Taxonomy and Evaluation of TCP-Friendly Congestion-Control Schemes on Fairness, Aggressiveness, and Responsiveness

Shih-Chiang Tsao, National Chiao Tung University
Yuan-Cheng Lai, National Taiwan University of Science and Technology
Ying-Dar Lin, National Chiao Tung University

Abstract

Many TCP-friendly congestion control schemes have been proposed to pursue the TCP-equivalence criterion, which states that a TCP-equivalent flow should have the same throughput with TCP *if it experiences identical network conditions as TCP*. Additionally, the throughput should converge as fast as TCP when the packet-loss conditions change. This study classifies eight typical TCP-friendly schemes according to their underlying policies on fairness, aggressiveness, and responsiveness. The schemes are evaluated to verify whether they meet TCP-equivalence and TCP-equal share. TCP-equal share is a more realistic but more challenging criterion than TCP-equivalence and states that a flow should have the same throughput with TCP *if competing with TCP for the same bottleneck*. Simulation results indicate that one of the selected schemes, TCP-friendly rate control (TFRC), meets both criteria under more testing scenarios than the others. Additionally, the results under non-periodic losses, low-multiplexing, two-state losses, and bursty losses reveal the causes that bring fault cases to the schemes. Finally, appropriate policies are recommended for an ideal scheme.

Real-time streaming media, such as video/audio conversations and movies online, often are transmitted over the Internet. Because the available bandwidth in the Internet is dynamic, a congestion control mechanism is required to prevent the media flow from suffering serious packet losses. A flow carried over Transmission Control Protocol (TCP) generally is subject to such a congestion control mechanism. TCP is the most widely-used transport protocol in the Internet and embeds an additive-increase and multiplicative-decrease (AIMD) congestion control mechanism.

The throughput controlled by AIMD in TCP changes dramatically and frequently, which may not satisfy real-time streaming media. Many AIMD-variant and other-style congestion control schemes have been proposed to solve this problem [1–6]. In addition to being smooth, these schemes are said to be TCP-friendly [7], because their controlled traffic is expected to coexist with TCP traffic in the Internet. *TCP-friendly* is a generic term describing a scheme that aims to use no more bandwidth than TCP uses. This study discusses in detail the proper behaviors of a TCP-friendly scheme in view of the following three criteria: TCP-compatibility, TCP-equivalence, and TCP-equal share.

TCP-compatibility is defined in RFC 2309 [7], which says that a TCP-compatible flow, in the steady state, should use no more bandwidth than a TCP flow under comparable conditions, such as packet-loss rate and round-trip time (RTT), where RTT means the time required for a packet to travel from the source to the destination and back. However, a TCP-compatible congestion control scheme is not preferred if it always offers far lower throughput than a TCP flow. Hence, a better congestion control scheme must not only meet TCP-compatibility but also pursue *TCP-equivalence*. A TCP-equivalent flow has the same throughput as a TCP flow if it experiences identical network conditions, which means the same patterns of packet-loss occurrences and RTT changes. Most current schemes tend to provide TCP-equivalence rather than just TCP-compatibility. However, TCP-equivalence in *all* network conditions is hard to achieve. Various studies have described schemes that achieve compatibility without always achieving equivalence [1–3, 8–10].

Although a TCP-equivalent scheme consumes TCP-equivalent bandwidth when working by itself, it may not coexist well with TCP in the Internet. A TCP-equivalent scheme merely ensures the same throughput between TCP and TCP-equivalent flows when both experience identical conditions, but not when both compete for the same bottleneck, which is the actual situation on the Internet. Competing for the same bottleneck does not imply experiencing identical network conditions [10]. Therefore, this study defines a new criterion, namely *TCP-equal share*. This criterion is more realistic than TCP-equivalence, because the most important concern is

0890-8044/07/$20.00 © 2007 IEEE

Criterion	Network premise	Proper behaviors of a scheme		
		Steady state	Transient state	
		Fairness	Aggressiveness	Responsiveness
TCP compatibility	Comparable conditions	Less bandwidth	Don't care	As fast as TCP
TCP equivalence	Identical conditions	Equal bandwidth	As fast as TCP	
TCP equal share	Same bottleneck			

Table 1. *The premises and proper behaviors in three criteria.*

whether flows with different controls can co-exist and equally share bandwidth in the same bottleneck; whereas coexistence is not in the picture of TCP-equivalence. Moreover, TCP-equal share is also more challenging than TCP-equivalence because a TCP-equivalent flow may not be TCP-equal share, but vice versa is true.

This study has three objectives. The first objective is to be a guide for selecting from existing TCP-friendly schemes, based on the proposed taxonomy and evaluation. The second objective is to indicate the potential fault cases and causes of the eight schemes evaluated, thus helping designers to realize what must be enhanced. The third objective is to recommend policies for designing an ideal scheme to meet all TCP-friendly criteria. Unlike the survey of Widmer *et al.* [11] that compares the functionality of various schemes, this study tests the selected schemes for the TCP-friendly criteria. Unlike Bansal *et al.* [8], who compare the transient behaviors of various schemes, this study additionally investigates these schemes under the steady state to reveal that they may use bandwidth unequal to TCP even in this case. In addition, this study investigates the bandwidth sharing between TCP and TCP-friendly flows (inter-fairness), differing from Tsaoussidis *et al.* [12], who study the bandwidth sharing among a group of homogeneous flows (intra-fairness).

For TCP-friendly schemes, Table 1 summarizes the proper behaviors for the three TCP-friendly criteria in three aspects, namely fairness, aggressiveness, and responsiveness, as explained further. Table 2 shows the eight typical TCP-friendly schemes selected for this study. The behaviors of these schemes are classified according to their key operational characteristics to realize how they meet the three criteria. The evaluation results verify whether these schemes meet the criteria and also reveal additional issues. Next, related work is discussed. Finally, we make recommendations about the preferred schemes and policies, based on the observed results.

Notably, although TCP-friendly rate control protocol (TFRCP) is simply the predecessor of TFRC, it is selected in this study due to its simplicity, which may be preferred by programmers of real-time applications. Moreover, Bansal *et al.* [8] defined a TCP-equivalent scheme differently from this study, as a scheme with the same AIMD as TCP, but without packet-loss recovery or fast retransmission.

