

Robust EKF-Based Wireless Congestion Control

Xiaolong Li, *Member IEEE* Homayoun Yousefi'zadeh, *Senior Member, IEEE*

Abstract—The variation of bandwidth in wireless networks imposes significant challenges to the operation of congestion control protocols, especially, those relying on estimations of link bandwidth. For example, TCP CUBIC probes the end-to-end available link bandwidth while XCP and VCP require an explicit knowledge of the available link bandwidth at intermediate nodes. Thus, these protocols are subject to oscillatory behavior and serious performance deterioration in wireless networks without properly compensating against the fluctuations of the bandwidth. In this paper, we propose a bandwidth estimation scheme utilizing Extended Kalman Filtering (EKF) to which we refer as EBE. Rather than directly measuring bandwidth in real-time, EBE monitors either per flow states at a sender or persistent queue sizes to predict the available bandwidth. EBE is utilized to compensate against the impact of bandwidth variations thereby stabilizing and improving the performance of congestion control protocols in wireless networks. We implement EBE in NS2 and integrate it with XCP, VCP, TCP CUBIC, and few other TCP alternatives. Through extensive simulation studies, we demonstrate significant performance improvements of these protocols in wireless networks as the result of using EBE.

Index Terms—Wireless Congestion Control, Time-Varying Bandwidth Estimation, Extended Kalman Filtering, Oscillatory Behavior, TCP CUBIC, XCP, VCP, NS2.

I. INTRODUCTION

With the explosive growth of the Internet, performing effective congestion control has turned into a critically important issue. The problem of congestion control is nonlinear in nature, regardless of the use of wired or wireless transmission media. Most proposed congestion control protocols have been developed using nonlinear design principles. Examples of such design principles include two phase slow start, congestion avoidance dynamic windows, binary feedback, additive-increase, multiplicative-decrease, and multiplicative-increase. The use of wireless transmission media only exacerbates nonlinearity effects considering fading, shadowing, path loss, and other signal attenuation factors that are absent in wired transmission environments but have a direct impact on the effectivity of congestion control protocols.

As the defacto standard protocol of congestion control, TCP reveals efficiency problems in high Bandwidth Delay Product (BDP) networks due to its conservative approach in updating the congestion window (*cwnd*) size. Over time, many alternative protocols have been proposed to cope with this problem. These protocols roughly fall into two categories, end-to-end probing protocols and router-assisted explicit feedback protocols. While protocols of the former category such as [1],

[2], [3] perform aggressive probing of the value of *cwnd* in response to a congestion event, protocols belonging to the latter category such as eXplicit Congestion-control Protocol (XCP) [4], Variable-structure Congestion-control Protocol (VCP) [5], and Multi Packet Congestion-control Protocol (MPCP) [6] rely on an accurate knowledge of the link capacity for performing congestion control. Albeit following different operating mechanisms, the performance of protocols in both categories above can be significantly degraded in wireless networks because of frequent bandwidth variations. In order to accommodate protocols belonging to both categories, the concept of the link bandwidth used in our paper is two-fold. For router-assisted protocols, the concept of available link bandwidth is not the bandwidth as seen by a connection. Rather, it is the available physical link capacity monitored by network routers. For end-to-end protocols, the concept of available bandwidth is the available end-to-end bandwidth as seen by a TCP flow.

As the main motivation of this paper, the use of bandwidth estimation schemes can help relieve the problem of congestion control protocols associated with bandwidth variations. In recent years, many bandwidth estimation schemes have been proposed. Proactive probing schemes discussed in [7], [8], [9], [10], [11] estimate bandwidth by injecting probing packets into network. While these schemes may gain well on precision, packet injection may exacerbate congestion events. Packet dispersion techniques such as [12], [13], [14], [15] may somewhat alleviate the problem of active probing techniques by sending a smaller number of packets back-to-back into the network. Nonetheless, they are classified as active probing techniques. On the other hand, passive measurement schemes such as [16], [17], [18], [19], [20], [21], [22], [23] rely on passive measurements of the available bandwidth at routers. Also belonging to passive measurement schemes, certain TCP variants such as TCP Westwood (TCPW) [24] and TCP Vegas [25] estimate the available bandwidth based on the variance of the arriving rate and the Round Trip Time (RTT). Although these schemes avoid the disadvantage of proactive probing schemes, their accuracy is typically not as good as active measurement schemes.

In this paper, we propose a novel protocol-agnostic EKF-based Bandwidth Estimation (EBE) scheme that operates with TCP/VCP/XCP at the TRANSPORT layer. Utilizing EBE, both end-to-end protocols and router-assisted protocols can achieve significant performance improvements. In particular, EBE provides congestion control protocols with an accurate estimation of the actual link bandwidth by monitoring either per flow states or router related information whichever is of significance. In this paper the terms capacity and bandwidth are used interchangeably. Sender perceived available bandwidths are monitored in the case of end-to-end protocols, while

The authors ([xiaolonl,hyousefi]@uci.edu) are with the Center for Pervasive Communications and Computing at the University of California, Irvine. A preliminary part of this work was appeared in the Proceedings of IEEE ICC 2010. This work was sponsored in part by a research grant from the Boeing Company.

routers' persistent queue sizes are used for router-assisted protocols. While EBE predicts the *physical* link capacity for router-assisted protocols as seen by routers, it predicts end-to-end available bandwidth seen by end nodes for end-to-end protocols.

