

# Active Queue Management, But Only At Network Edges!

## ABSTRACT

Congestion control in the internet is usually decomposed into an end-system controller (eg: TCP) and a bottleneck controller (AQM or Drop-Tail queueing) which generates feedback. This paper asks an architectural question: can the active queue management (AQM) function be moved to the edges of the network so that the core routers focus simply on packet forwarding? The question is interesting in the light of recent trends to consolidate traffic management functions at network edges. We develop a theoretical framework to answer this question in the affirmative. The idea is to use a notion of path capacity and path demand to build an edge-based virtual queue AQM (called EVQ) at the network edges. We show that such a scheme can achieve the usual steady state objectives: high utilization, low persistent queue length, negligible packet drop rate and max-min fairness. We also establish the global asymptotic stability for arbitrary number of flows and time delays, characterize its equilibrium fairness to be max-min and not proportional. For real networks, the realization of the framework and the transient performance depend upon robust and short-time-scale estimation of edge-to-edge parameters: path capacity and available bandwidth. This work uses recent advances in active measurements (e.g., CapProbe, pathChirp) and can leverage future advances in this area. We demonstrate the measurement tradeoffs and parameter sensitivities through simulations of multi-bottleneck scenarios.

## 1. INTRODUCTION

Internet congestion control involves the combination of end-system control (eg: TCP) and a bottleneck feedback generator (active queue management (AQM) or Drop-Tail queueing). Though there has been a tremendous amount of research in AQM design [1, 2, 3, 4, 5, 6], and RED [7] is widely implemented, ISPs tend not to enable such AQM schemes. Reasons include uncertainty in parameter configuration guidelines and unwillingness to drop packets until

buffers really overflow. Minor concerns include the computational load in the critical forwarding path. Recent industry trends (eg: differentiated services, deep packet inspection) have involved consolidation of traffic management functions at network edges. Given this state of affairs, this paper asks an architectural question (see Figure 1): *can the active queue management (AQM) function be moved and consolidated at the edges of the network so that the core routers focus simply on packet forwarding?*

We develop a theoretical framework to answer this question. The framework involves active measurement of notional quantities called “*path capacity*” and “*path demand*” and the design of a new edge based AQM using these quantities as the input. We show that the resulting scheme (Edge-Based Virtual Queue or EVQ), in combination with TCP achieves good utilization, low persistent queues, max-min fairness and negligible packet drop rate. We also develop parameter guidelines and proofs for global asymptotic stability for arbitrary network topologies and time delays.

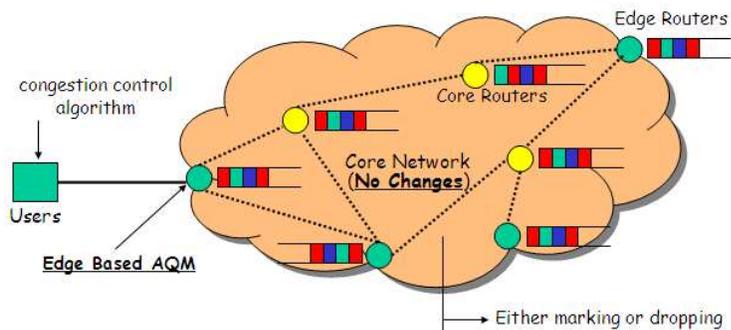


Figure 1: Relocating the AQM function From Network Cores to Network Edges.

EVQ operates at the network edges and abstracts an edge-to-edge path as a virtual link. A path is the series of bottlenecks between the ingress and egress router. The flows in the same edge-to-edge path are managed together: they contribute a virtual demand to this virtual link which builds up a virtual queue, leading to congestion signals. The *path capacity* is defined as the minimum of link capacities on the

path. The *path demand* is defined as the difference between the path capacity and available bandwidth on the path. Note that path demand may not correspond to the demand (sum of flow rates) at any bottleneck: it is just a worst case estimate of congestion at the smallest capacity link. Our theoretical framework shows that this estimate is sufficient to build a stable AQM scheme from the edge of the network. We build our EVQ by modifying the Adaptive Virtual Queue (AVQ) algorithm by Kunniyur et. al. [2] which also uses the concepts of virtual capacity and virtual queue per link (determined by demand). In EVQ, a virtual queue is maintained for every path instead of every link based upon the notions of path capacity and path demand.

Since EVQ operates a flow control scheme edge-to-edge, the fairness amongst different end-to-end TCP flows in equilibrium is determined not only by TCP but also by EVQ. Like TCP, EVQ uses a *path utility function* to influence the dynamics and steady state of virtual capacity. We show that this results in max-min fairness (unlike the regular proportional fairness achieved in end-to-end schemes), a fact also verified by subsequent ns-2 simulations.

Our initial implementation in ns-2 uses the “*CapProbe mode*” technique from UCLA [8] to estimate path capacity, and “*pathChirp*” [9] technique from Rice University to estimate the available bandwidth on the path. These techniques use correlated and carefully timed packet probes to perturb the path and make inferences from the time-series of the received probes. The time-scale of robust active measurement matters: our ns-2 simulations currently assume that demand/capacity conditions do not change during the time-scale of active measurements – an assumption that must be revisited in real-world design. Our primary focus in this paper is on the theoretical feasibility, design and stability analysis of this new architectural concept.

Section II presents the key ideas of this framework and EVQ algorithm outline. Section III analyzes the stability, equilibrium fairness and parameter guidelines for the EVQ algorithm. Section IV, provides preliminary MATLAB and ns-2 simulations to support our theoretical results, and comparisons of “EVQ at edges only” vs AVQ or Drop-Tail queue management at all routers. Section V presents conclusions and future work.

## 2. DESIGN GUIDELINES

This section develops the basis of the assertion that the AQM function can be moved to network edges.

### 2.1 Key Design Ideas

#### #1. Decouple active queue management (AQM) from its placement in network

Consider a scenario where a network *edge router* (along with an egress edge router) estimates the notional path capacity  $C_I$  and corresponding path demand  $D_I$  of path  $I$ . Here, path capacity is defined as  $C_I = \min_{l \in T_I} \{c_l\}$ , where  $T_I$  is the set of links over which path  $I$  traverses and  $c_l$  is the link capacity at link  $l$ . The path demand is *artificially* defined as the

difference between path capacity and available bandwidth on the path:

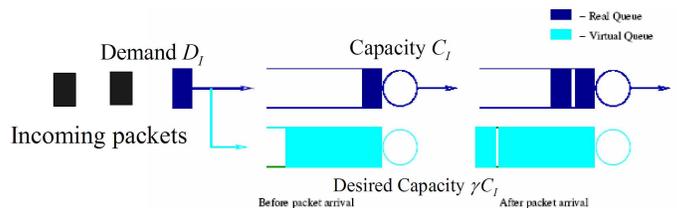
$$D_I = \min_{l \in T_I} \{c_l\} - \min_{l \in T_I} \{c_l - y_l\}, \quad (1)$$

where  $\min_{l \in T_I} \{c_l - y_l\}$  denotes the available bandwidth for path  $I$  and  $y_l$  is the aggregate rate at link  $l$ .

If the demand  $D_I$  equals the virtual path capacity defined as  $\gamma C_I$ , where  $0 < \gamma < 1$  is the desired *target utilization* for the network (see Figure 2), then the steady state queue for the path  $i$  will be zero. This is because

$$D_I - \gamma C_I = 0 \Leftrightarrow \min_{l \in T_I} \{c_l - y_l\} = (1 - \gamma) \min_{l \in T_I} \{c_l\}, \quad (2)$$

which implies that at the equilibrium the real demand at *any* node in the path is less than its capacity, i.e.,  $y_l \leq c_l, \forall l \in L$ , implying that the steady state of queue length will be zero. If this can be realized at the *edge* of the network, we can essentially decouple the task of managing bottleneck queues and the placement of this management function. In other words, we do not require an active queue management component to be present at every bottleneck and could move this function to the network edges.