TCP-Friendliness

Steady State and Transient State

As shown in Table 1, the term steady state is used in the description of the three criteria. A steady-state network originally meant a network with negligible change over an arbitrarily long period. By this definition, the Internet would not be in the steady-state condition unless the term arbitrarily long is removed from the definition. The measured result in [13] reveals that the packet-loss condition experienced by an Internet flow may consist of multiple *minute-scale* steady-state regions, and the time interval between any two consecutive losses may be mutually independent and have the same probability distribution, that is, be independently and identically distributed (i.i.d.), within a region. Thus, a TCP-friendly scheme should use the same bandwidth as TCP in a steady-state region, while being *aggressive* enough to capture the available bandwidth and being *responsive* enough to protect itself from congestion, as the packet-loss condition changes across regions (the transient state). Notably, a packet loss (event) in this study denotes an event causing a TCP flow halving its

Scheme	Full name	Parameters	Reference
GAIMD	General additive inc./multiplicative-dec.	$\alpha = 0.2$, $\beta = 0.125$	[1]
IIAD	Inverse-inc./additive-dec.	$\alpha = 1.0$, $\beta = 0.67$, $k = 1$, $l = 0$	[5]
SQRT	Square-root inc./dec.	$\alpha = 1.0$, $\beta = 0.67$, $k = 0.5$, $l = 0.5$	[5]
SIMD	Square-inc./multiplicative-dec.	$\beta = 0.0625$, $k = -0.5$, $l = 1$	[4]
AIAD/H	Additive inc./dec. with history	$\beta = 0.25$, $k = 0$, $l = 0$	[4]
TFRCP	TCP-friendly rate control protocol	Interval = 5 seconds	[6]
TFRC	TCP-friendly rate control	The number of samples = 8	[2]
TEAR	TCP-emulation at receiver	The number of samples = 8	[3]

Table 2. *The control parameters used in each scheme.*

Policy	Fairness	Aggressiveness		Responsiveness
Aspect	***Throughput adjusting***	***Step of each inc.***	***Curve type***	***Life cycle of loss statistics***
GAIMD	Window-based	Nonhistorical	Linear	Variable history
IIAD	Window-based	Historical	Sublinear	Nonhistorical
SQRT	Window-based	Historical	Sublinear	Variable history
SIMD	Window-based	Historical	Superlinear	Variable history
AIAD/H	Window-based	Historical	Linear	Nonhistorical
TFRCP	Rate-based	Nonhistorical	Superlinear	Fixed history
TFRC	Rate-based	Historical	Linear	Fixed history
TEAR	Rate-based	Historical	Linear	Fixed history

Table 3. *Taxonomy in fairness, aggressiveness, and responsiveness policies.*

congestion window. Such an event may imply that multiple, consequent packets are discarded. For convenience, this study, like other studies [1–6], ignores the term *event*.

TCP-friendly Criteria

This study uses the following three criteria to describe the proper behaviors of a TCP-friendly scheme. Vojnovic *et al.* presented a criterion, named *conservative* [10]. However, this criterion is suitable only for evaluating schemes that use a TCP throughput formula and therefore, is not considered herein.

TCP-Compatibility — The basic criterion, introduced in RFC 2309 [7], is defined as, "A TCP-compatible flow is responsive to congestion notification and uses no more bandwidth in the steady state than a conformant TCP flow running under comparable conditions (e.g., packet-loss rate, RTT)." As shown in Table 1, this criterion forbids a scheme from providing a flow with more bandwidth than TCP to protect TCP flows from starvation. Based on this definition, a TCP-compatible flow should decrease the throughput at least *as fast* as TCP when the packet-loss condition becomes severe, that is, responsive but not necessarily aggressive. Otherwise, the compatibility criterion would be violated during the long convergence time of the flow.

TCP-Equivalence — This study defines the criterion as, "If given identical network conditions, then a TCP-equivalent flow uses the same bandwidth as a TCP flow when the network condition is either in the steady or transient state." This criterion, unlike *TCP-compatibility*, requires the same bandwidth, not just *no more* bandwidth than TCP. Therefore, a TCP-equivalent scheme is more desirable for transmitting media traffic, because it provides more bandwidth than a TCP-compatible scheme. Moreover, to meet the criterion in the transient state, a TCP-equivalent scheme must consider aggressiveness in addition to responsiveness, that is, if more bandwidth becomes available, then a TCP-equivalent scheme should increase the throughput of its controlled flow as fast as TCP. Finally, TCP-equivalence requires *identical network conditions*, rather than comparable conditions, to ensure the same patterns of packet-loss occurrences and RTT changes. The requirement is necessary to test a scheme as to whether to have the same throughput as TCP, because TCP has different throughputs under the same mean but different variances of loss rate or RTT [14].

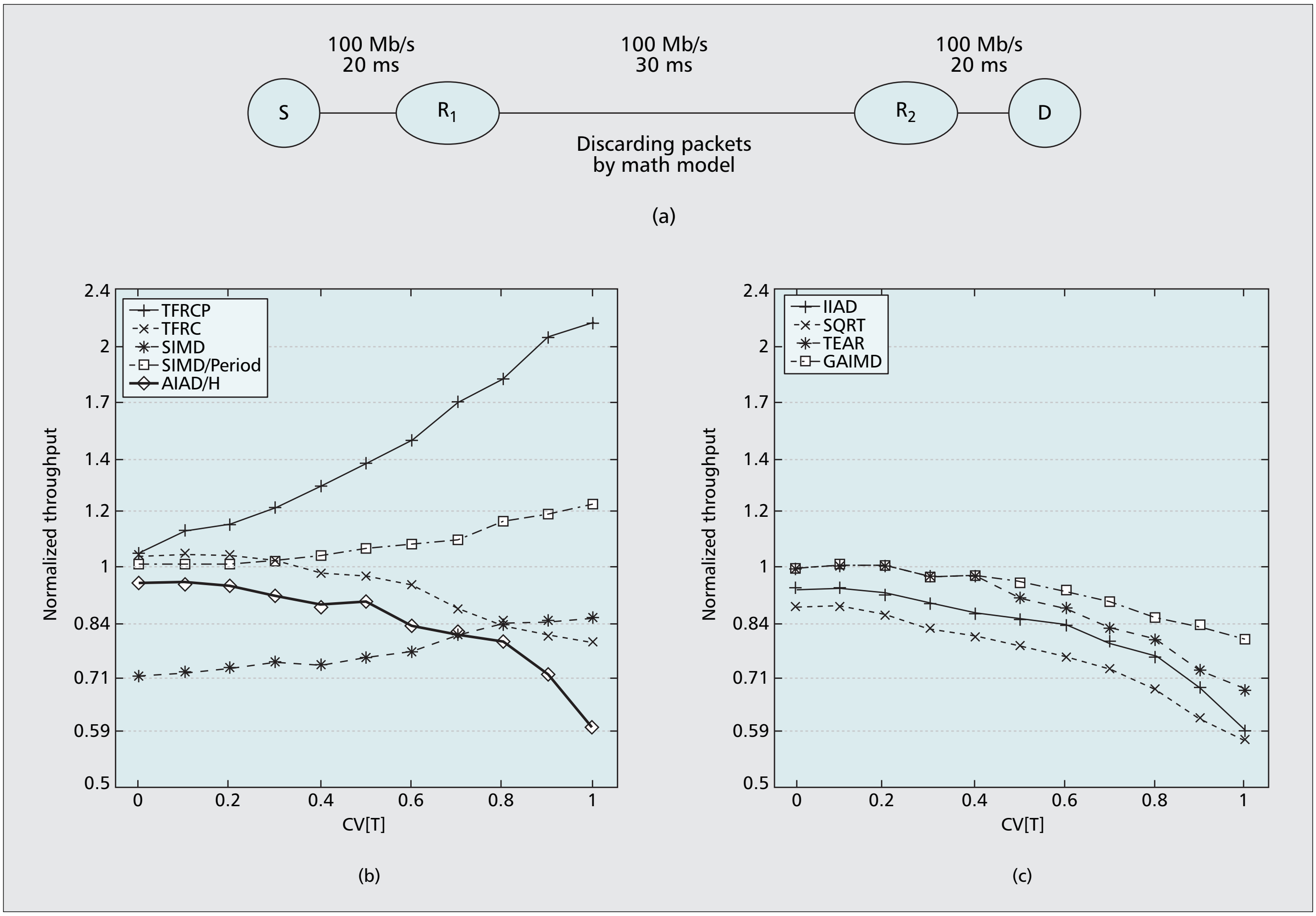
A TCP-equivalent scheme may work well in routers that use well-designed active queuing management (AQM) algorithms to manage their bottleneck links, because such routers may offer the required *premise*, namely *given identical network conditions* to TCP and TCP-equivalent flows. However, if this premise is not supported, then a TCP-equivalent flow may have *more* throughput than a TCP flow when the TCP-equivalent flow experiences fewer packet losses from the routers. To support the premise, these AQMs apply equal packet-loss rate on flows of the same throughput, with the loss rate being directly proportional to the throughput. Since TCP and TCP-equivalent flows adjust the throughput based on their loss rates regulated by the AQM, finally they would have the same throughput and loss rate. Readers interested in this issue can refer to Gwyn *et al.* [15].

