streaming. Another example is a network service provider(NSP)'s network which provide contracted transport service to their ISPs. In both cases, the guaranteed and fair bandwidth allocation may be very significant issues. We believe our scheme is appropriate to solve the problem.

The ingress edge nodes aggregate end-to-end user flows, such as TCP and UDP flows, having the same ingress-egress edge pair into a separate VL according to ISP's service policy. An ingress edge node maintains multiple VLs in connection with several egress edge nodes. Over these VLs, *control packets* are transmitted by the ingress edge nodes into the network together with user data packets and travel along the forward path down to the egress edge nodes and then are returned to the ingress edge node along the backward path, which may not be same with forward path. *Forward control packets* are those flowing from the ingress edge node to the egress edge node while *backward control packets* are those returning from the egress edge node to the ingress edge node.

2.1 Node Functional Architecture

Figure 2 illustrates the functional architecture of an edge node handling multiple aggregate flows(AFs), and a core node, respectively. In the architecture, all the nodes use FIFO queueing and drop-tail packet discard policy, which are most commonly used in the Internet. According to some set of pre-defined rules, incoming user flows are classified into different aggregate flows based on the contents of their packet header, which include source and destination address, protocol ID, source and destination port numbers, and other information such as input interface. The aggregate flow is mapped onto a VL. The classified user flows are multiplexed to be a single aggregate flow and then wait for forwarding into an output queue in the per-aggregate queue, which is an input queue at the input port.

The Rate Control Unit(RCU) regulates the transmission

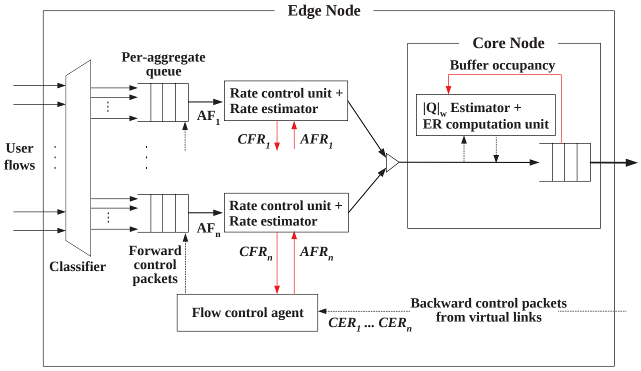


Figure 2: The functional architecture of an edge node, and a core node, respectively.

rate of the aggregate flow using the allowed flow rate(AFR) value. The AFR is the rate at which a VL is allowed to send. The regulated aggregate flow are directly forwarded into an output queue via the switching fabric and transmitted into the network without any loss and delay if the output link is not a bottleneck. In practice, multiple aggregate flows may be forwarded into an output queue. It is possible for their bottlenecks to be different because of their different routing paths or different destination nodes. If the incoming traffic from all the aggregate flow arrives at a higher rate than the output link capacity, the output link is source-bottlenecked, what we call. The explicit rate flow control algorithm is employed in the output queue of the edge node to solve the problem.

The Rate Estimator measures the actual output rate of RCU, what we call current flow rate(CFR). we use exponential averaging to estimate the rate. The estimated rate is updated periodically. The FCA generate a control packet each time NCP bytes of Data is transmitted, where we define NCP as the bytes number of data transmitted between two adjacent control packets, and determines the AFR value using the common explicit rate(CER) value carried by backward control packet, as detailed in Section 2.2.

In the core node, the explicit rate flow control algorithm is implemented at the output port, like the edge node. It consists of two functionality, a ER Computation Unit(ECU) and a $|Q|_w$ estimator. Their algorithms are described in detail in Section 3.5 and Section 3.6, respectively.

The ECU periodically computes the common explicit rate. The ECU intercept a control packet from the incoming traffic and updates the control packet's CER field. All the control packet is updated by the same CER value no matter which aggregate flow each of them belongs to. The $|Q|_w$ estimator estimates at the periodic time interval the weighted number of locally-bottlenecked VLs on a bottleneck link, which is used by ECU for calculating CER. $|Q|_w$ estimator and ECU can be easily added to the nodes.

2.2 Weighted max-min Fair Bandwidth Allocation

In the bandwidth allocation scheme, fairness is always important issue. max-min fairness is one of the most well-known concept and discussed in many literatures. Fairness with minimum rate guarantee was discussed in [9]. Weighted max-min fairness is a generalized extension of max-min fairness. Recently, the weighted max-min fair bandwidth allocation was studied by several research works [18, 19, 8] and employed in the QoS architecute researches[14, 16].

In our scheme, the available bandwidth of a bottleneck is allocated fairly among the competing aggregate flows based on the weighted max-min fairness, which enables ISPs to provides their customer with fine-grained service differentiation. More specifically, the minimum guaranteed rates(MGRs) of all the aggregate flows passing through a bottleneck link are guaranteed and the excess bottleneck bandwidth, excepting the sum of all the MGRs but including the unused bandwidth, is distributed in an weighted max-min sense. The fair share of each aggregate flow is the sum of a MGR and the weighted portion of the excess bottleneck bandwidth.

Each aggregate flow has an weight associated with its class of service and the excess bottleneck bandwidth proportional to their weights is distributed among competing flows. We don't place limits on the range of the discrete weights that can be supported. If all the weights are equally assigned(e.g., all 1s), max-min fair bandwidth allocation is achieved.

The weighted max-min fair bandwidth allocation is simply obtained by regulating the transmission rate of each VL using its AFR. Upon receipt of backward control packet at a time t, the common explicit rate from the network is notified and AFR for VL i is computed as follows.

$$AFR_i(t) = min[PFR_i, w_iCER_i(t) + MGR_i] \qquad (1)$$

where w_i denotes the predetermined weight of the VL i, CER_i denotes the value in the CER field of the control packet, MGR_i and PFR_i respectively denote the minimum guaranteed rate and the peak flow rate(PFR) of the VL i. The PFR is the maximum rate at which the VL is allowed to send.

By the above simple source operation together with the next explicit flow control algorithm, we can make sure that the rates of all the VLs converge to the weighted max-min fair bandwidth allocation. Concurrently, the bottleneck queues are stabilized at the target queue length and there is no drop in the core nodes.

3. EXPLICIT RATE FLOW CONTROL ALGORITHM

In this section, we present the distributed algorithm to compute common explicit rate, which is the core part of the our scheme. We construct a new common ER allocation algorithm on a solid analytical basis and the novelty of our proposed algorithm is an explicit control of both rate and queue dynamics.

Benmohamed and Meerkov in their pioneering work [4] formulated the rate-based flow control problem as a discrete-time feedback control problem with delays. Based on this formulation, they derived a control-theoretic ER allocation algorithm which not only achieves asymptotic stability of the closed-loop system but also allows for arbitrary control of the closed-loop performance. Its complete controllability of the closed-loop performance, however, comes at a high cost. That is, it requires long memory of the queue lengths and the ER values at present and in the past, and requires a large number of floating point multiplications every discrete time slot. Therefore, its practical use is limited as the round-trip delay increases [12].

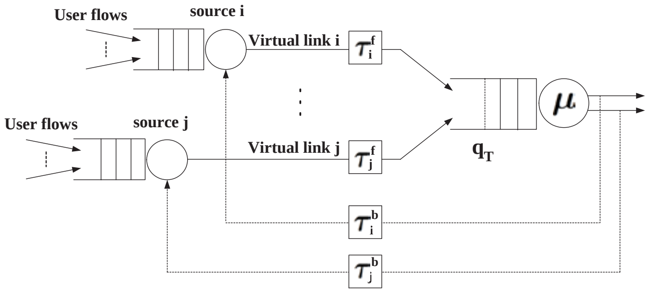


Figure 3: Network model with a bottleneck node of interest.

We take a different approach. We aim to design an ER allocation algorithm which allows for low degree of implementation complexity but with an *acceptable* level of control rather than arbitrary control for the closed-loop performance. More specifically, we trade off the capability of arbitrary control of the closed-loop performance for low degree of implementation complexity by removing the long memory of past queue lengths and ER values. Our proposed discrete-time algorithm is as follows.

$$r[k+1] = r[k] - \frac{A}{|Q|_w}(q[k]-q[k-1]) - \frac{BT}{|Q|_w}(q[k]-q_T) \quad (2)$$

where A and B are positive controller gains, T is the duration of update interval, Q denotes the set of locally-bottlenecked VLs at the bottleneck link and $|Q|_w$ is the weighted cardinality of Q, where we define the weighted cardinality as the term referring to the weighted total number of elements, or members, in a set. The weighted cardinality of Q is the weighted total number of locally-bottlenecked VLs. It means that, for example, A locally-bottlenecked VL having an weight of 2 can be regarded as two links having normalized fair share in MAX-MIN fair sense. If all the VLs have a weight value of 1, $|Q|_w$ is the number of locally-bottlenecked VLs.

By the term an acceptable level of control, we mean that by properly choosing the controller gains for a given round-trip delays, one can completely control the *asymptotic* behavior of the closed-loop system. An explicit condition to achieve this level of control is given in the paper. On the other hand, the available bottleneck link bandwidth has to be allocated in the weighted max-min fair sense to the individual VL. It is shown that this happens automatically in the steady state by virtue of the equation (2). Another notable feature of the proposed algorithm is the normalization of the controller gains by the number of locally-bottlenecked VL's $|Q|_w$. We show that this normalization is indeed beneficial such a way that it makes the closed-loop performance to be virtually independent of the weighted number of locally-bottlenecked VLs on a bottleneck link.

