

Enhancing Fairness and Throughput of TCP in Heterogeneous Wireless Access Networks

Fei Peng^{1,2} and Victor C. M. Leung^{1,2}

Most of the recent research on TCP over heterogeneous wireless networks has concentrated on differentiating between packet drops caused by congestion and link errors, to avoid significant throughput degradations due to the TCP sending window being frequently shut down, in response to packet losses caused not by congestion but by transmission errors over wireless links. However, TCP also exhibits inherent unfairness toward connections with long round-trip times or traversing multiple congested routers. This problem is aggravated by the difference of bit-error rates between wired and wireless links in heterogeneous wireless networks. In this paper, we apply the TCP Bandwidth Allocation (TBA) algorithm, which we have proposed previously, to improve TCP fairness over heterogeneous wireless networks with combined wireless and wireline links. To inform the sender when congestion occurs, we propose to apply Wireless Explicit Congestion Notification (WECN). By controlling the TCP window behavior with TBA and WECN, congestion control and error-loss recovery are effectively separated. Further enhancement is also incorporated to smooth traffic bursts. Simulation results show that not only can the combined TBA and WECN mechanism improve TCP fairness, but it can maintain good throughput performance in the presence of wireless losses as well. A salient feature of TBA is that its main functions are implemented in the access node, thus simplifying the sender-side implementation.

KEY WORDS: Transmission Control Protocol; wireless networks; Explicit Congestion Notification; fairness

1. INTRODUCTION

Wireless communication technology has been making significant progress recently and will be playing an increasingly important role in access networks, resulting in the wide deployment of heterogeneous wireless and wired environment in the next-generation all-IP networks for Internet access. It is very important that access nodes provide end-to-end quality of service (QoS), including fairness in bandwidth allocation, for both wired and wireless users.

Since many Internet applications run on top of the transmission control protocol (TCP), a number of techniques have been developed to address the issue

of TCP performance over heterogeneous wireless/wireline networks. However, most of them focus on enhancing TCP by distinguishing between transmission and congestion losses since inability to do so is widely known as the main reason of TCP performance degradation in such networks. The solutions are roughly classified into link layer protocols, split-connection protocols and end-to-end protocols [1]. The link layer approaches aim to repair transmission losses locally using Forward Error Correction (FEC) or Automatic Repeat request (ARQ). The cost of such repair may be a loss of available bandwidth or the introduction of new path dynamics (extra delay, jitter, packet reordering, etc). Therefore, corruption losses still remain a major concern for transport layer. Split-connection protocols isolate mobility and wireless related problems by splitting a TCP connection between the mobile host and fixed host into wired connection and wireless connection. While

¹ Department of Electrical and Computer Engineering, University of British Columbia, 2332 Main Mall, Vancouver, BC, Canada V6T 1Z4,

² E-mails: feip@ece.ubc.ca; vleung@ieee.org

effective, these protocols violate the end-to-end semantics. End-to-end protocols adjust TCP operations to improve performance in the presence of wireless losses in a way that maintains end-to-end semantics. Several end-to-end solutions have recently proposed to estimate the cause of packet losses such as TCP VenO [2], TCP-Jersey [3], Jitter-based TCP [4], TCP Westwood [5].

Besides the problem of misinterpreting transmission losses as indications of network congestion, it is well known that TCP suffers from inherent unfairness toward connections with long round trip times (RTTs) and connections that have to traverse multiple congested routers. These problems would exist in both wired and wireless environments. So far, TCP fairness over heterogeneous wireless networks has not received much attention in the research community. Most proposals such as TCP Vegas [6], Fast TCP [7] and a Fast TCP related enhancement [8] aimed at improving TCP fairness have been evaluated in wired networks where transmission losses are not significant; but application of these techniques to wireless access networks requires further investigations. As well, TCP VenO [2] employs a mechanism similar to TCP Vegas but for the purpose of differentiating the causes of packet loss. Since both TCP Vegas and Fast TCP can improve TCP fairness in wired networks, it is reasonable to consider them for the wireless environment. However, as they both rely on propagation delay measurements, the issues of delay variations due to transmission losses and handoffs in wireless networks should be addressed when these techniques are considered. The enhancement for Fast TCP in [8] can also be applied to wireless networks to improve TCP fairness performance; however, the estimation of marking statistics and appropriate choice of universal parameter for the proposed algorithm are major concerns in wired networks, which are exacerbated by the characteristics of wireless networks. All of these mechanisms [6–8] require global modifications of TCP in end-hosts, such that compatibility with older TCP versions would be a concern. Techniques that implement TCP enhancements in the access node with minimal modifications to TCP in end-hosts would be more attractive from a practical perspective.

To address unfairness suffered by connections with long RTTs or traversing multiple congested routers, we have proposed the TCP Bandwidth Allocation (TBA) algorithm and illustrated its effectiveness in wired networks [9]. In this paper, we apply TBA to heterogeneous wireless access networks in the

presence of high bit-error rates. In contrast to other end-to-end proposals [2–5], the TBA algorithm is implemented in the access node, which does not require modifications at the end-systems. It is therefore easy to implement. Particularly, our scheme uses the receiver's advertised window in each ACK to change the TCP window behavior at the source, in a manner similar to F-Snoop [10]. However, F-Snoop only guarantees fairness among connections through a common base station, which are assumed to have the same RTTs. As it assumes that total bandwidth occupancy would not exceed the available bandwidth, F-Snoop may not be able to cope with large and dynamic fluctuations of bandwidth demands. In contrast, our TBA bandwidth allocation algorithm takes the dynamic oscillation of bandwidth occupancy into account. To differentiate wireless packet losses from congestion packet losses, we further propose to combine the TBA algorithm with Explicit Congestion Notification (ECN) [11] that has been modified to avoid window reduction caused by wireless errors in [12] and we employ it as wireless ECN (WECN), therefore, the combined approach referred as TBA/WECN, is proposed. Since ECN is a "Proposed Standard" recommended by the Internet Engineering Task Force (IETF), it is expected to be widely used in the future for TCP congestion control. Our proposed mechanism is therefore very practical. Further enhancement is also incorporated to smooth traffic bursts over wireless links that is implemented at the access node to simplify the implementation at the source node. Simulation results illustrate the effectiveness of the combined TBA/WECN approach.

The organization of the paper is as follows. Section 2 presents the TBA algorithm. Implementation of the TBA algorithm in the access node and its combination with WECN at the end nodes are described in Section 3. Network configurations of the simulations and the metrics for performance evaluations are given in Sections 4 and 5, respectively. The simulation results are analyzed in Section 6. We conclude the paper in Section 7.

2. TCP BANDWIDTH ALLOCATION ALGORITHM

Consider a set of J links with bandwidth c_j for link $j \in J$ in the network where congestion could occur. Define route r as a path with links in J and let R be the set of possible routes. Set $A_{jr} = 1$ if $j \in r$ so that link j lies on route r , and set $A_{jr} = 0$ otherwise.

