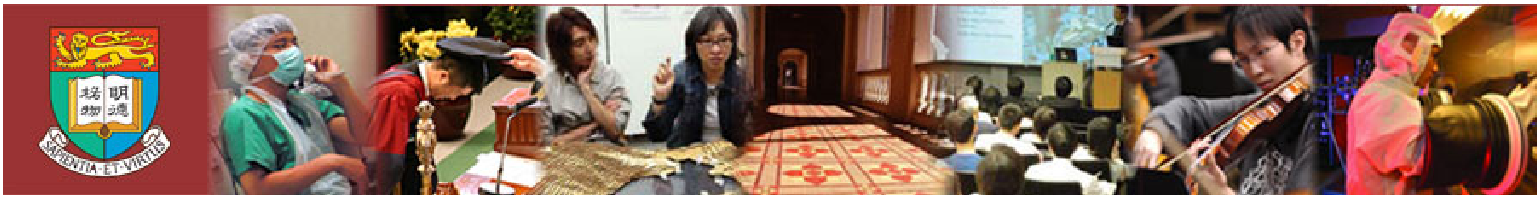


Enhancing AQM to Combat Wireless Losses

Chengdi Lai, Ka-Cheong Leung, and Victor O.K. Li
Department of Electrical and Electronic Engineering
The University of Hong Kong
Pokfulam Road, Hong Kong, China
E-mail: {laichengdi, kcleung, vli}@eee.hku.hk

***Abstract*—In order to maintain a small, stable backlog at the router buffer, active queue management (AQM) algorithms drop packets probabilistically at the onset of congestion, leading to backoffs by Transmission Control Protocol (TCP) flows. However, wireless losses may be misinterpreted as congestive losses and induce spurious backoffs. In this paper, we raise the basic question: Can AQM maintain a stable, small backlog under wireless losses? We find that the representative AQM, random early detection (RED), fails to maintain a stable backlog under time-varying wireless losses. We find that the key to resolving the problem is to robustly track the backlog to a preset reference level, and apply the control-theoretic vehicle, *internal model principle*, to realize such tracking. We further devise the integral controller (IC) as an embodiment of the principle. Our simulation results show that IC is robust against time-varying wireless losses under various network scenarios.**

I. Introduction

Congestion control regulates the amount of data traffic injected by end systems into communication networks, preventing persistent network overloading. In the Internet, it is a task typically implemented jointly by Transmission Control Protocol (TCP) and the queue management algorithms in a distributed manner. Running at the end systems, TCP treats packet losses as a signal of network overloading, and backs off when detecting any losses. Running at an intermediate router, the queue management algorithm monitors the queue length of a router buffer, and drops packets based on buffer occupancy.

Active queue management (AQM) is a class of queue management algorithms first proposed in the 1990s [4], [10]. Contrary to the traditional queue management algorithms that do not drop packets until buffer overflow, AQM probabilistically drops packets before buffer overflow based on a dropping rate determined from the past and/or present queue lengths. This keeps the backlog at router buffers small and desynchronizes the backoffs of the end systems. When operating effectively, AQM stabilizes the packet queue around a low level, so that 1) the end-to-end delay can be reduced and the delay jitter can be smoothed out; 2) sufficient buffer space is maintained to absorb bursty traffic, and 3) the bottleneck link is kept backlogged and thus fully utilized.

The next generation network is expected to be a heterogeneous network of networks, including both wired and wireless components. Wireless networks may extend beyond access networks to backhaul networks [13], and even backbone networks.

The characteristics of wireless links pose great challenges to the existing congestion control mechanisms, which are designed based on certain principles that may not hold in the wireless environment [17]. Most notably, signals propagating over wireless links are subject to severe interference, noise, and propagation loss. Packets transmitted over wireless links may be damaged to an extent beyond the recovery capability of error control codes (if any), and are thus discarded. This constitutes another cause of packet loss, in addition to congestive loss. The implications of wireless losses on the design of congestion control are two-fold. First, TCP tends to misinterpret wireless losses as congestive losses and backs off unnecessarily, thereby possibly underutilizing the network capacity. The problem has motivated numerous TCP variants in the literature and is not the focus of this paper.

Second, the extra source of packet losses may interfere with the normal operation of AQM, which communicates implicitly with TCP via active packet drops. When many TCP flows are sharing a bottleneck wireless link, it is unlikely that a large fraction of them will experience wireless losses simultaneously (unless a link is broken). Thus, spurious backoffs due to wireless losses would affect the aggregate transmission of all flows much less severely than when there are only a few flows. The wireless link will still be kept backlogged, as shown by our simulation results.[1] Yet, it casts doubt on the following basic question (BQ):

(BQ) Can AQM maintain a stable, small backlog under wireless losses?

The question is largely unexplored in the literature, and is the focus of this work. In our simulation experiments, we find that random early detection (RED) [4], one of the most representative AQM algorithms, fails to maintain a stable backlog under wireless losses.

Apparently, the problem may be made trivial by applying one of the following two approaches:

1. Design a TCP enhancement that can *fully* differentiate between congestive and wireless losses. Unfortunately, this is hardly attainable due to the limited network information available to the end systems. Our simulation experiments demonstrate that the performance problem of RED persists when the end systems are running wireless TCP enhancements like TCP-NCL [16], which has been shown to be very successful in differentiating congestive and wireless losses.

2. Use packet marking instead of packet dropping to signal network congestion, such as [3]. However, we argue that a globally enabled marking scheme is impractical due to heterogeneity of the Internet.[2]

We therefore opt to tackle the problem through the *loss-*

[1] The observation has also been reported in the literature and shaped the design principles of several enhancements for adapting TCP to high-speed and/or wireless networks. Interested readers can refer to [6], [25] and the references therein.

[2] For example, explicit congestion notification (ECN) [22] is the most widely adopted marking scheme, but is generally disabled by default in several versions of Microsoft Windows [7].

978-1-4673-1298-1/12/$31.00 © 2012 IEEE

based AQM enhancements, which make use of packet drops for signalling network congestion. We model the dynamics of the congestion control system around its equilibrium by a linear control system, and study the interaction of wireless losses and AQM in this control-theoretic framework. By employing the queue size as the system output, wireless losses are considered as *disturbances* to the system.

Furthermore, we propose a family of solutions for stabilizing the queue size against wireless losses based on the *internal model principle* [5]. The *internal model principle* rejects disturbances, or forces the part of the output due to disturbances to asymptotically converge to zero, by including a *model* of the disturbances, or poles of the disturbances, as part of the feedback control loop. In the case of the congestion control system, it can be realized by simply modifying the formula of the packet dropping rate in AQM. The application of the *internal model principle* in combating wireless losses offers the following advantages:

1. It enables AQM to adjust the packet dropping rate to compensate for the packet drops due to wireless losses without any specific knowledge about the wireless packet error rate (WPER);

2. It is realized as an independent module in AQM, and thus offers good extensibility to incorporate other AQM enhancements, such as [26], for improving inter-flow fairness in ad-hoc networks.

By assuming the intensity of wireless losses to be slowly time-varying, we devise the integral controller (IC) as an embodiment of the internal model principle. We have developed the design rules for IC. Our simulation results show that IC is capable of maintaining stable queue sizes and performing effective congestion avoidance against wireless losses. Its performance is maintained in the presence of perturbing traffic flows (e.g. HTTP traffic).

This paper is organized as follows. Section II reviews the related work in the literature and puts our work in perspective. Section III discusses BQ using several simulated examples, and exposes the performance problem of RED. Section IV presents our system model, a linear control system obtained by linearizing the non-linear control system of TCP/AQM, as the basis for our analysis. Section V proposes a family of solutions for enhancing AQM to combat wireless losses based on the internal model principle, including IC. The design rules are developed for IC. Section VI presents our simulation results. Section VII concludes the paper and discusses some possible extensions of our work.