**Figure 2: Virtual queue for each path with capacity  $C_I$  and arriving rates  $D_I$ .**

#### #2. Virtual queue and path utility function

In order to dynamically match path demand  $D_I$  with virtual path capacity  $\gamma C_I$ , we follow the AVQ algorithm [2]. The edge router maintains a virtual queue for each path  $I$  whose capacity is  $\gamma C_I$  and whose buffer size is equal to the buffer size of the real edge queuing buffer (see Figure 2). Upon each packet arrival, a *fictitious* packet is enqueued in the virtual queue if there is sufficient space in the real buffer. If the new packet overflows the virtual buffer, then the packet is discarded in the virtual buffer and the real packet is marked by setting its ECN bit or the real packet is dropped, depending upon the congestion notification mechanism used by the edge router. Since EVQ is performed at the edge of network, fairness amongst end-to-end flows will not only be determined by TCP flow control but also by EVQ. To guarantee fairness between different flows, EVQ incorporates a *path utility function*  $W_I(\varsigma_I)$  in the dynamic equation to determine  $\varsigma_I$  (see Line 15 of the pseudo-code of EVQ in Table 2).

#### #3. Active measurement of path Capacity available bandwidth

Path capacity is estimated periodically using the “CapProbe” technique, which combines delay as well as dispersion measurements of packet pairs to filter out samples distorted by cross-traffic and give a quick and accurate capacity estimation [8]. To estimate the path demand  $D_I$ , we first estimate the path available bandwidth and then subtract it out of the path capacity. We use the pathChirp [9] technique that sends exponentially spaced *chirp probing train* to infer the available bandwidth. PathChirp has several advantages over packet pairs or packet trains [8]. We understand that the time-scales and robustness of these estimates determine the nature of the transient performance and the *realized* stability of the scheme. We use our simulations only to make preliminary statements about steady state behavior (ignoring transients).

## 2.2 The EVQ Protocol: Outline and Key Features

The terminology used in the paper is summarized in Table 1:

$x_i$	Sending rate of user $i$
$y_l$	Aggregate rate at link $l$
$c_l$	Actual capacity of link $l$
$p_l$	Price at link $l$
$q_i$	Aggregate price received by user $i$
$U_i(\cdot)$	Utility functions of user $i$
$\pi_l$	Virtual price at link $l$
$C_I$	Capacity of Path I
$D_I$	Demand of Path I
$\varsigma_I$	(“Varsigma”): Virtual capacity for virtual link at the edge of path I
$B$	Buffer Size of virtual queue in packets
$VQ$	Size of Virtual Queue in packets
$W_I(\cdot)$	Utility functions of Path I
$\beta$	Fairness ratio of EVQ
$\lambda$	Damping factor of EVQ
$\alpha(\cdot)$	Barrier or forcing function to match path demand to capacity
$\gamma$	Desired utilization of Path I

Table 1: Terms used in the paper.

At each packet arrival epoch, EVQ updates the virtual queue capacity according to the following equation:

$$\varsigma_I(t) = \varsigma_I(s) + \beta W_I'(\varsigma_I(t)) + \alpha(\gamma C_I - D_I), \quad (3)$$

where  $t$  is current time and  $s$  is the arrival time of last packet.  $W_I(\cdot)$  is the path utility function corresponding to the virtual capacity  $\varsigma_I$  and  $\alpha(\gamma C_I - D_I)$  is the hypothetical price charged.  $\alpha(\cdot)$  can also be thought of as a *barrier function* [10] forcing  $D_I$  to approach  $\gamma C_I$ . For example, we use the following  $\alpha(\cdot)$

$$\alpha(\gamma C_I - D_I) = -\lambda(D_I - \gamma C_I) \quad (4)$$

If  $\lambda > 0$  becomes large enough, when  $t \rightarrow \infty$ ,  $D_I$  will stay close to  $\gamma C_I$ , that is,  $\min_{l \in T_I} \{c_l - y_l(t)\}$  will be forced to  $(1 - \gamma) \min_{l \in T_I} \{c_l\}$ . This in turn implies  $y_l(t) < c_l$  for all links when  $t \rightarrow \infty$ , i.e. no links inside the network will be congested except the virtual link at the edge. In EVQ,

we call  $\lambda$  the damping factor, which is set to determine how aggressive the marking or dropping should be when path demand is more than the desired utilization  $\gamma$  of path capacity  $C_I$ .

The following pseudo-code describes an implementation of EVQ scheme:

The EVQ Algorithm
1. $t$ : Current time
2. $s$ : Arrival time of the last packet enqueued
3. $T$ : Time when the last path demand estimate is taken
4. $D(T)$ : Path demand estimate at time $T$ in packets
5. $C(T)$ : Path capacity estimate at time $T$ in packets
6. $\xi$ : Time after which $D(T)$ is updated
7. <b>for</b> each packet arrival in $(t; t + \xi)$ <b>do</b>
8. $VQ \leftarrow \max\{VQ - \varsigma \cdot (t - s), 0\}$ /* Update virtual queue size */
9. <b>if</b> $VQ + 1 > B$ <b>then</b>
10. Mark (or drop) the packet in real queue
11. <b>else</b>
12. $VQ \leftarrow VQ + 1$
13. Enqueue the packet in the real queue.
14. <b>end if</b>
15. $\varsigma(t) = \varsigma(s) + (\beta W_I'(\varsigma(s)) + \lambda(\gamma C(T) - D(T))) \cdot (t - s)$ /* Update virtual path capacity */
16. $\varsigma(t) = \max\{\varsigma(t), 0\}$
17. $s \leftarrow t$ /* Update last packet arrival time */
18. <b>end for</b>

Table 2: Pseudo-code of EVQ

We note the following features of the EVQ scheme:

**1. Parameters  $(\beta, \lambda, \gamma)$  and Utility Function  $W_I(\cdot)$ :** EVQ uses three parameters that completely determine the performance of EVQ along with the path utility function  $W_I(\cdot)$ . The desired utilization  $\gamma$  determines the robustness to the presence of uncontrollable short flows. It allows an operator to tradeoff utilization against queue length. The damping factor  $\lambda$  determines the aggressiveness of packet marking. The fairness ratio  $\beta$  can be set differently for each path and is then the relative weight assigned to a path that determines the weighted fairness in equilibrium. The path utility function  $W_I(\cdot)$  is any increasing, strictly concave, and continuously differentiable function which determines the global stability of EVQ.

**2. Link utilization vs Path Capacity Utilization:** For every link  $l$ , if we define a virtual price  $\pi_l$  as

$$\pi_l = \max\{0, y_l - \gamma' c_l\}, \quad (5)$$

where

$$\gamma' = 1 - \frac{(1 - \gamma) \min_{l \in T_I} \{c_l\}}{c_l} \quad (6)$$

we can rewrite EVQ algorithm as

$$\varsigma_I(t) = \varsigma_I(s) + \beta W_I'(\varsigma_I(t)) + \lambda \max_{l \in T_I} \{\pi_l\} \quad (7)$$

Thus, virtual capacity  $\varsigma_I$  is adjusted by a max virtual price along the path, instead of the sum of the prices at every link.

Noting that unlike the sum feedback, which in general gives proportional fairness [10], we will argue in the next section that this feature of EVQ results in *max-min fairness*. That means in a single link case, each flow gets an equal share of the link capacity. [11]. Observe that:

$$1 > \gamma' \geq \gamma \quad (8)$$

Therefore the achieved link utilization at any link in the path (including the dominant bottleneck) is *no less than* path utilization parameter  $\gamma$ .

**3. The steady queue length is zero except at the edge-based virtual queue:** When  $t \rightarrow \infty$ , path demand will approach path capacity due to barrier function  $\alpha(\cdot)$ , that is

$$D_I - \gamma C_I \rightarrow 0 \Leftrightarrow \min_{l \in T(I)} \{c_l - y_l\} \rightarrow (1 - \gamma) \min_{l \in T(I)} \{c_l\}. \quad (9)$$

It implies that for every link  $l$ ,  $y_l < c_l$  and flows gets congested only at the edge-based virtual link with virtual capacity  $\varsigma_I$ .