This defines a 0–1 matrix $\mathbf{A} = (A_{jr}, j \in J, r \in R)$. We associate a route r with a TCP connection or user, and denote x_r as the bandwidth (rate) allocated to connection r . Define vectors of link rates $\mathbf{c} = (c_j, j \in J)$ and connection rates $\mathbf{x} = (x_r, r \in R)$, we have:

$$\mathbf{Ax} \leq \mathbf{c}, \mathbf{x} \geq \mathbf{0} \quad (1)$$

The constraint (1) says that at each link j , the aggregate source rate $\sum_{r \in J} A_{jr} x_r$ does not exceed the capacity c_j . To make the proportional share of bandwidth for each flow, the resource allocation problem can be formulated as the following nonlinear program:

$$\max \sum_{r \in R} p_r \log x_r. \quad (2)$$

where $p_r > 0$ represents the amount that user r chooses to pay per unit time to receive in return a bandwidth (rate) x_r proportional to p_r . Hence, p_r is under user control. Since the objective function (2) is strictly concave and the feasible region given by (1) are compact, the optimal solution exists and is unique. Suppose that the network knows the value of p_r , the optimal solution x_r^* to (2) for the proportional rate allocation is:

$$x_r^* = p_r \frac{c_j}{\sum_{j \in R} \sum_{r \in R} A_{jr} p_r} \quad (3)$$

Thus the following continuous-time version of a rate control algorithm can be used to solve the proportional rate allocation problem in (2) and achieve the optimal rate in (3):

$$\frac{d}{dt} x_r(t) = k_r p_r - k_r p_r \frac{x_r(t)}{x_r^*} \quad (4)$$

It has been proven [13] using the Lyapunov function that (4) converges to the fair point (3).

Associate route r with its Round Trip Time (RTT) T_r and its TCP congestion window size w_r that changes with time, the relationship of w_r with rate x_r and T_r is given by the following equation:

$$x_r \times T_r = w_r, \quad x_r \geq 0, T_r \geq 0 \quad (5)$$

Identity (5) arises if a full window of packets can be sent in each RTT for any TCP version. Later in Section 3, we will define the window size in each RTT in detail. The congestion window is also bounded by the Receiver's advertised window (RCVW) field in TCP option, that is, w_r in (5) should be no more than the window size in RCVW field (denoted as w_{\max}), otherwise, it is equal to w_{\max} . Due to this characteristic, we propose to use the RCVW field to carry w_r for our TBA

algorithm. Generally, current TCP versions reserved enough bits in RCVW field to provide large w_{\max} so that the congestion window would not be constrained by small w_{\max} that would otherwise limit the bandwidth utilization. Here we didn't consider networks with high bandwidth-delay products where even larger w_{\max} would be required for full bandwidth utilization and there are proposals to extend more bits in RCVW field that is out scope of our paper. Allowed by large enough w_{\max} in general networks, the equilibrium rate (3) related to w_r in (5) can be reached. In the following, we will introduce how to determine w_r based on (4) and (5).

Differentiating (5) on both sides, we get:

$$T_r dx_r + x_r dT_r = dw_r \quad (6)$$

We then combine (4) and (6) to obtain the window size dynamics in below, in which TBA algorithm is represented:

$$\frac{dw_r(t)}{dt} = \left(k_r p_r - k_r p_r \frac{x_r(t)}{x_r^*} \right) T_r + x_r \frac{dT_r(t)}{dt} \quad (7)$$

To explore the stability of system (7) in the presence of time lags, we linearize (4) and obtain:

$$x_r(t+1) = x_r(t) + k_r \left[p_r - \frac{p_r}{x_r^*} x_r(t - T_r) \right] \quad (8)$$

Employing the method in [14], the corresponding characteristic equation to (8) is thus

$$\lambda^{T_r+1} - \lambda^{T_r} + k_r \cdot \frac{p_r}{x_r^*} = 0 \quad (9)$$

The fixed point is locally stable if (9) has all roots with moduli less than unity. Let the root be $e^{i\theta}$ where $0 \leq \theta \leq \pi$ (this is sufficient since $-\pi \leq \theta \leq 0$ just gives the conjugate root), replace λ with $e^{i\theta}$ in (9), we deduce that θ needs to satisfy the equations

$$k_r \cdot \frac{p_r}{x_r^*} = 2 \sin\left(\frac{\theta}{2}\right) \text{ and } (2T_r+1)\theta = (4n+1)\pi \quad (10)$$

Where n is an integer. We choose $n = 0$ in (10) to look for the smallest positive k_r such that λ is of unit modulus. Hence, substituting for θ in (10), we see that there are no solutions for θ if

$$k_r < 2 \frac{x_r^*}{p_r} \sin\left(\frac{\pi}{2(2T_r+1)}\right) \quad (11)$$

Under the above condition, the system (8) is then proven to be locally stable. A larger value of k_r in (11) would generally result in quicker convergence to stability.

Equation (4) with condition (11) shows that the bandwidth and sum of prices have a greater influence on the rate of convergence than the propagation delay. The data rate will be stable if k_r satisfies (11). So, the total rate across the bottleneck node j will converge to c_j . The queuing delay would oscillate around a small value and the equilibrium point for end-to-end round trip time will converge to the propagation delay. Therefore, the convergence of window size could also be determined by (11). In the following section, we will describe how TBA algorithm is implemented by (7) in discrete time with time lags, where the parameter k_r is chosen according to (11) to ensure system stability.

3. IMPLEMENTATION OF TBA ALGORITHM

3.1. Support from TCP layer

For the purpose of effectively decoupling congestion control and loss recovery, the basic idea proposed in [12] use ECN and an adaptive algorithm instead of packet losses as indication of congestion. Employing the method in [12], we use only ECN as indication of congestion and refer it as wireless ECN (WECN). When combine TBA with WECN, denoted as TBA/WECN, the source node initiates congestion control by reducing its congestion window in response to ECN only, but not packet losses in order to avoid unnecessary throughput degradation. In this case ECN has a higher priority in window control than ACK option field provided by the TBA algorithm. Implementing TBA algorithm in access node is described in detail later. For comparison, we consider another configuration denoted as TBA/ECN. With TBA/ECN, normal ECN is applied in the end nodes, and the source node reduces its congestion window in response to both packet losses and ECN that have higher priority than the value set from the access node. In this way, the congestion control and loss recovery are not separated. The TCP retransmission mechanism does not have to be modified in the access node or TCP sender; therefore any TCP version can be supported by our combined TBA/WECN algorithm.

Here we briefly discuss stability of the TBA/ECN or TBA/WECN enhancement. The congestion window will follow TBA control implemented at the access node. However, when the end system reduces the window size according to the arrival of ECN or detection of packet losses, the equilibrium point established by the TBA algorithm is no longer valid,

thus necessitating the computation of a new equilibrium point. In TBA/WECN, although packet losses would not cause reductions of the congestion window, they would delay the time for the oscillating window size to return to the optimal point because of retransmissions. Since the system would always establish a new equilibrium point, the TBA algorithm could maintain stability even with packet losses and ECN.