In comparison with the existing bandwidth estimation schemes, our work shares some of its features with those of XCP-b [16] and QFCP [20]. However, the absence of the major drawbacks of those schemes noted below significantly differentiates our work from them. First, the accuracy of the probing mechanism of XCP-b is negatively impacted in high BDP networks with small network buffer sizes. Moreover, XCP-b is a specific modification of XCP and not applicable to other congestion control protocols. As an alternative to XCP, QFCP provides per flow feedback based on flow rate instead of XCP's per packet feedback based on window adjustment. Using a simple linear increase by a fixed factor growth model for link capacity, QFCP responds very slowly to a large bandwidth increase. Further, using the fixed growth factor can eventually increase the link capacity estimate to a very large value (infinity in theory) when a link is always idle or underutilized. As such, QFCP has to place an upper bound on the estimate preventing it from working efficiently when bandwidth increases to a value beyond the set upper bound. Most importantly, both protocols represent reactive mechanisms to mitigate congestion events while our proposed work represents a proactive mechanism to prevent congestion events. As such, we argue that the proposed EBE scheme is more effective than those protocols in using queue status information that may be made available at routers.

This paper makes several contributions. First, it utilizes an EKF to provide a passive yet precise estimate of the available bandwidth. In doing so, it develops the system and measurement model of the EKF for end-to-end protocols of interest. To the best of our knowledge, this is the first work on applying EKF to available bandwidth estimation for congestion control protocols without introducing probing traffic. Second and unlike previously proposed schemes, the proposed scheme in this paper, EBE, is not limited to a specific congestion control protocol and can be used in conjunction with any congestion control protocol of interest. Third, this paper proposes utilizing an EBE maintainer module capable of effectively detecting both link bandwidth increases and decreases. Most existing approaches perform well on link bandwidth decreases, but none of them can *proactively* predict link bandwidth increases. Finally, the paper reports sample results of extensive performance evaluation experiments illustrating improvements for a wide range of congestion control protocols.

The rest of this paper is organized as follows. In Section II, we explain the detail functionality of our proposed EBE module based on a theoretical description of the EKF utilized in this study. Section III describes the individual modules used in the design and implementation of EBE. Simulation results and the analysis of those results are provided in Section IV. Finally, Section V concludes this paper.

II. EKF-BASED BANDWIDTH ESTIMATION

In this section, a brief mathematical description of the operating principles of the EKF used in our work is provided. The discussion is then followed by introducing the EBE system mapping into the context of the current discussion.

A. Extended Kalman Filter

Both Extended Kalman Filtering (EKF) [26] and Particle Filtering (PF) [27] are widely used tools for solving nonlinear state estimation problems. While an EKF linearizes the nonlinear dynamics of a system assuming the process and sensor noise have Gaussian distributions, a PF does not require the noise to be Gaussian. However, the advantage of PF over EKF is typically offset by its higher complexity. Considering its practical advantage, this study uses EKF as the estimation tool of choice after experimentally verifying the associated underlying assumptions. An EKF requires to define both a system state and a measurement model. Specifically, a system model can be described as

$$\underline{x}(t_i) = \underline{f}(t_i, t_{i-1}, \underline{x}(t_{i-1})) + \underline{w}(t_i) \quad (1)$$

where $\underline{x}(t_i)$ is an n -dimensional vector describing the state of the system at time t_i and \underline{f} is the n -dimensional nonlinear system state transition vector over the time interval $[t_{i-1}, t_i]$. The n -dimensional zero-mean discrete-time white Gaussian noise vector \underline{w} represents the unknown system dynamics not included in \underline{f} . It is assumed to have a covariance kernel

$$E \{ \underline{w}(t_i) \underline{w}^T(t_j) \} = \begin{cases} \Psi_\omega, & t_i = t_j \\ 0, & t_i \neq t_j \end{cases} \quad (2)$$

where Ψ_ω is an $n \times n$ matrix representing the strength of \underline{w} . The measurement model is defined as

$$\underline{z}(t_i) = H \underline{x}(t_i) + \underline{v}(t_i) \quad (3)$$

where \underline{z} is the m -dimensional measurement vector and H is the $m \times n$ measurement matrix. The measurement noise \underline{v} representing the uncertainty of the measurement is an m -dimensional vector containing discrete-time white Gaussian noise of zero mean and covariance kernel

$$E \{ \underline{v}(t_i) \underline{v}^T(t_j) \} = \begin{cases} R, & t_i = t_j \\ 0, & t_i \neq t_j \end{cases} \quad (4)$$

where R is an $m \times m$ matrix representing the strength of \underline{v} . EKF operates in two cycles. The first cycle known as the "prediction" cycle, projecting the current state $\hat{\underline{x}}(t_{i-1})$ and error covariance $P(t_{i-1})$ from t_{i-1} to t_i , obtains an apriori estimate $\hat{\underline{x}}(t_i^-)$ and error covariance $P(t_i^-)$ for t_i as

$$\hat{\underline{x}}(t_i^-) = \underline{f}[t_i, t_{i-1}, \hat{\underline{x}}(t_{i-1})] \quad (5)$$