TCP-Equal Share — This study defines the criterion as, "A TCP-equal share flow uses the same bandwidth as a TCP flow if both flows compete for the same bottleneck." This criterion should hold regardless of whether the network conditions experienced by the two flows are identical. This criterion differs from TCP-equivalence in its premise, "competing for the same bottleneck," which implies "*competing for the shared bandwidth resources*," but it is not necessary for TCP-equivalence.

TCP-equal share is more realistic than TCP-equivalence. A new scheme is safe to deploy if it provides the same bandwidth as TCP when competing for the same bottleneck, not just when it has identical network conditions. However, achieving TCP-equal share is more challenging than achieving TCP-equivalence, because competing for the same bottleneck does not imply experiencing identical network conditions [10]. Therefore, a TCP-equivalent flow may not be TCP-equal share if it experiences different network conditions from a TCP flow. However, a TCP-equal share flow should have the same bandwidth as a TCP flow, regardless of network conditions, implying that it is also TCP-equivalent.

Taxonomy in Fairness, Aggressiveness, and Responsiveness

The following section investigates the fairness, aggressiveness, and responsiveness policies taken by the selected schemes, as summarized in Table 3.



■ Figure 1. *The throughputs of TCP-friendly schemes normalized with the throughput of TCP, under the loss link whose interloss time has a general exponential distribution. a) shows the artificial loss topology used in the article. For clarity, results are separately shown in b) and c).*

Fairness Policy

The fairness policy of a scheme describes how the scheme adjusts a flow to have equivalent throughput to a TCP flow in the long term under the steady state. As shown in Table 3, the selected schemes use two fairness policies, window-based (WB) and rate-based (RB).

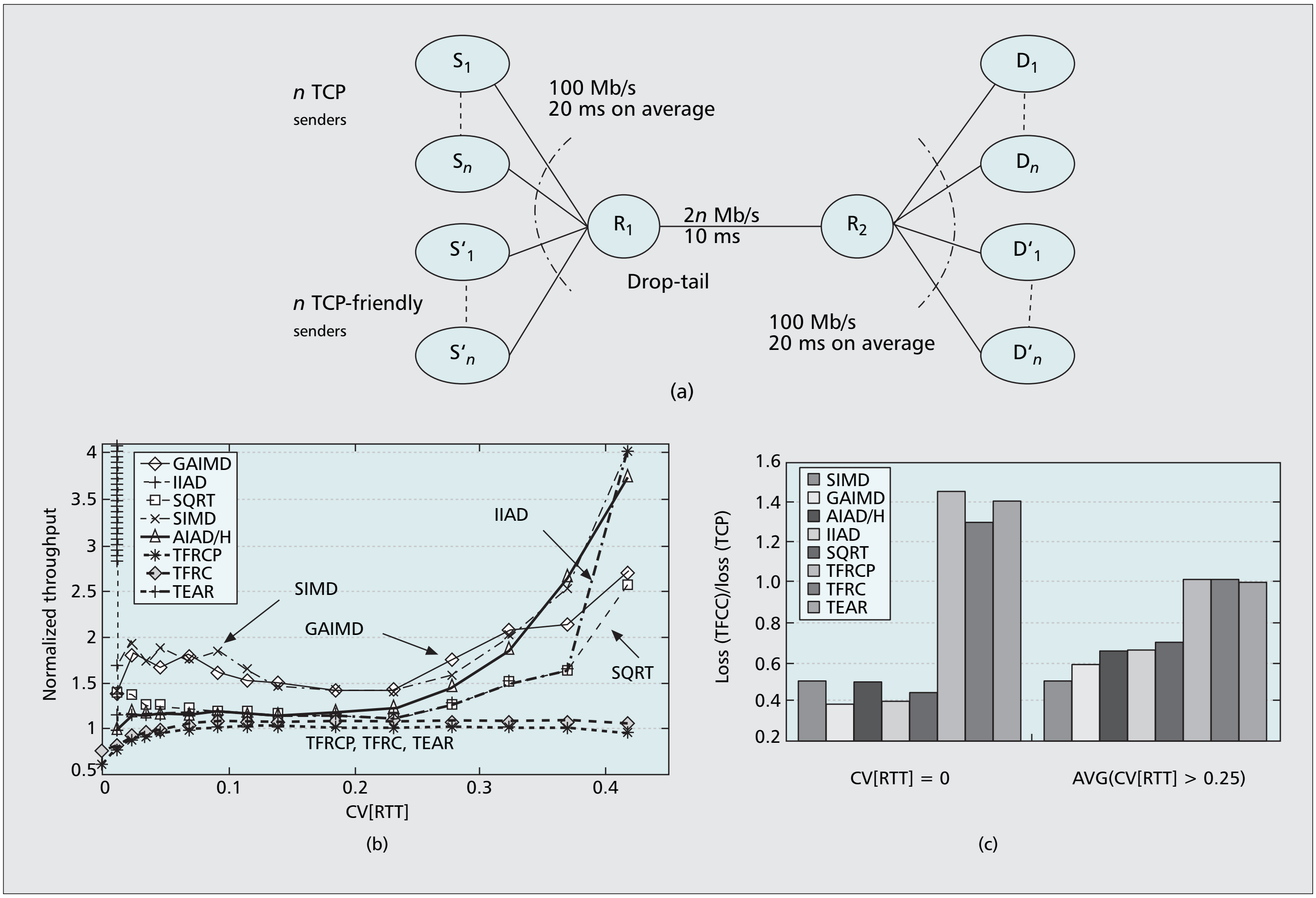
The WB fairness policy controls the throughput by adjusting the congestion window (CWND). The CWND represents the number of packets that can be sent freely without waiting for their acknowledgements and is updated by a set of control parameters (Table 2). A specific relationship exists between the parameters, giving a scheme equal throughput to the TCP. Applying this policy requires the development of control parameters and their specific relationship. For instance, general additive increase/multiplicative decrease (GAIMD) uses two parameters, α and β, to control its CWND, increasing CWND by α for every RTT and decreasing CWND by β if a packet loss occurs. A specific relationship $\alpha = 3\beta/(2 - \beta)$ exists between α and β for achieving the same throughput as TCP. Five of the selected schemes, GAIMD, square-root increase/ decrease (SQRT), inverse-increase/additive-decrease (IIAD), square-increase/multiplicative-decrease (SIMD), and additive increase/decrease with history (AIAD/H), apply the WB policy.