3.2. Support in Access Node

We assume that ECN is implemented in network nodes for congestion feedback. Since the access node has the characteristics that data packets and their corresponding ACKs are forwarded by the same node in opposite directions, we implement the TBA algorithm at the access node as shown in Figure 1. For clarification, we separate the algorithm into three functional parts, i.e., *TBA Algorithm Calculation*, *ACK Identification* and *Writing ACK Option*. *ACK Identification* separates the ACK flow from normal data traffic, identifies the connection each ACK belongs to, and detects the ECN bit set by downstream network nodes. After the appropriate window size is determined by *TBA Algorithm Calculation*, the new window size is written into the receiver's advertised window (RCVW) field in each ACK by *Writing ACK Option*, which is then inserted in the return link. In the following we describe the calculation of window size by the TBA Algorithm in more detail.

The TBA algorithm described in (7) is in continuous time formation, to make it applicable to real implementation, we divide time into discrete window cycles. Corresponding to w_r , T_r and x_r in (5), the window size, RTT and average rate of user r in cycle i are denoted, respectively, as $w_r(i)$, $T_r(i)$, and $x_r(i)$, where $w_r(i)$ is obtained by detecting the number of packets in the window when the sequence number of the backward ACK is equal to that of the first packet sent in this window, $T_r(i)$ is the duration that $w_r(i)$ is recorded and $x_r(i)$ is calculated by the formula $w_r(i)/T_r(i)$. Suppose the change in window size during $T_r(i)$ cycles is $\Delta w_r(i) = w_r(i) - w_r(i-1)$, formula (7) is rewritten as,

$$\Delta w_r(i) = \left(k_r p_r \left(1 - \frac{x_r(i-1)}{x_r^*} \right) T_r(i-1) \right) / \left(T_r(i-1) + x_r(i-1)(T_r(i) - T_r(i-1)) \right) \quad (12)$$

Each ACK should cause the window size to be changed by $\Delta w_r(i)/w_r(i-1)$, since there are $w_r(i-1)$

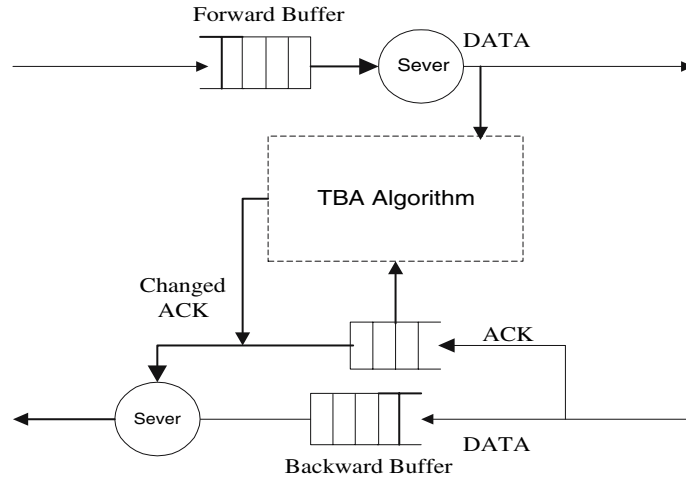


Fig. 1. TBA implementation in the access node.

ACKs from the previous cycle. To get the value of the changed window size from (12), that is, to calculate the changed window size in i -th cycle for connection i , the average rate $x_r(i-1)$ and the round-trip time $T_r(i-1)$ in $(i-1)$ th cycle are required. Each time a new window cycle begins, $x_r(i-1)$ and $T_r(i-1)$ are updated.

Due to cumulative characteristics of ACKs, a large number of data segments could be acknowledged by one new ACK after the loss recovery phase and a burst of data packets could be inserted into the network that results in bursts of traffic. In wireless network, recovery of transmission losses can contribute to more traffic bursts if the congestion window is allowed to be large for high bandwidth utilization. To smooth the traffic bursts we modify the TBA algorithm using a variable *THRESH*. The main purpose of *THRESH* at the access node is to keep track of the slow start threshold (*ssthresh*) at the TCP sender. The value of *ssthresh* MUST be set to no more than the value given by $\max(\text{FlightSize}/2, 2 \cdot \text{SMSS})$, when a TCP sender detects congestion by ECN or segment loss using the retransmission timer or three duplicate ACKs. The FlightSize is the amount of outstanding data in the network and can be estimated as $w_r(i)$ as above at the access node. Therefore, for TBA/ECN, *THRESH* is set to half of current window size (*CWND*) whether ECN or packet loss is indicated, as packet losses are considered to be caused by congestion; for TBA/WECN, *THRESH* is set to half *CWND* only when ECN is detected, as packet losses are considered to be caused by transmission errors. Having determined the value of *THRESH*, we will describe an algorithm to smooth burst of data in the next section. Since the access node is not responsible

for loss recovery, this method can be applied to any TCP version.

4. SIMULATION CONFIGURATION

The topology, the link bandwidths and propagation delays in the simulated network scenarios are shown in Figures 2 and 3. We use topology 1 shown in Figure 2 to examine the effects of many competing connections over a single bottleneck link. Consider several clients ($D1-D6$) connecting to a base station ($n1$) using wireless and wired links. The based station is connected to the backbone network through an access node ($n0$) where the TBA algorithm is implemented and the congestion occurs. Then, we use topology 2 shown in Figure 3 to examine the effects of a wireless link on a TCP connection with long RTT that must also traverse a number of congested hops shared by TCP connections over wired or wireless links with short RTTs. In this case, the congested hops are configured at the access nodes $n1-n4$, where TBA is implemented. According to (4) and (12), the sending window for each TCP source in topology 2 is determined by the sum value of window sizes calculated by the TBA algorithm in each access node along the path. We consider two-way traffic and each access node works in bi-direction. In each direction, 10 connections are running between each source and client in topologies 1 and 2 for simulation. In both topologies, the access node is associated with mixed wire/wireless links. We focus on the access node and simplify the backbone network as a single link. Congestions for data transfer are assumed to

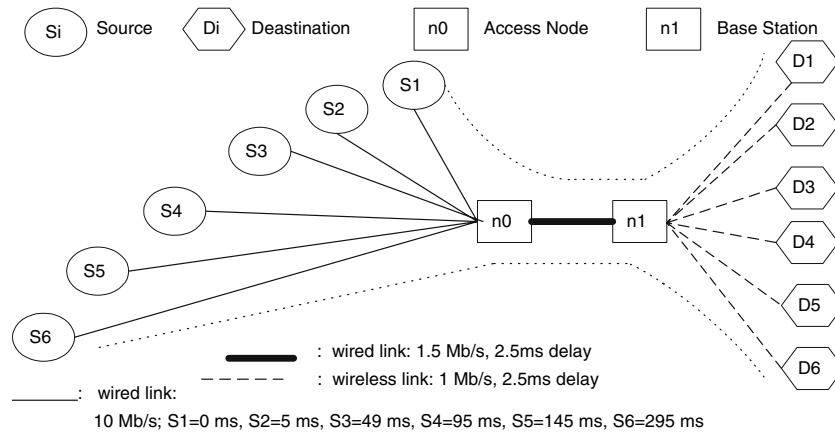


Fig. 2. Network topology 1 of simulation model (Config1).