$$P(t_i^-) = \nabla \underline{f} P(t_{i-1}) \nabla \underline{f}^T + \Psi_\omega \quad (6)$$

where $\nabla \underline{f}$ is the Jacobian matrix of the partial derivatives of \underline{f} with respect to \underline{x} . The second cycle known as the "update" cycle, incorporating a new measurement $\underline{z}(t_i)$ into the apriori estimate, obtains an improved aposteriori estimate $\hat{\underline{x}}(t_i)$ as

$$\hat{\underline{x}}(t_i) = \hat{\underline{x}}(t_i^-) + K(t_i) [\underline{z}(t_i) - H \hat{\underline{x}}(t_i^-)] \quad (7)$$

$$P(t_i) = [I - K(t_i)H] P(t_i^-) \quad (8)$$

In the equation above, the Kalman gain $K(t_i)$ is given by $K(t_i) = P(t_i^-)H^T [HP(t_i^-)H^T + R]^{-1}$. Given equations (1) through (8), the quantities \underline{x} , \underline{z} , \underline{f} , $\nabla \underline{f}$, Ψ_ω , H , and R have to be determined in order to utilize EKF.

B. EBE Mapping of Router-Assisted Protocols

In this section, we develop bandwidth estimation models for router-assisted protocols. As the protocols of interest in this category, this section focuses on XCP [4] and VCP [5].

Fundamentally, both XCP and VCP require all intermediate routers to monitor network congestion status and send feedback to the sender. While XCP directly controls the sender's sending rate, VCP only signals the sender with the value of load factor representing the ratio of demand to capacity. In XCP, feedback is calculated through the following equation

$$\Phi = a \cdot d \cdot S - b \cdot \zeta \quad (9)$$

where a and b are constant parameters, d is the average RTT , S represents the spare bandwidth calculated by subtracting the input traffic rate from the link capacity value, and ζ is the persistent queue size computed as the minimum queue size observed for an arriving packet in the last propagation delay. Once XCP converges, the value of ζ is supposed to be 0 and nearly 100% link bandwidth utilization achieved given an accurate estimate of the link capacity. In contrast, VCP calculates the value of Load Factor using equation

$$LF_l = \frac{T_l + \kappa_q \cdot \zeta}{U_t \cdot C_l \cdot t_I} \quad (10)$$

where T_l is the input traffic load during time interval t_I , κ_q controls how fast the persistent queue drains, C_l is the link capacity, U_t is the target utilization rate, and ζ is the persistent queue size.

Similar to XCP, the performance of VCP relies on an accurate estimation of the link capacity under which persistent queue size is designed to be nearly 0 when VCP converges. Since link capacity oscillates in wireless networks, the performance of both protocols can be significantly deteriorated without properly compensating against capacity oscillations.

Both XCP and VCP require individual link capacity values which are not directly measurable. However, given a router, the utilization of its downstream link can be approximated with a function of its persistent queue length [5]. The spare bandwidth can be derived from the router persistent queue length for XCP according to the work of [16]. EBE attempts at predicting the router queue length and translating the prediction result into the value of link capacity. Hence, router queue length and packet arrival rate are selected as state variables and the queue length is selected as the measurement for router-assisted protocols. We adopt the system and the measurement model utilized in [28] in which the underlying assumptions for using EKF in our work are validated. Specifically, $\underline{x}(t_i) = [x_1(t_i) \quad x_2(t_i)]^T$ where x_1 represents the queue length in packets and x_2 represents packet arrival rate in packets per second. Further, the measurement variable $z(t_i)$ is a scalar representation of the queue length. The

measurement matrix is then given by $H = [1 \quad 0]$. The vector $\underline{f}(\cdot) = [f_1(\cdot) \quad f_2(\cdot)]^T$ contains elements f_1 and f_2 representing state transition functions of x_1 and x_2 , respectively.

For $\hat{x}_1(t_{i-1}) \geq 1$

$$\begin{aligned} f_1 &= \frac{\rho}{1-\rho} (1 - Q_{\hat{x}_1(t_{i-1})+2}(\alpha, \\ &\quad - \frac{1}{\rho^{\hat{x}_1(t_i)}(1-\rho)} (1 - Q_{\hat{x}_1(t_{i-1})+2}(\beta, \alpha)) \\ &\quad + \hat{x}_1(t_{i-1})Q_{\hat{x}_1(t_{i-1})+1}(\alpha, \\ &\quad + \rho\mu\tau Q_{\hat{x}_1(t_{i-1})+2}(\alpha, \beta) - \mu\tau Q_{\hat{x}_1(t_{i-1})}(\alpha, \beta) \end{aligned} \quad (11)$$

For $\hat{x}_1(t_{i-1}) = 0$

$$\begin{aligned} f_1 &= \frac{\rho}{1-\rho} (1 - Q_2(\alpha, \beta)) \\ &\quad - \frac{1}{1-\rho} (1 - Q_2(\beta, \alpha)) + \rho\mu\tau Q_1(\alpha, \beta) - \mu\tau (1 - Q_2(\beta, \alpha)) \end{aligned} \quad (12)$$

where $\tau = t_i - t_{i-1}$, $\rho = \frac{1}{\mu} \hat{x}_2(t_{i-1})$, $\alpha = \sqrt{2\rho\mu\tau}$, $\beta = \sqrt{2\mu\tau}$, and μ is the service rate updated according to the output of EBE.

