Using Adaptive Rate Estimation to Provide Enhanced and Robust Transport over Heterogeneous Networks

Ren Wang, Massimo Valla, M.Y. Sanadidi, and Mario Gerla
UCLA Computer Science Department, Los Angeles, CA 90095, USA
{renwang,mvalla,medy,gerla}@cs.ucla.edu

Abstract

The rapid advancement in wireless communication technology has spurred significant interest in the design and development of enhanced TCP protocols. Among them, TCP Westwood (TCPW) is a sender side only modification to improve TCP performance particularly over heterogeneous networks. The key idea of TCPW is to use rate estimation methods to set the congestion window and slow start threshold after a packet loss. When packet losses are not only due to buffer overflow, but random errors as well, TCPW estimation methods have been shown to provide significant performance improvement. The earliest estimation method, called Bandwidth Estimation (BE), however, may result in over-estimation under certain circumstances, and thus may be unfriendly toward non-TCPW traffic. TCPW CRB (Combined Rate and Bandwidth estimation) and TCPW ABSE (Adaptive Bandwidth Share Estimation), have been later introduced to address this concern. The schemes provide better control of the tradeoffs among efficiency, friendliness, and implementation complexity. CRB may slightly sacrifice the efficiency gain to ensure friendliness. ABSE adaptivity mechanisms are more sophisticated and provide both better efficiency and friendliness. In this paper, we summarize ABSE, which adapts to congestion level, as well as round drip time, and other network dynamics, thus providing enhanced and robust performance under various network conditions. Extensive experiments show that TCPW ABSE is able to enhance TCP performance significantly over "large leaky pipes", while maintaining friendliness toward TCP NewReno. In this paper we show that TCPW ABSE is robust to packet and ACK compression due to cross traffic on forward and backward paths. We also show that ABSE is robust to buffer size variations, which are inevitable in today's networks.

Keywords: TCP Westwood, End-to-End Congestion Control, Rate Estimation, ACK compression, Random Errors.

This research was supported by NSF under Grant ANI-9983138

1. Introduction

The majority of data services in the Internet are carried by TCP (Transmission Control Protocol), with applications ranging from bulk data transmission (FTP) to web browsing (HTTP). The TCP congestion control mechanism has evolved over time, from TCP Tahoe to the currently widely used TCP NewReno. Originally designed for the traditional "wired" environment of the Internet, where congestion accounts for most packet losses, it is well known that the current TCP throughput deteriorates in high-speed heterogeneous networks including wired and wireless links. Unlike wired networks, in wireless networks, many of the packet losses are due to noise and external interference over wireless links. Congestion control schemes in current TCP assume that a packet loss is invariably due to congestion and reduce their congestion window by half, thus the performance deterioration mentioned above.

Many research efforts have been undertaken to adapt TCP to the new wired/wireless environment [GMLW99][KVM99] [GMPG00][BPSK97][BK98]. Such work can be classified, according to the protocol level the schemes are operating on, into three main categories: (1) Schemes relying on link layer enhancements proposing ARQ schemes at the wireless link layer, exemplified by SNOOP [BSAK95] Explicit Loss Notification (ELN) [BK98]; (2) Network layer features beneficial to TCP performance, or providing to TCP explicit congestion information, exemplified by Explicit Congestion Notification (ECN) [Floy94]; (3) End-To-End schemes at the Transport layer, requiring no support form lower layers.

In the third category, TCP Vegas [BP95] and Packet Pair flow control [Kesh91] are general enhancements to TCP and are not intended specifically for heterogeneous paths. The basic idea in TCP Vegas congestion avoidance is that the sender infers network congestion level from changes in round trip time (RTT) and a mismatch of "sending rate" and current window setting. If the observed mismatch becomes larger than a given threshold, the source will decrease its congestion window (*cwnd*), thus reducing its transmission rate. Some fairness issues related to TCP Vegas, however, were studied and reported in [HMM98] [BB00] [Bona99]. The Packet-Pair

(PP) flow control scheme estimates the bottleneck backlog. The larger the backlog, the more severe is the congestion. The estimated backlog is used to adjust the proposed rate congestion control scheme. However, the PP scheme is known to work only under round-robin scheduling at the routers – a feature not available in many commercial routers.

Our research has been focused on an end-to-end approach to provide enhanced transport over heterogeneous networks. To handle wireless losses in such networks, in [CGLMS00][CGMSW01] we have proposed TCP Westwood (TCPW, for short). TCPW design adheres to the end-to-end transparency guidelines set forth in [Clar88] and requires slight modifications, only at the sender side. A TCPW sender attempts to estimate the sending rate to which a connection is eligible. We call this rate the Eligible Rate Estimate (ERE).Using the ERE, a connection is to converge over time to its fair share of the network bandwidth.. Of course it is difficult to determine an accurate ERE and achieve perfect fair sharing. TCPW techniques aim to determine an ERE that provides good fairness and friendliness profiles in sharing network bandwidth in most conditions. By properly monitoring the stream of arriving ACKs, and mining the data in these ACKs, the sender determines its ERE. More specifically the sender learns the amount of data delivered as reported in the ACKs , and from this information gets a sample of the ERE, which is then appropriately averaged using an adaptive discrete low-pass filter. After a packet loss, the congestion window *(cwin)*and slow start threshold *(ssthresh)* are set based on ERE .

TCPW estimation has evolved starting with a method we call Bandwidth Estimation, or BE for short. TCPW BE strategy provides significant throughput gains when error loss is as likely as congestion loss [CGMSW01]. The performance of TCPW BE has been promising, exceeding that of TCP NewReno in large leaky pipes. Consider the situation where TCPW and TCP NewReno connections coexist and share common bottlenecks. Friendliness in this shared environment may be as important as efficiency. Under certain conditions TCP NewReno may experience some performance degradation since TCPW BE estimates may actually exceed the fair share of the connection.

To manage the efficiency/friendliness tradeoffs, we have proposed in [WVNMG02] a new estimation technique called the Combined Rate and Bandwidth estimation (CRB) scheme. CRB includes two estimation methods, the first is BE, mentioned above. The second method is called Rate Estimation, and it differs from BE in the sample definition. In RE the sample is calculated by considering ACKs that arrived over a fixed interval of time, while in BE the sample is calculated using the last two ACKs. RE provides better estimates under congestion condition. However, RE may underestimate the eligible rate when packet losses are due to random errors. We have shown that RE works best when packet loss is mostly due to congestion. If, on the other hand, packet loss is mostly due to link errors, BE gives better performance. In CRB, a connection first infers the predominant cause of packet loss (buffer congestion or random error) and then uses the most appropriate estimation method. Simulation shows that the adaptive CRB provides an effective compromise between efficiency and friendliness. CRB is friendlier to NewReno, but this is achieved at the cost of reduced efficiency.