3.1 Fluid Network Model

First, consider a network model in Figure 3 where we model a single bottleneck node explicitly and the other nodes implicitly to simplify the analysis. We then use fluid flow analysis which is fairly standard [17].

Assume that the round-trip delay, τ_i, of a VL i, which is the sum of forward-path delay τ_i^f and backward-path delay τ_i^b, is constant and the sources of VLs 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 infinite.

Let $a_i(t)$, $r(t)$, $b(t)$, and p_i respectively denote the rate at which VL i transmits data at the source time t, the common explicit *rate* of VL i computed by the node of interest at the node time t, the latest minimum value of the common explicit *rates* allocated to VL i by the nodes along the VL i's path except the one allocated by the bottleneck node of interest, and the peak rate constraint of VL i (i.e., PFR of VL i). Also, let $r_i^w(t)$ and $b_i^w(t)$ respectively denote the weighted explicit rate of VL i which are computed by an ingress edge node at the node time t as follows:

$$r_i^w(t) = w_i r(t) + m_i, \quad \forall\ i \in N \quad (3)$$

and

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

where w_i and m_i denote a weight value and the minimum data rate which VL i guarantees(i.e., MGR), respectively. The value of w_i and m_i is available from either the control packet being arrived or the MGR table being maintained in the node, depending on the implementation.

The source behavior of VL i can be modeled by

$$a_i(t) = \min[\ r_i^w(t-\tau_i^b),\ b_i^w(t),\ p_i\ ], \quad \forall\ i \in N \quad (5)$$

where N denotes the set of all the VLs whose route includes the bottleneck node of interest. This model implies that a VL transmits data at the smallest value among the weighted explicit rates allocated by the nodes along the route and the PFR of the VL.

The dynamics of the bottleneck queue of interest are given by

$$\dot{q}(t) = \begin{cases} \sum_{i\in N} a_i(t-\tau_i^f) - \mu, & q(t) > 0 \\ [\ \sum_{i\in N} a_i(t-\tau_i^f) - \mu\ ]^+, & q(t) = 0. \end{cases} \quad (6)$$

where $[\cdot]^+ = \max[\cdot, 0]$.

The proposed ER allocation algorithm is a distributed algorithm which runs independently and identically at each node based on the current network state including the queue length, $q(t)$, the derivative of the queue length, $\dot{q}(t)$, and the estimate of the weighted number of locally-bottlenecked VLs, $|\hat{Q}|_w$. The algorithm is given by the following equations in continuous time.

$$\dot{r}(t) = \begin{cases} -\frac{A}{|\hat{Q}|_w}\dot{q}(t) - \frac{B}{|\hat{Q}|_w}(\ q(t) - q_T\ ), & r(t) > 0 \\ [\ -\frac{A}{|\hat{Q}|_w}\dot{q}(t) - \frac{B}{|\hat{Q}|_w}(\ q(t) - q_T\ )\ ]^+, & r(t) = 0 \end{cases} \quad (7)$$

where $A,\ B > 0$. We assume that $m_i \le p_i,\ \forall\ i \in N$ and there exists an administration which guarantees $\sum_{i\in N} m_i < \mu$. Note that $r(t)$ is the common part of per-VL ER allocations, $r_i(t),\ \forall\ i$, and no per-VL computation is required. This is why this algorithm is scalable in terms of computational complexity with increasing number of VLs.

A notable feature of the proposed algorithm is the normalization of the controller gains, A and B, by the estimate of the weighted number of locally-bottlenecked VLs, $|\hat{Q}|_w$. This normalization is optional, i.e., it is not absolutely necessary but it is recommended since, as will be discussed in Section 3.4, it makes the closed-loop dynamics to be virtually independent of the weighted number of locally-bottlenecked VLs on the bottleneck link.

The terms, remotely-bottlenecked VL and locally-bottlenecked VL, are defined in the steady state for a given network loading. Locally-bottlenecked VLs at a bottleneck link are defined to be those VLs whose fair share is determined at this link. In the same way, remotely-bottlenecked VLs at a bottleneck link are defined to be those VLs whose fair share is determined at other places because either their data transfer rate is limited by their PFR or they are bottlenecked at some other link in the path. Let $a_{is} = \lim_{t\to\infty} a_i(t)$, $r^w_{is} = \lim_{t\to\infty} r^w_i(t)$ and $b^w_{is} = \lim_{t\to\infty} b^w_i(t)$. Then, more formally, the set of all the locally-bottlenecked VLs, Q, at the bottleneck link of interest is given by $Q = \{i | i \in N \text{ and } a_{is} = r^w_{is}\}$ and the set of all the remotely-bottlenecked VLs, $N - Q$, at the bottleneck link of interest is given by $N - Q = \{i | i \in N \text{ and } a_{is} = \min[b^w_{is}, p_i]\}$.

3.2 Steady State and Fairness

In this section we study the steady state characteristics of the closed-loop dynamics when our ER allocation algorithm is applied. Suppose that the closed-loop dynamics have an equilibrium point at which the derivatives of the system variables are zero, i.e., $\lim_{t\to\infty} \dot{q}(t) = 0$ and $\lim_{t\to\infty} \dot{r}(t) = 0$. Let $r_s = \lim_{t\to\infty} r(t) > 0$. Then, from (3), (5) and (7), we have

$$a_{is} = \min[r^w_{is},\ b^w_{is},\ p_i], \quad r^w_{is} = w_i r_s + m_i, \qquad \forall\ i \in N, \quad (8)$$

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

$$\sum_{i\in N} a_{is} = \mu. \quad (9)$$

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

$$\sum_{i\in Q} w_i r_s + \sum_{i\in Q} m_i + \sum_{i\in N-Q} \min[b^w_{is}, p_i] = \mu \quad (10)$$

which implies that

$$r_s = \frac{\mu - \sum_{i\in N-Q} \min[b^w_{is}, p_i] - \sum_{i\in Q} m_i}{|Q|_w}. \quad (11)$$

The following proposition states the result.

Proposition 3.1. *For $\sum_{i\in N} m_i < \mu$ and $\min[b^w_{is}, p_i] \geq m_i$, there exists a unique steady state solution (equilibrium point) at which* (i) *the queue length is equal to the target queue length ($q_s = q_T$),* (ii) *the available bandwidth at the link is fully utilized ($\sum_{i\in N} a_{is} = \mu$), and* (iii) *individual MGRs are guaranteed at the link and the bandwidth subtracted by the sum of MGRs, $\mu - \sum_{i\in N} m_i$, is allocated in the weighted max-min fair sense to the individual sources. That is,*

$$a_{is} = \begin{cases} \frac{w_i(\mu - \sum_{i\in N-Q} \min[b^w_{is}, p_i] - \sum_{i\in Q} m_i)}{|Q|_w} + m_i, & i \in Q \\ \min[b^w_{is}, p_i], & i \in N - Q. \end{cases} \quad (12)$$

This proposition implies that when our ER allocation algorithm is applied, the closed-loop system has a unique equilibrium point at which the weighted max-min fairness with MGR guarantee is achieved and the queue length is equal to the target value q_T no matter what the network loading is.

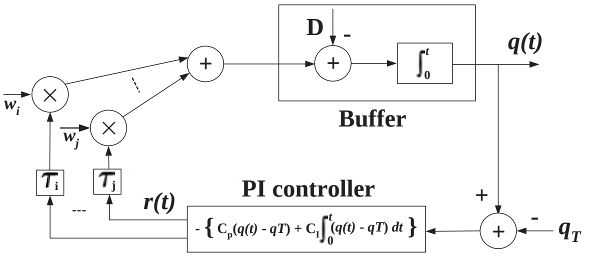


Figure 4: The system model in the neighborhood R.

3.3 Asymptotic Stability

Suppose that there exists a neighborhood of the equilibrium point in which the followings are satisfied: a) $b(t) = b_s$ and $b^w_i(t) = b^w_{is}$, $\forall\ i \in N$, i.e., the dynamics of the other nodes are in steady state; b) $\{i | i \in N \text{ and } a_i(t) = r^w_i(t - \tau^b_i)\} = Q$ and $\{i | i \in N \text{ and } a_i(t) = \min[b^w_{is}, p_i]\} = N - Q$, i.e., the locally-bottlenecked VLs transmit data at $r^w_i(t - \tau^b_i)$ and the remotely-bottlenecked VLs transmit data at $\min[b^w_{is}, p_i]$; c) the saturation nonlinearities in (6) and (7) are not activated, i.e., both $q(t)$ and $r(t)$ are positive-valued; d) the $|Q|_w$-estimation process is in steady state, i.e., $\hat{|Q|}_w$ is constant. Then, in this neighborhood, we can simplify the dynamic equations (5), (6) and (7) as follows.

$$a_i(t) = \begin{cases} r^w_i(t - \tau^b_i), & i \in Q \\ \min[b^w_{is}, p_i], & i \in N - Q, \end{cases} \quad (13)$$

$$\dot{q}(t) = \sum_{i\in N} a_i(t - \tau^f_i) - \mu \quad (14)$$

and

$$\dot{r}(t) = -\frac{A}{\hat{|Q|}_w}\dot{q}(t) - \frac{B}{\hat{|Q|}_w}(\ q(t) - q_T\ ). \quad (15)$$