The function Q_n is given by

$$Q_n(\alpha, \beta) = \exp\left(-\frac{\alpha^2 + \beta^2}{2}\right) \sum_{k=1-n}^{\infty} \left(\frac{\alpha}{\beta}\right)^k I_k(\alpha\beta).$$

in which I is the modified Bessel function of the first kind defined as

$$I_n(y) = \sum_{k=0}^{\infty} \frac{(y/2)^{n+2k}}{k! \Gamma(n+k+1)} \quad (13)$$

Due to the unknown nature of the packet arrival rate, the state transition function of x_2 is approximated as a Brownian motion process [29]. Such approximation results in identifying f_2 as

$$f_2 = \hat{x}_2(t_{i-1}) \quad (14)$$

Note that since f is not continuous, the partial derivatives of \underline{f} must be approximated by a two-sided difference equation. Therefore,

$$\nabla_{\underline{f}_{-11}} = \frac{f_1\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \Delta x_1 \\ 0 \end{bmatrix}\right) - f_1\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} \Delta x_1 \\ 0 \end{bmatrix}\right)}{2\Delta x_1} \quad (15)$$

Further,

$$\nabla_{\underline{f}_{-12}} = \frac{f_1\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta x_2 \end{bmatrix}\right) - f_1\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 0 \\ \Delta x_2 \end{bmatrix}\right)}{2\Delta x_2} \quad (16)$$

In the equations above, Δx_1 and Δx_2 represent small perturbations from the value of x_1 and x_2 , respectively. Because x_1 represents the queue size and must be an integer, Δx_1 is chosen to be 1, the smallest perturbation allowed. In addition, Δx_2 is chosen to be $\frac{x_2}{100}$ in order to provide a scaled perturbation that was two orders of magnitude smaller than

the state. Accordingly and given $\underline{x} = \hat{\underline{x}}(t_{i-1})$, the Jacobian matrix of partial derivatives of \underline{f} is approximated by

$$\nabla \underline{f} = \begin{bmatrix} \nabla f_{11} & \nabla f_{12} \\ \nabla f_{21} & \nabla f_{22} \end{bmatrix} = \begin{bmatrix} \nabla f_{11} & \nabla f_{12} \\ 0 & 1 \end{bmatrix} \quad (17)$$

where

$$\nabla f_{11} = \frac{f_1 \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) - f_1 \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right)}{2}$$

$$\nabla f_{21} = \frac{f_1 \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{x_2}{100} \end{bmatrix} \right) - f_1 \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 0 \\ \frac{x_2}{100} \end{bmatrix} \right)}{\frac{x_2}{50}} \quad (18)$$

The two dynamic noise signals are treated as uncorrelated and thus Ψ_ω is diagonal. The values in the following equation are obtained from our experiments which result in the best performance improvement.

$$\Psi_\omega = \begin{bmatrix} 1.0 \times 10^{-5} & 0 \\ 0 & 5.0 \times 10^{-6} \end{bmatrix} \quad (19)$$

Finally, the measurement noise R is a scalar determined by iteratively calculating the variance of z .

C. EBE Mapping of End-to-End Protocols

In this section, we develop bandwidth estimation models for end-to-end protocols. Most end-to-end congestion control protocols perform an estimation of the available bandwidth in order to set the parameters *ssthresh* (threshold of slow start) and *cwnd* after observing a packet loss. As the main protocol of interest in this category outperforming other alternatives such as TCP Westwood and TCP Vegas, this section focuses on TCP CUBIC [2]. In contrast to XCP and VCP, TCP CUBIC grows its congestion window size only depending on two consecutive congestion events. The window growth expression (*cwnd*) in CUBIC uses the function

$$W(t) = \eta(t - K)^3 + W_{max} \quad (20)$$

where η is a scaling factor, t is the elapsed time from the last window reduction caused by a packet loss event, W_{max} is the last value of *cwnd* that caused packet loss, and $K = \sqrt[3]{\frac{\beta W_{max}}{C}}$. In the latter expression, β is a constant multiplication decrease factor applied to window reduction at the time of a loss event.

In TCP CUBIC, the estimation of available bandwidth is mapped to W_{max} . Our previous experimental results [30] have revealed that TCP CUBIC performs best in various wireless scenarios among recent TCP variants because of the utilization of the cubic function and the introduction of W_{max} . Thus in this paper, we integrate EBE with the TCP CUBIC showing how it can improve bandwidth utilization by providing a more accurate estimate of the available bandwidth to end-to-end protocols. Nonetheless, we note that EBE can be integrated with any congestion control protocol.