Notation conventions: we use bold fonts to denote matrices and vectors, as in "$\mathbf{x}$(t)", except for the diagonal matrix Λ. When the time index is not specified, such as "$\mathbf{x}$", we refer to an equilibrium value. We use $\tilde{x}(t)$ and $\tilde{\mathbf{x}}(t)$ to represent $(x(t) - x)$ and $(\mathbf{x}(t) - \mathbf{x})$, respectively. $\lambda(.)$ and $\sigma(.)$ denotes the eigenvalue and singular value of a matrix, respectively.

II. Related Work

In this section, we summarize the related work on the design and analysis of AQM, focusing on those applying the control-theoretic approaches and/or adapting AQM to wireless networks.

First proposed in 1993, RED [10] demonstrates the inherent advantage of AQM in helping the network operate in the optimal region of high throughput and low delay. However, parameter tunings and new variants of RED invariably adopted the trial-and-error approach due to the difficulties in understanding the dynamics of TCP/AQM.

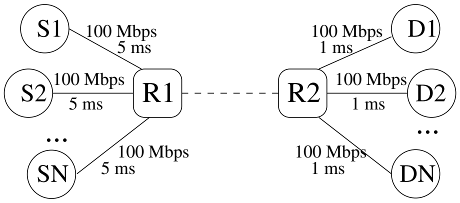


Fig. 1. A wireless bottleneck link.

The fluid models of TCP/AQM, as presented in, for example, [14], [19], [21], provide a basis for systematic design and analysis of AQM. The optimization-based approach [14], [18] interprets TCP/AQM as a distributed algorithm for solving a utility maximization problem, subject to capacity constraints. The major focus is on the optimal solutions attained at the equilibria. However, transient responses of AQM are mostly ignored.

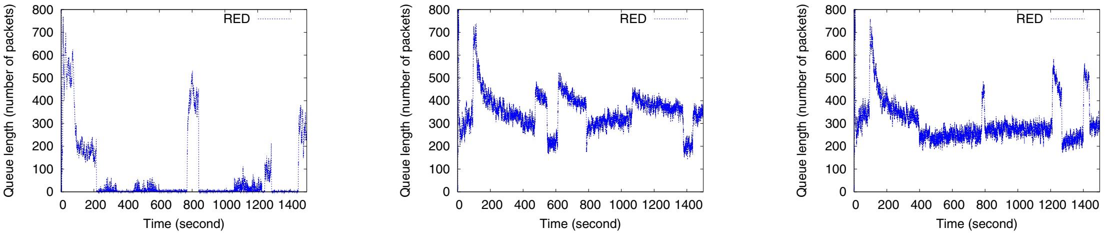
The control-theoretic approach views Internet congestion control as a non-linear control system. The system is first linearized around its equilibrium so as to analyze the dynamics of TCP/AQM around the equilibrium. This enables parameter tuning of RED and design of new AQM algorithms based on frequency-domain analysis [11], [12], [20], which is powerful in improving the transient response of AQM and guaranteeing the stability of the linear system. Interested readers can refer to [23] for a survey. The work in this area closest to our study is [11], which designs a proportional-integral (PI) controller for the general network topology. Generally, IC in our work can be considered a special case of the PI controller with the proportional part being set to zero. However, the PI controller considered in [11] assumes the proportional part to be non-zero, so that it cannot be reduced to IC. Moreover, we apply IC to combat wireless losses, whereas [11] focuses on designing a PI controller in wired networks.

The *global stability* and *region of attraction* of Internet congestion control have been studied in the context of a non-linear system. Please refer to [8], [27] and the references therein. In a nutshell, these are concerned with whether the system will converge to the equilibrium starting from a feasible initial status not necessarily close to the equilibrium [15]. Among them, Fan et al. [8] examined the robustness of the network flow control against disturbances. Yet, the AQM in [8] is *static rate-based*, which determines the packet dropping rate as a function of the instantaneous incoming data rate. Our work considers *buffer-based* AQM variants, which are widely used in the Internet to determine the packet dropping rate from the queue length.

In adapting AQM to wireless networks, the inter-flow fairness in ad-hoc networks can be improved [26] by virtually aggregating packets queued in those interfering nodes to one queue, and applying AQM to manage the queue. The fairness issue is complementary to the focus of this work, i.e. stability and robustness against wireless losses. It is also worth noting that our proposal shares the common feature of AQM that packets are dropped roughly proportionally to the bandwidth share of each flow, thereby improving fairness over the traditional queue management algorithms on the per-link basis. Moreover, as noted earlier in Section I, it offers good extensibility to accommodate enhancements, such as [26].

III. AQM under Wireless Losses

In this section, based on a simulation study, we discuss BQ raised in Section I: *Can AQM maintain a stable, small backlog*

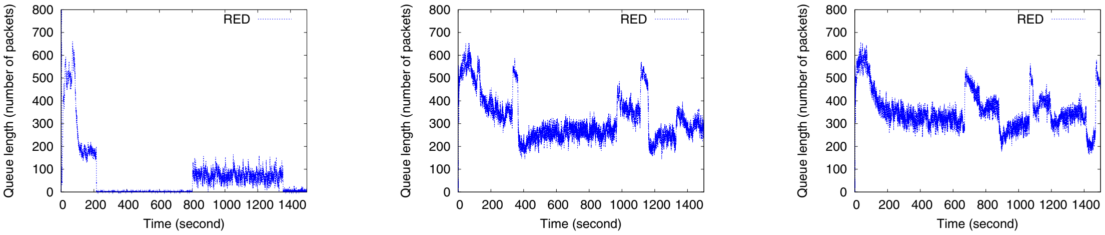


(a) $N = 20$, $M = 1$.

(b) $N = 100$, $M = 1$.

(c) $N = 20$, $M = 5$.

Fig. 2. Queue length dynamics under varying wireless losses in the dumbbell topology with TCP NewReno/RED.



(a) $N = 20$, $M = 1$.

(b) $N = 100$, $M = 1$.

(c) $N = 20$, $M = 5$.

Fig. 3. Queue length dynamics under varying wireless losses in the dumbbell topology with TCP-NCL/RED.

under wireless losses?

We use the simulation topology as exhibited in Fig. 1, similar to [1], [6]. It models the scenario in which two wired LANs are connected via a full duplex wireless link. A total of N pairs of source and destination are connected via the wireless bottleneck link with capacity 15 Mbps, and propagation delay 50 ms. Within each pair (S_i, D_i), M long-lived TCP flows are set up from S_i to D_i. The flows are simulated for 1400 seconds. The size of a link buffer is 800 packets. We vary WPER of the bottleneck link. Time is divided into periods, each of exponential distribution, with a mean of 100 seconds. At the beginning of each period, WPER of the bottleneck link is determined from a uniform distribution over $[0, 4\%]$, and is then kept constant during that period.[3]

RED is employed as the AQM algorithm in this simulation study. The parameters are set based on the result of the frequency domain analysis in [12].

We first consider the following question: *Is a wireless link backlogged under wireless losses?* Fig. 2 shows the queueing dynamics when the end systems are running TCP NewReno [9], a standardized TCP variant. When there are only 20 flows traversing the bottleneck link as shown in Fig. 2(a), spurious backoffs of some flows due to wireless losses affect the aggregate transmission of the flows considerably. The aggregate connection goodput is measured to be around 12 Mbps, or 80% of the link capacity, and the backlog is zero most of the time. In this case, the major concern is to enhance wireless TCP for improving link utilization.