The RB fairness policy directly adjusts the throughput by finely controlling the time between sending two packets and thus has a smoother rate than the WB policy. The RB policy continues to estimate the potential throughput of a TCP flow during its lifetime and repeatedly adjusts the sending rate according to this estimated TCP throughput, enabling a flow to have equal throughput to TCP. Applying this policy requires the development of schemes for estimating the TCP throughput and determining when to adjust the sending rate. The RB policy is applied in three schemes, TFRCP, TFRC, and TCP-emulation at receiver (TEAR).

Aggressiveness Policy

The aggressiveness policy of a scheme describes how the scheme increases the throughput of a flow before encountering the next packet loss. As shown in Table 3, the non-historical policy is taken by GAIMD and TFRCP. The step of increase is *independent of the history of packet losses* and is thus fixed during the whole life of the flow. Unfortunately, this behavior brings the trade-off between aggressiveness and smoothness. For instance, when GAIMD employs a small step for smoothness, a slow rate of increase may prohibit GAIMD from achieving either TCP-equivalence or TCP-equal share when the loss condition changes dramatically. Conversely, TFRCP doubles its rate if it does not encounter any loss during a fixed-time interval, which makes it super-linear, that is, fast and aggressive, but possibly causes large oscillation, that is, poor smoothness.

By contrast, the historical policy has a variable step. For example, to achieve smoothness, SIMD initially takes a smaller increasing step than TCP after encountering a packet loss. SIMD then enlarges the step according to the historical maximum CWND, to increase the aggressiveness before encountering the next loss. The historical policy also enables AIAD/H to dynamically determine a step for linearly increasing the



(a)

(b)

(c)

■ Figure 2. n *TCP-friendly and* n *TCP flows compete for the bottleneck link. The propagation delays among each set of* n *flows are distributed uniformly with* CV*[RTT] = 0~0.42: a) dumbbell topology; b)* n *= 8; c) comparison loss rate.*

throughput. AIAD/H seems to be more adaptive than GAIMD.

Three of the schemes with the historical policy, namely SQRT, IIAD, and SIMD, have non-linearly increasing curves between packet losses, because they change their steps per RTT, instead of per loss. SQRT and IIAD have sub-linearly increasing curves, because they shorten the step inversely with increasing

$$\sqrt{CWND}$$

and CWND, respectively. In contrast, SIMD has a super-linear behavior and thus has the fastest increasing rate, because the step in SIMD is enlarged with the time escaped from the latest loss.

Responsiveness Policy

The responsiveness policy of a scheme describes how the scheme decreases the throughput of a flow when the packet-loss condition becomes severe. The key difference among the policies is the life cycle of the loss statistics used in adjusting the new throughput. The loss statistics include the number of inter-loss packets (the received packets between two losses), the inter-loss time, or the loss rate measured in an interval. There are three policies, namely, non-historical, fixed-history, and variable history, as shown in Table 3.

The non-historical policy ignores the historical packet-loss statistics in decreasing throughput and thus, decreases the throughput at a *constant* speed, thereby producing a trade-off between responsiveness and smoothness. For example, to ensure smoothness, IIAD and AIAD/H employ a small decreasing speed, leading to a long convergence time and the violation of all three criteria, particularly when a significant change of loss condition occurs.

The other two policies consider the historical packet-loss statistics in order to decrease the throughput. In the variable history policy, loss statistics of a large value may have a longer duration to affect the throughput than that of a small value. For example, CWND in GAIMD controls the throughput and can be regarded as a weighted average over all historical values on inter-loss time, where the values obtained earlier have smaller weights [14]. Therefore, early but large values still affect the throughput, even when their weights are small. However, schemes with fixed history consider only the latest *n* loss statistics when computing the new throughput. A loss statistic, regardless of its value, is eliminated from the computation if it is not among the latest *n* values. Fixed-history schemes include TFRC, TEAR, and TFRCP.

Fairness Evaluation

We use ns-2 simulation [16] and examine the fairness of eight different schemes to determine whether they meet the TCP-equivalence and TCP-equal share criteria. The source codes of TEAR, TFRCP, SIMD, and AIAD are not included in the package of ns-2 simulation, but instead are published individually on the Web sites of their authors. Also, this study, similar to [2, 4, 5] uses selective acknowledgment (SACK) options [17] as the TCP version and assumes no delayed acknowledgments. For the simulation, we use packets that were 1,000 bytes long and a maximum window size of 200 packets.

TCP-Equivalence: Artificial-Losses Testing Scenario with Identical Network Conditions

A link with artificial packet losses was used to test for TCP-equivalence. The link discards the passing packets with a specific mathematical model. Such a link guarantees that any two passing flows experience identical loss conditions, thus satisfying the premise in TCP-equivalence, making this link suitable for the test of TCP-equivalence. Sufficient bandwidth was allocated for this link to prevent the packets from being dropped due to overflow.

The selected schemes were tested to determine whether they are robust enough to have the same throughput as TCP under varied artificial links that have different means or coefficient-of-variations (CVs) of inter-loss time. The two statistics were varied because both affect the TCP throughput [14]. A general exponential random variable allows its coefficient-of-variation to be changed while fixing its mean, or vice versa, so it is employed to drop packets at the link. The time between two packet losses thus forms a general exponential distribution, which also is used in [10] to investigate the conservativeness of TFRC. Only the testing result under links with different coefficient-of-variations is shown herein. The result with different means already has been obtained [4, 5].

The artificial link, plotted as the link R_1–R_2 in Fig. 1a, drops one packet every T seconds. T denotes a general exponential distributed random variable where $E[T]$ is fixed at 5 and $CV[T]$ uniformly increases from 0 to 1. The results in Fig. 1 were averaged from five runs of 5200 seconds each, where the data within the first 200 seconds were discarded, and the mean coefficient-of-variation of the simulation results between the five runs was 0.025. Because this coefficient-of-variation is small, it is ignored in the plot to improve the clarity of the figure.

Observation 1: Non-Periodic Losses Should Be Considered in Adopting WB/RB Fairness Policies — Figure 1b–c reveal that none of the WB/RB schemes meet TCP-equivalence under *non-periodic* packet loss ($CV[T] > 0$). When $CV[T] = 1$, GAIMD and TFRC only have 80% throughput of TCP; whereas TEAR, IIAD, SQRT, and AIAD/H have 60% on average, because all schemes, except SIMD, were proposed based only on the periodic-loss assumption, that is, the packet losses occur periodically. The unfairness under $CV[T] = 1$ should be handled by these schemes because the inter-loss time in the Internet may approximate an i.i.d. exponential distribution equivalent to the link with $CV[T] = 1$, according to the observation in [13].

Notably, the TFRCP and SIMD flows exhibit a different trend from other flows in Fig. 1b–c. The difference of TFRCP is due to the convex TCP throughput equation and the fixed rate-adjusting period [10]; whereas that of SIMD occurs because its specific relationship between parameters is based on the packet-loss model with $CV[T] \simeq 1$ [4]. Figure 1b also plots the curve of SIMD variant, SIMD/period, with this design based on $CV[T] = 0$. Unfortunately, SIMD/period violates the TCP-compatibility criterion under non-periodic conditions.