### 3. THEORETICAL DEVELOPMENT OF EVQ

The starting point for the analysis of such a scheme is the fluid model of the network flow control [12, 13, 14, 15, 16, 17]. A theoretical justification of how a stochastic discrete-time equation can be approximated by a fluid model is shown in [18]. We then incorporate the virtual capacity update equation with this model and study the global stability of the entire system and the fairness of the equilibrium.

#### 3.1 Model

Consider a network model [10] shown in Figure 3. Packets from each user (with sending rate  $x_i$ ) are routed through the links with the aggregate link rate  $y = Rx$ , where  $R$  is the routing matrix. Each link  $j$  has a fixed capacity  $c_j$ , and based on its congestion and queue size, a link price  $p_j$  is computed:

$$p_l = h_l(y_l, c_l), \quad l = 1, \dots, L. \quad (10)$$

If  $p_j$  is the loss probability at link  $j$ , for analytical tractability, we assume that

$$h(y_l, c_l) = \frac{\max\{0, (y_l - c_l)\}}{y_l}. \quad (11)$$

In general,  $h_l(y_l)$  is the function of the link arrival rate. We also consider a more general form of the pricing function (used in our analysis):

$$h(y_l, c_l) = \left(\frac{y_l}{c_l}\right)^\mu. \quad (12)$$

The price each source  $i$  receives has two parts. The first one is the aggregate source price from network,  $q = R^T p$ , which models the interior network prices implied by Drop-Tail queueing (eg: the delay penalties due to TCP self-clocking). Recall that we *do not* have an *explicit* AQM scheme implemented inside the network. EVQ also marks

packets when the virtual buffer overflows. In the paper, we denote this part of price by  $v$ , modeled as:

$$v_I = h(x_I, \varsigma_I), \quad (13)$$

where the subscript  $I$  denotes the path and  $x_I = \sum_{i \in I(i)} x_i$  is the sum of the rates for the flows sharing the path. To make the stability and fairness analysis applicable to diverse types of network protocols, we consider a *general* source flow control algorithm

$$\dot{x}_i = \kappa_i(x_i) (U'_i(x_i) - q_i), \quad i = 1, \dots, N \quad (14)$$

or in vector form,

$$\dot{x} = \kappa(x) (U'(x) - q), \quad (15)$$

where  $x = [x_1 \ x_2 \ \dots \ x_N]^T$  and the price due to drop and marking is modeled as

$$q = R^T p + v \quad (16)$$

#### 3.2 Stability: Sketch of Proof

We use the following utility functions for the source flow control and edge queue management:

$$U_i(x_i) = \frac{x_i^{1-n}}{1-n}, \quad n > 0 \quad (17)$$

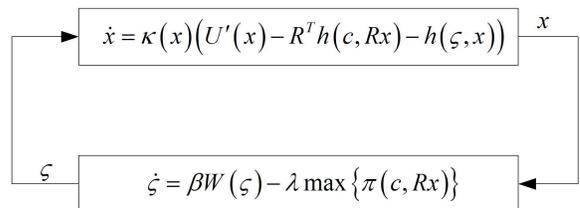
$$W_I(\varsigma_I) = \frac{\varsigma_I^{1-m}}{1-m}, \quad m > 0, \quad (18)$$

which corresponds to TCP-like congestion control algorithm [19]. The conditions for global asymptotic stability for other utility functions can be derived in a similar way. To demonstrate stability, we substitute pricing functions (11) or (12) into the dynamics of send rates and virtual capacity, and rewrite the network fluid model with EVQ as:

$$\dot{x} = \kappa(x) \left( U'(x) - R^T h(c, Rx) - h(\varsigma, x) \right) \quad (19)$$

$$\dot{\varsigma} = \beta W(\varsigma) - \lambda \max\{\pi(c, Rx)\}, \quad (20)$$

where for brevity, we represent every variable in the vector form. Noting the fact that the rate subsystem is perturbed by  $\varsigma$ , whose dynamics is in turn perturbed by the EVQ algorithm with  $x$  as inputs, we can represent the fluid model as the *feedback interconnection to two subsystems* and will show its global asymptotic stability with the input-to-state (ISS) small gain theorem [20, 21].



**Figure 4: Feedback interconnection of rates dynamics and virtual queue dynamics.**

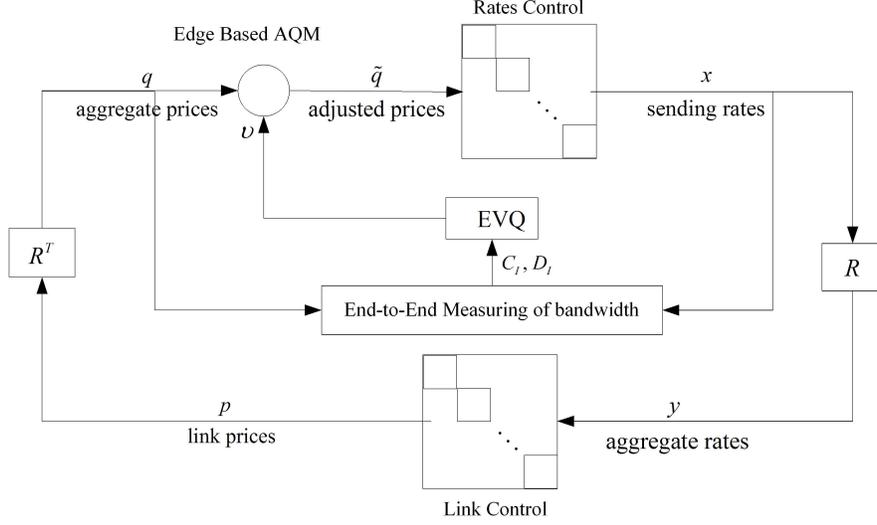


Figure 3: Fluid network model For EVQ

First we demonstrate that there exists a ISS gain  $\gamma_1$  from  $\tilde{x} = [\tilde{x}_1 \ \tilde{x}_2 \ \cdots \ \tilde{x}_N]^T$  to  $\tilde{\zeta} = [\tilde{\zeta}_1 \ \tilde{\zeta}_2 \ \cdots \ \tilde{\zeta}_N]^T$ , where  $\tilde{x}$  and  $\tilde{\zeta}$  are new, *transformed* state variables:  $\tilde{x}_i = \ln \frac{x_i}{x_i^*}$ ,  $\tilde{\zeta}_I = \ln \frac{\zeta_I}{\zeta_I^*}$  and where  $x_i^* > 0$ ,  $\zeta_I^* > 0$  are the equilibrium of source sending rates  $x_i$  and virtual path capacities  $\zeta_I$ . Then, we show another gain  $\gamma_2$  from  $\tilde{\zeta}$  to  $\tilde{x}$  for the rate subsystem. Finally, we show *global asymptotic stability from the ISS small gain condition*:

$$\gamma_1 \gamma_2 < 1. \quad (21)$$

Because  $\gamma_1$  and  $\gamma_2$  is independent of the number of flows and time delays, we conclude that when the small gain condition (21) is satisfied, global asymptotic stability is preserved:

**THEOREM 1.** *Consider the EVQ model where  $U_i(x_i)$  and  $W_I(\zeta_I)$  are as in (17) and (18). If the pricing function  $h(\cdot)$  is as in (11), then the network is globally asymptotically stable if*

$$n > 1, \quad m > \frac{1}{n-1} \quad (22)$$

*If the pricing function  $h(\cdot)$  is as in (12), then the network is globally asymptotically stable if*

$$n > \mu, \quad m > \frac{\mu}{n-\mu} \quad (23)$$

Due to space limits, we only prove the single case where the pricing function takes the exponential form as in (12) and each path only has only one flow. This proof can be generalized.