When the number of flows increases to 100 by increasing either the number of source-destination pairs N, or the number of flows per pair M, the aggregate transmission of the flows is able to keep the bottleneck link backlogged and fully utilized, despite the presence of wireless losses. This is clearly exhibited in Fig. 2(b),(c). A large N corresponds to a large wired LAN. A large M can correspond to multiple simultaneous file transfers, or TCP enhancements that use parallel TCP sockets for better performance over wireless networks, such as [6]. In this case, our major concern shifts from link utilization to the backlog dynamics under wireless losses. We note that the backlog fluctuates severely across time due to the time-varying wireless losses. Thus, *the answer to BQ for RED is negative.*

To investigate the cause of the backlog fluctuations under time-varying wireless losses, we have conducted another set of simulations under *time-invariant* wireless losses. We have observed that different WPERs essentially cause the backlog to converge to different levels. Thus, time-varying wireless losses effectively force the backlog to switch among different equilibrium levels from time to time. Moreover, the sluggish transient response of RED in attaining such switching induces further backlog fluctuations.

Now, we explore some feasible directions for reducing the observed backlog fluctuations. If TCP can be enhanced to fully differentiate between congestive losses and wireless losses, the impact of wireless losses on the congestion control system will be minimal and the fluctuations can possibly be reduced. Fig. 3 exhibits the queueing dynamics when the end systems are running TCP-NCL, a wireless TCP enhancement that has been shown to very successful in differentiating between congestive and wireless losses. When there are only 20 flows as shown in Fig. 3(a), we note that TCP-NCL has improved the link utilization over TCP NewReno. The link is backlogged from the 800th second to the 1300th second, and the aggregate connection goodput is measured to be around 14 Mbps, or 93% of the link capacity.

When the number of flows increases to 100 as shown in Fig. 3(b),(c), the link is always backlogged. Again, we note that the backlog fluctuates severely. Although we can enhance TCP to attain more accurate differentiation between congestive losses and wireless losses, the accuracy in differentiation is nevertheless constrained by the limited information on the network status available to end systems. On the other hand, the equilibrium backlog level is very sensitive to the change in WPER. The fluctuation will persist even when only a small fraction of wireless losses are misinterpreted by TCP as congestive losses.

Therefore, *the backlog fluctuation cannot be reduced signifi-*

[3] We present simulation results with more exhaustive parameter settings in Section VI. Nevertheless, the discussion in this section generally applies.

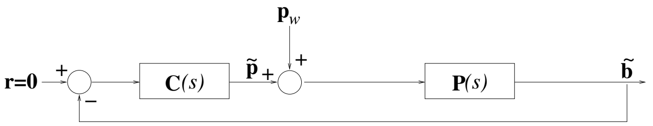


Fig. 4. Control block diagram of the linearized congestion control system.

cantly by simply enhancing TCP only. The key to maintaining a small, stable backlog under wireless losses is for AQM to direct the backlog to track a fixed reference level despite the variations in WPER. We now present a systematic approach to achieve this goal.

IV. System Model

In this section, we present our system model. We start by constructing a non-linear system of TCP/AQM. We then linearize the system around its equilibrium and develop its frequency domain representations. The latter will serve as the basis for further analysis in later sections.

We begin by considering a generic communication network with a set of TCP flows F and a set of bottleneck links L.[4] We confine our discussion to the case of unipath static routing in which a Flow f traverses one identified path $L_f \subseteq L$.

The non-linear system of TCP/AQM is a time-delay system. There are time lags between when a packet is dropped over Link l, and when the source of Flow f detects the loss, and between when the source of Flow f injects data into the network, and when Link l receives the data. For any time instant t, we define the former as *backward delay*, $\overleftarrow{d_{fl}}(t)$, and the latter as *forward delay*, $\overrightarrow{d_{fl}}(t)$. Denote the round-trip time of Flow F as $d_f(t)$, it can be established [19] that:

$$\overrightarrow{d_{fl}}(t) + \overleftarrow{d_{fl}}(t) = d_f(t) \tag{1}$$

We define the routing matrix associated with the equilibrium forward delays as:

$$[\mathbf{R}_F(s)]_{lf} \quad \triangleq \quad \begin{cases} e^{-s\overrightarrow{d_{fl}}} & \text{Flow } f \text{ traverses Link } l \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

and further denote $\mathbf{R} \triangleq \mathbf{R}_F(0)$.

Each TCP Flow f adjusts its congestion window $W_f(t)$ based on the aggregate packet dropping probability $q_f(t)$ directed by the additive-increase/multiplicative-decrease (AIMD) algorithm. Denote the instantaneous transmission rate of Flow f as $x_f(t)$. By [19], we have:

$$\begin{aligned} \dot{W}_f(t) \quad &= \quad x_f(t-d_f)(1-q_f(t))\frac{1}{W_f(t)} \\ &\quad -x_f(t-d_f)q_f(t)\frac{W_f(t)}{2} \end{aligned} \tag{3}$$

In wireless networks, $q_f(t)$ is determined as:

$$q_f(t) \quad \approx \quad \sum_{l \in L} \mathbf{R}_{lf}(p_l(t-\overleftarrow{d_{fl}}) + p_{wl}(t-\overleftarrow{d_{fl}})) \tag{4}$$

where $p_l(t)$ and $p_{wl}(t)$ are the packet dropping probabilities over Link l due to AQM and wireless losses, respectively. We note that a packet needs to be stored in the buffer managed by AQM, before being transmitted over the link subject to wireless losses. Thus, the delay between packet transmission and packet drop due to wireless losses is slightly greater than the delay between packet transmission and packet drop due to AQM. Nevertheless, we assume the difference to be negligible for simplicity.

For each link l, the instantaneous queue size $b_l(t)$ is determined by the aggregate incoming data rate $y_l(t)$ and the link capacity c_l, following the system dynamics:

$$\dot{b_l}(t) = y_l(t) - c_l \tag{5}$$

$y_l(t)$ is the sum of the time-delayed source rates of all flows traversing Link l:

$$y_l(t) \quad = \quad \sum_{f \in F} \mathbf{R}_{lf} x_f(t-\overrightarrow{d_{fl}}) \approx \sum_{f \in F} \mathbf{R}_{lf} \frac{W_f(t-\overrightarrow{d_{fl}})}{d_f(t)} \tag{6}$$

The approximation above is made based on the assumption that $d_f(t)$, which is determined by the collective transmission of many flows, varies in a larger timescale than $x_f(t)$.

Besides data transmission and packet dropping, the dynamics of flows and links are also coupled via RTTs of the flows. RTT is the sum of the queueing delays for all links traversed and the round-trip propagation delay, τ_f:

$$d_f(t) \quad = \quad \tau_f + \sum_l \mathbf{R}_{lf} \frac{b_l(t)}{c_l} \tag{7}$$

Denote the Laplace transforms of $(\tilde{p_l}(t) : l \in L)$, $(p_{wl}(t) : l \in L)$, and $(\tilde{b_l}(t) : l \in L)$ as $\tilde{\mathbf{p}}(s)$, $\mathbf{p}_w(s)$, and $\tilde{\mathbf{b}}(s)$, respectively. Following a procedure similar to [12], we linearize the non-linear system specified in (3), (4), (5), and (6) around its equilibrium attained with $\mathbf{p}_w(t) \equiv 0$, and apply the Laplace transform to the resulting linear system. Define a *plant transfer function matrix*, $\mathbf{P}(s)$, as the transfer function from $(\tilde{\mathbf{p}}(s) + \mathbf{p}_w(s))$ to $\tilde{\mathbf{b}}(s)$, we obtain:

$$\begin{aligned} \mathbf{P}(s) \quad &= \quad -\left(s\mathbf{I} + \mathbf{R}\,diag(\frac{W_f}{d_f^2})\mathbf{R}^T diag(\frac{1}{c_l})\right)^{-1} \\ &\quad \cdot\mathbf{R}_F(s)\,diag\left(\frac{e^{-sd_f}}{(s+\frac{q_f W_f}{d_f})q_f}\right)\mathbf{R}_F(-s)^T \end{aligned} \tag{8}$$

$\mathbf{P}(s)$ represents the interaction of the TCP dynamics, queueing dynamics, and routing. The stability of this open-loop system can be established by the following lemma [11].