TCP-Equal Share: Low-Multiplexing Testing Scenario with the Same Bottleneck

A dumbbell topology provides the premise of TCP-equal share, that is *competing for the same bottleneck* and thus, is used to verify the TCP-equal share of a scheme in the steady state. As shown in Fig. 2a, n TCP-friendly flows compete with n TCP flows for a single bottlenecked link. All flows have backlogged data for the whole testing period. In particular, this study investigates a low-multiplexing scenario [18], where n is small and Drop-Tail is deployed to manage the bottleneck link, because previous results [1–5] imply that a TCP-equivalent flow may violate TCP-equal share under such a scenario. Drop-Tail is a queuing management algorithm that discards new arrival packets when its managed queue is full.

To indicate the cause of the violation, the scenario used in [1–5] was slightly modified at two points. First, instead of using a fixed capacity, for example, 15 or 60 Mb/s, the link had $2n$ Mb/s. Such a link can provide on average 1 Mb/s of bandwidth for each flow, avoiding the influence of the TCP time-out handling mechanism, as expected from previous studies [1–5]. Second, although multiple rounds were tested for the same n, the RTT heterogeneity of n TCP flows and of n studied flows were enlarged equally over different rounds. The RTT heterogeneity of n flows represents the coefficient-of-variation of the RTTs of these flows, denoted as $CV[\text{RTT}]$. The mean end-to-end propagation delay was set to 50 ms for all rounds. The queue size was 1.5 times the bandwidth-delay product.

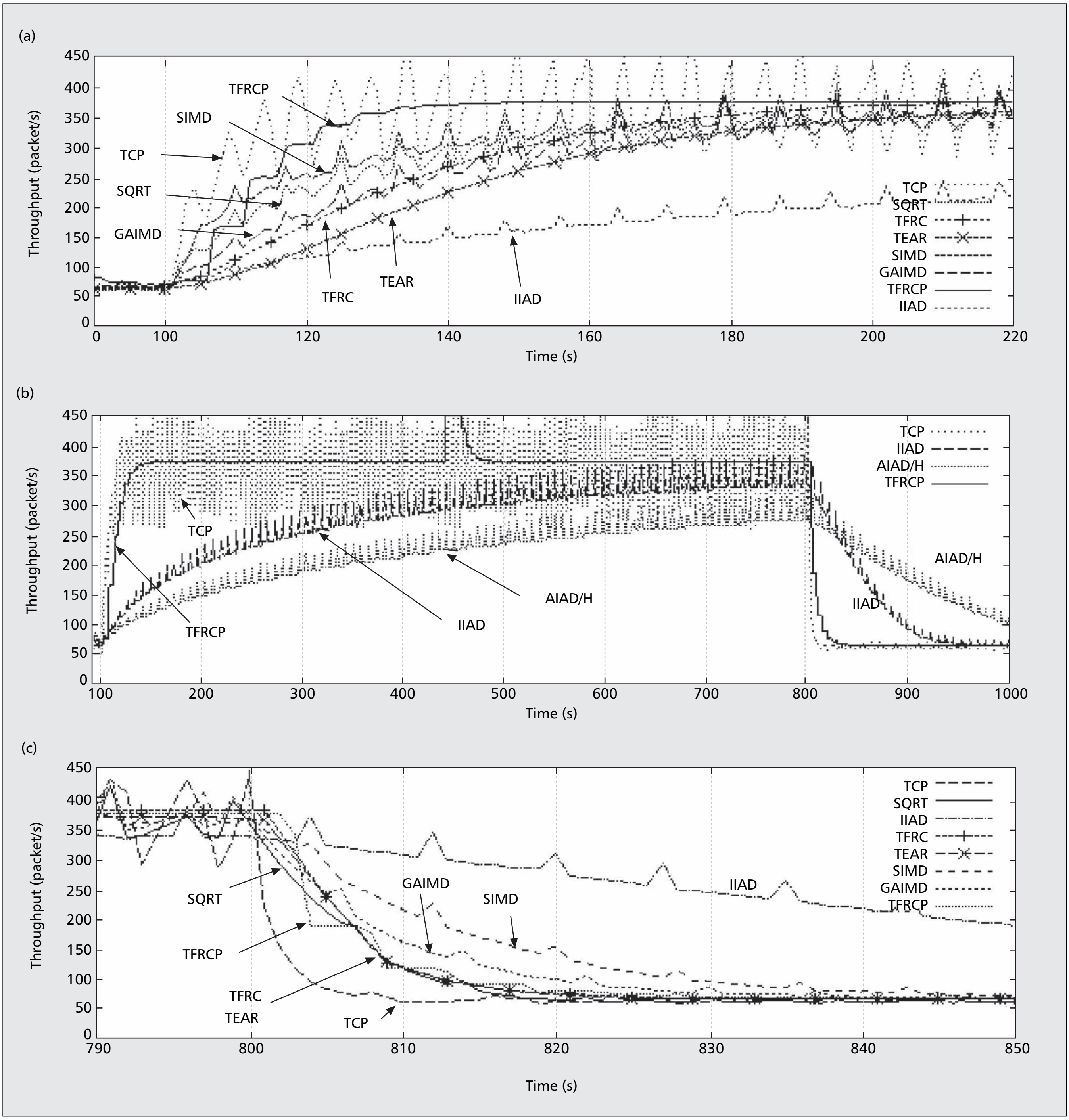
Observation 2: RB Fairness Policy Wins and RTT Heterogeneity Matters for TCP-Equal Share — Figure 2b indicates that the tested schemes do not always ensure TCP-equal share under the scenario, because they are based on the premise of TCP-equivalence, that is, "any two flows experiencing identical network conditions," but not that of TCP-equal share. Thus, these schemes cannot have the same throughput as TCP when the premise of TCP-equivalence is false, that is, they do encounter different numbers of packet losses.

To show that the premise of TCP-equivalence is false under the scenario, Fig. 2c plots the normalized packet-loss rate experienced by the TCP-friendly flows with the shortest RTT, compared with that of TCP flows. The loss rates of shortest-RTT flows are shown because their differences are the most significant among all flows. Three RB schemes, namely TFRCP, TFRC, and TEAR, clearly suffer a higher loss rate than TCP at $CV[\text{RTT}] = 0$, but an equal rate at $CV[\text{RTT}] > 0.25$ that explains their bandwidth sharing with TCP in Fig. 2b. Similarly, the other five schemes suffer a lower loss rate than TCP, so they occupy much more bandwidth than TCP.

Figure 2b also reveals that the *RTT heterogeneity* of the competing flows significantly affects *the fairness* between TCP and TCP-friendly flows. GAIMD and SIMD occupy more bandwidth on average than TCP flows (1.5~4 times), particularly when $CV[\text{RTT}] = 0$ (>10 times), where the number of competing flows is small ($n = 8$, total is 16). The seriously unfair situation at $CV[\text{RTT}] = 0$ also exists even when the total number of competitory flows is 64.

The unfair situation in the five WB schemes results from their exercising the packet acknowledgement mechanism. These schemes, like TCP, delay the transmission of the next data packet if the transmitter does not receive an ACK packet because the queue of a router in the transmission path has overflowed. By the delay, they encounter fewer packet losses and thus have higher throughput than the three RB schemes. Moreover, because the overflow is alleviated by TCP significantly reducing its CWND, these five schemes that slowly reduce their CWNDs, may monopolize the link until the queue is overflowing again. Thus, they have higher average throughput than TCP.