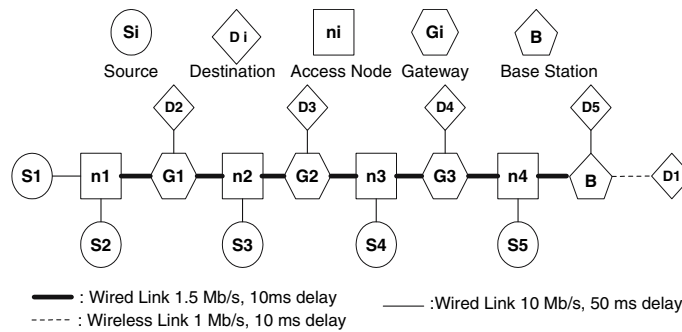


Fig. 3. Network topology 2 of simulation model.

occur only at the access nodes, which set the ECN bits in forwarded packets upon detection of congestion.

We employ Random Early Detection (RED) that is currently deployed over the Internet backbone [15] for congestion detection. The RED queue at each output link uses two thresholds: *minthresh* and *maxthresh*. The RED queue also maintains an average queue length Q_{avg} estimated by the moving average of the instantaneous queue length Q . When $minthresh \leq Q_{avg} \leq maxthresh$, incoming packets are randomly marked by setting ECN with a probability that increases with Q_{avg} . Incoming packets are randomly dropped with a probability that increases with Q_{avg} when $Q_{avg} > maxthresh$. The average queue length is largely dependent on a long-term averaging value of the instantaneous queue length, the instantaneous queue length, which is obtained by detecting buffer occupancy on every incoming packet provides the sender with more accurate buffer information. Experiment show that better performance is obtained by using the instantaneous queue length rather than using the average queue length in the RED algorithm when RED and

ECN are used in wireless networks. To improve robustness by avoiding the loss of ECN packets due to link errors or congestion, we set ECN with probability 1 when instant queue length $Q > minthresh$ instead of setting ECN with a lower probability, that is, at the congested node $n0$ in Figure 1 and G_i and B in Figure 2, we set ECN in every incoming packet when the buffer size exceeds *minthresh*, which is set to half the buffer capacity. The *maxthresh* is set as buffer capacity.

To smooth traffic bursts, we use the parameter *THRESH* defined in Section 3.2. the increment of window size by the TBA algorithm is limited to 4 packets per normal ACK if $CWND < 4 \times THRESH$; otherwise the window increment is limited to 1 packet per normal ACK. For a partial ACK, we increase window size by 3/4 of the value calculated by the TBA algorithm. The parameters of 4, 1 and 3/4 are evaluated and chosen from simulation to serve the purpose of illustrating the effectiveness of the proposed method to smooth traffic bursts. Optimization of these parameters according to different configuration environment is left for further research.

Since transmission losses have a significant performance impact on the transport layer, we assume the wireless links in the TCP connections have the same network conditions as the wired ones except that the wireless links have $BER > 0$ whereas no bit errors occur in the wired links. In the simulations, bit errors over each wireless link occur at random and independently with a probability given by the BER. Since the gain parameter k_r is more dependent on bandwidth and price than propagation delay from (11), though propagation delays vary widely between the different connections, each connection would have approximately the same k_r value. Larger k_r satisfying (11) could generally improve the speed of convergence. In this paper, we choose k_r as 50 for both topologies, which is large enough for rapid convergence.

In topology 1 (Figure 2), to test the fairness of wireless connections with different RTTS, Config1 employs wireless links between $n1$ and all receivers Di . To test the fairness between wireline and wireless connections, Config2 employs a wireless link between $n1$ and $D6$ only, and wired links between $n1$ and the other destination nodes. There are also two configurations in topology 2 (Figure 3). In Config3, wireless links are employed between all the receivers Di and gateways Gi or B . Config4 employs wireless links only between receiver $D1$ and B . Each simulation runs for 15 seconds. We work with time intervals that are assumed to be short enough to ignore changes in traffic characteristics but at the same time long enough compare with the RTT to allow TCP to adjust traffic levels. For this reason, short-lived connection is not our focus in the paper. Each TCP segment is 1000 bytes long. We first examine a range of queue sizes (buffer capacity) from 20 to 100 packets at a high BER of 10^{-5} . Then, we examine a range of BER values with a fixed buffer capacity.

Five TCP mechanisms are compared by simulations using the ns-2 simulator [16]. The benchmark mechanism, denoted as Reno, is standard TCP Reno. ECN employs the ECN mechanism on top of TCP Reno. WECN employs the same mechanism as ECN except that window control in response to packet losses is disabled. TBA/ECN and TBA/WECN, respectively, combine TBA with ECN and WECN. In Config2 of topology 1 and Config4 of topology 2, TBA/WECN and WECN are evaluated by applying the respective mechanisms only in the connection with the wireless link, while the corresponding wired connections employ TBA/ECN and ECN, respec-

tively. However, in Config1 of topology 1 and Config3 of topology 2, since all the connections incorporate wireless links, the same mechanism is applied to all the connections for comparisons.

5. PERFORMANCE METRICS

A number of metrics for quantifying fairness have been proposed, but no single metric has common acceptance. In this paper, assuming $p_r = 1$ in (2) in Section 2, each user has the right to $1/n$ -th of the capacity of the bottleneck link for n users proportionally sharing that link. Jain's metric is applicable in this context. Consider connection r receiving a fraction b_r of the bandwidth on a given link, in which b_r is defined as the number of source data bits (i.e., not counting retransmissions) successfully transferred over a link during some time interval (i.e., the actual goodput) divided by the product of link rate and time interval. The fairness of the allocation is defined as:

$$Fairness = \frac{(\sum_{i=1}^n b_i)^2}{n \times (\sum_{i=1}^n b_i^2)}$$

This metric ranges continuously in value from $1/n$ to 1, with 1 corresponding to equal allocation for all users, which is the best result in proportional fairness.