By combining (13) and (14), we obtain

$$\dot{q}(t) = \sum_{i\in N-Q} \min[b^w_{is}, p_i] + \sum_{i\in Q} r^w_i(t - \tau_i) - \mu. \quad (16)$$

Define an error function by $e(t) = q(t) - q_T$. By combining (15), the differentiation of (16) and the differentiation of (3), we obtain the following closed-loop equation

$$\ddot{e}(t) + \frac{A}{\hat{|Q|}_w}\sum_{i\in Q} w_i \dot{e}(t - \tau_i) + \frac{B}{\hat{|Q|}_w}\sum_{i\in Q} w_i e(t - \tau_i) = 0 \quad (17)$$

which is a second-order retarded differential equation. The characteristic equation of the closed-loop equation is given by

$$s^2 + \frac{A}{\hat{|Q|}_w}\sum_{i\in Q} w_i s e^{-s\tau_i} + \frac{B}{\hat{|Q|}_w}\sum_{i\in Q} w_i e^{-s\tau_i} = 0 \quad (18)$$

which has infinite number of roots. For the asymptotic stability of the closed-loop equation (17), all the roots of the characteristic equation (18) must have negative real parts [3]. It is possible to find a necessary and sufficient condition for exponential polynomials to have stable roots [15]. However, deriving an explicit form of such a condition is extremely complicated especially for the case with a large number of heterogeneous delays.

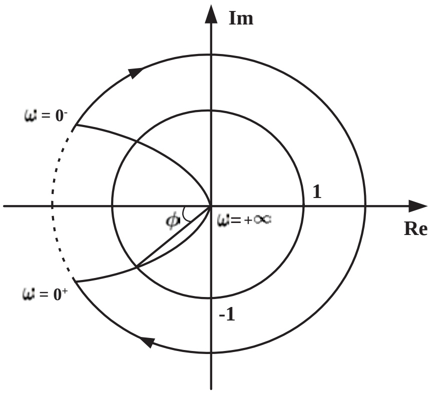


Figure 5: Nyquist plot of $G(j\omega)$.

On the other hand, by using Nyquist stability criterion, Blanchini et al. have found an explicit form of the stability condition of the discrete-time closed-loop system which involves PID controller used for congestion control and has heterogeneous round-trip delays [5]. They showed that the closed-loop system with a single source is stable if and only if the round-trip delay is strictly bounded by a known quantity which can be obtained easily. For the case of multiple sources, they proved that the closed-loop system is stable if and only if the single source case is stable while the round-trip delay is equal to the known quantity. Nevertheless, the condition they found is hard to be applied for the controller gains directly. By appealing to this result, we find the asymptotic stability condition of the continuous-time closed-loop system in an easily usable form.

Figure 4 shows the closed-loop system model associated with a link in the neighborhood R, where (14) can be written as

$$\dot{q}(t) = \sum_{i\in Q} w_i r(t-\tau_i) + \underbrace{\sum_{i\in Q} m_i + \sum_{i\in N-Q} \min[b^w_{is}, p_i] - \mu}_{constant} \quad (19)$$

The constant terms in (19) can be considered as a disturbance, so we integrated them into the constant disturbance D. For a single source case, the open-loop transfer function is given by

$$F(s) = \underbrace{\left(\frac{C_P}{s} + \frac{C_I}{s^2}\right)}_{G(s)} w\, e^{-\tau s} \quad (20)$$

In our work, $C_I = B/|\hat{Q}|_w$ and $C_P = A/|\hat{Q}|_w$, but we can set $C_I = B$ and $C_P = A$ in single source case. Letting $s = j\omega$, we obtain

$$F(j\omega) = \left(-\frac{B}{\omega^2} - j\frac{A}{\omega}\right) we^{-j\omega\tau} \quad (21)$$

The Nyquist plot of $G(j\omega)$ is shown in Figure 5. It describes a parabola in $\omega \in (0, +\infty)$. By inserting the small semicircular detour $\{s = \epsilon e^{j\alpha}, -\pi/2 \le \alpha \le +\pi/2\}$ around the pole at the origin, we obtain an corresponding infinite circle whose phase changes from $+\pi$ to $-\pi$.

Obviously, the Nyquist plot of $F(j\omega)$ can be obtained if $G(j\omega)$ is rotated by $\omega\tau$ in the clockwise direction. Thus the Nyquist stability criterion requires $\bar{\omega}\tau < \phi$ where $\bar{\omega}$ is $\omega > 0$ such that $|G(j\omega)| = 1$(the point P in Figure 5) and $\phi = \arccos(-Re[G(j\bar{\omega})])$. The condition is more formally stated in the following proposition.

PROPOSITION 3.2. *The closed-loop system is asymptotically stable if and only if the delay is bounded by*

$$0 \le \tau < \frac{\arccos\left(\frac{Bw}{\bar{\omega}^2}\right)}{\bar{\omega}} = \tau_{max} \quad (22)$$

PROOF. *Sufficiency: We can easily see from Figure 5 that the Nyquist plot has a unique intersection with the unit circle in $0 < \omega < \infty$. More formally, we can show that there exists a unique $\bar{\omega} > 0$ as follows.*

$$|F(j\omega)|^2 = |G(j\omega)|^2 = \left(\frac{Bw}{\omega^2}\right)^2 + \left(\frac{Aw}{\omega}\right)^2 = 1 \quad (23)$$

which is equivalent to

$$\omega^4 - (Aw)^2\omega^2 - (Bw)^2 = 0$$

Since we assume the positive controller gains, the above equation has a unique solution $\bar{\omega} > 0$.

This result means that once the Nyquist plot starting from $-\infty^2 - j\infty$ goes into the unit circle, it never leaves it. Therefore, if the delay is bounded by τ_{max}, the Nyquist plot of $F(j\omega)$ does not encircle the point $-1 + j0$, which means the closed-loop system is asymptotically stable.

Necessity: It can be easily shown that if $\tau \ge \tau_{max}$, the Nyquist plot of $F(j\omega)$ touches or encircles the point $-1+j0$. Thus if the closed-loop system is asymptotically stable, the delay is bounded as shown in (22). □

However, it is difficult to apply the delay bound condition itself (22) to the design of a controller. We rearrange the condition into an easily usable form in the following corollary.

COROLLARY 3.1. *Let $U = A\tau$ and $V = B\tau^2$. Then the closed-loop system is asymptotically stable if and only if*

$$0 < U < \pi/2 \text{ and } 0 < V < \omega_1^2 \cos\omega_1 \quad (24)$$

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

PROOF. *See the Appendix.* □

For better understanding, we provide the stable region of U and V in Figure 6.

We have found the asymptotic stability condition for the case of a single source. As mentioned above, it is not a simple problem to find an explicit form of stability condition for the multiple source system with heterogeneous round-trip delays. However, we show that the stable gain for the case of multiple sources can be easily found from (24) by appealing to the proposition below.

PROPOSITION 3.3. *Consider the closed-loop system with multiple sources in which all the delays are bounded by a maximum delay $0 \le \tau_i \le \tau_{max}, \forall i \in Q$. Then, the controller gain (A, B) is a stable gain of the multiple source system if and only if the controller gain stabilizes the single source system with $\tau = \tau_{max}$.*

PROOF. *See the Appendix.* □

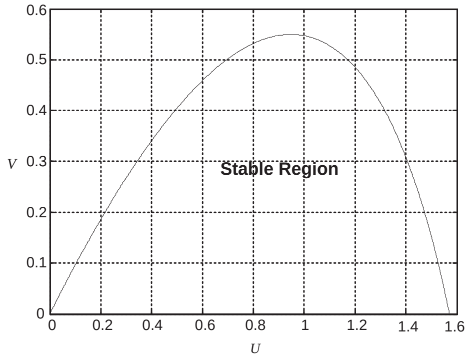


Figure 6: Stable region with respect to U and V

Accordingly, once the maximum of all the round-trip delays τ_{max} is known, the stable gain for the multiple source system can be obtained from $A = U/\tau_{max}$ and $B = V/\tau_{max}^2$ where U and V satisfies (24).

3.4 Principal Root and Asymptotic Decay Rate

In the stable region depicted in Figure 6, we find (U, V) at which the asymptotic decay rate or the convergence speed of the closed-loop system is maximized. The asymptotic decay rate is dominated by the principal eigenvalue which has the largest real part of the system poles. Thus, the maximum asymptotic decay rate is achieved when the real part of the principal eigenvalue is minimized. The characteristic equation of the closed-loop system in Figure 4 is given by

$$s^2 + \left(\frac{A}{|\hat{Q}|_w} s + \frac{B}{|\hat{Q}|_w} \right) \sum_{i \in Q} w_i e^{\tau_i s} = 0 \tag{25}$$

It is too complex to find and minimize the real part of the principal root of (25) since it has heterogeneous delays. Instead of investigating (25), we analyze the homogeneous delay system, i.e., the system with $\tau_i = \tau, \forall i$. Although this can cause an error in the optimization, the gain (A, B) is still a stable gain if $\tau \leq \tau_{max}$ ($\tau_{max} = \max_{i \in Q} \tau_i$) because it stabilizes the system with $0 \leq \tau_i \leq \tau_{max}, \forall i \in Q$. Hence, setting $\tau_i = \tau$ is acceptable. Moreover, if $|\hat{Q}|_w \approx |Q|_w$ and s is replaced by s/τ in (25), we obtain

$$s^2 e^s + Us + V = 0 \tag{26}$$

where $U = A\tau$ and $V = B\tau^2$.