Lemma 1: The system specified in (8) is stable if $\mathbf{R}$ is of full row rank.

It is time to include AQM into the model. In the time domain, AQM determines $\mathbf{p}(t)$ based on $\mathbf{b}(t)$. In the frequency domain, this corresponds to taking $\tilde{\mathbf{b}}(s)$ as input and producing $\tilde{\mathbf{p}}(s)$ as output, that can be represented by a transfer function matrix of dimension $|L| \times |L|$, denoted as $\mathbf{C}(s)$. The control loop is therefore closed, as shown in Fig. 4.

The stability of this closed-loop system can be established by the following lemma.

Lemma 2: The system in Fig. 4 is stable if $\det(\mathbf{I}+\mathbf{P}(s)\mathbf{C}(s))$ contains zeros on the open left half plane (OLHP) only.

Proof: The proof is omitted due to space limitations. ■

Finally, the role of wireless losses is represented by $\mathbf{p}_w$ in Fig. 4. Together with $\tilde{\mathbf{p}}(s)$, $\mathbf{p}_w(s)$ determines the system output $\tilde{\mathbf{b}}(s)$ via $\mathbf{P}(s)$. However, while $\tilde{\mathbf{p}}(s)$ is in turn determined by $\tilde{\mathbf{b}}(s)$ via $\mathbf{C}(s)$ and thus part of the feedback control loop, $\mathbf{p}_w(s)$ is not in the loop but serves as an *external disturbance* to the system.

[4] A non-bottleneck link delays packets for almost a constant amount of time and can thus be modelled as part of the propagation delay experienced by a traffic flow traversing that link.

V. Robust AQM based on Internal Model Principle

In this section, we develop an approach to track the backlog to a reference level despite the presence of wireless losses, based on the frequency domain representation of the TCP/AQM system under wireless losses as shown in Figure 4. Per our discussion in Section III, the objective is to maintain a stable, small backlog under wireless losses.

In the time domain, the prescribed objective is met if $\tilde{\mathbf{b}}(t)$, the backlog deviation from the equilibrium level, converges asymptotically to zero. This means that $\tilde{\mathbf{b}}(s)$ does not contain any unstable poles in the frequency domain.

We begin by noting the following relationship between $\tilde{\mathbf{b}}$ and $\mathbf{p}_w$:

$$\tilde{\mathbf{b}}(s) \quad = \quad (\mathbf{I}+\mathbf{P}(s)\mathbf{C}(s))^{-1}\mathbf{P}(s)\mathbf{p}_w(s) \tag{9}$$

On the other hand, without loss of generality, we can express the disturbance of wireless losses, $\mathbf{p}_w$, as:

$$\mathbf{p}_w(s) = \frac{1}{\phi(s)}\mathbf{k}(s) \tag{10}$$

where $\phi(s) \in \mathbb{F}$ contains all the unstable poles of $\mathbf{p}_w$ and $\mathbf{k}(s) \triangleq \phi(s)\mathbf{p}_w(s)$. $\mathbb{F}$ denotes the set of the rational functions in s. Obviously, all entries in $\mathbf{k}$(s) will not contain any unstable poles. The unstable poles determine the asymptotic behaviour of $\mathbf{p}_w(s)$. Thus, $\frac{1}{\phi(s)}$ is often referred to as a *model* of the disturbance. It follows that:

$$\tilde{\mathbf{b}}(s) \quad = \quad \frac{1}{\phi(s)}(\mathbf{I}+\mathbf{P}(s)\mathbf{C}(s))^{-1}\mathbf{P}(s)\mathbf{k}(s) \tag{11}$$

When the system in Fig. 4 is stable, the unstable poles of $\tilde{\mathbf{b}}(s)$, if any, come from the unstable poles contained in $\phi(s)$. To attain disturbance rejection of $\mathbf{p}_w$, or to prevent the unstable poles contained in $\phi(s)$ from becoming poles of $\tilde{\mathbf{b}}(s)$, the *internal model principle* proposes to insert the model of disturbance $\frac{1}{\phi(s)}$ inside the feedback loop [5].

The design of AQM determines $\mathbf{C}(s)$, whereas the *plant transfer function* $\mathbf{P}(s)$ is invariant across different AQM algorithms. Therefore, we put $\frac{1}{\phi(s)}$ inside the feedback loop by selecting $\mathbf{C}(s)$ as:

$$\mathbf{C}(s) = \frac{1}{\phi(s)}\Lambda_N(s)\Lambda_D(s)^{-1} \tag{12}$$

where $\Lambda_N(s), \Lambda_D(s) \in \mathbb{F}^{|L|\times|L|}$ are coprime diagonal matrices. $\det(\Lambda_N(s))$ does not share roots with $\phi(s)$, and $\det(\Lambda_D(s))$ does not have any roots on the closed right half plane (CRHP). We constrain $\mathbf{C}(s)$ to be diagonal so that each intermediate router can execute the AQM algorithm distributedly without incurring the need of exchanging the instantaneous queue sizes with each other.

Theorem 1: Consider the system in Fig. 4 with $\mathbf{P}(s)$ and $\mathbf{C}(s)$ as specified in (8) and (12), respectively. Suppose the conditions of Lemmas 1 and 2 hold so that both open-loop and closed-loop stabilities are established. Let $\mathbf{P}(s) = \mathbf{D}^{-1}(s)\mathbf{N}(s)$ be a left coprime factorization. The system attains disturbance rejection of (10) if $\det(\mathbf{N}(s))$ does not contain zeros at the roots of $\phi(s)$.

Proof: Substituting (12) into (11) gives:

$$\tilde{\mathbf{b}}(s) = \Lambda_D(s)(\phi(s)\mathbf{D}(s)\Lambda_D(s)+\mathbf{N}(s)\Lambda_N(s))^{-1}\mathbf{N}(s)\mathbf{k}(s) \tag{13}$$

Disturbance rejection is attained if none of the zeros of $\det(\phi(s)\mathbf{D}(s)\Lambda_D(s) + \mathbf{N}(s)\Lambda_N(s))$, which include the poles of $\tilde{\mathbf{b}}(s)$ as a subset, are on CRHP. For the roots of $\phi(s)$ (which are unstable poles of $\mathbf{p}_w(s)$ and thus on CRHP), the determinant reduces to:

$$\det(\mathbf{N}(s)\Lambda_N(s)) = \det(\mathbf{N}(s))\det(\Lambda_N(s)) \tag{14}$$

This is non-zero since neither $\det(\Lambda_N(s))$ nor $\det(\Lambda_N(s))$ contains zeros at the roots of $\phi(s)$ by assumption.

For $s \in \{s|\Re(s) \geq 0 \text{ and } \phi(s) \neq 0\}$:

$$\begin{aligned}
&\det(\phi(s)\mathbf{D}(s)\Lambda_D(s)+\mathbf{N}(s)\Lambda_N(s)) \\
= \;&\det(\mathbf{D}(s))\det(\Lambda_D(s)) \\
&\cdot\det(\phi(s)\mathbf{I}+\mathbf{P}(s)\Lambda_N(s)\Lambda_D(s)^{-1}) \\
= \;&\phi(s)^{|L|}\det(\mathbf{D}(s))\det(\Lambda_D(s)) \\
&\cdot\det(\mathbf{I}+\mathbf{P}(s)\mathbf{C}(s))
\end{aligned} \tag{15}$$

We now consider the three determinants in the final expression above one by one. As $\mathbf{R}$ is of full row rank, $\mathbf{P}(s)$ does not have any unstable poles and thus $\det(\mathbf{D}(s))$ does not contain any zeros on CRHP. Neither $\det(\Lambda_D(s))$ nor $\det(\mathbf{I}+\mathbf{P}(s)\mathbf{C}(s))$ contain zeros on CRHP by assumption.