In [WVSG02], we have introduced a new estimation method that achieves both efficiency and friendliness. The new method, the Adaptive ABSEBandwidth Share Estimation (ABSE) provides continuous adaptivity to the congestion level, thus maintaining both efficiency and friendliness in a wide range of conditions.

BE also suffers performance deterioration under certain conditions such as: (1) Very small bottleneck buffer size relative to the bandwidth delay product, (2) Significant packet and ACK compression due to cross traffic on forward and backward paths. In this paper, we summarize the TCPW ABSEABSE scheme introduced in [WVSG02], and investigate the different estimation techniques and reveal the rationale behind the new estimation method. We study the robustness of the efficiency, fairness and friendliness gain thatABSE provides, showing that it is able to achieve robust and enhanced transport over wired/wireless networks, via adapting to network congestion level and automatically tuning the estimator parameters according to round trip time and network stability. As a result the ABSEABSE estimator is agile enough to react to persistent changes, while tolerating transient noise.

It is worth noting that all TCPW variants (including TCPW ABSE) rely only on information readily available at the sender, using the current TCP header, and do not require any support from receivers or any network component.

The paper is structured as follows. For "self containment", in Section 2, we briefly describe TCPW ABSE protocol, and in section 3, we discuss the ABSE estimation algorithm and how it addresses the limitations in BE/RE methods. In Section 4, we evaluate the robustness of TCPW ABSE performance, in terms of estimation accuracy and end-to-end throughput, against packet and ACK compression, various buffer capacities and channel loss rates. In Section 5, we provide a fairness and friendliness evaluation of TCPW ABSE. Finally Section 6 concludes the paper.

2. TCPW ABSE Protocol

Under ABSE, the sender adaptively determines the Eligible Rate Estimate (ERE) of a connection, based on information in the ACKs, and the rate at which the ACKs are received. After a packet loss indication, which could be due to either congestion or link errors, the sender uses the estimated rate ERE to properly set the congestion window and the slow start threshold.

Further details regarding rate estimation are provided in following Sections. For now, let us assume that a sender has determined the connection ERE as mentioned above. Upon a packet loss indication (3 DUPACKS or a timeout), the sender sets *cwin* and *ssthresh* as follows (For a more complete description of the protocol, please refer to [CGMSW01][WVSG02]):

```
if (3 DUPACKs are received)
  ssthresh =  (ERE * RTTmin) / seg_size;
  if (cwin > ssthresh) /* congestion avoid. */
    cwin = ssthresh;
  endif
endif

if (coarse timeout expires)
  cwin = 1;
  ssthresh = (ERE * RTTmin) / seg_size;
  if (ssthresh < 2)
    ssthresh = 2;
  endif;
endif
```

3. Adaptive Sampling and Filtering in ABSE

In this section, we Provide more details regarding ABSE, and discuss how ABSE addresses the inherent limitations in BE and RE.

3.1. Adapting to Network Congestion Level in ABSE

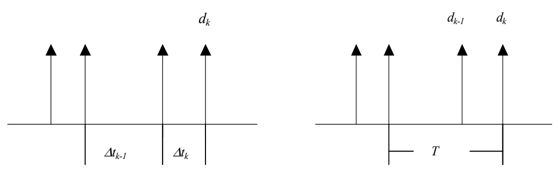
The Eligible Rate Estimates (ERE) are determined using atime-varying coefficient, exponentially-weighted moving average (EWMA) filter, which has both adaptive gain and adaptive sampling. Let t_k be the time instant at which the k_{th} ACK is received at the sender. Let s_k be the ERE sample, and $\hat{s}_k$ the filtered estimate of the ERE at time t_k. Let α_k be the time-varying coefficient at t_k. The ABSE filter is then given by:

$$\hat{s}_k = \alpha_k \hat{s}_{k-1} + (1-\alpha_k) s_k \tag{1}$$

where $\alpha_k = \frac{2\tau_k - \Delta t_k}{2\tau_k + \Delta t_k}$, and τ_k is a filter parameter which determines the filter gain, and varies over time adapting to RTT and other path conditions. In this Section we assume a fixed τ_k and focus on adaptive sampling, leaving the discussion of τ_k adaptation to Section 3.2.

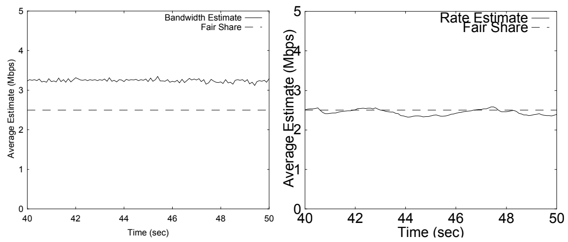
In the filter formula above, the ERE sample at time k is $s_k = \frac{\sum_{t_j > t_k - T_k} d_j}{T_k}$, where d_j is the number of bytes that have been reported delivered by the j_{th} ACK, and T_k is an interval over which the ERE sample is calculated.

3.1.1. Estimation Limitations in BE and RE. In TCPW BE estimation, the BE sample is obtained from the most recent pair of ACKs. Since TCP traffic tends to be bursty (clustering), BE may overestimate the connection bandwidth share. In (fixed-interval) Rate Estimation (RE) the RE sample is computed based on the amount of data acknowledged during the latest interval of time T, i.e., a train of packets in the interval T. The resulting RE estimate is generally lower, since the calculation in RE has the effect of equally spacing the ACKs over the interval T. A long time interval T in RE, thus, produces a more conservative estimate that overcomes the effect of packets clustering due to congestion and compression. On the other hand, RE fails to measure a more appropriate eligible rate when the link is underutilized due to random errors [WVSNG02]. Fig. 1(a) and Figure 1(b) illustrate the sampling time interval used in BE and RE methods respectively.