We find the asymptotic decay rate numerically. Suppose that $s = \alpha + j\beta, (\alpha < 0)$ is a root of (26), then inserting $s = \alpha + j\beta$ into (26) yields

$$\begin{array}{l} e^{\alpha}\{(\alpha^2 - \beta^2)\cos\beta - 2\alpha\beta\sin\beta\} + U\alpha + V = 0 \\ e^{\alpha}\{(\alpha^2 - \beta^2)\sin\beta + 2\alpha\beta\cos\beta\} + U\beta = 0 \end{array} \tag{27}$$

from which we can get a biquadratic equation of β as

$$e^{2\alpha}\beta^4 + (2\alpha^2 e^{2\alpha} - U^2)\beta^2 + \alpha^4 e^{2\alpha} - (U\alpha + V)^2 = 0 \tag{28}$$

For given (U, V) satisfying (24), we repeatedly solve (28) by decreasing α from zero. Let α_s be the value of α at current iteration. If the solution contains a real number β_s and (α_s, β_s) satisfies (27), α_s is taken as the real part of the principal root for given (U, V). Figure 7 is the result of this numerical approach. The asymptotic decay rate is maximized approximately at $(U, V) = (0.5, 0.1)$. Hence, $(A, B) = 0.5/\tau_{max}, 0.1/\tau_{max}$ is a stable and optimal controller gain.

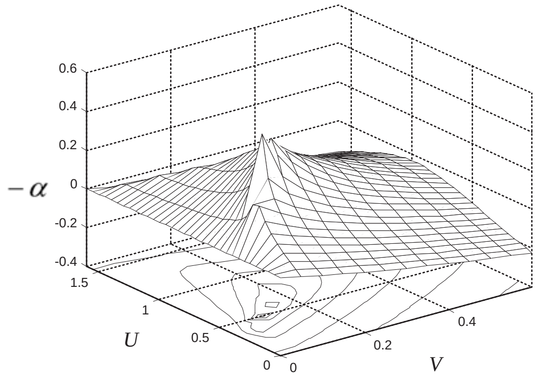


Figure 7: Asymptotic decay rate $-\alpha$ as a function of U and V

3.5 Discrete-Time Implementation

A recommended discrete-time implementation of the proposed common ER allocation algorithm, (7), at a node is as follows. Update the common ER periodically with an update interval T by

$$r[k+1] = [\ r[k] - \frac{A}{|\hat{Q}|_w}(\tilde{q}[k] - \tilde{q}[k-1]) - \frac{BT}{|\hat{Q}|_w}(\tilde{q}[k] - q_T)\ ]^+ \tag{29}$$

where $\tilde{q}[k]$ denotes the low-pass filtered queue length, which is estimated in terms of bytes instead of in terms of packets because data packets have a variable size. Particularly in our simulation studies in Section 4 we use a periodic-averaging filter such that $\tilde{q}[k] = \frac{1}{T}\int_{(k-1)T}^{kT} q(t')dt'$. Note that (29) corresponds to (7) as $T \rightarrow 0$ if $\tilde{q}[k] \approx q[k]$. In contrast to the periodic computation of the common ER, we recommend that per-VL common ER allocation be performed aperiodically upon arrival of the corresponding control packet in either forward path or backward path depending on the implementation. That is, upon arrival of VL i's control packet at time t, the node writes $r(t)$ on the CER field of that control packet where $r(t)$ is the present value of $r[k]$. Therefore, no per-VL operation is required in our discrete-time common ER allocation algorithm.

3.6 $|Q|_w$ Estimation

As seen in the above, if $|Q|_w \approx |\hat{Q}|_w$, i.e., $|Q|_w$ is estimated correctly, the closed-loop system is virtually independent of $|Q|_w$ and the optimal controller gain can be found irrespective of $|Q|_w$. Furthermore, from Appendix B, we can infer that the overestimation or correct estimation of $|Q|_w$ enables us to find the stable gain of the multiple source system from the single source case, which also means that the underestimation should be avoided since it can make the system unstable. This is why we introduce a certain margin in the $|Q|$ estimator design in the next.

The basic idea of $|Q|_w$ estimation is from the Su, de Veciana and Walrand's algorithm [17], which estimates the number of ON sources sharing a link without per-VL accounting in ATM network. We modify the algorithm to estimate the number of locally-bottlenecked VLs without doing per-VL accounting.

Suppose that the jth control packet arrives at a node at the node time t^j. if the jth control packet happens to be a control packet of VL i, it carries the value $a_i(t^j\text{-}\tau_i^f)$ in the CFR and the value m_i in the MGR field. The node monitors the control packet arrivals in a synchronous fashion over fixed-length intervals of W seconds. For the lth interval, the number of locally-bottlenecked VLs can be approximated by

$$|Q|_w^l = \sum_{t^j \in ((l-1)W, lW]} \frac{\text{NCP} + \text{HS}}{W \cdot CFR_b(t^j)} w\{CFR(t^j) \\ -MGR(t^j) \geq \delta \cdot w \cdot r(t^j)\}, \quad 0 < \delta < 1 \qquad (30)$$

where $w\ \{\ \cdot\ \}$ is the indicator function of the weight, HS is the byte size of a control packet, $CFR(t^j)$ and $MGR(t^j)$ respectively denote the value in the CFR field and the value in the MGR field of the jth control packet, and $r(t^j)$ is the latest value of the common ER at time t^j. And $CFR_b(t^j)$ denotes the value of $CFR(t^j)$ represented in terms of bytes per sec. HS can be ignored because it is very small value compared to NCP. Upon arrival of the jth control packet, if the current data rate of a VL subtracted by the MGR is greater than or equal to the latest value of common ER at the node, the VL to which the jth control packet belongs is counted as a locally-bottlenecked VL. Otherwise, it is treated as a remotely-bottlenecked VL. Here δ is the margin to avoid the underestimatation of the number of locally-bottlenecked VLs particularly near the steady state. As the system approaches the steady state, the current data rate of a locally-bottlenecked VL stays around the sum of the MGR and the common ER multiplied by the weight. Thus without the margin δ the VL could be counted wrongly as a remotely- bottlenecked VL even for small perturbation in the current data rate of the VL. By having this margin, however, one can effectively avoid this type of underestimation. Through simulations, we found that $\delta = 0.9$ is the recommended choice. Also note that the value of the indicator function is normalized by the expected number of control packet arrivals of the VL within W seconds, $(W \cdot CFR_b)/(\text{NCP+HS})$, so that the summation of these values over a W-second interval gives a correct estimate of the number of locally-bottlenecked VLs. Based on this estimate for each interval, the recursive estimate is computed at the end of every interval as follows.

$$|\hat{Q}|_w = \text{sat}_1^{|N|_w}[\lambda|\hat{Q}|_w((l-1)W) + (1-\lambda)|Q|_w^l], \\ 0 < \lambda < 1 \qquad (31)$$

where λ is an averaging factor and the saturation function ensures that $1 \leq |\hat{Q}|_w\ (t) \leq |N|_w$ for all t. Through simulations we found that $\lambda = 0.98$ yields stable and effective estimation of $|Q|_w$ for a wide range of number of VLs sharing a link and the available bandwidth, irrespective of the choice of W.

4. SIMULATION RESULTS

In this section, we use simulations to verify and demonstrate the performance of our scheme as described in the previous sections. All the simulation are performed in the $ns-2$[1] environment. We consider two different network configurations to perform the simulations, the single bottleneck configuration and the multiple bottleneck configuration with multiple bottleneck links, which are fairly standard.

Table 1: Parameters for explicit rate flow control Algorithm

ER Allocation Algorithm				$\|Q\|_w$-Estimation Algorithm		
A	B	q_T	T	W	δ	λ
$\frac{0.5}{\tau_{max}}$	$\frac{0.1}{\tau_{max}^2}$	256Kbytes	30Δ	300Δ	0.9	0.98

(τ_{max}=max$\{\tau_i,\ i{\in}N\}$, Δ=one virtual packet transmission time)

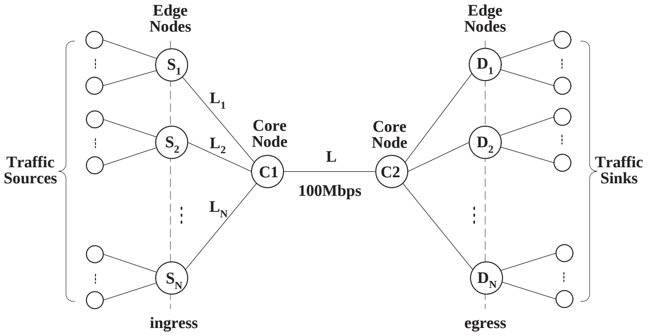


Figure 8: Single bottleneck configuration.