Note that CRHP is a subset of

$$\{s|\phi(s) = 0\}\bigcup\{s|\Re(s) \geq 0 \text{ and } \phi(s) \neq 0\} \tag{16}$$

Thus, we conclude that $\det(\phi(s)\mathbf{I}+\mathbf{P}(s)\Lambda_N(s)\Lambda_D(s)^{-1})$ does not contain zeros on CRHP. ■

Remark 1: When $\det(\mathbf{N}(s))$ *does* contain zeros at some roots of $\phi(s)$, there will be some cancellations of unstable zero-pole pairs between $\mathbf{C}(s)$ and $\mathbf{P}(s)$. This violates the condition of *total stability*, which means that every possible input-output pair of the system is bounded-input-bounded-output (BIBO) stable. It is one of the physical constraints that a practical system should meet. Interested readers can refer to [5] for details.

Remark 2: The attained disturbance rejection is robust against perturbation to $\mathbf{P}(s)$, $\Lambda_N(s)$, and $\Lambda_D(s)$, as long as the mild assumptions of Theorem 1 continue to hold after the perturbation. In practice, the TCP AIMD parameters, routing topologies, and network capacities may change over time, leading to such perturbation. The AQM design specified in (12) can nonetheless maintain the robustness against wireless losses.

Remark 3: The attained disturbance rejection is not robust against errors in implementing $\phi(s)$. Consider when $\phi(s)$ is inaccurately implemented as $\tilde{\phi}(s)$, $\mathbf{C}(s)$ thus becomes:

$$\mathbf{C}(s) = \frac{1}{\tilde{\phi}(s)}\Lambda_N(s)\Lambda_D(s)^{-1} \tag{17}$$

It follows that we have $\tilde{\mathbf{b}}(s)$ as (18). In (18), $\tilde{\mathbf{b}}_1(s)$ contains the stable poles only while $\tilde{\mathbf{b}}_2(s)$ contains the unstable poles of $\phi(s)$, and represents the steady state errors in tracking the backlog to a reference queue size due to the inexact cancellation of zeros between $\tilde{\phi}(s)$ and $\phi(s)$. Nevertheless, it is still very meaningful to apply the internal model principle with inexact implementation of $\phi(s)$. This reduces the steady state errors compared to when the principle is not applied. For example, consider the simple case that $\phi(s) = s - a$ and $\tilde{\phi}(s) = s - a + \epsilon$, where a and ϵ represent

$$\tilde{\mathbf{b}}(s) = \overbrace{\Lambda_D(s)(\tilde{\phi}(s)\mathbf{D}(s)\Lambda_D(s) + \mathbf{N}(s)\Lambda_N(s))^{-1}\mathbf{N}(s)\mathbf{k}(s)}^{\tilde{\mathbf{b}}_1(s) \triangleq} + \overbrace{\Lambda_D(s)(\tilde{\phi}(s)\mathbf{D}(s)\Lambda_D(s) + \mathbf{N}(s)\Lambda_N(s))^{-1}\mathbf{N}(s)\mathbf{k}(s)\frac{\tilde{\phi}(s) - \phi(s)}{\phi(s)}}^{\tilde{\mathbf{b}}_2(s) \triangleq} \tag{18}$$

an unstable pole and the error involved in implementing $\phi(s)$, respectively. $\tilde{\mathbf{b}}_2(s)$ then reduces to:

$$\begin{aligned} \tilde{\mathbf{b}}_2(s) &= \Lambda_D(s)(\tilde{\phi}(s)\mathbf{D}(s)\Lambda_D(s) + \mathbf{N}(s)\Lambda_N(s))^{-1} \\ &\cdot \mathbf{N}(s)\mathbf{k}(s)\frac{\epsilon}{s-a} \end{aligned} \tag{19}$$

which is proportional to ϵ. Therefore, we can keep the steady state error minimal with a good approximate implementation of $\phi(s)$.

Generally, Theorem 1 can be applied in two directions. On the one hand, when $\phi(s)$ is known,[5] we construct an appropriate $\mathbf{C}(s)$ to yield a desired disturbance rejection. For example, wireless losses may vary periodically at a frequency, say, $\frac{1}{T}$, resulting in $\phi(s) = 1 - e^{-Ts}$. An AQM with $\mathbf{C}(s)$ containing $(1 - e^{-Ts})$ as the denominator can potentially help maintain a stable, small backlog under wireless losses with a periodic time-varying packet loss rate. This deserves further study in the future.

On the other hand, without any specific knowledge of $\phi(s)$, $\phi(s)$ can be designed to serve as a good approximation for the general scenarios. We are going to investigate the design of such a controller in the next subsection.

A. Design of IC

Here, we approximate the disturbance $\mathbf{p}_w$ as a step input function. This approximation is valid as long as the linear system in consideration converges much faster than the variations in the WPER. It follows that $\phi(s) = s$. From (12), we can choose the diagonal element of $\mathbf{C}(s)$ as:

$$\mathbf{C}_{ll}(s) = \frac{g_l}{s}, \quad 1 \le l \le |L| \tag{20}$$

which is essentially an IC. It converts its input into the integral of that input in the time domain. The proposed IC will be part of any other $\mathbf{C}(s)$ satisfying (20) and $\phi(s) = s$, and is the essential element fulfilling the *internal model principle*. Denote:

$$\hat{\mathbf{R}}_F(\cdot) \triangleq diag\{c_l\}^{-1}\mathbf{R}_F(\cdot)diag\{\frac{W_f}{d_f}\} \tag{21}$$

$$\omega_g \triangleq 0.1 \min_{l \in L, f \in F} \left(\frac{2}{d_f W_f}, \frac{1}{d_f}, \frac{\sigma^2_{\min}(\hat{\mathbf{R}}_F(0))\, c_l}{W_f} \right) \tag{22}$$

The following proposition gives the design rules for the proposed IC.

Proposition 1: Consider the system in Fig. 4 with $\mathbf{P}(s)$ and $\mathbf{C}(s)$ as specified in (8) and (20), respectively. Suppose that $\mathbf{R}$ is of full row rank. Let g_l be chosen as $\frac{g}{c_l}$. The system is closed-loop stable and attains disturbance rejection of $\mathbf{p}_w$ as specified in (10) with $\phi(s) = s$ if:

$$g \le \omega_g \cdot \frac{4\,\sigma^2_{\min}(\hat{\mathbf{R}}_F(0))}{\|\hat{\mathbf{R}}_F(0)\|_1} \cdot \frac{1}{(\max_{f\in F} d_f\, W_f)\max_{f \in F} W_f} \tag{23}$$

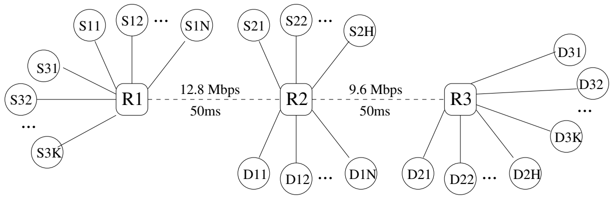


Fig. 5. Two wireless bottleneck links.

In the case of homogeneous flows sharing a bottleneck link with $\max_f W_f \ge 2$, a simplified condition can be obtained as:

$$g \le \frac{0.8(N_-)^3}{d_+^5 c^3} \tag{24}$$

where N_-, d_+, and c correspond to the lower bound for the number of flows, the upper bound for the round-trip time (RTT), and the link capacity, respectively.