Although neither the WB and RB fairness policies can ensure TCP-equal share, the RB flows would experience similar packet-loss rate to TCP flows and can meet TCP-equal share in most cases, that is, under $CV[\text{RTT}] > 0.05$. By contrast, the WB flows may severely starve TCP flows. Therefore,



■ Figure 3. *The comparison of the slowly convergent behaviors between TCP-friendly schemes under the two-state loss condition.*

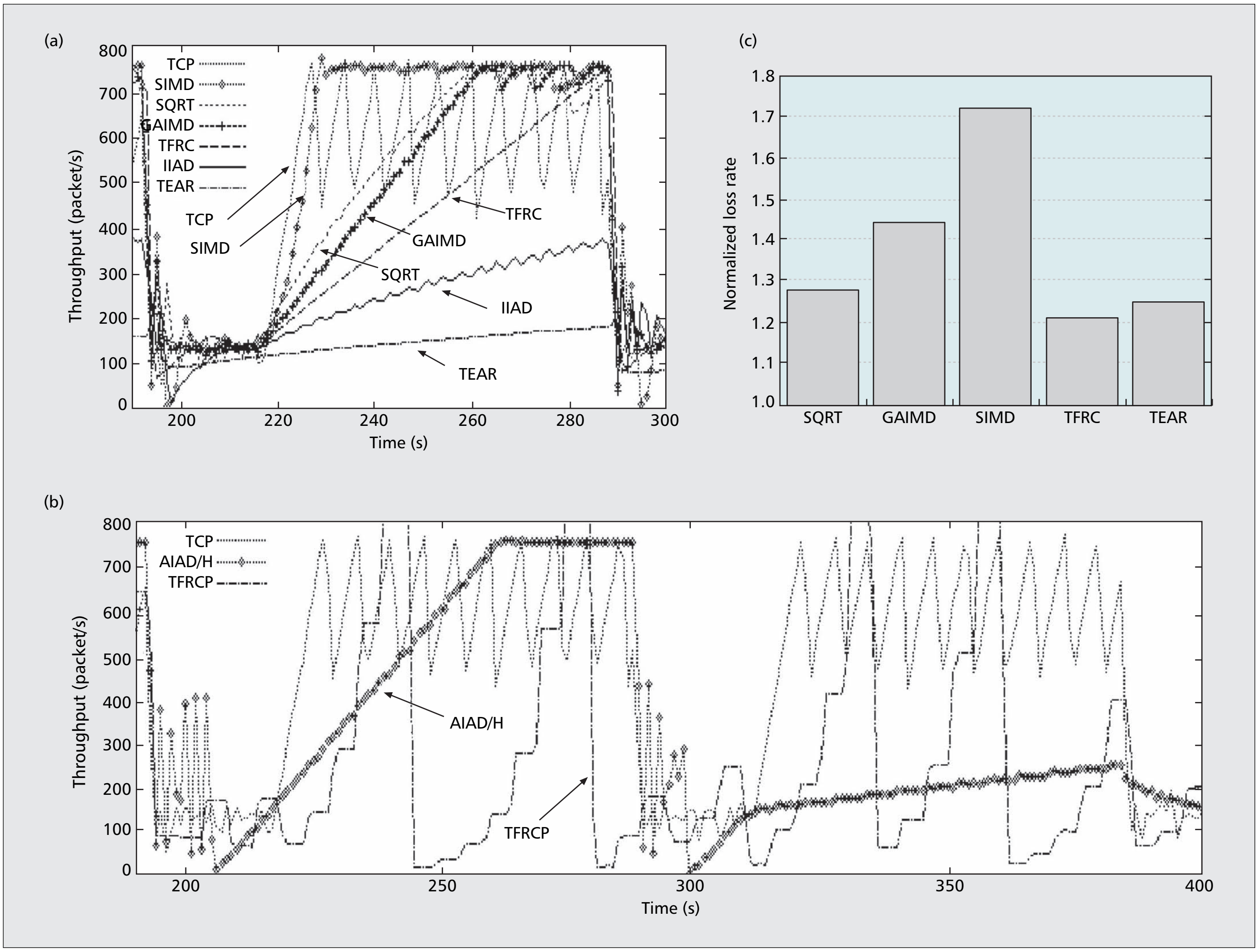
the RB fairness policy should have a better chance than WB of meeting the TCP-equal share. Notably, these TCP-friendly schemes also were tested under a topology with multiple bottlenecks, but the results reveal that their TCP-equal share is unrelated to the number of bottlenecks, when this number increases from 1 to 10.

Evaluation on Aggressiveness and Responsiveness

This section evaluates the selected schemes on their aggressive and responsive behaviors to verify whether they meet the TCP-equivalence and TCP-equal share criteria.

TCP-Equivalence: Two-State Artificial-Losses Testing Scenario with Transient Convergence

The objective of the testing is to observe whether the throughput of the schemes converge as fast as TCP. An artificial-loss link was used, as previously, because it satisfies the premise of TCP-equivalence. However, a two-state packet-loss model was adopted in the link to simulate large changes in the loss conditions. A packet was dropped every five seconds during the 100th~800th seconds and every one second at other times. The result after 100 seconds exhibits aggressive behavior and after 800 seconds exhibits responsive behavior. The RTT of the testing flow was about 140ms.



■ Figure 4. *a) and b) The slowly aggressive behaviors of TCP-friendly schemes under the bursty losses network. b) has a longer timescale than a) to show that TFRCP and AIAD/H have different behaviors in each on/off period. c) The number of loss events encountered by TCP-friendly schemes, normalized to that by TCP, under the low-available bandwidth case.*

Observation 3: Throughput-Inversed Aggressive, Defined in the Following, and Non-Historical Responsive Policies are Inadequate — Figure 3a and the left part of Fig. 3b reveal that IIAD and AIAD/H take 700 seconds to increase their throughput to the new steady throughput. Such a long time is unacceptable, particularly since the other six schemes reach steady throughput within 100 seconds. Surprisingly, although AIAD/H has a linearly increasing curve between two packet losses as mentioned earlier, it has a slower convergence than IIAD. Under this scenario, the reason that both schemes seriously violate TCP-equivalence is their slowly increasing behaviors across over multiple losses, instead of between two losses. Both schemes shorten the increasing step inversely with their throughput per loss. Herein such an unfavorable, slow, and aggressive behavior is called a *throughput-inversed aggressiveness policy*.

Figure 3c and the right part of Fig. 3b verify that the *non-historical responsiveness policy* does not satisfy the TCP-equivalence criterion. The policy brings IIAD and AIAD/H longer convergence time than the other schemes. Figure 3c reveals that the *fixed-history policy* usually takes a shorter time to converge than the variable history policy. TFRC, TFRCP, and TEAR take 20 seconds to converge, which is half the time of GAIMD and SIMD. However, the results also reveal that SQRT, which has a variable history policy, also has a short convergence time. Further analysis indicates that the control parameters used in SQRT have the advantage of a short convergent time.

TCP-Equal Share: Bursty-Loss Testing Scenario with the Same Bottleneck

To test whether a scheme in the transient state meets the TCP-equal share criterion, a two-state constant-bit rate (CBR) arrival traffic with obviously different rates between on and off periods was applied to the dumbbell bottleneck scenario used previously. The oscillating CBR traffic emulates the arrival of a group of TCP flows, significantly changing the packet-loss condition of the bottleneck, and thus providing the required transient-state scenarios. Such traffic in [8] is used to observe how a GAIMD, TFRC, IIAD, or SQRT flow competes with a bursty arrival of TCP traffic.