Unless specified otherwise, we use the following parameters for simulations in this section. The capacity of each output link is equally 100Mbps. All the links except for the link L have the propagation delay of 1ms. The propagation delay of the link L is 10ms. The buffer size of the edge nodes is 64 Kbytes and the NCP is set to 30. All the simulations use a fixed packet size of 1Kbyte unless specified otherwise. All the sources used in this simulation are persistent. In the case of TCP sources, they use TCP Reno algorithm and their maximum window size is set to 64 Kbytes. To decide the T and W, we first define a *virtual packet* with the packet size of 1000 bytes, since the explicit rate flow control algorithm is operated in byte mode and a variable-sized packets stream is regarded as a fix-sized bytes stream. We obtain T by multiplying NCP with one virtual packet transmission time and through simulations, found W which yields stable and efficient estimation of $|Q|_w$. In Table 1, we summarize the recommended values for the simulation parameters in our explicit rate flow control algorithm. Note that the target queue length of an edge node is set to 128Kbytes.

4.1 Single Bottleneck Scenario

We first consider the single bottleneck configuration, shown in Figure 8. If we consider the propagation delay between ingress and egress node and the maximum queueing delay(i.e., 256Kbytes/100Mbps $\doteqdot$ 21ms), τ_{max} is roughly 45 ms. The VL models used in this simulation configuration are summarized in Table 2. Note that we vary the MGR value, and the arrival and departure times of the VLs in order to investigate the impact of the differences in MGR, and the transient activities on the network performance.

4.1.1 Support for weighted max-min fairness

We establish one VL to each edge node for the convenience of simulation and the bottleneck link L is shared by six VLs,

Table 2: VL Models in the single bottleneck configuration.

VL#	PFR	MGR	Arr.	Dept.	Weight	
	(Mbps)	(Mbps)	(sec)	(sec)	Case I	Case II
VL1	100	5	0	∞	1	1
VL2	100	5	0	∞	2	1
VL3	100	5	0	∞	3	1
VL4	100	10	50	∞	1	1
VL5	100	10	0	∞	1.5	1
VL6	20	10	25	75	2.5	1

Table 3: the fair rates satisfying the weighted max-min fairness with minimum guaranteed bandwidth in the single bottleneck configuration.

VL#	Fair rate/Actual rate			
	0∼25	25∼50	50∼75	75∼∞
VL1	15/15.11	12.33/12.36	10.29/10.28	12.65/12.64
VL2	25/25.03	19.67/19.70	15.59/15.56	20.29/20.25
VL3	35/34.72	27/26.89	20.88/20.83	27.94/27.89
VL4			15.29/15.28	17.65/17.65
VL5	25/24.99	21/20.93	17.94/17.92	21.47/21.42
VL6		20/19.96	20/19.97	

N=6. Each VL has the weight value of case I in Table 2. The traffic sources attached to each VL consists of 50 TCP flows.

For comparison purpose, we have computed the theoretical fair rates satisfying the weighted max-min fairness with minimum rate guarantee for the given simulation scenario based on Proposition 3.1, and include the results in Table 3. Table 3 also includes the actual rate of each VL averaged over a 10sec interval of all the transient period. Observe that the fair rate of each VL varies in time according to the arrivals and departures of the other VLs and that the VL6 are bottlenecked at the core node C1.

Figure 9 shows the simulation results with single bottleneck configuration. The units of the fair rates are Mb/s and the units of arrival and departure times are in seconds. Observe from Figure 9(e) and actual rates given in Table 3 that the actual source transmission rates perfectly agree with the theoretical fair rates given in Table 3. The initial transient behavior is due to our initial condition that $r(0) = 0$ at both C1 and C2, i.e., it takes a time for the common ER value to ramp up to the operating point. The queue length at the bottleneck node, C1, is shown in Figure 9(d). The join of VL6 at 25 sec and VL4 at 50 sec results in the surge of the queue length and the leave of VL6 at 75 sec results in the sudden drop of the queue length. The flow control algorithm, however, rapidly recovers the queue length to the target value q_T(=256 Kbytes) and restabilizes it at the value. Figure 9(a) shows the estimate of the weighted number of locally bottlenecked VLs, $|\hat{Q}|_w\ (t)$, at the node C1. Note that except the initial transient period this estimate perfectly agrees with the true value, $|Q|_w\ (t)$, which is shown to be 7.5 in [0, 50) s and 8.5 in [50, ∞) s in Table 2.

Figure 9(f) shows the normalized throughputs of all the TCP flows belonging to each VL. The TCP flows share its per-aggregate bandwidth allocated in their normal way. However, the unfairness results from random drop in per-aggregate queue. Figure 9(b) shows the buffer occupancy and packet drops in per-aggregate queue and the packet that arrives at a full buffer is dropped with drop-tail.

Table 4: the fair rates satisfying the max-min fairness with minimum guaranteed bandwidth in the single bottleneck configuration.

VL#	Fair rate/Actual rate			
	0∼25	25∼50	50∼75	75∼∞
VL1	23.75/23.76	18.75/18.72	14.17/14.15	18/18.03
VL2	23.75/23.80	18.75/18.74	14.17/14.15	18/18.03
VL3	23.75/23.74	18.75/18.74	14.17/14.15	18/18.03
VL4			19.17/19.12	23/22.88
VL5	28.75/28.56	23.75/23.69	19.17/19.14	23/22.89
VL6		20/19.98	19.17/19.13	

4.1.2 Support for max-min fairness

This simulation shows that max-min fair bandwidth allocation is achieved if all the VLs have the same weight. Each VL has the weight value of case II in Table 2.

For comparison purpose, we have computed the theoretical fair rates satisfying the max-min fairness with minimum rate guarantee for the given simulation scenario based on Proposition 3.1, and include the results in Table 4. Table 4 also includes the actual rate of each VL averaged over a 10sec interval of all the transient period.

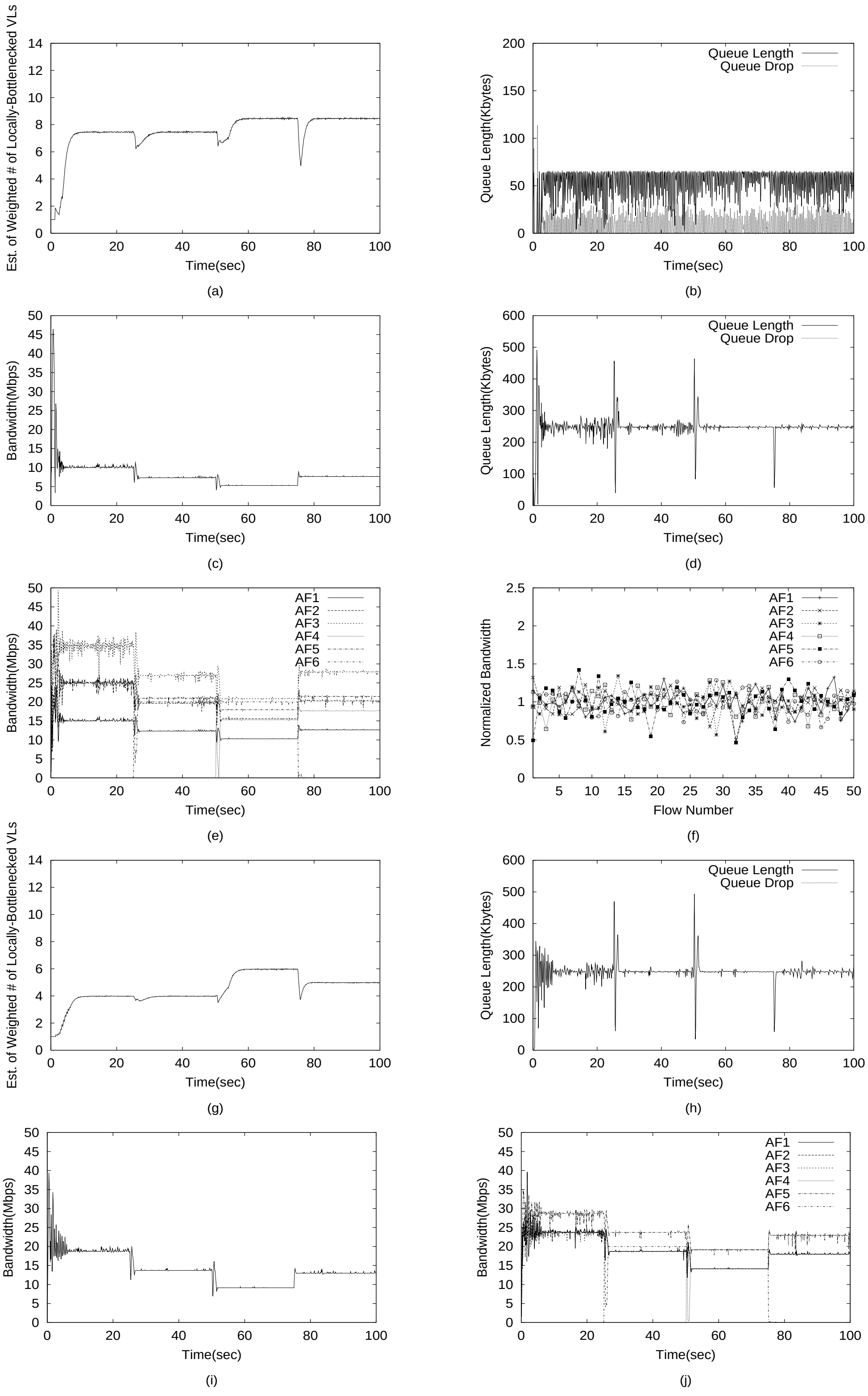
Observe from Figure 9(j) and actual rates given in Table 4 that the actual source transmission rates perfectly agree with the theoretical fair rates given in Table 4. The queue length at the bottleneck node, C1, is shown in Figure 9(h). The join of VL6 at 25 sec and VL4 at 50 sec results in the surge of the queue length and the leave of VL6 at 75 sec results in the sudden drop of the queue length. The flow control algorithm, however, rapidly recovers the queue length to the target value q_T and restabilizes it at the value. Figure 9(g) shows the estimate of the weighted number of locally bottlenecked VLs, $|\hat{Q}|_w\ (t)$, at the node C1. Note that except the initial transient period this estimate perfectly agrees with the true value, $|Q|_w\ (t)$, which is shown to be 4 in [0, 50) sec, 6 in [50, 75) sec, and 5 in [75, ∞) sec in Table 2.