TCP CUBIC registers the value of W_{max} . When the value of *cwnd* is close to W_{max} , the cubic function reaches its plateau around W_{max} . Since the value of W_{max} remains

unchanged until the next packet loss event occurs, TCP CUBIC is not able to efficiently detect the variations of the available bandwidth. Thus, EBE focuses on adaptively predicting the proper value of W_{max} based on RTT variations. When a packet loss event occurs due to congestion and under FIFO queuing scheme, each flow experiences its maximum RTT. In case Active Queue Management (AQM) schemes such as RED are deployed along the path of a flow, a packet loss event might occur before the router queue overflows. Note that packet loss still occurs when the router queue builds up to certain level, which makes the flow experience an RTT close to its maximum RTT. Thus, in the rest of the paper, we do not differentiate a FIFO queue from an AQM queue. Denoting R_{max} as the value of RTT when TCP CUBIC updates W_{max} , an accurate estimate of W_{max} puts the RTT of the flow close to R_{max} when *cwnd* is close to W_{max} . Hence, a significant difference between the values of RTT and R_{max} implies either an increase or decrease of the available bandwidth. When a packet loss happens, EBE registers the values of *cwnd* and RTT as W_{max} and R_{max} . To simplify the system model, EBE only provides the estimate at the plateau of the cubic function. More specifically, W_{max} is decreased when $\left| \frac{W(t_i)}{W_{max}} - 1 \right| < \epsilon$ and if $0 < \frac{RTT(t_i)}{R_{max}} - 1 < \Omega$. W_{max} is increased if $1 - \frac{RTT(t_i)}{R_{max}} > \nu$, where $0 < \Omega < \nu < 1$. Next, we discuss how EBE adjusts the value of W_{max} based on the variations of RTT. Define the expected maximum throughput for a flow without packet loss as $T_{max} = \frac{W_{max}}{R_{max}}$. Since W_{max} is the expected maximum value of *cwnd*, we approximate the projected maximum throughput by $T_{actual}(t_i) = \frac{W_{max}(t_i)}{RTT(t_i)}$ at t_i when the actual *cwnd* is close to W_{max} . Then, the variation in the value of $T_{max}(t_i)$ is determined by the difference between the $T_{max}(t_{i-1})$ and $T_{actual}(t_{i-1})$. The rationale is that when EBE needs to decrease the value of W_{max} , the difference between $T_{actual}(t_i)$ and T_{max} is supposed to be consumed by other flows and thus the value of W_{max} has to be reduced to $W_{max} - \delta$, where δ is defined as $(T_{actual}(t_i) - T_{max})R_{max}$. In contrast, when EBE needs to increase the value of W_{max} , the value of T_{actual} is larger than T_{max} . Since the flow has already consumed more bandwidth than the estimated maximum available bandwidth, the value of W_{max} can be set as the current *cwnd* without causing congestion.

Based on the explanation above, we define the system and measurement models of EKF. There are two state variables defined as $\underline{x}(t_i) = [x_1(t_i) \quad x_2(t_i)]^T$ where x_1 represents W_{max} in packets, x_2 represents the RTT of the flow, and $z(t_i)$ is the scalar measurement variable representing W_{max} . The measurement matrix is given by $H = [1 \quad 0]$.

Let t_L denote the time at which the most recent loss event happened, then $\lambda = \frac{\hat{x}_2(t_{i-1})}{\hat{x}_2(t_L)} - 1$. Defining the logical operator \clubsuit as

$$\clubsuit = \left(\left(\left| \frac{W(t_{i-1})}{\hat{x}_1(t_{i-2})} - 1 \right| < \epsilon \right) \& (0 < \lambda < \Omega) \right) \mid (-\lambda > \nu) \quad (21)$$

the state transition function of x_1 is described as follows.

$$f_1 = \hat{x}_1(t_{i-1}) + f_1^\dagger \quad (22)$$

where

$$f_1^\dagger = \begin{cases} \left(1 - \frac{\hat{x}_2(t_{i-1})}{\hat{x}_2(t_L)}\right) \hat{x}_1(t_{i-2}), & \text{if } \clubsuit \\ 0, & \text{otherwise} \end{cases} \quad (23)$$

The transition function of x_2 is represented as

$$f_2 = \hat{x}_2(t_{i-1}) \quad (24)$$

Accordingly, the Jacobian matrix of partial derivatives of \underline{f} is given by

$$\nabla \underline{f} = \begin{bmatrix} 1 & \nabla f_{12} \\ 0 & 1 \end{bmatrix} \quad (25)$$

where

$$\nabla f_{12} = \begin{cases} -\frac{\hat{x}_2(t_{i-1})}{\hat{x}_2(t_L)}, & \text{if } \clubsuit \\ 0, & \text{otherwise} \end{cases} \quad (26)$$

The definition of Ψ_ω is the same as that of Equation (19) and the scalar measurement noise R is determined by iteratively calculating the variance of z . In our work, we set $\epsilon = 0.3$, $\Omega = 0.2$, and $\nu = 0.5$ reflecting our best experimental findings.

III. SUPPORTING EBE MODULES

In this section, we describe individual modules used in the design and implementation of EBE.