**Proof:** We use the small gain theorem to show the global asymptotic stability of the system:

Step 1: For the EVQ algorithm

$$\dot{\zeta} = \beta W'(\zeta) - \lambda \max\{\pi(c, Rx)\}, \quad (24)$$

We rewrite it in the following form:

$$\dot{\zeta} = \beta (W'(\zeta) - \eta) \quad (25)$$

$$\eta = \frac{\lambda}{\beta} \max\{\pi(c, y)\} \quad (26)$$

$$y = Rx. \quad (27)$$

From Lemma 1, 2 and 3, we have the ISS gain of 1 from  $\tilde{x}$  to  $\tilde{y}$ ; ISS gain of 1 from  $\tilde{y}$  to  $\tilde{\eta}$  and ISS gain of  $\frac{1}{m}$  from  $\tilde{\eta}$  to  $\tilde{\zeta}$ :

$$\|\tilde{y}(t)\|_\infty \leq \|\tilde{x}(t)\|_\infty \Rightarrow \|\tilde{y}(t)\|_{L_\infty} \leq \|\tilde{x}(t)\|_{L_\infty} \quad (28)$$

$$\|\tilde{\eta}\|_\infty \leq \|\tilde{y}\|_\infty \Rightarrow \|\tilde{\eta}\|_{L_\infty} \leq \|\tilde{y}\|_{L_\infty} \quad (29)$$

$$\|\tilde{\zeta}(t)\|_{L_\infty} \leq \max\left\{\|\tilde{\zeta}(0)\|, \frac{1}{m} \|\tilde{\eta}\|_{L_\infty}\right\} \quad (30)$$

$$\|\tilde{\zeta}(t)\|_a \leq \frac{1}{m} \|\tilde{\eta}\|_a. \quad (31)$$

Thus, the ISS gain from  $\tilde{x}$  to  $\tilde{\zeta}$  is

$$\gamma_1 = \frac{1}{m}, \quad (32)$$

which implies that:

$$\|\tilde{\zeta}(t)\|_{L_\infty} \leq \|\tilde{\zeta}(0)\| + \gamma_1 \|\tilde{x}\|_{L_\infty} \quad (33)$$

$$\|\tilde{\zeta}(t)\|_a \leq \gamma_1 \|\tilde{x}\|_a. \quad (34)$$

Step 2: As in Step 1, in order to use the lemmas in the Appendix to show that there exists a gain  $\gamma_2$  from  $\tilde{\zeta}$  to  $\tilde{x}$  for the rate subsystem (19), we represent it as

$$\dot{x} = \kappa(x) (U'(x) - q) \quad (35)$$

$$q = R^T p + v = \begin{bmatrix} R^T & I_{N \times 1} \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix} \quad (36)$$

$$\begin{bmatrix} p \\ v \end{bmatrix} = \begin{bmatrix} h(c, y) \\ h(\varsigma, x) \end{bmatrix} \quad (37)$$

$$\begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} R \\ I_{N \times 1} \end{bmatrix} x. \quad (38)$$

Following the same steps as in Step 1, we have

$$\|\tilde{x}\|_{L^\infty} \leq \|\tilde{x}(0)\| + \frac{\mu}{n} \|\tilde{x}\|_{L^\infty} + \frac{\mu}{n} \|\tilde{\zeta}\|_\infty \quad (39)$$

$$\|\tilde{x}(t)\|_a \leq \frac{\mu}{n} \|\tilde{x}\|_a + \frac{\mu}{n} \|\tilde{\zeta}\|_a, \quad (40)$$

which implies that the gain from  $\tilde{\zeta}$  to  $\tilde{x}$  is

$$\gamma_2 = \frac{\mu}{n - \mu}, \quad (41)$$

that is

$$\|\tilde{x}\|_{L^\infty} \leq (1 + \gamma_2) \|\tilde{x}(0)\| + \gamma_2 \|\tilde{\zeta}\|_\infty \quad (42)$$

$$\|\tilde{x}\|_a \leq \gamma_2 \|\tilde{\zeta}\|_a. \quad (43)$$

*Step 3:* Global asymptotic stability follows from the ISS small gain theorem because (23) implies

$$\gamma_1 \gamma_2 = \frac{\mu}{nm - m\mu} < 1. \quad (44)$$

□

Note that Theorem 1 does not take transmission and queuing delays into account. These time delays will mean that rates and virtual capacities get updated based upon delayed feedback, i.e.,  $x_\tau = x(t - \tau_1)$  and  $\varsigma_\tau = \varsigma(t - \tau_2)$  instead of  $x(t)$  and  $\varsigma(t)$ . Thus, the fluid model with time delays will be

$$\dot{x} = \kappa(x_\tau) \left( U'(x) - R^T h(c, Rx_\tau) - h(\varsigma_\tau, x_\tau) \right) \quad (45)$$

$$\dot{\varsigma} = \beta W(\varsigma) - \lambda \max\{\pi(c, Rx_\tau)\}. \quad (46)$$

However, *Theorem 1 still holds true in the time-delayed model* because delays do not introduce any extra gain to the system (time-delays have amplitude of 1). That is, the ISS gain from  $x_\tau$  to  $x$  is equal to 1, and so is the gain from  $\varsigma_\tau$  to  $\varsigma$ . Thus, global asymptotic stability holds for arbitrary number of flows and any amount of time delays.

### 3.3 Fairness Characteristics in Equilibrium

In this section, we characterize the equilibrium and show that EVQ displays max-min fairness when  $\lambda$  is arbitrarily large.

**THEOREM 2.** *In a multiple-bottleneck topology, if there exists a unique equilibrium, then the EVQ algorithms allocate a set of max-min fair path rates  $x_I$  when  $\lambda$  is arbitrarily large.*

*Sketch of Proof:* In the steady state, we already argue that when  $\lambda$  large enough,  $y_l < c_l$  for all link  $l$  and the virtual link at the edge of network is the only congested link along the path  $I$ , which implies

$$x_I^* = \varsigma_I^*. \quad (47)$$

Thus, at the equilibrium,

$$W_I'(x_I^*) = W_I'(\varsigma_I^*) = \frac{\lambda}{\beta} \max_{i \in T(I)} \{\pi_i^*\}, \quad (48)$$

which implies that a corresponding max price along the path is fed back, instead of their sum. Thus, using the same argument as in [22, 23], we claim that path rate  $x_I$  has max-min fairness.

## 4. SIMULATION RESULTS

This section presents MATLAB and ns-2 simulations.

### 4.1 MATLAB simulations

The objectives of the MATLAB fluid flow simulation are to:

- 1) Illustrate the stability and fairness of EVQ.
- 2) Investigate the sensitivity of EVQ to the damping factor parameter  $\lambda$ .
- 3) Study the robustness to random noise in path capacity and path demand estimates.

The link capacity  $C_l = 1$  for all  $l$ . The desired utilization  $\gamma = 0.9$ , and routing matrix  $R$  is

$$R = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad (49)$$

This routing matrix  $R$  corresponds to a multi-bottlenecked topology shown in Figure 5, where a long flow [S2-D2] (going through two bottlenecks) competes with two short flows [S0-D0] and [S1-D1] (each going through one bottleneck).

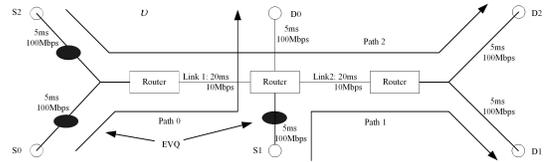
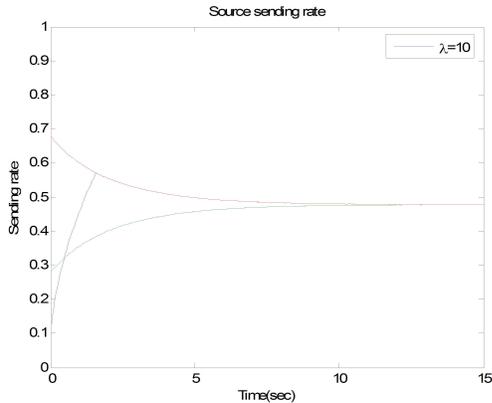


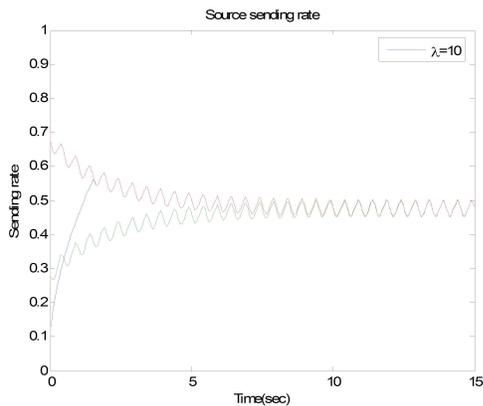
Figure 5: Topology used in the simulations.