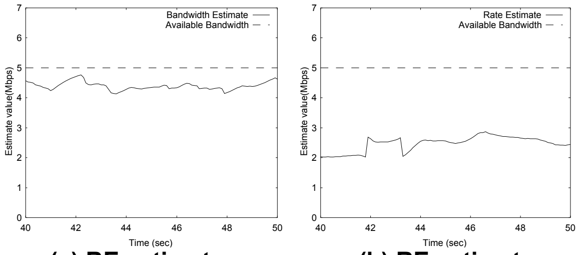
(a) BE (pair of packets) (b) RE (train of packets)
Fig. 1. BE and RE sampling illustration

We evaluate BE and RE using simulation. All the results presented in this paper have been obtained using the Network Simulator [ns2]. In Fig. 2 we compare the BE and RE estimates under network congestion. The estimate traces are produced in a scenario with 2 TCP connections competing for a single 5Mbps bottleneck link, the RTT is 70ms, and the bottleneck buffer is set to equal to pipe size (bandwidth RTT production). From Fig. 2(a), we can see that BE has an average estimate of 3.2Mbps, which is an overestimate of the fair share (2.5Mbps). Figure 2(b) shows that RE provides a right-on-target estimate slightly oscillating around the fair share.



(a) BE estimate (b) RE estimate (sampling interval T =RTT)
Fig. 2. BE and RE under Network Congestion (5Mbps Link, 70ms RTT, 2 competing TCP connections)

The next simulation experiments aim to evaluate the performance of BE and RE in the presence of random link errors. Using the same network configuration except that the bottleneck link experiences a 1% random packet error rate. In the simulations, only one TCP connection is active, thus the available bandwidth (and the fair share) is 5Mbps. Fig. 2 shows that RE estimate settles at about 2.4 Mbps, under estimating its fair share; while BE is able to obtain a more accurate estimation when the link is under utilized.



(a) BE estimate **(b) RE estimate**

Fig. 3. BE and RE in the presence of random errors (5Mbps link, 70ms RTT, 1 TCP, 1% packet loss rare)

3.1.2. ABSE adaptive sampling. We have proposed the Combined RE and BE (CRB) scheme [WVSNG02], which switches between RE and BE, after identifying whether the predominant cause of packet loss is error or congestion. The interval T over which RE is calculated is fixed, and the discriminator of cause of loss relies on a threshold mechanism θ. When the ratio of expected traffic to achieved rate exceeds the threshold θ, and a loss is detected, the predominant cause of loss is estimated to be congestion. With fixed T and θ, CRB suffers efficiency degradation in order to achieve friendliness to NewReno connections.

To preserve both efficiency and fairness, we have introduced the use of a continuously adaptive sampling interval T. The more severe the congestion, the longer T should be. ABSE provides an adaptive sampling scheme, in which the time interval T_k associated with the k_{th} received ACK is appropriately chosen between two extremes (as illustrated in Fig. 4), depending on the network congestion level. The sampling interval ranges between T_{min} and T_{max}. T_{min} is the ACK interarrival time, while T_{max} is set to RTT. This adaptation idea of the sampling interval is illustrated in Fig. 4.

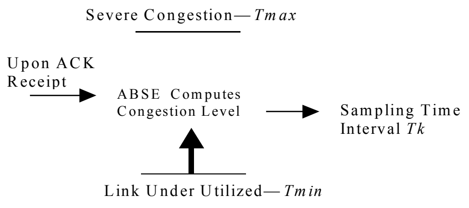

Severe Congestion— Tmax
Upon ACK Receipt
ABSE Computes Congestion Level
Sampling Time Interval Tk
Link Under Utilized— Tmin

Fig 4. Illustration of sampling time interval Tk adapting to network congestion level

To determine the network congestion level, the ABSE estimator compares the Estimated Eligible Rate (ERE) with the instantaneous sending rate obtained from *cwin/RTTmin*. A measure of the path congestion level is thus obtained. The difference between the instantaneous sending rate and the achievable rate, clearly feeds the bottleneck queue, thus revealing that the path is becoming congested. The larger the difference, the more severe the congestion, and the larger the new value of T_k should be.

When the k_{th} ACK arrives, the estimator first check the relation between ERE estimate $\hat{s}_{k-1}$ and the current *cwin* value. When $\hat{s}_{k-1} * RTT_{\min} \geq cwin$, indicating a path without congestion, T_k is set to T_{min}. Otherwise, T_k is set to:

$$T_k = RTT * \frac{cwin - (\hat{s}_{k-1} * RTT_{\min})}{cwin} \qquad (2)$$

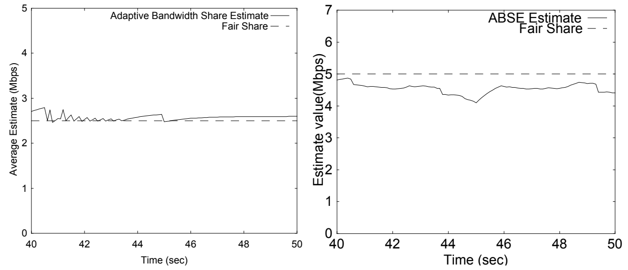
Or upon rearrangement:

$$T_k = RTT * (\frac{cwin}{RTT_{\min}} - \hat{s}_{k-1}) / \frac{cwin}{RTT_{\min}} \qquad (3)$$

In equation (4), $cwin / RTT_{\min}$ is the expected sending rate, while $\hat{s}_{k-1}$ is the estimated rate the network allowed. After T_k is chosen, the ERE sample associated with k_{th} received ACK is then expressed by:

$$s_k = \frac{\sum_{t_j > t_k - T_k} d_j}{T_k} \qquad (4)$$

Using the same network configuration as in Section 3.1.1, we evaluate the accuracy of ABSE estimates, both in the presence of network congestion (Fig. 5(a)), and in the presence of link errors (Fig. 5(b)).



(a) Network congestion **(b) Link is under utilized due to random errors (1%)**

Fig. 5. ABSE estimate (5Mbps Link, 70ms RTT)

3.2. Adapting to RTT and Network Instability in ABSE

The EWMA (Exponentially Weighted Moving Average) filter $\hat{s}_k = \alpha_k \hat{s}_{k-1} + (1 - \alpha_k) s_k$ (1), used in TCPW, places more importance on recent data and discounts older data in an exponential manner. The value of the parameter α_k, dictates the degree of filtering. The smaller α_k the more agile the filter, and the larger α_k the more stable the filter. The relationship between the filter behavior and the parameter τ_k is analyzed in detail in [WVSG02]. Basically, when τ_k is larger, α_k will be larger and the filter tends to be more stable and less agile.