Utilization is another important metric since a high fairness is of little use if the link capacity is grossly underutilized. Utilization is defined as the sum of b_i of all connections over a link. The closer the utilization metric approaches 1, the higher the link capacity is utilized.

$$Utilization \equiv \frac{(segments_acked) * (segment_size)}{(link_rate) * time}$$

The third metric is the utilization deviation. This metric is used in the evaluation of topology 2. The utilization deviation is defined as,

$$Utilization_Deviation \equiv \frac{\sum_{i=1}^{Connections} |i - 0.5|}{Connections}$$

where *Connections* represents the number of connections in topology 2 connected to the same gateway. From Figure 3, we see that the ideal bandwidth shared by each sender is half the bandwidth of the bottleneck link. If both the utilization and the fairness are good, the *Utilization_Deviation* is very small; otherwise, it can be very large.

6. SIMULATION RESULTS AND ANALYSIS

Figures 4 and 5 compare the (a) fairness, (b) utilization and (c) congestion drop rate (percentage) of different schemes over a range of queue size for Config1 and Config2, respectively, in topology 1 with $BER = 10^{-5}$ over the wireless links. These figures show that considering both fairness and utilization, TBA/WECN outperforms the other schemes over a wide range of queue sizes. Figure 4 shows that TBA/WECN and TBA/ECN have better fairness than the other schemes, but the utilization of TBA/ECN is much lower than TBA/WECN since TBA/ECN cannot avoid throughput degradation caused by link errors. Likewise TBA/WECN and WECN have better utilization than the other schemes, but TBA/WECN has much better fairness than WECN. While WECN avoids unnecessary window reductions due to wireless losses, it suffers from poor fairness typical of TCP due to its inherent bias toward connections with long RTT or traversing multiple congested routers. The TBA algorithm in combination with either WECN or ECN can alleviate this bias and give significantly better fairness performance.

Applying TBA/WECN to the single wireless connection in Config2 yields a more consistent improvement in fairness over TBA/ECN, as shown in Figure 5(a), compared to the case of multiple wireless connections in Config1 shown in Figure 4(a). This shows that TBA/WECN is more helpful to a wireless connection competing for bandwidth with wired connections than several wireless connections sharing the same congested link. Figure 5(b) shows that the utilization of TBA/ECN and ECN is much closer to that of TBA/WECN and WECN since the wired connections can occupy the bandwidth lost by the shared wireless connection. Figure 5(c) shows that the congestion drop rate of TBA/WECN is much closer to other schemes as compared to Figure 4(c) since wireless connection sharing with wired connections is less aggressive than multiple wireless connections sharing a same bottleneck link.

Comparing Figures 6 with 7(a) and (b) for topology 2, we observe similar results as refer to the comparison of topology 1 in Figures 4 and 5(a) and (b). TBA/WECN in the single wireless connection case is more robust to obtain better fairness than other schemes compared to TBA/WECN in the multiple wireless connection case. The *utilization deviation* of TBA/WECN is also much closer to TBA/ECN in single wireless connection case than multiple wireless connection case. Additionally, compare topology 1

with topology 2, Figures 5 with 7(a) and (b) show that TBA/WECN yield significant better fairness and utilization where the wireless connection traverses multiple nodes and shares links with wired connections in topology 2 than wireless connection traverse one node situation in topology 1. This illustrates that TBA/WECN is quite robust in heterogeneous wireless networks where connections suffer from extremely unfairness due to high BER, long transmission delay and traversing multiple nodes. Overall, Figures 4–7 show that the performance of TBA/WECN is much better than the other schemes.

From Figures 4–7, it is apparent that to realize the performance advantages of TBA/WECN, a sufficiently large buffer size (> 50) should be used. When buffer size is smaller than 50, it is difficult for TBA/WECN to effectively utilize bandwidth and TBA/WECN yield more congestion drops. However, a buffer size that is too large (> 90) is not helpful as the congestion windows can get very large resulting in an increased congestion losses shown in Figure 7(c) and decreased utilization shown in Figure 7(b), and this even influences fairness shown in Figure 7(a).

The congestion drop rate is calculated as number of dropped packets due to buffer overflow divided by total arrival at the congested node. The multiple wireless connection case in Figures 4 and 6(c) show that TBA/WECN suffers more congestion losses than other schemes. However, single wireless connection case in Figures 5 and 7(c) show that the congestion drop rate of TBA/WECN is not as higher than other schemes as in Figures 4 and 6(c) since multiple n wireless connections sharing a bottleneck link got n times bit error rate of which single wireless connection sharing with wired connections got. Due to the higher bit error rate, wireless losses would become more and cause severer burst of traffic that would introduce heavier congestion losses. However, since our algorithm smooth the traffic burst, the congestion packet losses is not significant even in Figures 4 and 6(c) and the utilization of TBA/WECN doesn't suffer much. The throughput of TBA/WECN could still comparable to WECN as shown in Figure 4(b) or even better than WECN and other schemes as shown in Figure 6(b).

We further evaluate performance of different schemes over a range of wireless link BER from 10^{-5} to 10^{-9} . Figure 8(a–c) shows the fairness, utilization and congestion drop rate, respectively, for Config3 in topology 2 with buffer size = 50. Results for the other configurations and topologies are very similar. At high BER, TBA/WECN has comparable fairness

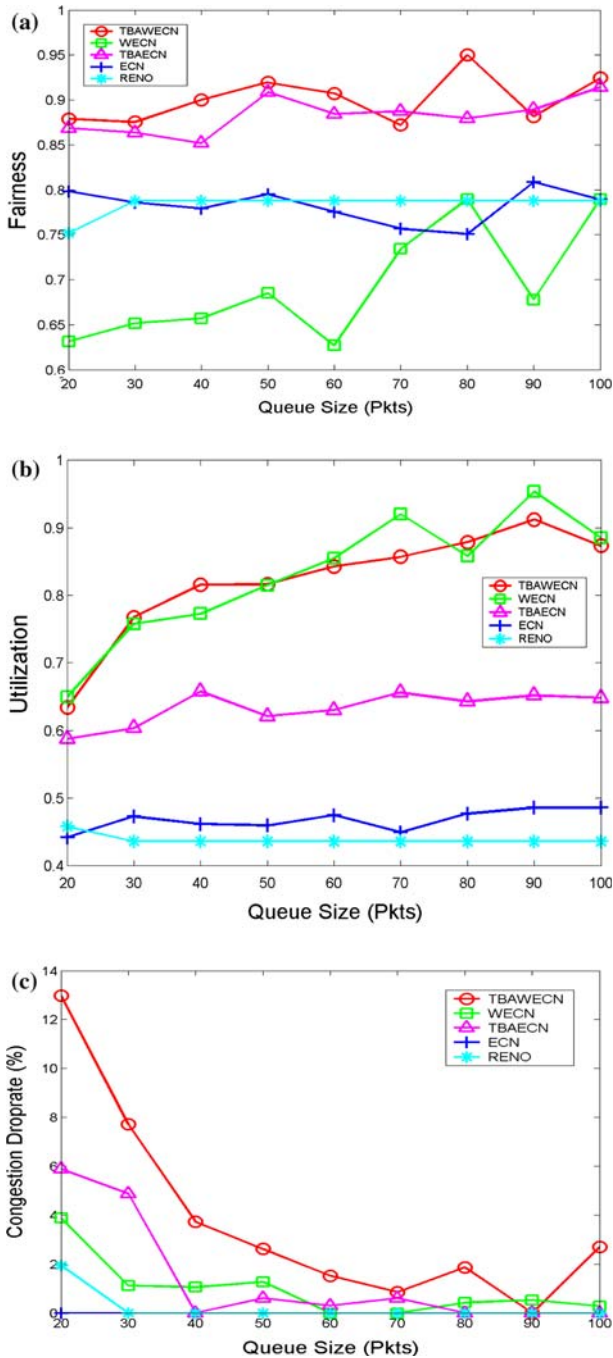


Fig. 4. (a) Fairness vs. queue size, topology 1, Config.1; (b) Utilization vs. queue size, topology 1, Config.1; (c) Congestion Drop Rate vs. queue size, topology 1, Config.1.