Proof: The proof follows a similar line of reasoning as [24]. ■

Remark 4: In [11], the PI controller is designed in the form of $\frac{k_l(s+z_l)}{s}$, which cannot be reduced to the proposed IC as specified in (20).

After the discretization via the bilinear transform for an IC, the packet dropping rate for each link l can computed as:

$$p_l[k] = p_l[k-1] + G_l(b_l[k] - b_l) \tag{25}$$

where $G_l = \frac{g_l}{f_l}$. f_l is the sampling frequency, i.e., the frequency at which the queue size is sampled and (25) is used to update the packet dropping rate. $p_l[k]$ and $b_l[k]$ are the step-wise packet dropping rate and queue size, respectively. The computation can be realized as an independent module in AQM, taking the queue length of a buffer as input and computing the packet dropping rate due to the IC as the output.

Example 1: Consider the scenario illustrated in Fig. 1, with the wireless link of capacity 15 Mbps and propagation delay 50 ms. A data packet is 1000 bytes long, and thus c_l = 1875 packets/second. A total of $N = 100$ homogeneous TCP flows share the link. Suppose that the equilibrium queue size is $b_l = 220$ packets[6]. It follows that d_f is around 0.2 seconds.

Applying (24) using $d_+ = 0.2$ seconds, $N_- = N = 100$, and $c = c_l = 1875$ packets,

$$g \le \frac{0.8 \cdot 100^3}{0.2^5 \cdot 1875^3} \approx 0.3793$$

Example 2: Consider another network scenario illustrated in Fig. 5, which consists of three wired local area networks (LANs) inter-connected to create a wide area network (WAN) via wireless links. Three sets of flows F_1, F_2, and F_3 share the network. There

[5] We note that $\phi(s)$ corresponds to the "pattern" in the variation of wireless losses. Knowledge about $\phi(s)$ is a much weaker assumption than knowledge about the exact WPER.

[6] For ease of comparison, b_l is determined in accordance with the equilibrium level achieved by RED without wireless losses. Nevertheless, for other reasonable values of b_l (generally, greater than N and less than the buffer size), all the computations in this section follow similarly and the discussions on IC in Section VI continue to hold.

are $N_1 = 60$, $N_2 = 50$, and $N_3 = 20$ flows in F_1, F_2, and F_3, respectively. Flows within the same set are homogeneous. Each of the flows in F_i $(i = 1, 2, 3)$ traverses from Sij to Dij, where $1 \le j \le 60$ for $i = 1$, $1 \le j \le 50$ for $i = 2$, and $1 \le j \le 20$ for $i = 3$. We thus have two wireless bottleneck links, whose dynamics are coupled via flows in F_3. Suppose that the equilibrium queue sizes of both wireless links are $b_1 = b_2 = 200$ packets. Let d_i and W_i be the equilibrium RTT and window size for each of the flows in F_i, respectively. It follows that:

$$\begin{aligned}
\mathbf{c} &= (1600, 1200) \text{ packets/second} \\
\mathbf{d} &\approx (0.18, 0.22, 0.35)^T \text{ seconds} \\
\mathbf{W} &\approx (4.2, 4.5, 3.2)^T \text{ packets}
\end{aligned}$$

and thus $\|\hat{\mathbf{R}}_F(0)\|_1 \approx 0.017$.

Now, the matrix $\hat{\mathbf{R}}_F(0)\hat{\mathbf{R}}_F(0)^T$ becomes:

$$\hat{\mathbf{R}}_F(0)\hat{\mathbf{R}}_F(0)^T \approx \begin{pmatrix} 0.014 & 0.00065 \\ 0.00065 & 0.015 \end{pmatrix} \qquad (26)$$

which gives $\sigma^2_{\min}(\hat{\mathbf{R}}_F(0)) \approx 0.0136$.

By (22) and (23),

$$\begin{aligned}
\omega_g &\approx 0.1 \min\left(\frac{2}{0.35 \cdot 3.2}, \frac{1}{0.35}, \frac{0.0136 \cdot 1200}{4.5}\right) \approx 0.1786 \\
g &\le 0.1786 \cdot \frac{4 \cdot 0.0136}{0.017} \cdot \frac{1}{(0.35 \cdot 3.2) \cdot 4.5} \approx 0.1134
\end{aligned}$$

VI. Performance Evaluation

In this section, we evaluate the performance of the proposed IC and compare it with RED using Network Simulator (ns) Version 2.29. This is necessary because the robustness and stability of IC have been established in the context of the linearized congestion control system, which is nonlinear in reality. Sections VI-A and VI-B present the simulation results using the topologies illustrated in Figs. 1 and 5, respectively.

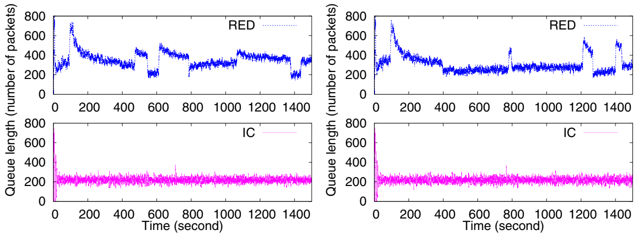
For IC, the packet dropping rate is updated at a frequency of 10 Hz. Nevertheless, we have repeated the experiments with a different implementation of IC, which updates the packet dropping rate at every packet arrival as RED does, with similar results reported.

The parameters of IC and RED follow Proposition 1 and [11], [12], respectively. For RED, $pMax$ and the queue weight are set to 0.1 and 2.6×10^{-6}, respectively. The parameters $minThresh$ and $maxThresh$ of RED, and the gain of IC, G_l, vary and will be specified in the discussion whenever necessary.

A. One Wireless Bottleneck Link

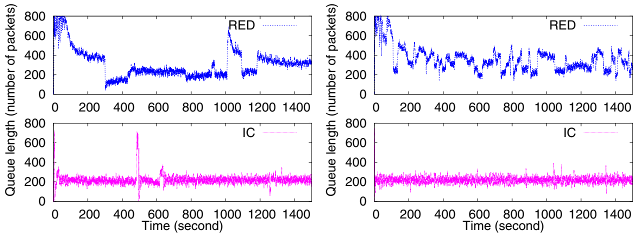
In the topology illustrated in Fig. 1, N source-destination pairs are connected via a wireless bottleneck link with capacity c Mbps, and propagation delay τ ms. Within each pair (S_i, D_i), M long-lived TCP Reno flows are sent from S_i to D_i. The size of the link buffer and the reference queue size of IC are set to 800 packets and 220 packets, respectively. Time is divided into periods, each of which follows an exponential distribution with a mean of *lossP* seconds. At the beginning of each period, the WPER of the bottleneck link is reset following a uniform distribution over $[0, L_m]$.

Figs. 6-9 plot the traces of the queue sizes under various configurations summarized in Table I. The "HTTP" and "CBR" columns in Table I indicate whether there are hypertext transfer protocol (HTTP) and constant bit rate (CBR) traffic flows coexisting with the long-lived TCP flows, respectively. The final three columns are the parameter settings of IC and RED, which are computed from the network settings.



(a) $N = 100$, $M = 1$.   (b) $N = 20$, $M = 5$.

Fig. 6. Comparison of queue length dynamics under different configurations of M and N.



(a) Larger variations in WPER.   (b) More frequent variations in WPER.

Fig. 7. Comparison of queue length dynamics under different wireless losses.