A. EKF Predictor

The EBE utilizes an EKF to provide a precise estimation of the quantities of interest to this study. For router-assisted protocols, EKF predictor runs on each router. For end-to-end protocols, EKF only runs on the sender side. While EKF predictor can be deployed in different places, it is designed to use a uniform sampling method. Specifically, there are two timers associated with the EKF predictor. The first timer to which we refer as the measurement timer is used for sampling the persistent queue size, i.e., arriving rate in the case of XCP and VCP; W_{max} and RTT in the case of TCP CUBIC. The timer expires once every 10 msec such that it can provide the underlying protocol with ample time to converge. The second timer to which we refer as the estimation timer is for making the estimation of the real-time persistent queue size and W_{max} . Importantly, the discrete-time points used by EKF are defined by the second timer. Note that this timer is independent of the timing of the congestion control algorithm. Thus, there is no correlation between these time points and the congestion control algorithm. As indicated in Internet measurements report [31], roughly 75% – 90% of the flows have RTTs less than 200 msec. Hence, we set the timer to expire once every 200 msec, which happens to be consistent with the sampling timers of XCP and VCP [4], [5]. Consequently, there are 20 available measurement samples between two consecutive timeouts of the estimation timer. The mean of these samples is delivered to EKF as z .

The EKF starts its first cycle, i.e., the prediction cycle, in which it gets an a priori state estimation x^- and an a priori covariance estimation P^- for the current estimate using Equation (5) and (6). Thereafter, using sampled measurement z of the queue size in the case of XCP and VCP or W_{max} value in the case of CUBIC, the EKF produces the Kalman gain,

the a posteriori state x , and the a posteriori covariance P as presented in subsection II.

Notably, the estimation of EKF is sensitive to the value of parameter Ψ_ω . In our experiments, we set Ψ_ω as that of Equation (19) experimentally yielding the most accurate estimate. The output of the EKF is delivered to the Link Capacity Monitor (LCM) in order to retrieve the instant link capacity as presented below. At this stage, the parameters of the EKF have been tuned for producing a precise prediction.

B. Link Capacity Monitor

The LCM primarily serves router assisted protocols. In the case of end-to-end protocols, the output of the EKF is the estimation of the maximum available bandwidth for a flow which can be directly used in the control algorithm for regulating *cwnd*. The following discussion is focused on LCM for router assisted protocols. To feed the EKF, EBE keeps track of the persistent queue size during the last estimation interval. According to the control algorithm of XCP and VCP and under the assumption of having an accurate estimate of the link capacity, the expected persistent queue size is close to zero when either protocol converges. If the actual link capacity is lower than the estimate, the queue will build up to compensate against the estimation error [16]. In fact, nearly all congestion control protocols have the design goal of forming nearly zero persistent queue sizes. The link capacity estimation error is reflected in a built up queue. Thus, for router-assisted explicit feedback protocols, EBE derives the spare bandwidth from the queue variation in each estimation interval instead of calculating the difference between the link capacity and the input traffic rate following Equation (6) of [16].

Further, EBE retrieves the link capacity by adding up the input traffic rate and the spare bandwidth.

C. EBE Maintainer

The EBE maintainer component is designed to increase the responsiveness of EBE in the case of a bandwidth increase. Specifically, when link bandwidth increases, the persistent queue size and RTT are supposed to decrease. However, if the previous persistent queue size and RTT remain low or nearly zero, such bandwidth increase will not affect the queue size and RTT significantly. As a result, EBE may not be able to drive the EKF in order to make an accurate prediction. Thus, the EBE maintainer is introduced to force an increase in the sending rate and to build a reasonable value for the persistent queue and RTT. Specifically, the EBE maintainer monitors the ratio of the average queue size and RTT in the current estimation interval and in the last estimation interval.

Queue Maintainer: For XCP and VCP, we maintain a queue length that equals to 15% of the queue buffer size in our simulation. Therefore, for instance, if the ratio is smaller than or equal to 1.0 and both two average queue sizes are below the threshold value of 15% of the buffer size, the queue maintainer assumes the link bandwidth has increased. It, then, outputs a factor of Υ to the congestion control protocol yielding a *shrunk* load factor or *inflated* feedback until it detects a ratio larger than 1.0 and an average queue

sizes larger than the threshold value. Note that the threshold value represents a tradeoff between the sensitivity of EBE to bandwidth variations and the efficiency of congestion control. The higher the threshold, the more sensitive the EBE to bandwidth increases. However, a high threshold value yields a high delay and a lower robustness to traffic bursts. We would like to note that the threshold value also depends on the system configurations, i.e., bottleneck queue capacity. In our experiments, EBE delivers its best performance in terms of bandwidth utilization with a threshold in the range of [13%–18%]. In Section IV, we provide experimental evidence to our argument.

The choice of Υ will vary slightly depending on the choice of the protocol. We set Υ to 1.14 for VCP and to 1.40 for XCP indicating our best experimental findings.

RTT Maintainer: For TCP CUBIC, the same logic is followed. Note that when a loss event happens, the queue is full and a flow is experiencing its maximum RTT. Under the assumption that the queuing delay is dominant when congestion happens, we set the threshold of RTT at $R_{max} \times 15\%$. If $RTT(t_i)/RTT(t_{i-1}) \leq 1.0$ and both $RTT(t_i)$ and $RTT(t_{i-1})$ are below the threshold, the RTT maintainer assumes that the link bandwidth has increased. Then, the same value of Υ as that in Queue Maintainer is used to *inflate* the $cwnd$ until a ratio larger than 1.0 and average RTTs larger than the threshold value are detected. In our simulation, we set Υ to 2.0 for TCP CUBIC. It is important to note that the choice for the value of Υ represents a tradeoff. A high value for Υ yields better bandwidth efficiency and a higher persistent queue size imposing negative effects on tolerance to bursty traffic. To the contrary, a low value for Υ yields a low speed of converge and a smaller persistent queue size. It is also important to note that such design does not require support from the congestion control protocol itself.