Whereas Bansal *et al.* [8] showed the statistical behavior for the selected schemes, this study reveals their micro behavior in one on/off period. Additionally, this study tested four schemes, SIMD, AIAD/H, TFRCP, and TEAR that were not tested in [8] but are included here. The bottleneck in the test was a 15 Mb/s link managed with Drop-Tail, where the rate of the two-state CBR traffic oscillated between two values, 14 Mb/s and 9 Mb/s, to vary the bandwidth available for the TCP-friendly flow to 1 Mb/s and 6 Mb/s, respectively. The propagation delay of flows was 60 ms, and the queue size was set to 1.5 times the bandwidth-delay product [8].

Behavior	Fairness			Aggressiveness		Responsiveness	
Criterion	TCP-eq (TCP-comp)[1]	TCP eq-share		TCP-eq (TCP-comp)	TCP eq-share	TCP-eq (TCP-comp)	TCP eq-share
Scenario	Nonperiodic losses	Low multiplexing		Two-state losses	Bursty losses	Two-state losses	Bursty losses
		Homogeneous RTTs	Heterogeneous RTTs				
GAIMD	Δ(O)	X	X	Δ(O)	Δ	Δ(Δ)	Δ
IIAD	X(O)	Δ	X	X(O)	X	X(X)	X
SQRT	X(O)	Δ	X	O(O)	Δ	O(O)	O
SIMD	Δ(O)	X	X	O(O)	O	Δ(Δ)	X
AIAD/H	X(O)	Δ	X	X(O)	X	X(X)	X
TFRCP	X(X)	Δ	O	Δ(O)	X	O(O)	O
TFRC	Δ(O)	Δ	O	Δ(O)	Δ	O(O)	O
TEAR	X(O)	Δ	O	X(O)	X	O(O)	O

O: Satisfactory Δ: Acceptable X: Unacceptable
TCP-eq: TCP equivalence TCP-comp: TCP compatibility TCP eq-share: TCP equal share

[1] The evaluating results on TCP compatibility are shown in the parentheses.

■ Table 4. *Comparison on fairness, aggressive and responsive behaviors among schemes.*

Observation 4: Historical/Super-linearly Aggressive and Fixed-History Responsive Policies are Satisfactory — Figure 4a indicates that *historical/super-linear aggressiveness* is the preferred policy, because it enables SIMD to use the available bandwidth as quickly as TCP, that is, to meet the TCP-equal share criterion and to have a smooth rate after the convergence. By contrast, as shown in Fig. 4b, the non-historical/super-linear policy of TFRCP is not recommended, because a non-historical policy does not change the increasing step to a small value after the convergence, thus causing large oscillations in TFRCP. Notably, care should be taken when using the history. AIAD/H also uses a historical aggressiveness policy but takes too short a history to allow a stable increase during the testing time. TFRCP and AIAD/H are not recommended because of their instability. The *historical/super-linear aggressiveness* policy has the fastest rate of increase and provides both smoothness and aggressiveness, making it most likely to meet the TCP-equivalence and TCP-equal share.

Figure 4c indicates that the *fixed-history responsiveness policy* meets TCP-equal share in terms of responsiveness by *encountering fewer packet losses* than other policies. Although all schemes reduce their throughput within about 15 seconds, Fig. 4c shows that the fixed-history schemes, such as TFRC and TEAR, encounter fewer losses during convergence than variable history schemes, such as SIMD and GAIMD. Therefore, the *fixed-history responsiveness* policy appears to have the best chance of meeting the three criteria, because it considers *bounded* statistics and thus may reach convergence with fewer packet losses or shorter time than other policies, particularly when the loss statistics change significantly.

Related Work

Many AIMD variants have been proposed for different purposes. This study evaluates variants that aim to have a throughput smoother than, but equivalent on average, to that of TCP. Therefore, this study does not evaluate some schemes, for example, AIMD-FC [9] and [19], that stress fast convergence in high-speed links. Additionally, this study focuses on inter-fairness, that is, whether a scheme shares the same bandwidth with TCP. Intra-fairness, that is, the fairness among the flows controlled by the same scheme, is discussed in [12, 20]. Moreover, the schemes selected herein detect congestion only by packet losses as TCP Reno and SACK [17] do. Actually, the RTT variation can be used for the detection, as in TCP Vegas [21], which also provides a smooth rate. However, RTT-based schemes may share bandwidth unfairly in the Internet where most traffic is still controlled by loss-based versions of TCP. C. Zhang and V. Tsaoussidis [22] recently proposed a scheme using both packet losses and RTTs, which may be the solution to the unfairness problem.

Although the topologies discussed have appeared in the literature, for example, [8, 10], *this study revises the simulation scenarios and compares additional schemes to reveal undiscovered phenomena*. For example, this study uses a common topology — dumbbell — to investigate the TCP-equal share of the schemes but changes the RTT heterogeneity to display the difference between WB and RB schemes. Tsaoussidis *et al.* considered the RTT-heterogeneity in [12] but for the intra-fairness of GAIMD flows. Moreover, this study, like [8], uses the oscillating CBR traffic but includes four extra schemes to show three interesting results; that is, SIMD has the fastest aggressiveness, AIAD/H and TFRCP are the most unstable, and TEAR has the slowest aggressiveness. Additionally, this study used the general exponential distribution, as used by Vojnovic *et al.* [10] who show that TFRC may have a lower throughput than TCP under non-periodic losses due to its design. However, this study reveals that schemes other than TFRC have the same unfairness phenomenon, although they control the throughput with methods different from TFRC.

In addition to the congestion control, other factors also must be considered when designing a protocol for carrying streaming traffic. E. Kohler *et al.* [23] discussed these factors in depth and proposed the Datagram Congestion Control Protocol (DCCP). DCCP allows free selection of a congestion control scheme and therefore is the most realistic means for practical use of schemes addressed in this study. The protocol currently includes only two schemes, namely TCP-like and TFRC. We strongly encourage the addition of other schemes to the protocol.

Conclusions

For a TCP-friendly congestion control scheme, meeting TCP-compatibility protects only TCP flows from starvation and a network from congestion but cannot guarantee that the media flow obtains equal throughput to TCP. A good scheme should use the same throughput as TCP in the steady state, but as aggressive and responsive as TCP in the transient state. To examine whether the present TCP-friendly schemes meet the TCP-equivalence and TCP-equal share criteria, we classify the behaviors of eight typical schemes in terms of fairness, aggressiveness, and responsiveness. Additionally, we test the conformance of these schemes to the criteria under four scenarios, namely non-periodic losses, low-multiplexing, two-state losses, and bursty-losses.

Table 4 summarizes the evaluation result for the eight selected schemes for fairness, aggressiveness, and responsiveness. A comparison of the results in this table with the taxonomy results shown in Table 3 demonstrates that a TCP friendly scheme may have desirable TCP-equivalence and TCP-equal share in a general network condition, if it takes the rate-based fairness, historical/super-linear aggressiveness, and fixed history responsiveness policies. The evaluation results of the three recommended policies are shaded in Table 4, which are obviously more satisfactory than those of other policies.