In BE, the sender set τ to a fixed constant, which is used by all TCP connections initiated by this sender, despite the variance in their RTT and path instability. The drawback of this setting is if τ is too large, the filter will be very slow in following the change of path. In ABSE, we propose that the parameter τ_k adapts to network conditions to dampen estimates when the network exhibits very unstable behavior, and react quickly to persistent changes. A stability detection filter can be used to dynamically change the value of τ_k. We measure the network instability U with a time-constant EWMA filter [KN01]:

$$U_k = \beta U_{k-1} + (1-\beta)\left|s_k - s_{k-1}\right| \qquad (5)$$

In (6), s_k is the k_{th} rate sample, and β is the gain of this filter, which is set to be 0.6 in our experiments. When the network exhibits high instability, the consecutive observations diverge from each other, as a result, U_k increases. Under this condition, increasing the value of τ_k makes the ABSE filter, as in Equation (1), more stable. The adaptation of parameter τ_k is illustrated in Fig. 6.

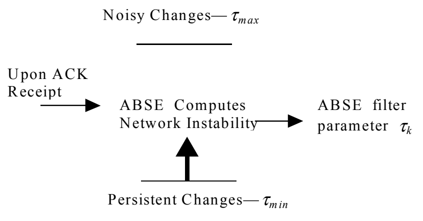

Noisy Changes— τ_{max}
Upon ACK Receipt
ABSE Computes Network Instability
ABSE filter parameter τ_k
Persistent Changes— τ_{min}

Fig. 6. Illustration of ABSE Filter τ_k Adaptation

When a TCP connection is operating normally, the interval between the consecutive acknowledgements are likely to vary between the smallest the bottleneck capacity allows, and one RTT. Therefore, τ_k should be larger than one RTT, thus $\tau_{min} = RTT$. We set τ_k to be:

$$\tau_k = RTT + N * RTT\frac{U_k}{U_{\max}} \qquad (6)$$

The value of RTT in the expression above is obtained from the smoothed RTT estimated in TCP. The factor N is set to be 10 in our experiments, which gives good performance under various scenarios. U_{max} is the largest instability in the ten most recent observations as in [KN01].

The τ_k adaptation algorithm described above is able to achieve agility to persistent changes while retaining stability against noise. In Fig. 7, we show the ERE estimated by applying adaptive τ_k. The sampling time interval T_k is fixed to the interval of the last two ACKs in this set of experiments. The simulation configuration features a 5Mbps bottleneck link with a one-way propagation time of 35ms. The bottleneck router is FCFS Drop-tail with buffer capacity equal to the pipe size (i.e. the bandwidth-delay product). The TCPW connection shares the bottleneck with on-off non-adaptive UDP traffic with time varying intensity. The actual available bandwidth for the TCP connection is shown in dotted line. In comparison to the results obtained by fixed τ in [WVSG02], the estimator with adaptive $\tau_{k,}$ can achieve faster response with less oscillations due to noise (Fig. 7).

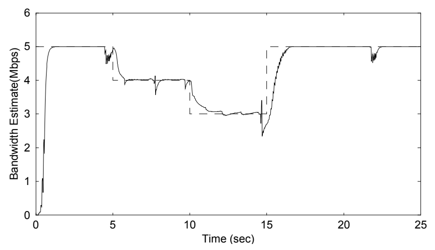


Fig. 7. ABSE estimation with adaptive τ_k

To conclude this Section, Fig. 8 shows a block diagram summarizing the different functions in ABSE.

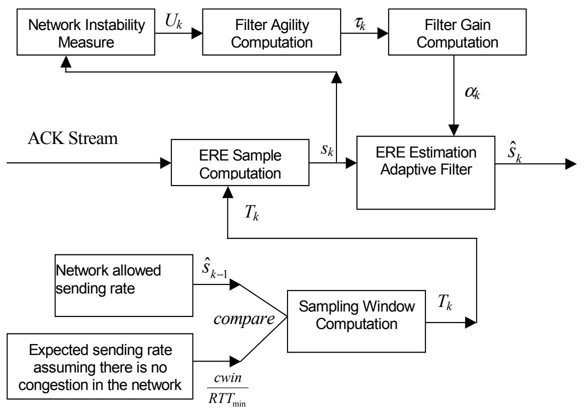

Network Instability Measure
U_k
Filter Agility Computation
τ_k
Filter Gain Computation
α_k
ACK Stream
ERE Sample Computation
s_k
ERE Estimation Adaptive Filter
$\hat{s}_k$
T_k
Network allowed sending rate
$\hat{s}_{k-1}$
compare
Sampling Window Computation
T_k
Expected sending rate assuming there is no congestion in the network
$\frac{cwin}{RTT_{\min}}$

Fig. 8. Block Diagram of ABSE Algorithm

4. Robustness of TCPW ABSE

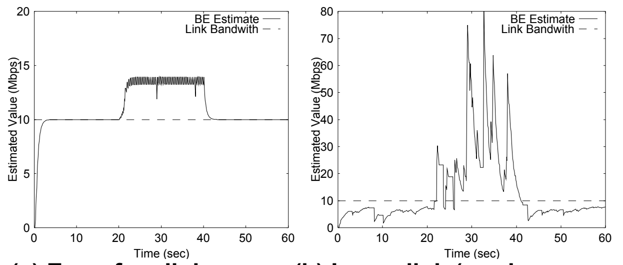
In this Section, we evaluate the robustness of TCPW ABSE under various network conditions, including packet and ACK compression due to cross and two-way traffic, different router buffer capacities, and random errors on wireless links. The results show that TCPW ABSE is able to provide enhanced and robust transport over heterogeneous networks

4.1. Two-way traffic and ACK compression

With one-way traffic, ACK packets are spaced no closer than the transmission time of the data packets, and this spacing remains unchanged as long as there is no congestion on the return path. With two-way data traffic, however, the ACK compression occurs when the spacing between the ACK packets is reshaped due to the data packets queued on the return path [ZSC91]. Recall that

TCP traffic tends to be bursty, when a cluster of ACK packets encounters a nonempty queue due to backward traffic, they lose their original time spacing, and leave the queue separated only by the transmission time of the ACK packet. Since ACK packets are typically much smaller than data packets, this causes a cluster of ACK packets to arrive at the source much more closely spaced, thus ACK compression.