as TBA/ECN but obtains much better utilization than TBA/ECN and other schemes. As we described before, TBA/WE-CN illustrate that the bandwidth is better utilized than other schemes, though it suffers from a few more packet losses. When BER of the

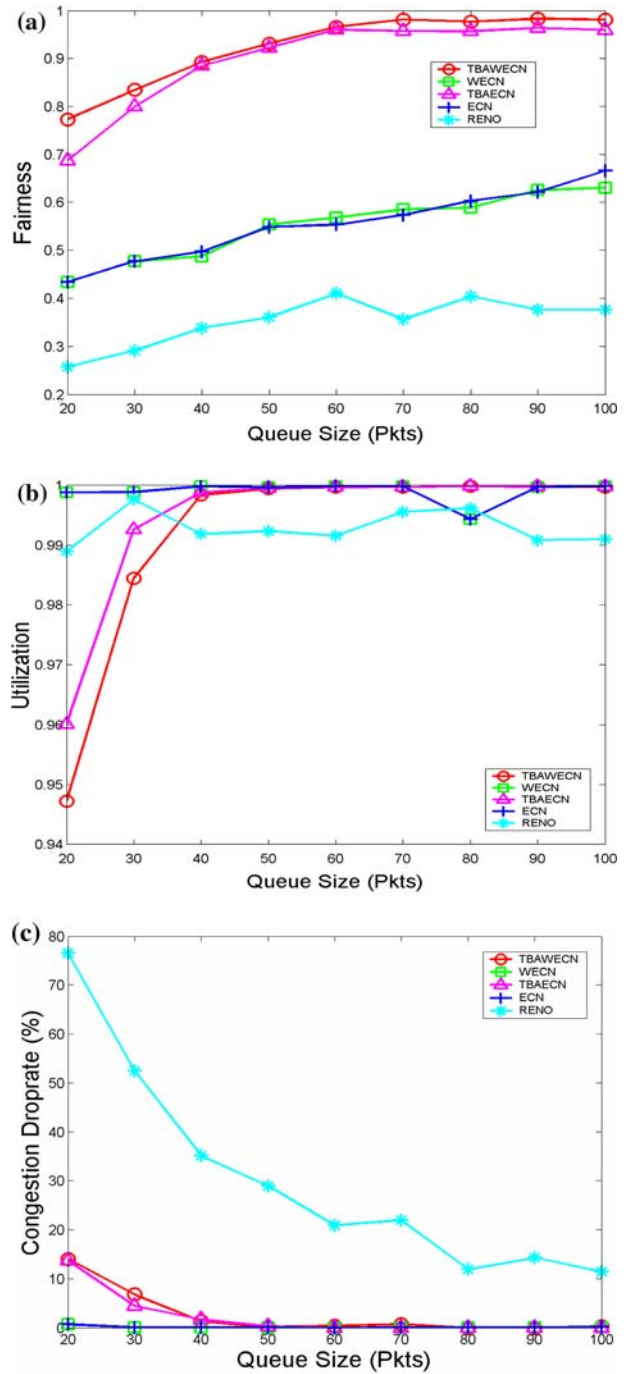


Fig. 5. (a) Fairness vs. queue size, topology 1, Config.2; (b) Utilization vs. queue size, topology 1, Config.2; (c) Congestion Drop Rate vs. queue size, topology 1, Config.2.

wireless link falls below 10^{-7} , TCP Reno and ECN encounter congestion losses since lower BER reduces the frequency of shrinking TCP window, however, this does little help for their utilization since the effect

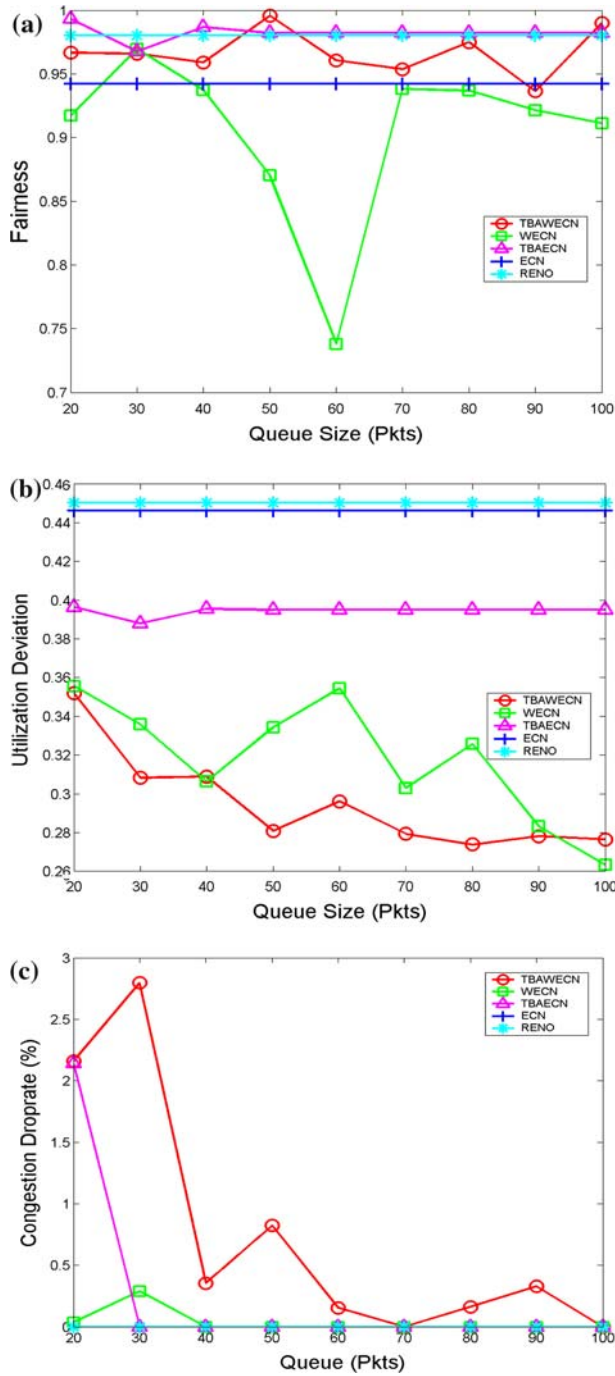


Fig. 6. (a) Fairness vs. queue size, topology 2, Config.3; (b) Utilization_deviation vs. queue size, topology 2, Config.3; (c) Congestion Drop Rate vs. queue size, topology 2, Config.3.