IV. PERFORMANCE EVALUATION

In this section, the NS2 performance evaluation results are presented. We implement EBE in NS2 and apply it to XCP, VCP¹, and CUBIC. We demonstrate EBE-based performance improvements of the protocols of interest measured in terms of bandwidth utilization, buffer occupancy, $cwnd$ dynamics, and the average FTP completion times. In our experiments, we refer to EBE-improved version of the protocols as XCP-EBE, VCP-EBE, and CUBIC-EBE. In the case of XCP, we also compare our results with those of XCP-b.

We have two simulation topologies, namely, dumbbell and parking-lot topologies. Both topologies share the following baseline configurations. In the case of XCP and VCP, all side links are configured to have a one way delay of 4 msec. The bandwidth of the bottleneck link(s) varies(vary) between 1 Mbps and 20 Mbps with a one way delay of 32 msec. Reverse path traffic exists in all simulations. A single simulation experiment has an overall duration of 120 sec.

¹It is worth noting that we have also applied EBE to MPCP as a multi-packet feedback congestion control protocol alternative to VCP. Considering the fact that EBE is agnostic to the use of multi-packets, the results are similar to those of VCP and are not reported here.

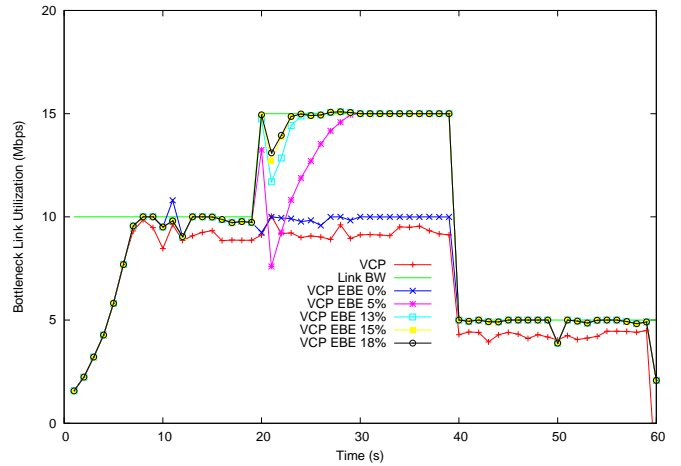


Fig. 1. The effects of EBE Queue threshold.

In the case of CUBIC, all side links are configured to have a one way delay of 25 msec. The bandwidth of the bottleneck link(s) varies(vary) between 1 Mbps and 400 Mbps with a one way delay of 200 msec. A single simulation experiment has an overall duration of 200 sec.

In all cases, an RTT difference of 10 msec is introduced among flows. Further, random bandwidth values are uniformly distributed² over the range of [1, 20] Mbps in the case of router-assisted protocols and [1, 400] Mbps in the case of TCP CUBIC. Each value holds for 10 seconds. Note that the chosen range of bandwidth variations affects the transient behavior of the protocols with which we experiment. While having a smaller maximum bandwidth reduces the reaction speed of EBE, having a larger maximum bandwidth may cause a more significant packet loss upon a bandwidth decrease.

Before providing specific results associated with the topologies of our experiment, we present experimental evidence as to why EBE delivers its best performance in terms of bandwidth utilization with a threshold in the range of [13%–18%].

Using the settings above, Fig. 1 illustrates the effects of the threshold on performance for the case of VCP. The initial bandwidth is set to 10Mbps. The bandwidth changes to 15Mbps at 20 seconds followed by 5Mbps at 40 seconds. As anticipated, when the threshold is set to 0, EBE fails to detect the bandwidth increase. When the threshold is set to 5%, EBE can gradually catch up with the bandwidth increase. However, the convergence speed is lower than the case associated with a threshold of 13%. Note that when the threshold is beyond 13%, there is no significant performance variation in terms of bandwidth utilization. However, when the threshold is higher than 18%, more oscillations are observed in terms of $cwnd$ dynamics affecting fairness. For example., while the average Jain fairness index is close to 0.98 with a threshold 18%, the fairness index reduces to under 0.9 with a more aggressive threshold. Moreover, it is desired to keep the queue length as low as possible in order to accommodate bursty traffic. Thus,

²While not reported here, we note that our experimental results with few other distributions have yielded comparable results. In fact, we have observed that the EKF can work with any distribution for as long as its time constant is relatively smaller than the change rate.

we observe that a choice of 15% for the value of threshold best addresses the factors above. While not reported due to shortage of space, the results associated with other protocols are similar to what is reported here.

A. Dumbbell Topology

Fig. 2(a) shows the dumbbell topology of our simulation study in which the bottleneck link bandwidth is assumed to be varying randomly according to a uniform distribution. There are two types of end-to-end aggregate FTP flows traversing the topology, namely, long-lived and short-lived flows. For XCP and VCP, there are 20 long-lived FTP flows traversing the bottleneck links in both directions and 10 short-lived FTP flows traversing the bottleneck link in the forward direction going from left to right³. All flows begin at a random start time between 0 sec to 300 msec. The buffer size of the bottleneck link is set to 575 KBytes. For CUBIC, there are 5 long-lived flows and 10 short-lived flows. All flows begin at a random start time between 0 sec to 100 sec. The buffer size of the bottleneck link is set to 3300 KBytes. CUBIC parameters are chosen to aid the protocol cope with its inherent significant performance degradation. For both XCP and VCP, the initial bandwidth is set to 10 Mbps. For CUBIC, we set the initial bandwidth to be 200 Mbps.