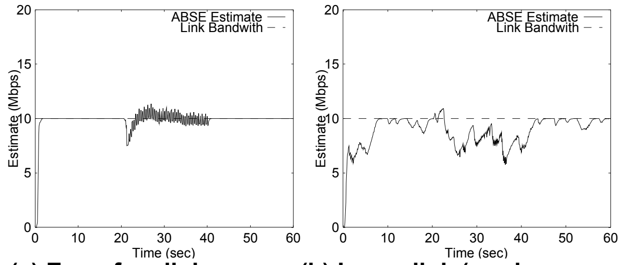
Therefore, if only the time interval between the consecutive ACK packets is used to obtain eligible rate samples, the ACK compression would lead to overestimation. Fig. 9 shows the time traces of BE estimates obtained considering a 10Mbps bottleneck link shared by two TCPW BE connections of opposite directions. The TCPW BE connection in the backward direction is active between 20 and 40 seconds. The results show BE overestimate the eligible rate in the face of ACK compression. Fig. 9(b) indicates that the overestimation is especially severe when the link is lossy. This is due to the frequent cumulative ACKs, which produce very large samples.



(a) Error-free link **(b) Lossy link (random error rate of 0.5%)**

Fig. 9. BE estimate in the presence of ACK compression due to two-way traffic

In ABSE, when congestion is detected, the algorithm uses a longer time interval to obtain a rate sample, buffering the ACK packets at the sender, and absorbing the effect of a cluster of closely spaced ACK packets. Moreover, when an extreme large sample due to cumulative ACK occurs, the ABSE filter detects such condition as high instability and more strongly filters such a sample. This suppresses the effect of large samples and maintains an accurate estimate.



(a) Error-free link **(b) Lossy link (random error rate of 0.5%)**

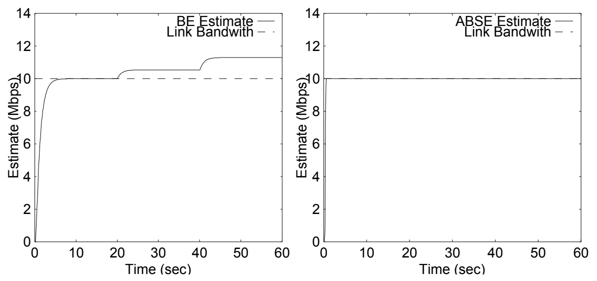
Fig. 10. ABSE estimate in the presence of ACK compression by two-way traffic

To evaluate the robustness of the ABSE estimation in the face of ACK compression, we use the same scenarios as above with two TCPW ABSE connections. The results in fig. 10(a) (error-free link) and Fig. 10(b) (lossy link) reveal that ABSE estimation method is able to sustain reasonably robust estimate against ACK compression.

4.2. Cross Traffic and Data Packet Compression

Data packet compression can reduce the estimates accuracy of in the same manner that ACK-compression does. Data packet compression happens when cross traffic enters the network path at *post-narrow links*, which are links after bottleneck link on the path but with greater transmitting capacities [DRM01]. This effect could cause the packets dispersion at the receiver to be less than the packets dispersion at the bottleneck.

To evaluate the impact of packet compression due to cross traffic on BE and ABSE, we run simulations with a 4-hop network path. The bandwidths of the 4 hops are 20, 10, 20, 30 Mbps respectively, with the second hop being bottleneck. A TCPW connection goes through all 4 hops, while UDP source 1 sends data at the third hop starting at 20sec, UDP source 2 sends data at the fourth hop starting at 40sec, both with rate of 5Mbps. Fig. 11 shows the time traces of BE and ABSE estimates in the presence of such cross traffic. The results show that BE overestimates the available bandwidth due to the same reasons as in ACK compression, while ABSE maintains an accurate estimation despite of the cross traffic (fig 11(b)).



(a) BE estimate **(b) ABSE estimate**

Fig. 11. BE and ABSE estimate in the presence of packet compression by cross traffic

4.3. Buffer Capacity Variation

It is well known that TCP performance can be impacted greatly by buffer capacity on the bottleneck router [BA00]. In this Subsection, we evaluate the performance of TCPW ABSE, TCPW BE and NewReno under various buffer capacities. The results show that TCPW ABSE provides relatively robust throughput performance in the face of small buffer size comparing to NewReno and TCPW BE, and better delay performance than NewReno when the buffer size is very large.

4.3.1 Single Connection. We first conduct experiments to test the performance of single TCP connection with different buffer sizes. The simulation configuration features a bottleneck link of 5Mbps bandwidth. The round trip propagation time is 70ms and the data packet size is 1000 bytes. Thus the pipe size, i.e. the bandwidth-delay product, is 42 packets. As can be observed from Fig. 12, NewReno utilization is severely degraded with small buffer capacity, and finally catches up when the buffer approaches the pipe size. TCPW ABSE and TCPW BE, on the other hand, can achieve much higher utilization despite the small buffer capacity.

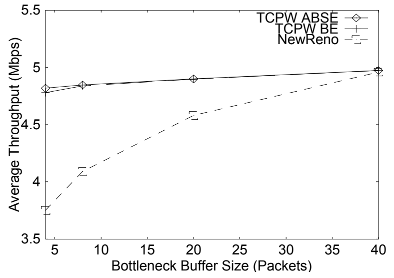
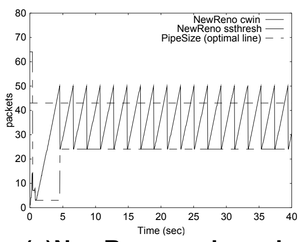
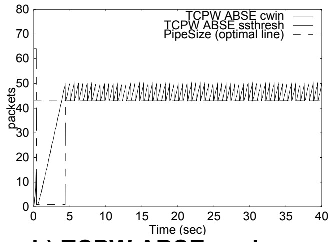


Fig. 12. Throughput vs. Buffer Size (single connection)

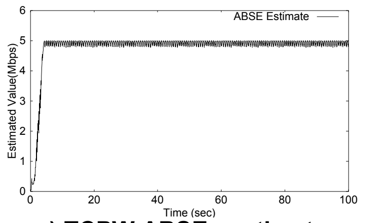
TCP behavior in Fig. 13 reveals the reason behind the performance degradation of NewReno and the robustness of TCPW with small buffer capacity. A buffer overflow occurs when the *cwin* exceeds 46 packets, which is the sum of the pipe and the buffer size. Upon the detection of a packet loss, the NewReno sender (see Fig. 13(a)) sets the new *cwin* and *ssthresh* to 23 packets (half of the old *cwin*). This value is much lower than the pipe size needed to fully utilize the link. Thus, setting *cwin* and *ssthresh* to half causes the router to be idle and the link under utilized.