of BER still exist. At the same time, TBA/WECN and TBA/ECN achieve the same fairness and utilization that is much better than other schemes. Therefore, with minimal wireless losses, TBA/ECN

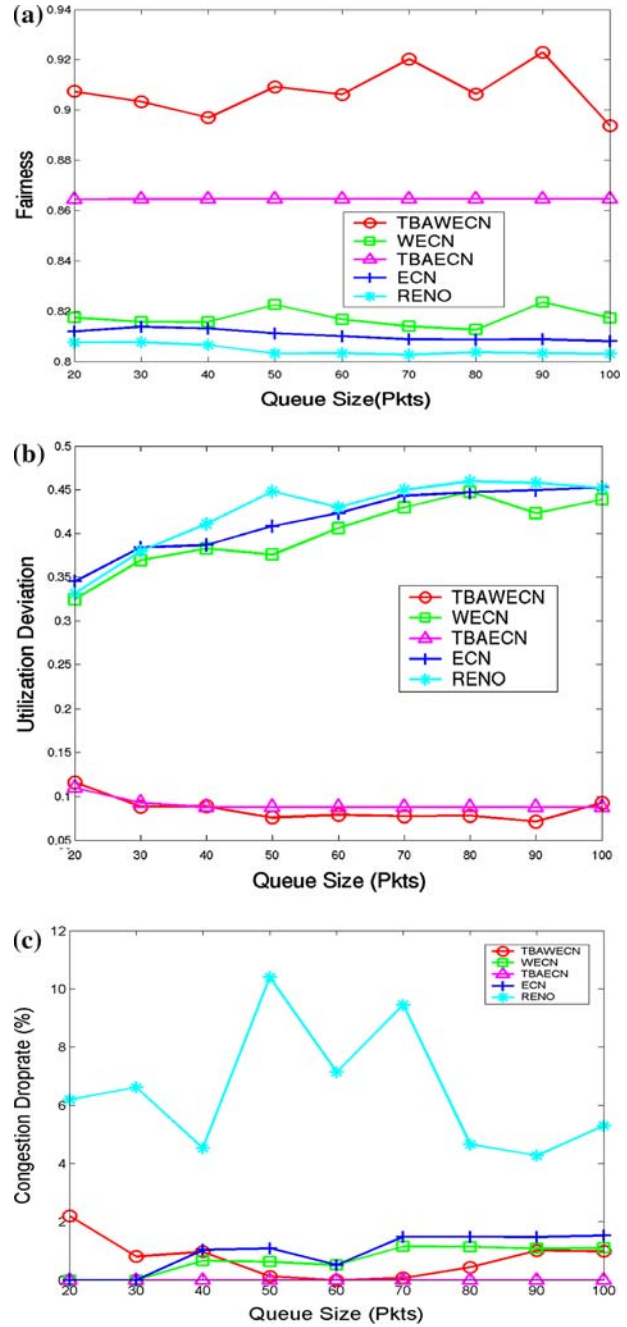


Fig. 7. (a) Fairness vs. queue size, topology 2, config. 4; (b) Utilization_deviation vs. queue size, topology 2, config. 4; (c) Congestion Drop Rate vs. queue size, topology 2, Config.4.

is just as effective as TBA/WECN, as both benefit from the fairness enhancement of TBA. Additionally, as we know, a lower BER could allow bigger window but more congestion losses. However, with the help of ECN, congestion losses are largely reduced. The drop rate in Figure 8(c) shows that all

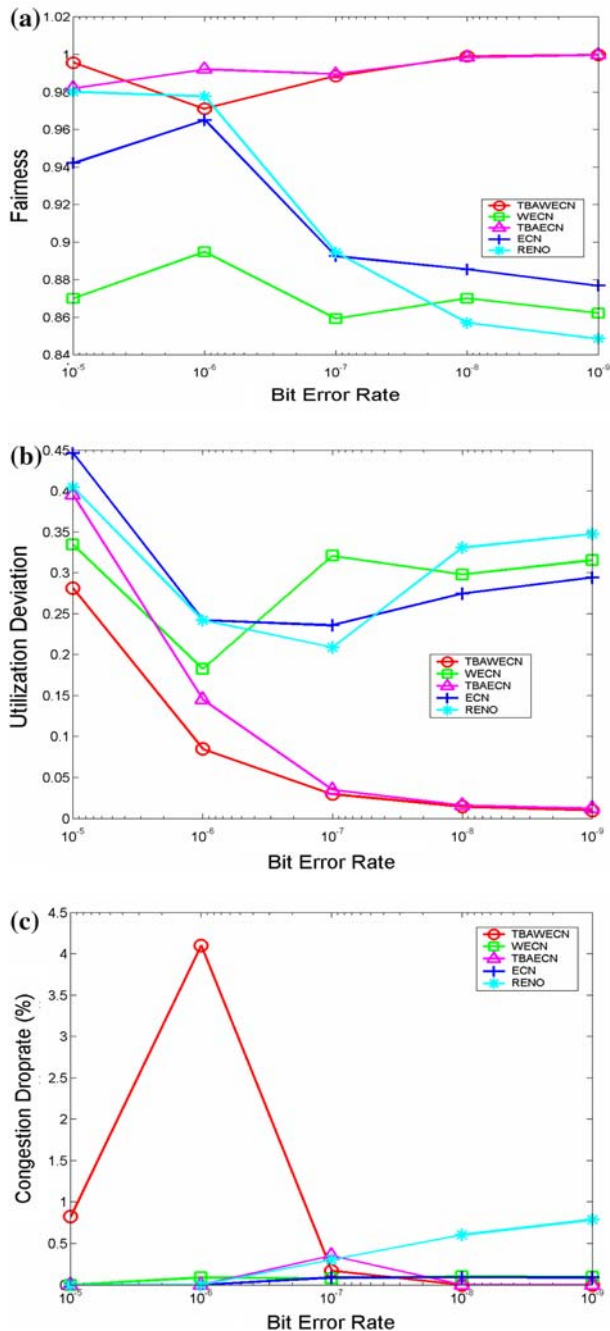


Fig. 8. (a) Fairness vs. queue size, topology 2, Config.3; (b) Utilization_deviation vs. queue size, topology 2, Config.3; (c) Congestion Drop Rate vs. queue size, topology 2, Config.3.

the schemes with ECN enhancement encounter less buffer overflow than RENO when BER drops below 10^{-8} . From the simulations, TCP Reno generally yields lower utilization and fairness performance in the presence of congestion or wireless losses.