4.1.3 Impact of different number of microflows

This simulation shows that weighted fair bandwidth allocation is independent of the number of microflows belonging to each aggregate flow. In this scenario, there are four aggregate flows with all 10Mbps of MGR. Aggregate flow 1 contains 50 TCP flows while the number of TCP flows in the other aggregate flows equally varies from 20 to 100. Aggregate flow 1 and 2 have a weight of 1, aggregate flow 3 has a weight of 2, and aggregate flow 4 has a weight of 3. Figure 10 shows normalized bandwidth of four aggregate flows averaged over 10 sec. The result shows aggregate flow 1 and other aggregate flows obtain their own fair rate regardless of differing the number of microflows.

4.1.4 Impact of different packet size

The aggregate flow that is sending larger packets will obtain more of the available bottleneck bandwidth, since congestion windows of its TCP flows grow more quickly. In this simulation, we shows that weighted fair bandwidth allocation is independent of the packet size of microflows belonging to each aggregate flow. In this scenario, there are four aggregate flows with all 10Mbps of MGR. Aggregate flow 1 transmits 512-byte packets while the packet size of the other three aggregate flows equally varies from 128 bytes to 1500



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 9: Performance of the single bottleneck configuration. I. Case of weighted max-min: (a) Estimate of the weighted number of locally-bottlenecked VLs at C1. (b) Queue length at the per-aggregate queue of S1. (c) Common ER at the C1. (d) Queue length at the C1. (e) The allocated bandwidth of all the VLs. (f) The normalized bandwidth of all the TCP flows. II. Case of MAX-MIN: (g) Estimate of the weighted number of locally-bottlenecked VLs at C1. (h) Queue length at C1. (i) Common ER at C1. (j) The allocated bandwidth of all the VLs.

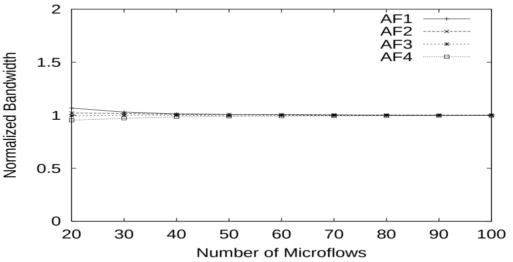


Figure 10: Impact of number of microflows

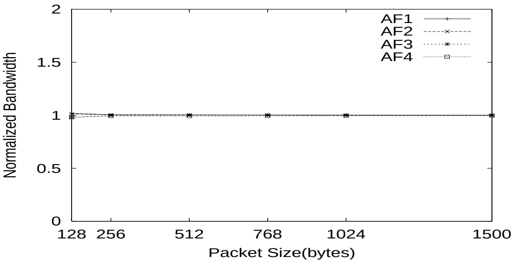


Figure 11: Impact of packet size

bytes. Aggregate flow 1 and 2 have a weight of 1, aggregate flow 3 has a weight of 2, and aggregate flow 4 has a weight of 3. Figure 11 shows that aggregate flow 1 and the other aggregate flows flows obtain its own fair rate regardless of differing packet sizes.

4.1.5 Protection of TCP flows from UDP flows

In this simulation, we shows that TCP flows can be protected from UDP flows in our scheme. In this scenario, there are four aggregate flows without MGR. Aggregate flow 1 contains one UDP flow with its sending rate increasing from 2Mbps to 24 Mbps while the other three aggregate flows consist of 50 TCP flows. Aggregate flow 1 and 2 have a weight of 1, aggregate flow 3 has a weight of 2, and aggregate flow 4 has a weight of 3. Figure 12 shows the bandwidth of each aggregate flow averaged over 10 sec. As the UDP rate increases, the bandwidth of TCP aggregate flows decrease because unused bandwidth of UDP flow is reduced. However, the UDP bandwidth is restricted to its own fair share, i.e., 14.28 Mbps.

4.2 Multiple Bottleneck Scenario

Finally, we study the multiple bottleneck configuration, shown in Figure 13, to study the case of multiple bottleneck nodes and VLs with different round-trip delays. Each edge node has one VL which contains 50 TCP flows. Eight VLs with different edge node locations are contained and the capacity of all the links is set equally 100 Mb/s, except that the link between C3 and C4 is 60 Mb/s. The link delay between C3 and C4 is 10ms and the link delay of the other links are 1ms. The VL models used in this simulation configuration are summarized in Table 5 and all the TCP sources are assumed to be persistent.

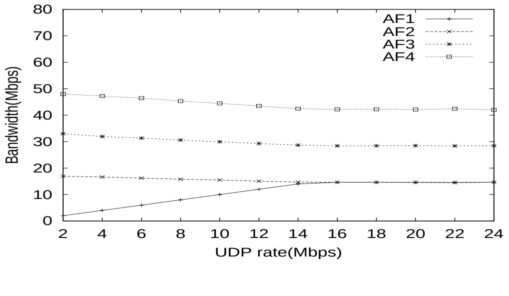


Figure 12: Protection of TCP aggregate flows from UDP flows

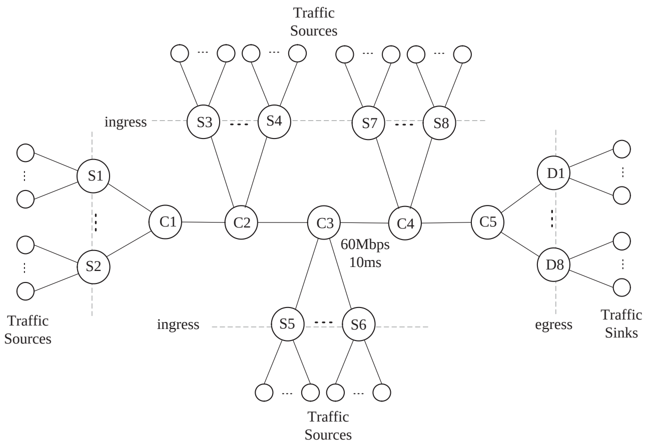


Figure 13: Multiple bottleneck configuration.

For comparison purpose, we also computed the theoretical fair rates satisfying the weighted max-min fairness with MGR guarantee for the given simulation scenario, we also include the theoretical bottleneck location of each VL in the table, which is the location at which each fair rate is determined. Figure 14 shows the simulation results. Observe from Figure 14(e), 14(f) that the actual transmission rates of edge nodes in steady state perfectly agree with the theoretical fair rates given in Table 5, irrespective of their round-trip delays and the bottleneck locations. The initial transient behavior is due to our initial condition that $r(0)=0$ at all the core nodes, which is again a phenomenon that hardly occurs during the normal operation. In the given scenario, there are two congested nodes, C3, C4. As expected, the queue length at these congested nodes converges to the target value, 256 packets, which is shown in Figure 14(c), 14(c). Figure 14(b) shows the estimate of the weighted number of locally bottlenecked VLs, $|\hat{Q}|_w\ (t)$, at the C3 and C4, respectively. We see that in the steady state the estimate stay around 6.5 and 3 at C3 and C4, respectively, which agree with the data in Table 5.

5. DEPLOYMENT ISSUES AND FURTHER WORK

We expect the initial deployment of the proposed scheme to be within large transit networks and then its deployment may also be extended to the smaller networks, such as campus network and corporate's network. Especially, Virtual Private Network(VPN), Voice over IP(VoIP) trunking, and Virtual LAN(VLAN) based on the proposed scheme will be able to provide better enhanced service to their customers.