Initial conditions (sending rates  $x_i$  and virtual capacity  $\varsigma_i$ ) are chosen randomly. The utility function  $W_i(\varsigma_i)$  in EVQ is  $-1/\varsigma_i$ ;  $U_i(x_i)$  is  $-1/x_i$ . Figure 5 shows the dynamics of sending rates of users when  $\lambda = 20$ . Although the fluid model is stable for arbitrary  $\lambda$ , a too large or too small value of  $\lambda$  will affect the transient performance. We choose  $\lambda$  in the range of 10 to 100 in all our simulations. The rate allocations shown in Figure 6 also confirms the max-min fairness [24], which implies every flow gets equal share of the bandwidth from the definition of max-min.



**Figure 6: User rates are max-min fair. The damping parameter  $\lambda$  is 10.**

To illustrate EVQ’s robustness, we introduce random noise to measurements, uniformly distributed between 0% to 10% of total capacity for both path capacity and path demand. Figure 6 suggests EVQ is robust to such short-time-scale measurement errors, which might be characterized under the framework of Noise-to-State stability known in control theory [25].



**Figure 7: Sending rates of users with capacity and demand measurement errors being up to 10% of total capacity.**

## 4.2 NS-2 simulations

The broad objectives of the ns-2 [26] simulations are as follows:

- 1) Performance evaluation: In the steady state, EVQ achieves low persistent queue length, low packet loss and acceptable utilization.
- 2) Fairness of EVQ: EVQ achieves max-min fairness instead of proportional fairness.

3) Comparison of EVQ with Drop-Tail and AVQ.

4) Measurement tradeoffs and sensitivity of EVQ to parameter configurations.

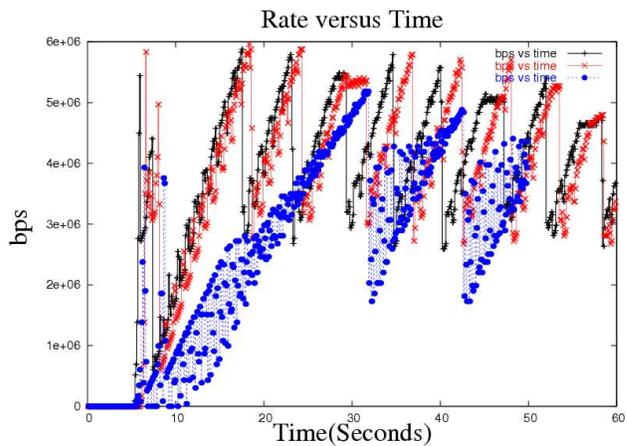
For all the results presented in this section we have used the CapProbe [8] and pathChirp [9] to estimate path capacity and available bandwidth. Throughout this section, we present simulation results for the multi-bottleneck topology, depicted in Figure 5, and assume that all the flows are persistent flows, i.e. they have infinite data to transfer. We assume that the link marks packets and thus, any packet loss is due to buffer overflow. All the access links are configured to have a capacity equal to 10 times that of bottleneck links. The bottleneck links capacity and delay is fixed at 100Mbps and 5ms respectively. Each router has a buffer equal to one bandwidth delay product. In setups, the bottleneck routers have Drop-Tail queue management deployed only.

### 4.2.1 Performance Evaluation of EVQ under NS-2

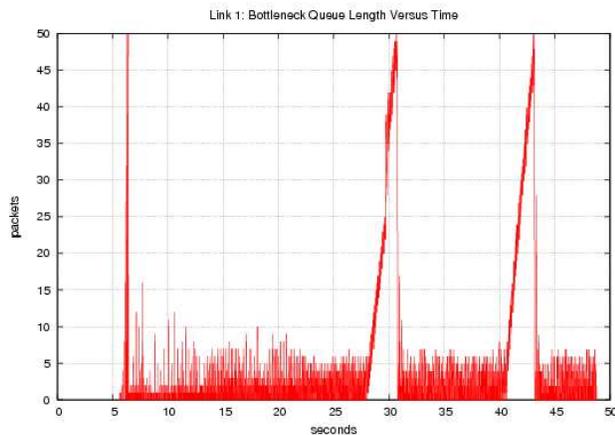
Figures 8 and 9 show the evolution of the queue length and the utilization at each router, respectively. We observe that in the steady state, EVQ has low persistent queue length and acceptable utilization. These results confirm the theoretical development of EVQ in Section 2, which mainly concentrate on the properties of the steady state response. However, we also note that our scheme is not optimized for the transient response of utilization. The oscillation around steady state and the long transience caused by some measurement artifacts need further investigations.

### 4.2.2 Fairness Characteristics of EVQ

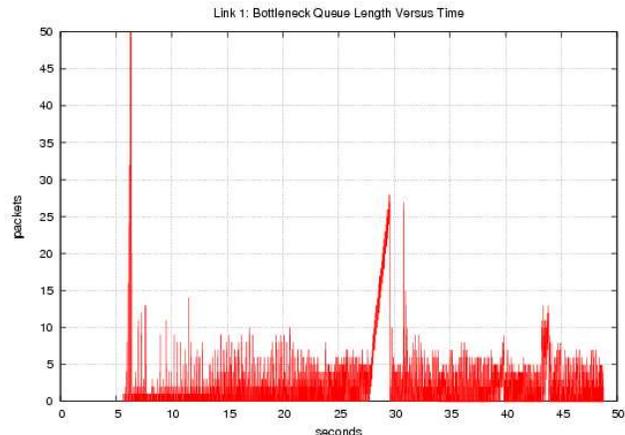
To evaluate the fairness characteristics of the EVQ, we can simply check the share of the bandwidth of all three flows. Figure 10 shows the sending rates for the three flows under consideration. Since all the three flows share the bandwidth equitably, thus demonstrating the max-min fairness of the algorithm [24].



**Figure 10: Sending rates (Mbps) vs time (seconds) for the EVQ (Simulation start at 5 second).**

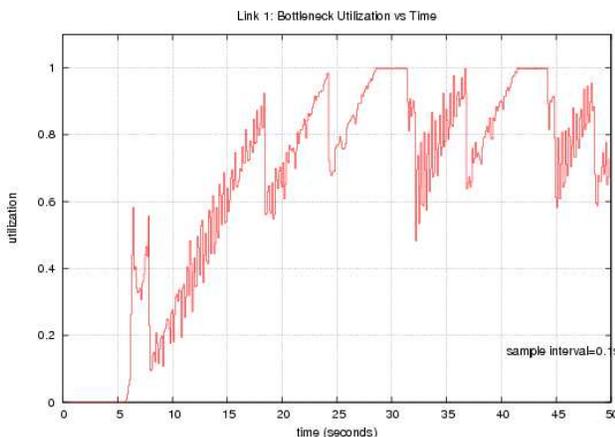


(a) Link 1

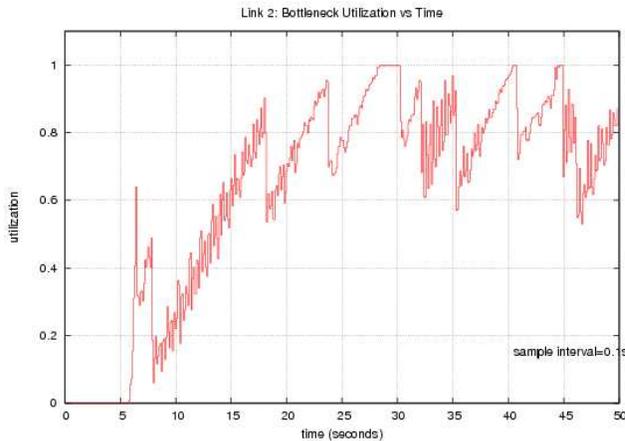


(b) Link 2

**Figure 8: Queue length (packets) vs time (seconds) for the EVQ (Simulation start at 5 second).**



(a) Link 1



(b) Link 2

**Figure 9: Link Utilization vs time (seconds) for the EVQ (Simulation start at 5 second).**

### 4.2.3 Comparison of EVQ with Drop-Tail and AVQ.

We compare EVQ with simple Drop-Tail and AVQ schemes. The desired utilization of the link is set to be 0.95 for both AVQ and EVQ schemes. Figures 11 and 12 show the evolution of the queue size for each of AQM scheme. Comparing them with Figure 8, we can see that as far as the queue length is concerned, EVQ performance is comparable with AVQ. Its transient performance, however, needs to study further.