(a)NewReno: cwin and ssthresh



b) TCPW ABSE: cwin and ssthresh



c) TCPW ABSE : estimate
Fig. 13 Illustration of TCPW ABSE and NewReno behavior with small buffer capacity: 4 packets)

On the other hand, Fig. 13(b) shows that TCPW ABSE reduces its *cwin* and *ssthresh* according to the eligible rate estimate (Fig. 13(c)), which is very close to the optimal value, i.e. equal to the pipe size. Therefore, TCPW ABSE can better utilize the link than NewReno despite of the small buffer. Note that TCPW BE also achieves good performance, since BE estimate is also very accurate in the case of single connection (thus no congestion).

Next we evaluate what TCPW can gain when the buffer capacity is very large, in which case, both TCPW and NewReno can fully utilize the link. In Fig. 14, we show the average packet queuing time in the case of large buffer sizes. We observe that TCPW ABSE achieves better delay performance than NewReno.

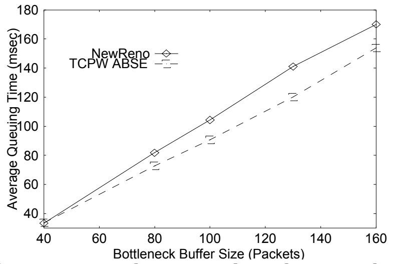
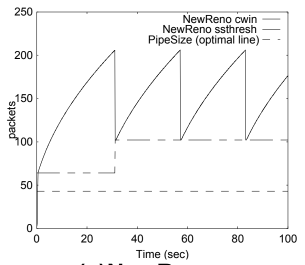
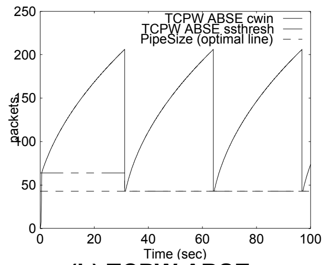


Fig. 14. Average packet queuing time vs. bottleneck buffer capacity

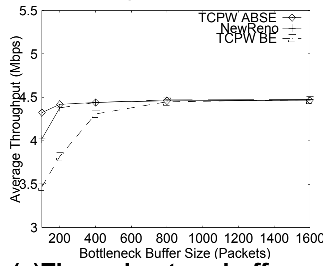
This again can be explained by the different behavior of the two protocols shown in fig. 15. In Figure 15(a), the NewReno sender, after a buffer overflow, reduces its *cwin* to half, which is much larger than the pipe size. The TCPW ABSE sender, however, reduces the *cwin* all the way down to the estimated pipe size (Fig. 15(b)), helping to drain the bottleneck buffer queue, and achieving a lower average packet queuing time.

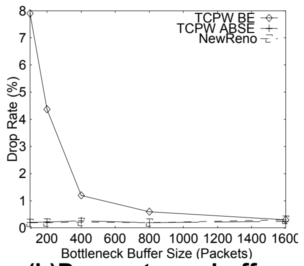




(a)NewReno **(b) TCPW ABSE**
Fig. 15 Illustration of TCPW ABSE and NewReno behavior with large buffer capacity (buffer = 160 packets)

4.3.2. Multiple Connections. Many connections competing for the same bottleneck link and buffer may cause utilization degradation [QZK01]. In this subsection, we evaluate the performance of TCPW BE, TCPW ABSE and NewReno in the face of multiple connections. We ran simulations with 10 TCP connections of TCPW ABSE, TCPW BE and NewReno respectively. The connections share a 45Mbps bottleneck link with a RTT of 70ms. The bottleneck buffer size varies from 100 packets (one forth of the pipe size) to 1600 packets (4 times of pipe size). The average throughput results are shown in fig. 16(a) and the drop rate (due to buffer overflow) results are shown in fig. 16(b).





(a)Throughput vs. buffer size **(b)Drop rate vs. buffer size**

Fig. 16. Throughput and drop rate vs. buffer size (pipesize = 400packets)

Fig. 16(a) shows that among the three protocols, TCPW ABSE achieves best utilization, while TCPW BE registers the worst performance with small buffer capacity. The reason behind the under utilization of NewReno with very small buffer is similar to the single connection analysis in Section 4.3.1. For TCPW BE, the overestimation of the bandwidth share (See Section 3.1) deteriorates the performance when the buffer is very small. After a buffer overflow, due to the overestimation, the *cwin* of TCPW BE is not reduced enough to eliminate the congestion, and the sender keeps up with the aggressive sending, causing more severe congestion and multiple drops, resulting in timeouts and thus poor link utilization. Fig. 16(b) shows the drop rate due to buffer overflow. We observe that TCPW BE has higher drop rates than NewReno and TCPW ABSE in the small buffer case, due to its aggressive estimation in the presence of congestion. As the buffer size grows, the congestion can be absorbed by queuing the packets in the large buffer, and TCPW BE achieves better utilization.

TCPW ABSE provides robust performance with small buffer, in terms of both high link utilization and low drop rate as shown in Fig. 16(b). This is because TCPW ABSE reduces the *cwin* according to a more accurate rate estimate. Fig. 16(b) also reveals that drop rate of TCPW ABSE is no higher than that of NewReno.

4.4. Random Errors on Lossy Links

In this subsection, we present a summary of performance evaluation to compare the performance of TCPW BE, CRB, ABSE, to that of NewReno in the wired/wireless configuration shown in Fig. 17. The wired portion has a capacity of 45 Mbps and one-way propagation time of 35ms (roughly the delay from West to East coast in the USA). The wireless portion is a very short 11-Mbps link with a negligible propagation time (e.g. WaveLAN link). The wireless link is assumed to connect a base station to a destination mobile terminal. The main lesson learned from this experiment is that ABSE, while achieving friendliness towards NewReno, does not suffer efficiency deterioration the same way CRB does.

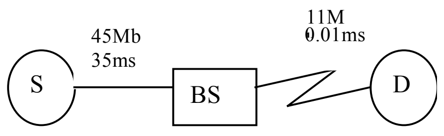


Fig. 17. Wired/Wireless Simulation Topology

The throughput of ABSE, BE, CRB and NewReno are compared under packet loss rate varying from 0 to 5%. The results in fig. 18 show that the throughput in ABSE, BE and CRB, are higher than that in NewReno. The largest improvement is obtained around 0.1% to 1% loss rate, where ABSE and BE throughput gain over NewReno is about 500%, while CRB gains only 300% over NewReno.