The network configurations for Fig. 6(a) and Fig. 6(b) are the same as those for Fig. 2(b) and Fig. 2(c), respectively. We observe that IC helps maintain a stable queue size throughout the simulation period. Despite the variations in WPER, the equilibrium queue size of IC tracks the reference level (220 packets) very well. We do observe occasional "spikes" in the queueing dynamics with IC. This is because IC needs to converge to a new equilibrium packet dropping rate to maintain the same equilibrium queue size when WPER changes. Nevertheless, the transition is much smoother than that of RED since the equilibrium queue size is unchanged and IC has better transient response than RED. Moreover, while the difference between the two equilibrium packet dropping rates should be approximately the change in WPER, IC is self-adaptive and does not require any knowledge about WPER for completing the transition. We note that this is an essential property of the wireless AQM enhancements based on the *internal model principle*.

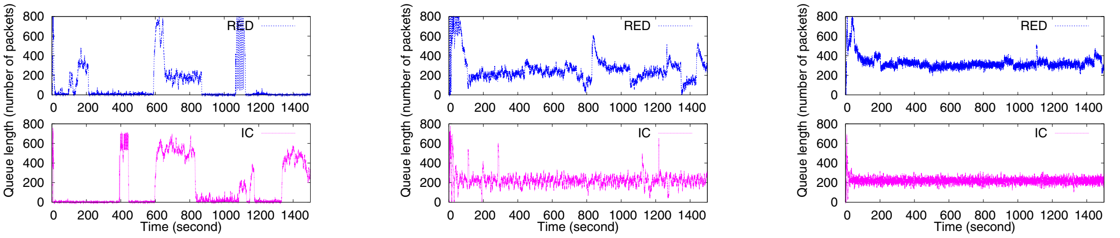
In addition, Fig. 6(a) and Fig. 6(b) exhibit similar queueing dynamics. Generally, in all of our conducted simulation experiments, the queueing dynamics are quite similar across different values of M for $1 \le M \le 10$, as long as the total number of flows, NM, is kept constant.

Fig. 7(a) and Fig. 7(b) exhibit the queueing dynamics against wireless losses with larger variations and more frequent variations in WPER, respectively. Correspondingly, RED exhibits more severe and more frequent fluctuations, respectively. IC basically maintains a stable queue size around the reference level in both cases, except that occasional spikes due to transitions become larger in Fig. 7(a). This is expected, since more abrupt "jumps" in WPER require IC to undergo greater changes in the equilibrium packet dropping rate.

Fig. 8(a)-(c) exhibit the queueing dynamics with "fatter" pipes, i.e., a bottleneck link with larger propagation delay or higher

TABLE I
NETWORK CONFIGURATIONS FOR NETWORKS WITH ONE WIRELESS BOTTLENECK LINK.

Figure	N	M	c	τ	L_m	*lossP*	HTTP	CBR	G_l	*minThresh*	*maxThresh*
Fig. 6(a)	100	1	15	50	4%	100	No	No	$2 \cdot 10^{-5}$	150	700
Fig. 6(b)	20	5	15	50	4%	100	No	No	$2 \cdot 10^{-5}$	150	700
Fig. 7(a)	20	5	15	50	8%	100	No	No	$2 \cdot 10^{-5}$	150	700
Fig. 7(b)	20	5	15	50	4%	20	No	No	$2 \cdot 10^{-5}$	150	700
Fig. 8(a)	20	5	15	200	4%	100	No	No	$2 \cdot 10^{-7}$	150	400
Fig. 8(b)	20	10	15	200	4%	100	No	No	$1.6 \cdot 10^{-6}$	150	400
Fig. 8(c)	20	10	30	50	4%	100	No	No	$1 \cdot 10^{-5}$	150	400
Fig. 9(a)	20	5	15	50	4%	100	Yes	No	$2 \cdot 10^{-5}$	150	700
Fig. 9(b)	20	5	15	50	4%	100	No	Yes	$2 \cdot 10^{-5}$	150	700
Fig. 9(c)	20	5	15	50	4%	100	Yes	Yes	$2 \cdot 10^{-5}$	150	700

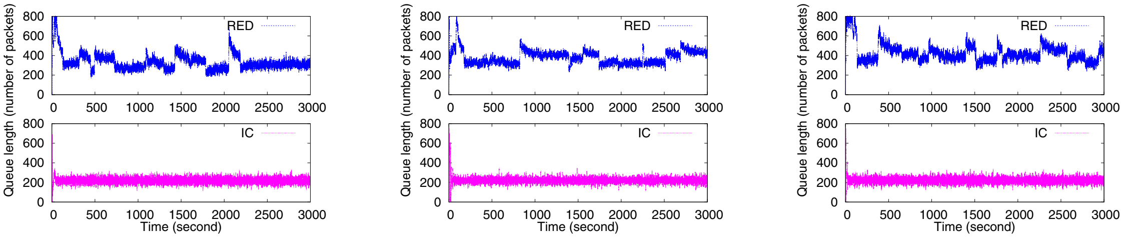


(a) Fatter pipe due to increased link delay.

(b) Fatter pipe due to increased link delay is shared by more TCP flows.

(c) Fatter pipe due to increased bandwidth is shared by more TCP flows.

Fig. 8. Comparison of queue length dynamics with different bottleneck link parameters.



(a) HTTP.

(b) CBR.

(c) HTTP and CBR.

Fig. 9. Comparison of queue length dynamics with the presence of other traffic.

bandwidth. Fig. 8(a) increases the propagation delay to 200 ms, approximately the two-way propagation delay with a geostationary satellite [2], and maintains the total number of flows to be 100. We observe that the queue length dynamics tend to fluctuate very severely for both IC and RED. In fact, the link is *not* backlogged for a significant portion of time.

The congestion window of a TCP flow regulates the number of packets in transit. It grows larger with a fatter pipe, since more packets in transit are needed to keep the pipe full. When a wireless packet loss occurs, the corresponding flow will spuriously halve its window. With a fatter pipe, the window reduction becomes larger. There are more severe fluctuations on the number of packets in transit, thereby possibly falling below the size of the pipe ($\approx 2c\tau$) and leaving the link not backlogged. We recall that, when a link is not backlogged, it is outside the operational region of the wireless AQM enhancements.

To mitigate the effect of fatter pipes, we can increase the number of TCP flows so that the congestion window of a single flow becomes smaller. Fig. 8(b) plots the queueing dynamics when the total number of flows is increased to 200. In this case, we observe that the link is backlogged throughout the simulation, and IC can effectively maintain a stable queue size.

Similarly, Fig. 8(c) plots the queueing dynamics when 200 flows are sharing a bottleneck link of doubled bandwidth. In this case, IC still demonstrates a robust tracking of the reference queue size. The backlog fluctuation with RED is not very significant due to the reduction in $maxThresh$, where $maxThresh$ is the maximum queue size allowed by RED. However, such reduction is generally not recommended in other scenarios due to the stability requirement of RED [12].

Fig. 9(a)-(c) exhibit the queueing dynamics with the presence of other traffic. Fig. 9(a) plots the queueing dynamics when we introduce the HTTP traffic into the dumbbell topology. In doing so, we attach 120 web client nodes to R1 and ten web server nodes to R2. We can see that the traces of the queue size are more "noisy", since the introduction of the HTTP traffic leads to a greater fluctuation on the network load. Nevertheless, IC demonstrates an effective and robust tracking of the reference queue level against the disruptions from both perturbing traffic and wireless losses.

Fig. 9(b) plots the queueing dynamics when we introduce CBR traffic into the dumbbell topology. In doing so, we attach 30 user datagram protocol (UDP) sources to R1 and their corresponding destinations to R2. Each UDP flow is driven by a long-lived CBR source at the constant rate of 0.1 Mbps. The UDP flows represent unresponsive flows that do not back off upon network congestion. With RED, the backlog generally shifts up due to the increased long-lived traffic load. On the other hand, IC still maintains a robust tracking of the reference queue level, despite the presence of unresponsive flows.

Fig. 9(c) plots the queueing dynamics when both HTTP and CBR traffics are present. We can identify the aforementioned impact of introducing HTTP and CBR traffics. With RED, the backlog trace becomes more "noisy" and shifts up as well. With IC, the backlog becomes more "noisy" but still tracks the reference level.