### 4.2.4 Measurement Tradeoff and Parameter Sensitivity of EVQ

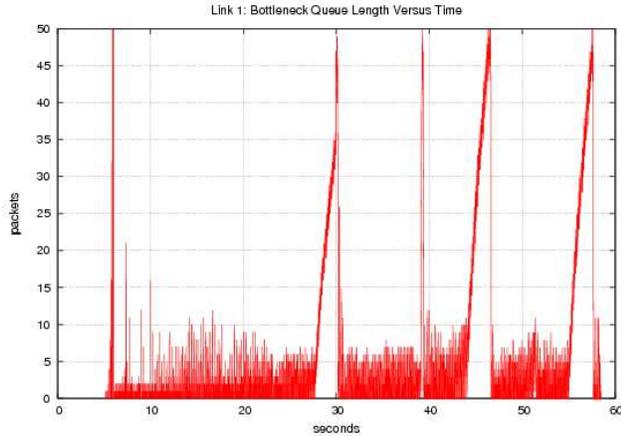
Although the implementation of EVQ does not depend upon the specific path measurement tools, however, poor estimation does affect the performance of EVQ. Figure 13 shows the steady queue length when the parameters of pathChirp are not well tuned as suggested in [9]. Thus, for real networks, the realization of the framework depends upon robust

and short-time-scale estimation of edge-to-edge parameters: path capacity and available bandwidth.

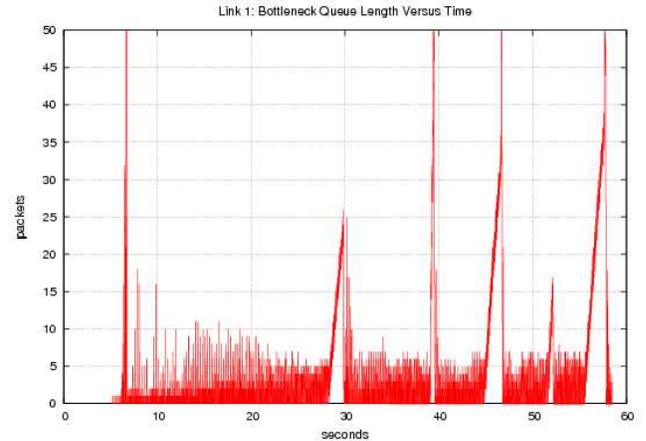
The ns-2 simulations also demonstrate again that the damping parameter  $\lambda$  is very crucial to the performance of EVQ (see Figure 14. For real networks, too large  $\lambda$  will result in aggressive dropping or marking so that utilization will be sacrificed. This dropping or marking saturation feature is not captured by the fluid model in Section 2. On the other hand, too small  $\lambda$  will lessen the functionality of EVQ and the performance of the network will be back to the one with TCP and simple Drop-tail. Through our simulations, the desire  $\lambda$  lies in  $[10, 100]$ .

## 5. CONCLUSION

This paper focussed on a theoretical answer to an architectural question: can the active queue management (AQM) function be moved to the edges of the network so that the core routers focus on packet forwarding? The novel ideas

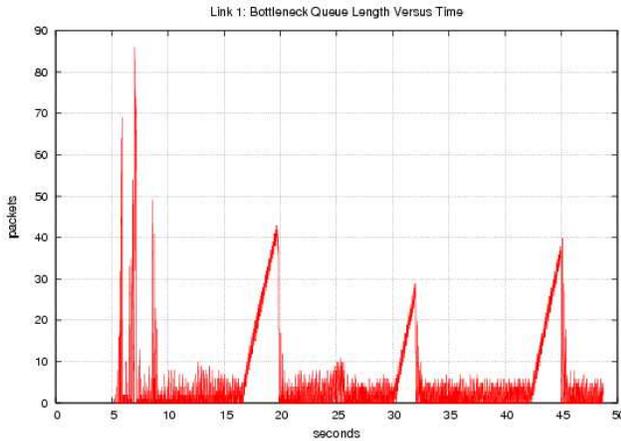


(a) Link 1

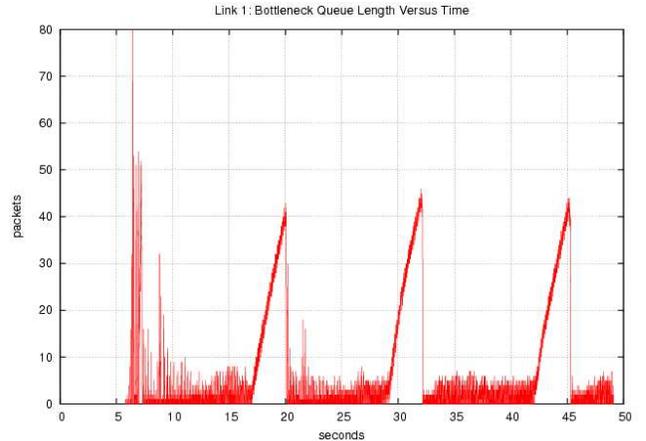


(b) Link 2

Figure 11: Evolution of the Queue Length (packets) for Drop-Tail at all links (Simulation start at 5 second).



(a) Link 1



(b) Link 2

Figure 12: Evolution of the Queue Length (packets) for AVQ at all links (Simulation start at 5 second).

and observations in this work include:

1. The idea that even an *artificial* measure of "worst" path capacity and "path demand" can be used as a proxy at network edges to construct an AQM scheme (called EVQ),
2. Such a scheme is global asymptotically stable, and achieves max-min fairness (instead of the usual proportional fairness),
3. we need to use a path utility function and a barrier function in the design in addition to the regular AVQ scheme (i.e. it is a hybrid of source-like controller and a link controller)
4. the time-scale and robustness of path capacity and available bandwidth estimation affects transient performance and steady state behavior.
5. Movement of functions in the network architecture can be facilitated by measurement advances. Though we

use current active measurement techniques (eg: PathChirp, CapProbe), the availability of more robust, shorter-time-scale estimators in the future will aid the implementation of EVQ.

This paper has focused on theoretical development of the edge-based AQM concepts. The ns-2 simulations presented are preliminary and illustrate the steady state behavior of our protocol realization, subject to measurement constraints. The design of better transient performance and superior robustness in the protocol realizations will be the focus of future work.

## 6. APPENDIX

In this section, we review the nonlinear control concepts used in the stability proof of EVQ fluid model and prove three important Lemmas of Theorem 1 in Section 3.