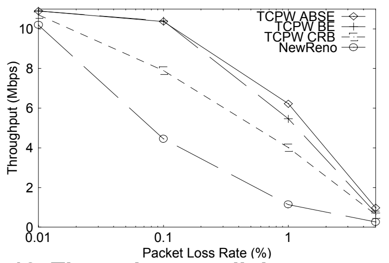


Fig. 18. Throughput vs. link error rate (%)

We have also assessed relation of the throughput gains to the E2E propagation time, and to the bottleneck link transmission speed. Results show that, when compared to NewReno, ABSE is able to maintain relatively robust performance with the increase of RTT . Further, ABSE is more effective than NewReno in utilizing bottleneck bandwidth, especially when the bandwidth is higher. For the details of the simulation results, please refer to [WVSG02].

5. Fairness and Friendliness evaluation

Fairness relates to the relative performance of a set of connections of the same TCP variant. Friendliness relates to how sets of connections running different TCP flavors affect the performance of each other. The simulation topology consists of a single bottleneck link with a capacity of 10 Mbps, and one-way propagation delay of 35ms. The buffer size at the bottleneck router is equal to the pipe size. The link is loss free except where otherwise stated.

5.1. Fairness

A set of simulations with 10 simultaneous flows was run to investigate fairness of ABSE. Throughput results are shown in Fig. 19. The Jain's fairness index [Jain91] of ABSE reached 0.9919, and that of NewReno is 0.9904.

Therefore, fairness of ABSE is comparable to that of NewReno.

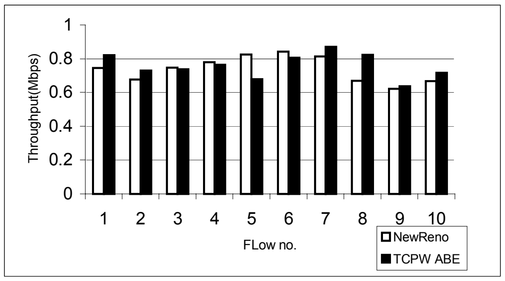
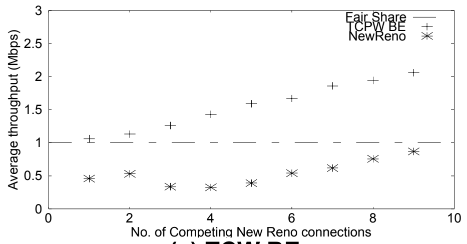


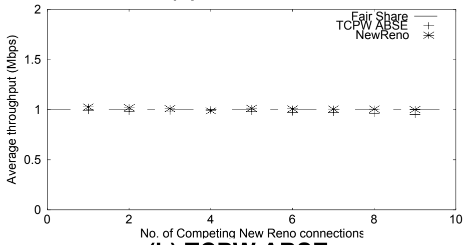
Fig. 19. Fairness comparison (10 connections of same TCP)

5.2. Friendliness

We evaluate below the friendliness of TCPW BE and ABSE towards NewReno. We ran simulations with total 10 TCP connections of various schemes sharing a 10Mbps bottleneck. In fig. 20(a) and (b), the horizontal axis represents the number of competing NewReno connections, the remaining connections being BE (in Fig. 20(a)) or ABSE (in Figure 20(b)). The vertical axis represents the average throughput of TCP NewReno and TCPW respectively. The results in Fig. 20(a) show that TCPW BE achieves higher throughput than its fair share, thus it is unfriendly to NewReno. This is because BE over-estimates the connection fair share [WVSNG02].



(a) TCW BE



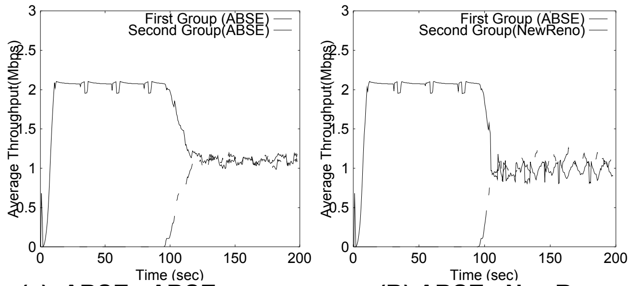
(b) TCPW ABSE

Fig. 20. TCPW BE and ABSE friendliness towards NewReno

In Fig. 20(b), we observe that the throughput achieved by both NewReno and ABSE are very close to the fair share, showing that ABSE achieves friendliness towards NewReno.

5.3. Transient Dynamics

To examine the transient behavior of new TCP connections against established connections, we simulate 10 flows sharing a 10Mbps bottleneck. In our experiments, 5 sources are established at the beginning; another 5 new sources become active after 100s. Fig. 21(a) shows the throughput averaged over a set of connections, when the new connections are ABSE, which join established ABSE connections. In Fig. 21(b), the new connections are TCP NewReno. The results show that the new connections, both ABSE and NewReno, readily acquire their fair share against the established connections.



(a) ABSE +ABSE (B) ABSE +NewReno

Fig. 21. Transient dynamics: instantaneous throughput of new and established connections

6. Conclusion

In this paper, after reviewing the estimation methods in TCPW, and elaborating on their strengths and limitations, we showed that TCPW ABSE estimation accuracy and throughput are robust in the face of ACK and data packet compression and varying buffer sizes., ABSE also maintains friendliness to TCP NewReno connections.

ABSE adapts the size of the observation interval, over which it obtains a Eligible Rate Estimate (ERE) sample, to the network congestion level. ERE samples are further smoothed through a low pass filter, whose agility is also adapted to measured network instability. By exploiting these two levels of adaptability, the proposed ABSE method produces a more accurate estimation of the connection bandwidth share, which turns out to be a critical factor to obtain efficiency as well as friendliness towards TCP NewReno connections. We have tested the proposed ABSE scheme in single and multiple connections scenarios, with and without random errors, with and without two-way traffic as well as cross traffic, and with various buffer capacities.