B. Two Wireless Bottleneck Links

In the topology illustrated in Fig. 5, the network configuration follows those in Example 2. The size of the two link buffers are both 600 packets. For both links, $minThresh$ and $maxThresh$ of RED are set to 150 and 500 packets, respectively. G_l is set to $7 \cdot 10^{-6}$ and $9.5 \cdot 10^{-6}$ for Links 1 and 2, respectively. We introduce time-varying losses into both links, with an average loss period of 100 s and a maximum WPER of 3 %. The simulation results are plotted in Fig. 10. Again, severe fluctuations can be observed in the queue sizes with RED over Link 1. On the other hand, IC is successful in maintaining stable queue sizes throughout the simulation. Similar results are reported for Link 2.

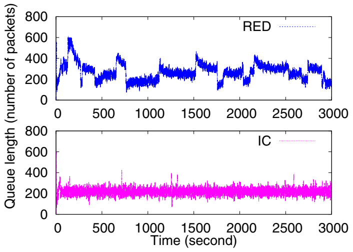


Fig. 10. Queue length dynamics of Link 1 in the topology with two wireless bottleneck links.

VII. Conclusions

This paper is centered around BQ: Can AQM maintain a small, stable queue under wireless losses? We carry out a series of study. We find that RED, one of the most representative AQMs, fails to maintain a stable queue size against time-varying wireless losses. The key to solving the problem is for AQM to force the queue size to track a reference queue level, despite the presence of wireless losses. Based on the frequency domain representation of TCP/AQM, we apply the internal model principle to reject the disturbances of wireless losses, thereby attaining such robust tracking. We devise the integral controller (IC) as an embodiment of the internal model principle, and develop a set of design rules for IC. We have conducted a series of experiments to examine the performance of IC and compare it with RED. In particular, we have evaluated the impact of severe variations of wireless losses, fat pipes, and the HTTP traffic on the two AQM algorithms. In general, IC demonstrates very robust performance in maintaining a small, stable backlog under wireless losses.

There are several possible extensions to our work, including:

1. developing a global stability analysis for the system of TCP/buffer-based AQM perturbed by wireless losses;

2. applying Theorem 1 to design AQM that attains clean disturbance rejection of wireless losses with known $\phi(s)$, as discussed in Section V, and;

3. developing a unified AQM solution for resolving both fairness and stability issues in ad-hoc wireless networks.

Acknowledgements

This research is supported in part by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Grant No. HKU 714510E.

References

[1] I.F. Akyidiz, C. Morabito, and S. Palazzo. TCP-Peach: A New Congestion Control Scheme for Satellite IP Networks. *IEEE/ACM Trans. Networking*, Vol. 9, No. 3, Jun. 2001.

[2] M. Allman, D. Glover, and L. Sanchez. Enhancing TCP over Satellite Channels using Standard Mechanisms. *IETF RFC 2488*, Jan. 1999.

[3] S. Athuraliya, S.H. Low, V.H. Li, and Q. Yin. REM: Active Queue Management. *IEEE Network*, Vol. 15, No. 3, pp. 48-53, May 2001.

[4] B. Braden, D. Clark, J. Crowsoft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Patridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang. Recommendations on Queue Management and Congestion Avoidance in the Internet. *IETF RFC 2309*, Apr. 1998.

[5] C.-T. Chen. Linear System Theory and Design. *New York: Oxford University Press*, 1999.

[6] M. Chen and A. Zakhor. Flow Control over Wireless Network and Application Layer Implementation. *Proc. of IEEE INFOCOM*, pp. 103-113, Mar. 2006.

[7] J. Davis. New Networking Features in Windows Server 2008 and Windows Vista. http://technet.microsoft.com/en-us/library/bb726965.aspx. Feb. 2008.

[8] X. Fan, M. Arcak, and J.T. Wen. Robustness of Network Flow Control against Disturbances and Time-Delay. *Syst. Contr. Lett.*, Vol. 53, No. 11, pp. 1329, Sep. 2004.

[9] S. Floyd, T. Henderson, and A. Gurtov. The NewReno Modification to TCP's Fast Recovery Algorithm. *IETF RFC 3782*, Apr. 2004.

[10] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE Trans. Networking*, Vol. 1, No. 4, Aug. 1993.

[11] H. Han, C.V. Hollot, Y. Chait, and V. Misra. TCP Networks Stabilized by Buffer-Based AQMs. *Proc. of IEEE INOFOCOM*, Mar. 2004.

[12] C.V. Hollot, V. Misra, D. Towsley, and W.-B. Gong. Analysis and Design of Controllers for AQM Routers Supporting TCP Flows. *IEEE Trans. Automatic Control*, Vol. 47, No. 6, Jun. 2002.

[13] Institute of Electrical and Electronics Engineers. IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems. *New York: Institute of Electrical and Electronics Engineers*, 2009.

[14] F. Kelly, A. Maulloo, and D. Tan. Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability. *J. of the Operational Research Society*, Vol. 49, No. 3, pp. 237-252, Mar. 1998.

[15] H.K. Khalil. Nonlinear Systems. *Upper Saddle River, N.J.: Prentice-Hall*, 2002.

[16] C. Lai, K.-C. Leung, and V.O.K. Li. Enhancing Wireless TCP: A Serialized-Timer Approach. *Proc. of IEEE INFOCOM*, Mar. 2010.

[17] K.-C. Leung and V.O.K. Li. Transmission Control Protocol (TCP) in Wireless Networks: Issues, Approaches, and Challenges. *IEEE Comm. Surveys and Tutorials*, Vol. 8, No. 4, pp. 64-79, Fourth Quarter 2006.

[18] S.H. Low. A Duality Model of TCP and Queue Management Algorithms. *IEEE/ACM Trans. Networking*, Vol. 11, No. 4, pp. 525-536, Aug. 2003.

[19] S.H. Low, F. Paganini, and J.C. Doyle. Internet Congestion Control. *IEEE Control Systems Magazine*, Vol. 22, No. 1, pp. 28-43, Feb. 2002.

[20] S.H. Low, F. Paganini, J. Wang, and J.C. Doyle. Linear Stability of TCP/RED and a Scalable Control. *Computer Networks Journal*, Vol. 43, No. 5, pp. 633-647, Dec. 2003.

[21] V. Misra, W.-B. Gong, and D. Towsley. Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED. *Comp. Comm. Rev.*, Vol. 30, No. 4, pp. 151-160, Oct. 2000.

[22] K.K. Ramakrishnan, S. Floyd, and D. Black. The Addition of Explicit Congestion. Notification (ECN) to IP. *IETF RFC 3168*, Sep. 2001.

[23] R. Srikant. The Mathematics of Internet Congestion Control. *Cambridge, MA:Birkhauser*, 2004.

[24] G. Vinnicombe. On the Stability of Networks Operating TCP-Like Congestion Control. *Proc. of the IFAC World Congress on Automatic Control*, Jun. 2002.

[25] J. Wang, J. Wen, J. Zhang, and Y. Han. TCP-FIT: An Improved TCP Congestion Control Algorithm and its Performance. *Proc. of IEEE INFOCOM*, Apr. 2011.

[26] K. Xu, M. Gerla, L. Qi, and Y. Shu. Enhancing TCP Fairness in Ad Hoc Wireless Networks Using Neighborhood RED. *Proc. of ACM MOBICOM*, pp. 16-28, Sep. 2003.

[27] L. Ying, G.E. Dullerud, and R. Srikant. Global Stability of Internet Congestion Controllers with Heterogeneous Delays. *IEEE/ACM Trans. Networking*, Vol. 14, No. 3, pp. 579-591, Jun. 2006.