7. CONCLUSIONS

We have presented simulation results to show that the TBA/WECN algorithm, enhanced by the traffic burst smoothing function, can improve TCP fairness and utilization in heterogeneous wireless networks in which the wireless link suffer from a high BER, long Round-trip time and traversing multiple congested routers. A desirable feature of our algorithm that eases implementation is that it does not depend on knowledge of the propagation delay, nor does it depend on loss estimation. Most of the algorithm is implemented at access node that requires minimal change of end systems. Although we have only presented results of applying these techniques to TCP Reno, simulations show that these techniques can similarly enhance the performance of other TCP versions.

The work presented in this paper is preliminary. In a number of ways the simulations presented in this paper are not complete. As an initial step this is a reasonable approach. However, additional simulations are needed to explore the parameter space, including: experiments with more complex and realistic traffic patterns and with more realistic corruption models. For the further improvement of fairness performance, other factors such as variation rapidity of wireless link characteristics should be considered. In addition, future work also needs to include tackling issues such as how to mitigate the practical security concerns introduced by the algorithm at the access node. It is also of interest to apply the algorithm to conditions where congestion occurs at other than access nodes.

ACKNOWLEDGMENT

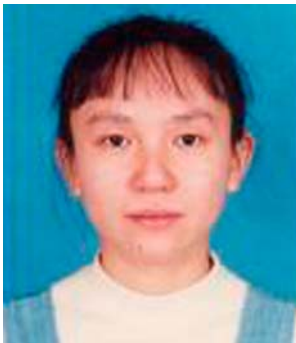
This work was supported by grants from Telus Mobility, Advanced Systems Institute of BC, and the Canadian Natural Sciences and Engineering Research Council under grant CRD247855-01. This paper is based in part on a paper presented at IEEE PIMRC 2003, Beijing, China, Sept. 2003.

REFERENCES

1. H. Elaarag, Improving TCP performance over mobile networks, *ACM Computing Surveys*, Vol. 34, pp. 357–374, 2002.
2. C. P. Fu, and S. C. Liew, TCP VenO: TCP enhancement for transmission over wireless access networks, *IEEE Journal of Selected Areas in Communications*, Vol. 21, pp. 216–228, 2003.
3. K. Xu, Y. Tian, and N. Ansari, TCP-Jersey for Wireless IP Communications, *IEEE Journal of Selected Areas in Communications*, Vol. 22, pp. 747–756, 2004.

4. E. H.-K. Wu, and M.-Z. Chen, JTCP: Jitter-based TCP for heterogeneous wireless networks, *IEEE Journal of Selected Areas in Communications*, Vol. 22, pp. 757–766, 2004.
5. G. Yang, R. Wang, M. Y. Sanadidi, and M. Gerla, TCPW with bulk repeat in next generation wireless networks. Proceedings of IEEE ICC, Anchorage, Alaska Vol. 1, pp. 674–678, 2003.
6. C. Q. Hua and T.-S. P. Yum, The fairness of TCP Vegas in networks with multiple congested gateways. Proceedings of IEEE International Conference on High Speed Networks and Multimedia Communications, pp.115–121, July 2002.
7. C. Jin, D. X. Wei, and S. H. Low, Fast TCP: Motivation, architecture, algorithms, performance. Proceedings of IEEE Infocom, Hong Kong, March 2004.
8. F. Paganini, Z. Wang, J. Doyle, and S. Low, Congestion control for high performance, stability and fairness in general networks, *IEEE Transaction on Networking*, Vol. 13, No. 1, pp. 43–56, 2005.
9. F. Peng and V. C. M. Leung, A novel fair bandwidth allocation algorithm for TCP window control. Proceedings of IEEE IPCCC'03, Phoenix, AZ, April 2003.
10. H.-J. Jang, and Y.-J. Suh, A flow control scheme for improving TCP throughput and fairness for wireless networks, *Proceedings of IEEE WCNC 2003*, Vol. 2, pp. 999–1003, 2003.
11. K. K. Ramakrishnan, S. Floyd, and D. Black, The addition of explicit congestion notification (ECN) to IP, RFC 3168, September 2001.
12. F. Peng, S. D. Cheng, and J. Ma, A proposal to apply ECN into wireless and mobile networks. Proceedings of INET, 2000.
13. F. P. Kelly, Models for a self-managed Internet, *Philosophical Transactions of the Royal Society*, Vol. 358, pp. 2335–2348, 2000.
14. R. Johari, and D. Tan, End-to-end congestion control for the Internet: delays and stability, *IEEE/ACM Transactions on Networking*, Vol. 9, pp. 818–832, 2001.
15. S. Floyd, R. Ramakrishna, and S. Shenker, Adaptive RED: An algorithm for increasing the robustness of RED's active queue management. Technical Report, ICSI, Aug. 2001.
16. The Network Simulator-ns-2, <http://www.isi.edu/nsnam/ns>.

BIOGRAPHY



Fei Peng received her M.S. degree and PH.D degree from Beijing University of Posts and Telecommunications, Beijing, China. she has been a Postdoc and work as a Research Associate with Electrical and Computer Engineering department at UBC. Her research interests include congestion and flow control, Wireless & Mobile Networks, etc.



Victor C. M. Leung (S'75–M'89–SM'97–F'03) received the B.A.Sc. (Hons.) and Ph.D. degrees in electrical engineering from the University of British Columbia (UBC), Vancouver, BC, Canada, in 1977 and 1981, respectively. From 1981 to 1987, he was a Senior Member of the Technical Staff at Microtel Pacific Research Ltd. (later renamed MPR Teltech Ltd.), Burnaby, BC, where he specialized in the planning, design, and analysis of satellite communication systems. He

also held a part-time position as a Visiting Assistant Professor at Simon Fraser University, Burnaby, in 1986 and 1987. In 1988, he was a Lecturer with the Department of Electronics, Chinese University of Hong Kong. He joined the Department of Electrical Engineering, UBC, in 1989, where he is a Professor, Holder of the TELUS Mobility Industrial Research Chair in Advanced Telecommunications Engineering, and a Member of the Institute for Computing, Information and Cognitive Systems. His research interests include the areas of architectural and protocol design and performance analysis for computer and telecommunication networks, with applications in satellite, mobile, personal communications, and high-speed networks. Dr. Leung is an Editor of the *IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS* and an Associate Editor of the *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*. He is the Technical Program Committee Vice-Chair in networking for the IEEE Wireless Communications and Networking Conference 2005 in New Orleans, LA, and has served on the TPCs of numerous international conferences. He was awarded the APEBC Gold Medal as the Head of the Graduating Class in the Faculty of Applied Science, UBC, in 1977. His graduate studies were sponsored by a Natural Sciences and Engineering Research Council Postgraduate Scholarship. He is a voting member of the Association for Computing Machinery.