As studied in [6], we know the unfairness and congestion

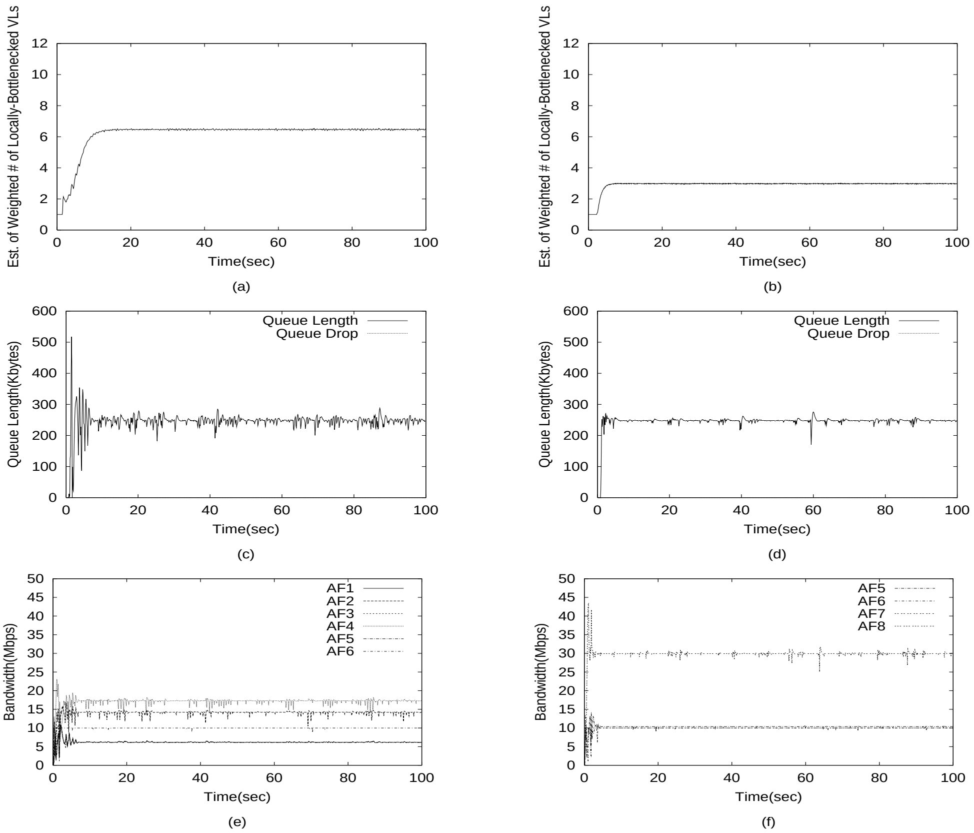


Figure 14: Performance of the multiple bottleneck configuration: (a) Estimate of the weighted number of locally bottlenecked VLs at C3. (b) Estimate of the weighted number of locally bottlenecked VLs at C4. (c) Queue length at C3. (d) Queue length at C4. (e) The allocated bandwidth of VLs bottlenecked at C3. (f) The allocated bandwidth of VLs bottlenecked at C4

Table 5: VL Models in the multiple bottleneck configuration.

VL#	PFR	MGR (Mbps)	Weight	Fair share/ Actual rate(Mbps)	Bottl.
VL1	100	0	1	6.15/6.19	C3
VL2	100	5	1.5	14.23/14.15	C3
VL3	100	0	1	6.15/6.19	C3
VL4	100	5	2	17.3/17.17	C3
VL5	100	0	1	6.15/6.19	C3
VL6	10	5	2.5	10/10	PFR
VL7	10	5	1	10/10	PFR
VL8	100	0	3	30/29.94	C4

collapse problem happens when TCP and UDP flows compete for given bandwidth, given nodes with FIFO scheduling. In our scheme, UDP flows can be separately handled in the per-aggregate queues, as experiments in [13], which used dedicated TCP trunks for UDP flows, because UDP flows do not perform congestion control by themselves.

The packet drops occurs only at the edge nodes because our scheme pushes the queueing beyond target queue length at the interior network out to the edge nodes. Even if not in our scheme, each edge node can employs queue management scheme, such as RED, FRED, to manage its per-aggregate queue.

We assume the controlled aggregate flows can not be mixed together with uncontrolled flows. The reason is why the competition between the controlled flows and uncontrolled flows may result in unfair bandwidth allocation of the controlled flows. Even though this problem can be avoided by allocating a special queue with a specified bandwidth at the core nodes as addresses in [7], such solution may require the more complexity and per-flow operation at the core nodes.

In our scheme, bandwidth allocation of individual microflows relies on the end-to-end user protocol, such as TCP congestion control and TCP friendly rate control(TFRC). we will extend our approach from edge-to-edge to end-to-edge to provide weighted max-min fair bandwidth allocation and minimum rate guarantee for the individual flows which constitute an aggregate flow. Such work requires the addition of two functionalities only in an edge node, the addition of explicit rate flow control algorithm at the input port to control the buffer occupancy of per-aggregate queue, and the addition of the explicit window adaptation scheme already studied by [11, 10], which is based on modifying the receiver's advertised window in TCP acknowledgement(ACK) returning to the TCP sender. We expect the explicit window adaptation scheme can control the buffer occupancy of the per-aggregate queue efficiently at the edge node, and results in significant improvement in packet loss rate, fairness, and goodput.

6. CONCLUSIONS

In this paper, we present a new scheme which supports per-aggregate QoS between pair-edges for the Internet. Edge nodes regulate the transmission rate of the aggregate flows using the explicit feedback rate from the network. Core nodes compute the common explicit rate using the explicit rate based flow control algorithm.

The proposed scheme has the following features: 1) It provides QoS for aggregate flows rather than individual flows. Accordingly, It can provide relatively different levels of service to each aggregate flow. 2) It employs edge-to-edge closed loop control mechanism between ingress node and egress node, which uses the virtual link concept we defined. 3) It is stateless-core approach in that no per-aggregate flow state is maintained at the interior network. Besides, it places only simple functionality within the network core, with more complex operations being implemented at the edge of the network. It also pushes the interior network congestion out to the network edges 4) It supports weighted max-min fair bandwidth allocation among aggregate flows and guarantees the minimum rate of each aggregate flow within the network. 5) we use a simple, scalable, and stable explicit rate allocation algorithm to operate the weighted max-min flow control with minimum rate guarantee among edges. The bandwidth allocated to each aggregate flow and network queues are asymptotically stabilized at the unique equilibrium point at which the weighted max-min fair bandwidth allocation with minimum rate guarantee and target queue length are achieved.

Based on all of the above key features, The proposed scheme enables different levels of service to be provided for aggregate traffic streams on a common network infrastructure. Accordingly, we believe that the proposed scheme will serve as an enhanced solution to provide edge-based best-effort service for the Internet. The simulation results show that our scheme can perform the excellent bandwidth allocation for the aggregate flows based on weighted max-min fairness.

7. REFERENCES

[1] ns-2 network simulator. http://www.isi.edu/nsnam/ns/, 2000.

[2] B. R. Barmish. *New Tools for Robustness of Linear Systems*. MacMillan, New York, 1994.

[3] R. Bellman and K. L. Cooke. Differential-difference equations. In *Academic Press*, New York, 1963.

[4] L. Benmohamed and S. M. Meerkov. Feedback control of congestion in packet switching networks: the case of single congested node. *IEEE/ACM Trans. on Networking*, 1:693–708, December 1993.

[5] F. Blanchini, R. L. Cigno, and R. Tempo. Robust rate control for integrated services packet networks. *IEEE/ACM Trans. on Networking*, 10(5):644–652, October 2002.

[6] S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the internet. *IEEE/ACM Trans. on Networking*, 7:458–472, 1999.

[7] D. Harrison and S. Kalyanaraman. Edge-to-edge traffic control for the internet. In *RPI ECSE Networks Laboratory Technical Report, ECSE-NET-2000-I*, January 2000.

[8] Y. Hou, H. Tzeng, and S. Panwar. A weighted max-min fair rate allocation for available bit rate service. In *Global Telecommunications Conference, 1997. GLOBECOM '97*, volume 1, pages 492–497.

[9] Y. Hou, H. Tzeng, and S. Panwar. A generalized max-min rate allocation policy and its distributed implementation using the abr flow control mechanism. In *Proceedings of IEEE Infocom' 98*, pages 1366–1375, March 1998.

[10] L. Kalampoukas, A. Varma, and K. Ramakrishnan. Explicit window adaptation: a method to enhance tcp performance. *IEEE/ACM Trans. on Networking*, 10:338–350, June 2002.

[11] S. Karandikar, S. Kalyanaraman, P. Bagal, and B. Packer. Tcp rate control. *Computer Communications*, 30(1):45–58, January 2000.

[12] A. Kolarov and G. Ramamurthy. A control-theoretic approach to the design of an explicit rate controller for abr service. *IEEE/ACM Trans. on Networking*, 7:741–753, October 1999.

[13] H. T. Kung and S. Y. Wang. Tcp trunking: Design,implementation and performance. In *Proceedings of the 7th International Conference on Network Protocols (ICNP'99)*, pages 222–231, October 1999.

[14] R. Sivakumar, T. Kim, N. Venkitaraman, and V. Bharghavan. Achieving per-flow weighted rate fairness in a core stateless network. In *IEEE Conference on Distributed Computing Systems 2000*, March 2000.

[15] G. Stépán. *Retarded dynamical systems: stability and characteristic functions*. White Plains, NY:Longman, New York, 1989.

[16] I. Stoica, S. Shenker, and H. Zhang. Core-stateless fair queueing: achieving approximately fair bandwidth allocations in high-speed networks. In *In Proceedings of ACM SIGCOMM Conference*, September 1998.

[17] C. F. Su, G. de Veciana, and J. Walrand. Explicit rate flow control for abr services in atm networks. *IEEE/ACM Trans. on Networking*, 8:350–361, June 2000.

[18] B. Vandalore, S. Fahmy, R. Jain, R. Goyal, , and M. Goyal. A definition of general weighted fairness and its support in explicit rate switch algorithms. In *in Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, pages 22–30, October.

[19] B. Vandalore, S. Fahmy, R. Jain, R. Goyal, and M. Goyal. General weighted fairness and its support in explicit rate switch algorithms. *Computer Communications*, 23(2):149–161, January 2000.