**Notation and definitions used in the proof:**

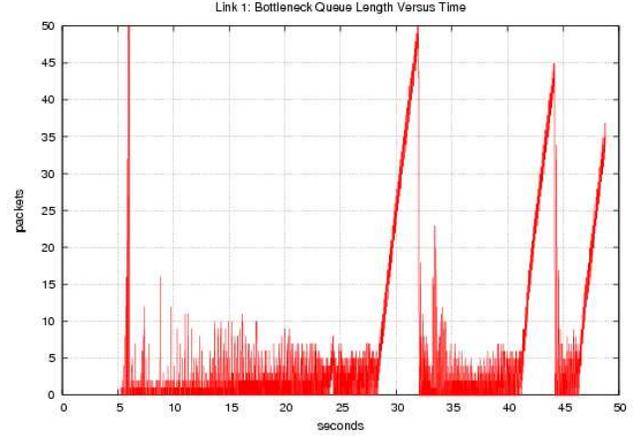
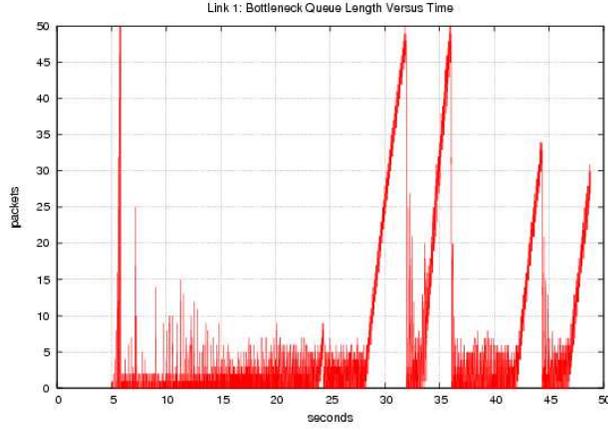
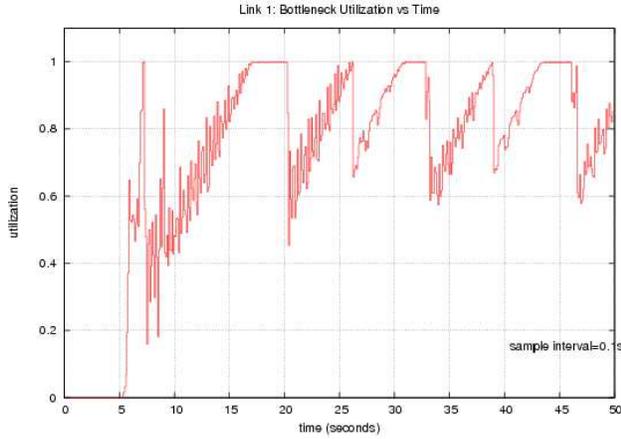
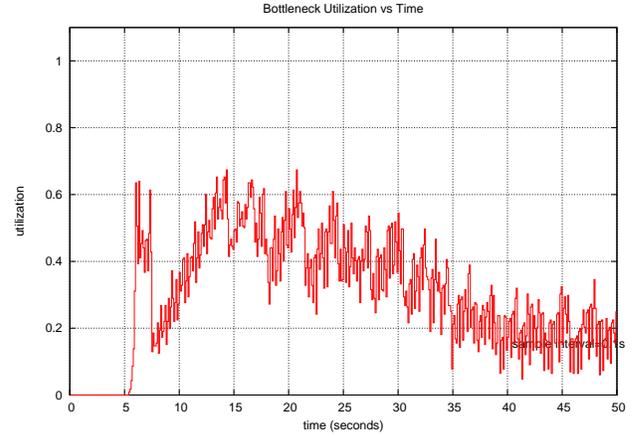


Figure 13: Evolution of the Queue Length (packets) when pathChirp is not well tuned (Simulation start at 5 second).



(a)  $\lambda = 5$



(b)  $\lambda = 500$

Figure 14: Evolution of the Queue Length (packets) for different  $\lambda$  (Simulation start at 5 second).

1) We denote by  $\|\cdot\|_p$  the  $p$ -norm of vectors and induced  $p$ -norm of matrices. Whenever the choice of  $p$  is unimportant, we drop the subscript. We let  $\|x\|_{L_p}$  denote the  $L_p$ -norm of  $x(t)$  on the interval  $[0, \infty)$ ,  $p \in (0, \infty]$ . For  $x \in L_\infty$ , we define  $\|x\|_a = \limsup_{t \rightarrow \infty} \|x(t)\|$ .

2) A function  $\gamma(\cdot) : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  is defined to be class- $K$  if it is continuous, zero at zero, and strictly increasing. It is said to be class- $K_\infty$  if it is class- $K$ , and grows unbounded.

3) A system  $\dot{x} = f(x, u)$  is said to be *input-to state stable* (ISS) if there exist class- $K_\infty$  functions  $\gamma_0(\cdot)$  and  $\gamma(\cdot)$  such that, for any input  $u(\cdot) \in L_\infty^m$  and  $x_0 \in \mathbb{R}^n$ , the response  $x(t)$  from the initial state  $x(0) = x_0$  satisfies

$$\|x\|_{L_\infty} \leq \max \gamma_0(\|x_0\|) + \gamma(\|u\|_{L_\infty}), \quad \|x\|_a \leq \gamma(\|u\|_a).$$

The function  $\gamma(\cdot)$  is referred to as the ISS gain.

LEMMA 1. For the rate subsystem system

$$\dot{x}_i = \kappa_i(x_i)(U_i(x_i) - q_i)_{x_i}^+, \quad (50)$$

where  $i = 1, \dots, N$ ,  $U_i(x_i) = \frac{x_i^{1-n}}{1-n}$  and  $(\cdot)_{x_i}^+$  is the project function to bound  $x_i$  in  $[x_{i \min}, x_{i \max}]$ , the ISS-gain from  $\tilde{q}_i = \ln \frac{q_i}{\tilde{q}_i}$  to  $\tilde{x}_i = \ln \frac{x_i}{\tilde{x}_i}$  is  $\frac{1}{n}$ , specifically

$$\|\tilde{x}_i(t)\|_{L_\infty} \leq \max \left\{ \|\tilde{x}_i(0)\|, \frac{1}{n} \|\tilde{q}_i(t)\|_{L_\infty} \right\} \quad (51)$$

$$\|\tilde{x}_i(t)\|_a \leq \frac{1}{n} \|\tilde{q}_i\|_a. \quad (52)$$

When  $L_\infty$  norm is define as  $\|\tilde{x}(t)\|_{L_\infty} = \sup_{t \geq 0} \|\tilde{x}(t)\|_\infty = \sup_{t \geq 0} \max |\tilde{x}_i(t)|$ , we also have

$$\|\tilde{x}(t)\|_{L_\infty} \leq \max \left\{ \|\tilde{x}(0)\|, \frac{1}{n} \|\tilde{q}\|_{L_\infty} \right\} \quad (53)$$

$$\|\tilde{x}(t)\|_a \leq \frac{1}{n} \|\tilde{q}\|_a, \quad (54)$$

where  $\tilde{x} = [\tilde{x}_1 \ \tilde{x}_2 \ \cdots \ \tilde{x}_N]^T$  and  $\tilde{q} = [\tilde{q}_1 \ \tilde{q}_2 \ \cdots \ \tilde{q}_N]^T$ .

**Proof:** With the nonlinear state transformation  $\tilde{q}_i = \ln \frac{q_i}{q_i^*}$  to  $\tilde{x}_i = \ln \frac{x_i}{x_i^*}$ , we obtain the transformed system as

$$\dot{\tilde{x}}_i = \left( \frac{1}{x_i^{n+1}} \left( 1 - e^{\tilde{q} + n\tilde{x}_i} \right) \right)_x^+ \quad (55)$$

The derivative of the ISS Lyapunov function [27]  $V_i(\tilde{x}_i) = \frac{1}{2} \tilde{x}_i^2$  along its trajectory is

$$\dot{V}_i = \tilde{x}_i \dot{\tilde{x}}_i = \frac{1}{x_i^{n+1}} \tilde{x}_i \left( 1 - e^{\tilde{q} + n\tilde{x}_i} \right). \quad (56)$$

When  $|\tilde{q}_i| \leq (1 - \beta)n|\tilde{x}_i|$ , where  $0 < \beta < 1$

$$\begin{aligned} \dot{V}_i &= \tilde{x}_i \dot{\tilde{x}}_i = \frac{1}{x_i^{n+1}} \tilde{x}_i \left( 1 - e^{\tilde{q} + n\tilde{x}_i} \right) \\ &\leq \frac{1}{x_{i\_max}^{n+1}} \frac{|\tilde{x}_i|}{e^{\beta|\tilde{x}_i|}} \left( 1 - e^{\beta n|\tilde{x}_i|} \right). \end{aligned} \quad (57)$$

Thus, we obtain:

$$|\tilde{x}_i(t)| \leq \beta (|\tilde{x}_i(0)|, t) + \frac{1}{n} |\tilde{q}_i|_{L_\infty}, \quad (58)$$

which implies (51) and (52). (53) and (54) follows from the definition of  $L_\infty$  norm.