In the near future we intend to address the performance of ABSE in short lived sessions. We will study the coexistence of ABSE with rate-adaptive real time UDP flows managed by TCP-like, equation driven rate control. We will address the efficient implementation of ABSE in popular systems such as Linux and Free BSD. Following that, measurements in laboratory settings, on the Internet, and combinations of both, will be carried out to assess ABSE behavior in actual network settings. We are also developing a control model to study the stability of TCPW.

References

[AP99] M. Allman and Vern Paxson, "On Estimating End-to-End Network Path Properties", *ACM/Sigcomm 1999.*

[BA00] C. Barakat and E. Altman, "Analysis of TCP in networks with small buffering capacity and large bandwidth-delay product," *Proceedings of Sigmetrics , Santa Clara, California, June 18-21, 2000*

[BB00] C. Boutremans, J.-Y. Le Boudec "A Note on the Fairness of TCP Vegas," *Proceedings of International Zurich Seminar on Broadband Communications, Zurich, Switzerland, February 2000, pp. 163-170*

[BK98] H. Balakrishnan and R. H. Katz, "Explicit Loss Notification and Wireless Web Performance," In *Proceedings of IEEE GLOBECOM'98 Internet Mini-Conference, Sydney, Australia, November 1998.*

[Bona99] T. Bonald, "Comparison of TCP Reno and TCP Vegas: Efficiency and Fairness," *In Proceedings of PERFORMANCE'99, Istanbul, Turkey, October 1999.*

[BP95] L.S. Brakmo and L.L. Perterson. "TCP Vegas: End-to-End Congestion Avoidance on a Global Internet." *IEEE Journal on Selected Areas in Communication, Vol. 13, Nov. 8, October 1995.*

[BPSK97] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," *IEEE/ACM Transactions on Networking, December 1997.*

[BSAK95] H. Balakrishnan, S. Seshan, E. Amir, and R. H. Katz, "Improving TCP/IP Performance Over Wireless Networks," *MOBICOM'95, Berkeley, CA, USA, November 1995.*

[CGLMS00] C. Casetti, M. Gerla, S. Lee, S. Mascolo, and M. Sanadidi, "TCP with Faster Recovery," *MILCOM 2000, Los Angeles, CA, October 2000.*

[CGMSW01] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links," *In Proceedings of Mobicom 2001, Rome, Italy, Jul. 2001.*

[CK74] V. C. Cerf and R. E. Kahn, "A Protocol for packet Network Interconnections," *IEEE Transactionsion Communications, vol. COM-22, no. 5, pp. 637-648, May 1974.*

[Clar88] D. Clark, "The design philosophy of the DARPA Internet protocols," *In Proceedings of Sigcomm'88 in ACM Computer Communication Review, vol. 18, no. 4, pp. 106 - 114, 1988.*

[DRM01] C. Dovrolis, P. Ramanathan, D. Moore, "what do packet dispersion techniques measure?", *in proc. of IEEE infocom 2001.*

[FF96] K. Fall and S. Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP," *Computer Communication Review, V. 26 N. 3, July 1996, pp. 5-21.*

[FJ93] S. Floyd and V. Jacobson, "Random Early Detection gateways for congestion Avoidance," *IEEE/ACM transactions on Networking, August 1993*

[Floy94] S. Floyd, "TCP and Explicit Congestion Notification," *ACM Computer Communication Review, V. 24 N. 5, pp. 10-23, Oct. 1994.*

[GMPG00] T. Goff, J. Moronski, D. S. Phatak, and V. Gupta, "Freeze-TCP: a True End-to-end TCP Enhancement Mechanism for Mobile Environments," *In Proceedings of IEEE INFOCOM'2000, Tel Aviv, Israel, March 2000.*

[HMM98] G. Hasegawa, M.Murata and H. Miyahara, "Fairness and Stability of Congestion Control Mechanism of TCP," *in Proceedings of 11th ITC Special Seminar, October, 1998*

[Hoe96] J. C. Hoe, " Improving the Start-up of A Congestion Control Scheme for TCP", *Proc. ACM SIGCOMM '96, pp. 270-280*

[Jaco88] V. Jacobson, "Congestion Avoidance and Control," ACM Computer Communications Review, 18(4) : 314 - 329, August 1988.

[Jaco90] V. Jacobson, "Berkeley TCP evolution from 4.3-Tahoe to 4.3 Reno," *Proceedings of the 18th Internet Engineering Task Force, University of British Colombia, Vancouver, BC, Sept. 1990.*

[Jain91] R. Jain, "The art of computer systems performance analysis," *John Wiley and sons, QA76.9.E94J32, 1991*

[Kesh91] S. Keshav "A Control-Theoretic Approach to Flow Control," *Proceeding of ACM SIGCOMM 1991*

[KN01] Minkyong Kim and Brian Noble, "Mobile Network Estimation," *In Proceedings of Mobicom 2001, Rome, Italy, Jul. 2001.*

[LB00] K. Lai and M. Baker, "Measuring Link Bandiwdths Using a Deterministic Model of Packet Delay", *Sigcomm 2000.*

[ns2] ns-2 network simulator (ver.2.) LBL, *URL:* http://www.mash.cs.berkley.edu/ns.

[QZK01] Lili Qiu, Yin Zhang, and Srinivasan Keshav. "Understanding the Performance of Many TCP Flows." *Computer Networks (formerly called Computer Networks and ISDN Systems), Vol. 37, pp. 227 - 306, 2001.*

[RFC2582] S. Floyd, and T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm," *RFC 2582, Experimental, April 1999.*

[RFC2883] S. Floyd, J. Mahdavi, M. Mathis and M. Podolsky, "An Extension to the Selective Acknowledgement (SACK) Option for TCP," *RFC 2883, July 2000.*

[TCPW] TCP Westwood modules for ns-2. *URL: http://www.cs.ucla.edu/~mvalla/tcpw*

[WVSG02] Ren Wang, Massimo Valla, M.Y. Sanadidi, and Mario Gerla, "Adaptive Bandwidth share Estimation in TCP Westwood", *UCLA Technical Report, also submitted to Globecom 2002*

[WVSNG02] Ren Wang, Massimo Valla, M.Y. Sanadidi, Bryan Ng and Mario Gerla, "Efficiency/Friendliness Tradeoffs in TCP Westwood", *IEEE Symposium on Computers and Communications, Taormina, Italy, July 2002.*

[ZSC91] Lixia Zhang, Scott Shenker and David Clark, "Observations on the Dynamics of Congestion Control Algorithms: The Effects of the Two-Way Traffic," *In Proc. SIGCOMM'91, pg 133-147.*