APPENDIX

A. THE PROOF OF COROLLARY 4.1

Proof. For given τ and A, denote B^* by the maximum value of B satisfying $\tau \leq \arccos(Bw/\bar{\omega}^2)/\bar{\omega} \doteq h$. We can see through differentiation that h is a monotonically decreasing function of B, thus $B = B^*$ when $\tau = \arccos(Bw/\bar{\omega}^2)/\bar{\omega}$ and $B < B^*$ should be satisfied for the stability. Let $\omega_1 = \bar{\omega}\tau$. Then, we obtain $\omega_1 = \arccos(B^* w/\bar{\omega}^2)$ and consequently, we can set $\sin\omega_1 = Aw/\bar{\omega}$ by (23). Noting that $0 < \omega_1 < \pi/2$, we can derive (24) as follows.

$$\begin{array}{l} 0 < A\tau = \omega_1 \frac{Aw}{\bar{\omega}} = \omega_1 \sin\omega_1 < \frac{\pi}{2} \\ 0 < B\tau^2 = \omega_1^2 \frac{Bw}{\bar{\omega}^2} < \omega_1^2 \frac{B^* w}{\bar{\omega}^2} = \omega_1^2 \cos\omega_1 \end{array} \qquad (32)$$

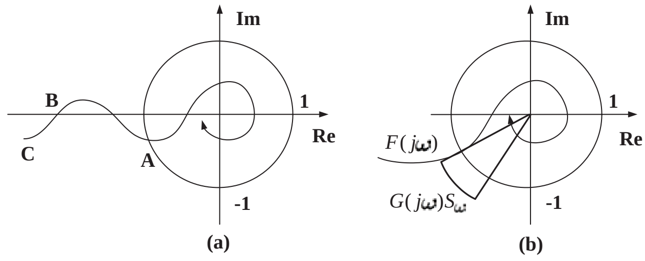


Figure 15: Nyquist diagram of $F(j\omega)$ - (a) Not possible. (b) Possible.

where ω_1 is the unique solution of $A\tau = \omega \sin \omega$ for $0 < \omega < \pi/2$. Sufficiency can be proven similarly. □

B. THE PROOF OF PROPOSITION 3.3

For the multiple source case shown in Figure 4, the open-loop transfer function is given by

$$F(s) = \left(\frac{A}{|\hat{Q}|_w} \frac{1}{s} + \frac{B}{|\hat{Q}|_w} \frac{1}{s^2} \right) \sum_{i \in Q} w_i e^{-\tau_i s} \qquad (33)$$

because $C_I = B/|\hat{Q}|_w$ and $C_P = A/|\hat{Q}|_w$ in our work. If $|Q|_w$ is overestimated or correctly estimated, i.e., $|\hat{Q}|_w \geq |Q|_w$, (33) is obviously included in the following equation.

$$F(s) = \underbrace{\left(\frac{A}{s} + \frac{B}{s^2} \right)}_{G(s)} \underbrace{\sum_{i \in Q} \rho_i e^{-\tau_i s}}_{Z(s)} \qquad (34)$$

where $\rho_i \geq 0$ and $\sum_{i \in Q} \rho_i = 1$. Note that $\rho_i = w_i/|\hat{Q}|_w$ in our work.

Setting $s = j\omega$ yields

$$\begin{aligned} &G(j\omega) = -\tfrac{B}{\omega^2} - j\tfrac{A}{\omega}, \;\; Z(j\omega) = \sum_{i \in Q} \rho_i e^{-j\omega\tau_i} \\ &F(j\omega) = G(j\omega) Z(j\omega) \end{aligned} \qquad (35)$$

Proof. **Necessity of Proposition 3.3**: If the controller gain stabilizes the multiple source system with $\rho_i \geq 0, \sum_{i \in Q} \leq 1$, it also stabilizes the system when $\tau_1 = \tau_{max}$, $\rho_1 = 1$, and $\rho_i = 0 \; \forall i \geq 2$. Thus the necessity is proved. □

To prove sufficiency, we need the following two lemmas

Lemma B.1. *Assume that the single source system is asymptotically stable. Let $\bar{\omega}$ be a unique ω such that $|F(j\omega)| = 1$, then $Im[F(j\omega)] < 0$ for $0 < \omega \leq \bar{\omega}$.*

Proof. *There exists ω^* (the point C in Figure 15(a)) such that $Re[F(j\omega)] < 0$ and $Re[F(j\omega)] < 0$ for $0 < \omega < \omega^*$ because $F(j\omega) \to -\infty - j\infty$ as $\omega \to 0^+$. We assume that the system is stable, and then there is a unique solution, $\bar{\omega}$, of the equation $|F(j\omega)| = 1$ by Proposition 3.2. Note that $\bar{\omega}$ is the point A in Figure 15(a). Thus we need to prove $Im[F(j\omega)], \omega^* \leq \omega \leq \bar{\omega}$. By contradiction, suppose that there exists ω satisfying $Im[F(j\omega)] = 0$ and denote by $\hat{\omega}$ its minimum value (the point B in Figure 15(a)). Obviously, we have $|F(j\hat{\omega})| > 1$, and there are two cases, $Re[F(j\hat{\omega})] > 1$ and $Re[F(j\hat{\omega})] < -1$. If $Re[F(j\hat{\omega})] > 1$, then $\angle F(j\hat{\omega}) = 0$. This is not possible because from (21) we have $Im[G(j\hat{\omega})] < 0$ and $\hat{\omega}\tau \leq \bar{\omega}\tau < \pi$, that is, $\angle F(j\hat{\omega}) < 0$. In the case of $Re[F(j\hat{\omega})] < -1$, we can summarize as*

$$\begin{cases} \angle F(j0^+) = -\pi \\ \angle F(j\omega^*) > -\pi \\ \angle F(j\hat{\omega}) = -\pi \\ \angle F(j\bar{\omega}) > -\pi \\ \lim_{n\to\infty} \angle F(j\frac{(2n+1)\pi}{\tau}) = -\pi \end{cases} \qquad (36)$$

which implies that there are at least 3 local extrema. However, the phase function and its derivative

$$\begin{aligned} \angle F(j\omega) &= \arctan\left(\tfrac{-r}{q\omega}\right) - \tfrac{\pi}{2} - \omega\tau \\ (\angle F(j\omega))' &= \tfrac{qr}{q^2\omega^2 + r^2} - \tau \end{aligned} \qquad (37)$$

show that there are at most one extremum. It contradicts our assumption, therefore $Im[F(j\omega)] < 0$ for $0 < \omega \leq \bar{\omega}$. □

Lemma B.2. *For given ω, define the set $V(\omega)$ as follows*

$$V(\omega) = \left\{ z = F(j\omega, \rho_i, \tau_i) \,\middle|\, \sum_{i \in Q} \rho_i \leq 1, 0 \leq \tau_i \leq \tau_{max} \right\}$$

The multiple source system is asymptotically stable $\forall \rho_i, \tau_i$ if and only if the two conditions below are met.

(i) There exists a choice of ρ_i and τ_i such that the multiple source system is asymptotically stable.

(ii) $V(\omega)$ does not include $-1 + j0$ for all $\omega \in [0, \infty]$.

Proof. *See [2].* □

Proof. **Sufficiency of Proposition 3.3**: (i) is self-evident since we are proving the sufficiency of Proposition 3.3. Now, the proof of the proposition is completed if (ii) is satisfied. Define a convex set $S_\omega^{\tau_{max}}$ by

$$S_\omega^{\tau_{max}} = \{ z \in C : |z| \leq 1, -\omega\tau_{max} \leq \angle z \leq 0 \}$$

For $Z(j\omega)$ in (35), it is intuitively noticed that $Z(j\omega) \in S_\omega^{\tau_{max}}$. In addition, $V(\omega) \subset G(j\omega) S_\omega^{\tau_{max}}, \forall \rho_i, \tau_i$ because we have

$$F(j\omega, \rho_i, \tau_i) = G(j\omega) Z(j\omega) \in G(j\omega) S_\omega^{\tau_{max}}$$

$G(j\omega) S_\omega^{\tau_{max}}$ is bounded by $F(j\omega) = G(j\omega) e^{-j\omega\tau_{max}}$ and $G(j\omega)$, and can be defined by

$$G(j\omega) S_\omega^{\tau_{max}} = \{ z | \angle F(j\omega) \leq \angle z \leq \angle G(j\omega), |z| \leq |G(j\omega)| \}$$

For better presentation, $G(j\omega) S_\omega^{\tau_{max}}$ is shown in Figure 15(b). For $\bar{\omega} < \omega \leq \infty$, $|G(j\omega)| < 1$ which implies that $G(j\omega) S_\omega^{\tau_{max}}$ always stays in the interior of the unit circle. Consequently, $V(\omega)$ is also in the interior of the unit circle because $V(\omega) \subset G(j\omega) S_\omega^{\tau_{max}}$. For $0 < \omega \leq \bar{\omega}$, $Im[F(j\omega)] < 0$ by Lemma B.1 and $Im[G(j\omega)] < 0$, thus we obtain $Im[G(j \omega) S_\omega^{\tau_{max}}] < 0$ which means that $G(j\omega) S_\omega^{\tau_{max}}$ does not include $-1 + j0$. For $\omega = 0$, $V(\omega)$ is at infinity. We have shown that the value set $V(\omega)$ does not include $-1 + j0$ for $\omega \in [0, \infty]$. Therefore, the gain (A,B) stabilizes the multiple source system if it stabilizes the single source system with $\tau = \tau_{max}$. □