LEMMA 2. For the linear mapping functions  $y_l = \sum_{i \in L(l)} x_i + \delta_l$  and  $y_l^* = \sum_{i \in L(l)} x_i^* + \delta_l$ , where  $x_i \geq 0$ ,  $\delta \geq 0$  and  $x_i^* > 0$ , the infinity gain from  $\tilde{x}_i = \ln \frac{x_i}{x_i^*}$  to  $\tilde{y}_i = \ln \frac{y_i}{y_i^*}$  is 1.

**Proof:** Noting that

$$\tilde{y}_i = \ln \frac{y_i}{y_i^*} = \ln \frac{\sum_{i \in L(l)} x_i + \delta_l}{\sum_{i \in L(l)} x_i^* + \delta_l} \quad (59)$$

and

$$\ln \frac{\sum_{i \in L(l)} x_i + \delta_l}{\sum_{i \in L(l)} x_i^* + \delta_l} \leq \ln \max \left\{ \max_{i \in L(l)} \left\{ \frac{x_i}{x_i^*} \right\}, 1 \right\} \quad (60)$$

$$= \max \left\{ \max_{i \in L(l)} \ln \left\{ \frac{x_i}{x_i^*} \right\}, 0 \right\}$$

$$\ln \frac{\sum_{i \in L(l)} x_i + \delta_l}{\sum_{i \in L(l)} x_i^* + \delta_l} \geq \ln \min \left\{ \min_{i \in L(l)} \left\{ \frac{x_i}{x_i^*} \right\}, 1 \right\} \quad (61)$$

$$\geq \min \left\{ \min_{i \in L(l)} \ln \left\{ \frac{x_i}{x_i^*} \right\}, 0 \right\},$$

we obtain,

$$\begin{aligned} \left| \ln \frac{\sum_{i \in L(l)} x_i}{\sum_{i \in L(l)} x_i^*} \right| &\leq \max \left\{ \left| \max_{i \in L(l)} \ln \left\{ \frac{x_i}{x_i^*} \right\} \right|, \left| \min_{i \in L(l)} \ln \left\{ \frac{x_i}{x_i^*} \right\} \right| \right\} \\ &= \max_{i \in L(l)} \left| \ln \left\{ \frac{x_i}{x_i^*} \right\} \right|. \end{aligned} \quad (62)$$

That is,  $|\tilde{y}_l(t)| \leq |\tilde{x}|_\infty$ . Thus,

$$|\tilde{y}|_\infty \leq |\tilde{x}|_\infty. \quad (63)$$

LEMMA 3. For the nonlinear mapping functions  $q_i = \max_{l \in I(i)} \left( \frac{y_l}{c_l} \right)^\mu$  and  $q_i^* = \max_{l \in I(i)} \left( \frac{y_l^*}{c_l^*} \right)^\mu$ , where  $q_i \geq 0$ ,  $q_i^* > 0$ ,  $c_l^* > 0$ ,  $c_l \geq 0$ , and  $\mu > 0$ ,

$$|\tilde{q}|_\infty \leq \mu |\tilde{y}|_\infty + \mu |\tilde{c}|_\infty. \quad (64)$$

Proof is omitted due to simplicity.

## 7. REFERENCES

- [1] C.V. Hollot, V. Misra, D. Towlsey, and W. Gong, On designing improved controllers for AQM routers supporting TCP flows, in Proceedings of INFOCOM, Alaska, Anchorage, April 2001.
- [2] S. Kunniyur and R. Srikant. Analysis and design of an adaptive virtual queue algorithm for active queue management, IEEE ACM Transactions on Networking, April 2004, pp. 286-299. An earlier version appeared in Proc. ACM Sigcomm 2001.
- [3] T. J. Ott, T. V. Lakshman, and L. H. Wong, SRED: Stabilized RED, in Proceedings of INFOCOM, New York, NY, March 1999.
- [4] W. Feng, D. Kandlur, D. Saha, and K. Shin, Blue: A new class of active queue management schemes, April 1999, Technical Report, CSE-TR-387-99, U. Michigan.
- [5] R.J. Gibbens and F.P. Kelly, Distributed connection acceptance control for a connectionless network, in Proc. of the 16<sup>th</sup> Intl. Teletraffic Congress, Edinburgh, Scotland, June 1999.
- [6] S. Athuraliya, D. E. Lapsley, and S. H. Low, Random early marking for Internet congestion control, in Proceedings of Globecom, 1999.
- [7] S. Floyd and V. Jacobson, Random early detection gateways for congestion avoidance, IEEE/ACM Transactions on Networking, August 1993.
- [8] Rohit Kapoor, Ling-Jyh Chen, Li Lao, Mario Gerla, and M. Y. Sanadidi. CapProbe: A Simple and Accurate Capacity Estimation Technique. ACM SIGCOMM 2004, Portland, USA, 2004.
- [9] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell. pathChirp: efficient available bandwidth estimation for network paths, Passive and Active Measurement Workshop, 2003.

- [10] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability, *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.
- [11] D. Bertsekas and R. Gallager. Data Networks. 2nd Ed., Simon & Schuster, December 1991.
- [12] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability, *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.
- [13] D.J. Hill and P.J. Moylan. Stability results for nonlinear feedback systems. *Automatica*, 13:377–382, 1977.
- [14] F. Paganini, J. Doyle, and S. Low, Scalable laws for stable network congestion control, *Proceedings of 2001 Conference on Decision and Control*, Orlando, FL, Dec. 2001, pp. 185-190.
- [15] C.I. Byrnes, A. Isidori, and J.C. Willems. Passivity, feedback equivalence, and global stabilization of minimum phase systems. *IEEE Transactions on Automatic Control*, 36:1228–1240, 1991.
- [16] F. Paganini, A global stability result in network flow control, *Systems and Control Letters*, vol. 46, pp. 165-172, 2002.
- [17] J. Wen and M. Arcak, A unifying passivity framework for network flow control, *IEEE Transactions on Automatic Control*, vol. 49, no. 2, pp. 162–174, 2004.
- [18] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability, *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.
- [19] G. Vinnicombe. On the stability of networks operating TCP-like congestion control. In Proceedings of the IFAC World Congress, Barcelona, Spain, 2002. University of Cambridge Technical Report CUED/F-INFENG/TR.398. Available at <http://www.eng.cam.ac.uk/gv>.
- [20] A. Teel, A nonlinear small gain theorem for the analysis of control systems with saturation, *IEEE Transactions on Automatic Control*, vol. 41, no. 9, pp. 1256–1271, 1996.
- [21] Z.-P. Jiang, and A. Teel, and L. Praly, Small-gain theorem for ISS systems and applications, *Mathematics of Control, Signals, and Systems*, Vol.7, 1994, 95-120
- [22] Bartek Wydrowski and Moshe Zukerman, MaxNet: A congestion control architecture for maxmin fairness, *IEEE Communications Letters*, vol. 6, no. 11, Nov. 2002, pp.512-514.
- [23] Yong Xia, Lakshminarayanan Subramanian, Ion Stoica and Shivkumar Kalyanaraman, One more bit is enough, ACM SIGCOMM, Philadelphia, PA, Aug. 22, 2005
- [24] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. Proceedings of the SPIE, vol.3530:55-63, Nov. 1998.
- [25] Hua Deng and, Miroslav Krstic. "Output-feedback stabilization of stochastic nonlinear systems driven by noise of unknown covariance". *Systems & Control Letters* 39 (2000) 173-182.
- [26] ns2 (online), <http://www.isi.edu/nsnam/ns>.
- [27] H. Khalil, *Nonlinear Systems*, 2nd ed. Englewood Clis, NJ: Prentice Hall, 1996.