Unfortunately, no scheme simultaneously takes the three recommended policies for meeting the three criteria. However, if protecting TCP flows from starvation, that is, meeting TCP-compatibility is the major concern, then TFRC is recommended. TFRC uses the rate-based fairness and fixed-history responsiveness policies, and therefore has better behaviors under most scenarios than others on average, as shown in the row TFRC of Table 4. However, if fast aggressiveness is the most important property, SIMD is recommended, because it takes the shortest time to converge and then maintains a stable throughput, due to its historical/super-linear aggressiveness policy. Nevertheless, SIMD violates TCP-compatibility under a low-multiplexing bottleneck, because of its window-based fairness policy. Moreover, SIMD spends a longer time or encounters more packet losses before reducing its throughput to the available bandwidth because of its variable, historical responsiveness policy.

As a result of this study, we also observed the following:

- A scheme should consider non-periodic loss models when taking any one of the fairness policies.
- The RTT heterogeneity between competitory flows influences the TCP-equal share of a scheme when the bottleneck is managed by the Drop-Tail algorithm.
- The throughput-inversed aggressiveness and non-historical responsiveness policies should not be taken, because they cannot adapt to the change of packet-loss conditions.

Acknowledgments

The authors would like to thank Dr. Bisdikian and the anonymous referees for their critical and constructive comments on this article. This work is supported in part by National Science Council Program for Promoting Academic Excellence of Universities, Cisco, Intel, and Chung-Hua Telecom.

References

[1] Y. Yang and S. Lam, "General AIMD Congestion Control," *Proc. IEEE ICNP 2000*, Nov 2000, pp. 187–98.

[2] S. Floyd *et al.*, "Equation-Based Congestion Control for Unicast Applications," *Proc. ACM SIGCOMM 2000*, Aug. 2000, pp. 43–56.

[3] I. Rhee, V. Ozdemir, and Y. Yi, "TEAR: TCP Emulation at Receivers-Flow Control for Multimedia Streaming," Tech. rep., Dept. of Comp. Sci., NCSU, Apr. 2000.

[4] S. Jin *et al.*, "A Spectrum of TCP-friendly Window-based Congestion Control Algorithms," *IEEE/ACM Trans. Net.*, vol. 11, no. 3, June 2003, pp. 341–55.

[5] D. Bansal and H. Balakrishnan, "Binomial Congestion Control Algorithms," *Proc. IEEE INFOCOM*, Apr. 2001, pp. 631–40.

[6] J. Padhye *et al.*, "A Model Based TCP-friendly Rate Control Protocol," *Proc. NOSSDAV*, June 1999.

[7] B. Braden, D. Clark, and J. Crowcroft, "Recommendations on Queue Management and Congestion Avoidance in the Internet," RFC 2309, Apr. 1998, http://www.ietf.org

[8] D. Bansal *et al.*, "Dynamic Behavior of Slowly-Responsive Congestion Control Algorithms," *Proc. ACM SIGCOMM '01*, Aug. 2001, pp. 263–74.

[9] A. Lahanas and V. Tsaoussidis, "Exploiting the Efficiency and Fairness Potential of AIMD-based Congestion Avoidance and Control," *Comp. Networks*, vol. 43, no. 2, Oct. 2003, pp. 227–45.

[10] M. Vojnovic and J.-Y. Le Boudec, "On the Long-Run Behavior of Equation-Based Rate Control," *IEEE/ACM Trans. Net.*, vol. 13, no. 3, June 2005, pp. 568–81.

[11] J. Widmer, R. Denda, and M. Mauve, "A Survey on TCP-friendly Congestion Control," special issue of the *IEEE Network Control of Best Effort Traffic*, vol. 15, no. 3, May–June 2001, pp. 28–37.

[12] V. Tsaoussidis and C. Zhang, "The Dynamics of Responsiveness and Smoothness in Heterogeneous Networks," *IEEE JSAC*, vol. 23, no. 6, June 2005, pp. 1178–89.

[13] Y. Zhang *et al.*, "On the Constancy of Internet Path Properties," *Proc. ACM SIGCOMM Internet Measurement Wksp.*, Nov. 2001, pp. 197–211.

[14] E. Altman, K. Avrachenkov, and C. Barakat, "A Stochastic Model of TCP/IP with Stationary Random Losses," *IEEE/ACM Trans. Net.*, vol. 13, no. 2, Apr. 2005, pp. 356–69.

[15] C. Gwyn, L. Miguel, and B. Sujata, "A Survey of TCP-friendly Router-based AQM Schemes," *Comp. Commun.*, vol. 27, no. 15, Sept. 2004, pp. 1424–40.

[16] Network Simulator — ns-2, 2.1b9, http://www.isi.edu/nsnam/ns/

[17] M. Mathis *et al.*, "TCP Selective Acknowledgment Options, " RFC 2018, Apr. 1996.

[18] S. Floyd, "Congestion Control Principles," RFC 2914, Sept. 2000.

[19] N. Itaya and S. Kasahara, "Dynamic Parameter Adjustment for Available-Bandwidth Estimation of TCP in Wired/Wireless Networks," *Comp. Commun.*, vol. 27 no. 10, June 2004, pp. 976–88.

[20] S. Gorinsky, "Feedback Modeling in Internet Congestion Control," *Proc. Next Generation Teletraffic and Wired/Wireless Advanced Networking*, Feb. 2004, pp. 231–34.

[21] L. Brakmo and L. Peterson, "TCP Vegas: End to End Congestion Avoidance on a Global Internet," *IEEE JSAC*, vol. 13, no. 8, Oct. 1995, pp. 1465–80.

[22] C. Zhang and V. Tsaoussidis, "TCP Smoothness and Window Adjustment Strategy," *IEEE Trans. Multimedia*, vol. 8, no. 3, June 2006, pp. 600–09.

[23] E. Kohler, M. Handley, and S. Floyd, "Designing DCCP: Congestion Control Without Reliability," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 36, no. 4, Sept. 2006.

Biographies

SHIH-CHIANG TSAO (weafon@cs.nctu.edu.tw) received a B.S. and an M.S. in computer science from National Chiao Tung University in 1997 and 1999, respectively. He is a Ph.D. candidate in computer science at National Chiao Tung University, where his advisor is Dr. Ying-Dar Lin. His research interests include TCP-friendly congestion control algorithms, fair-queuing algorithms, and Web QoS. He worked as an associate researcher at Chung-Hwa Telecom from 1999 to 2003, mainly in capturing and analyzing switch performance.

YUAN-CHENG LAI (laiyc@cs.ntust.edu.tw) received a Ph.D. degree in computer science from National Chiao Tung University in 1997. He joined the faculty of the Department of Information Management at National Taiwan University of Science and Technology in 2001 and has been an associate professor since 2003. His research interests include wireless networks, network performance evaluation, network security, and content networking.

YING-DAR LIN (ydlin@cs.nctu.edu.tw) received a B.S. from National Taiwan University in 1988 and a Ph.D. from the University of California Los Angeles in 1993. He is a professor of computer science at National Chiao Tung University, where he is also the director of the Computer and Network Center and Network Benchmarking Laboratory (NBL). His research interests include quality of service, deep-packet inspection, and hardware-software co-design. He is spending his sabbatical year, from July 2007, at Cisco, San Jose, California.