1) *Performance Comparison of VCP and VCP-EBE:* Fig. 3(a) compares the persistent queue size of the bottleneck link in the case of VCP and VCP-EBE. While VCP successfully maintains a low queue size when the link capacity is smaller than the configured capacity, severe oscillations and overflows are observed in the opposite case. Essentially, VCP is unaware of the underlying bandwidth variations and makes adjustments to its sending rate using the fixed advertised value of the link capacity. Accordingly, it does not make any specific adjustment as the link bandwidth increases and therefore maintains a low queue size as usual. VCP still attempts to achieve a 10 Mbps bandwidth throughput as the link bandwidth drops. Thus, the queue size increases and eventually overflows. Then, it reacts to overflow with a sending rate drop causing oscillation. In contrast, VCP-EBE can detect the change of link bandwidth and respond appropriately. Over the entire simulation period, VCP-EBE maintains a relatively steady queue size which is about 15% of the buffer size in our simulation. Although the average queue size for VCP-EBE is larger than that of VCP, the average queue size for VCP-EBE is at an acceptable range illustrating the tradeoff between efficiency and buffer occupancy.

Fig. 3(b) compares the bottleneck link utilization achieved by VCP and VCP-EBE. While the actual link bandwidth is larger than the configured value of 10 Mbps in the time interval from 10 sec to 60 sec, VCP fails to utilize the increased bandwidth efficiently. When the bandwidth drops, VCP still keeps increasing its sending rate in order to achieve the perceived utilization calculated using a fixed bandwidth value. The latter results in growing the queue size of the

bottleneck link and eventually making the link congested. To the contrary, VCP-EBE demonstrates near 100% utilization during the whole simulation period illustrating good sensitivity and responsiveness to bandwidth variations.

Fig. 4(a) and 4(b) illustrate the *cwnd* dynamics achieved by VCP-EBE and VCP. This experiment is performed using 5 long-lived flows. VCP-EBE shows responsiveness and efficiency as the result of maintaining a relatively steady value for *cwnd* and being able to catch up with the variations of the link bandwidth. In contrast, VCP fails to regulate the value of *cwnd* efficiently showing severe oscillations. Fig. 5(a) and 5(b) statistically illustrate the *cwnd* dynamics achieved by VCP-EBE and VCP using 50 long-lived flows. In the figures, *cwnd* dynamics of a sample flow, the 10-percentile, and 90-percentile of *cwnd* dynamics of all flows are plotted. It is observed that VCP-EBE consistently converges and the value of *cwnd* for all flows fall into a small range as the link bandwidth varies. In contrast, VCP fails to converge and demonstrates severe oscillations in *cwnd* values. Note that during the first 60 seconds when the link bandwidth increases, VCP converges in a relatively small range of *cwnd* values yielding inefficiency. Further, VCP fails to converge when link bandwidth significantly decreases after 60 seconds.

2) *Performance Comparison of XCP, XCP-b, and XCP-EBE:* In this section, we compare the performance of XCP, XCP-b, and XCP-EBE. Fig. 6(a) compares the persistent queue size of the bottleneck link for the three variants of XCP. In the case of XCP and similar to the case of VCP, the behavior of the persistent queue size is quite different when the link capacity is larger or smaller than the configured capacity. XCP could successfully maintain a low queue size in the former case, while oscillations are observed in the latter case based on the same reasons explained in the case of VCP. Both XCP-b and XCP-EBE show similar queue size characteristics. While both schemes maintain relatively steady queue sizes, their average queue sizes are larger than that of XCP over the entire simulation period. It is also observed that the average queue size of XCP-EBE is typically larger than that of XCP-b. Fig. 6(b) compares the bottleneck link bandwidth utilization achieved by XCP, XCP-b, and XCP-EBE. A close look at the results reveals the following observations. First, XCP fails to utilize the increased bandwidth efficiently when the actual link bandwidth is larger than the configured value. When the bandwidth drops, XCP “blindly” increases the sending rate eventually causing a significant packet loss. To the contrary, XCP-EBE demonstrates near 100% utilization during the whole simulation period illustrating good sensitivity and responsiveness to bandwidth variations. Moreover, XCP’s response to bandwidth is a little faster than both XCP-EBE and XCP-b at 80 sec and 100 sec marks. The reason is that the persistent queue size of XCP is larger than that of XCP-EBE and XCP-b. Hence, when the link bandwidth increases at 80 sec and 100 sec, the decrease in persistent queue size is more prominent in XCP than it is in XCP-EBE and XCP-b. In essence, XCP notifies the sender to increase its sending rate and has a faster response. After the queue size drains to a small value, XCP is no longer capable of knowing the actual link bandwidth and utilization will stay low. XCP-b shows

³While not shown here, we have observed that our proposed scheme works for other scenarios in which a large number of short-lived flows dominate the traffic mix.

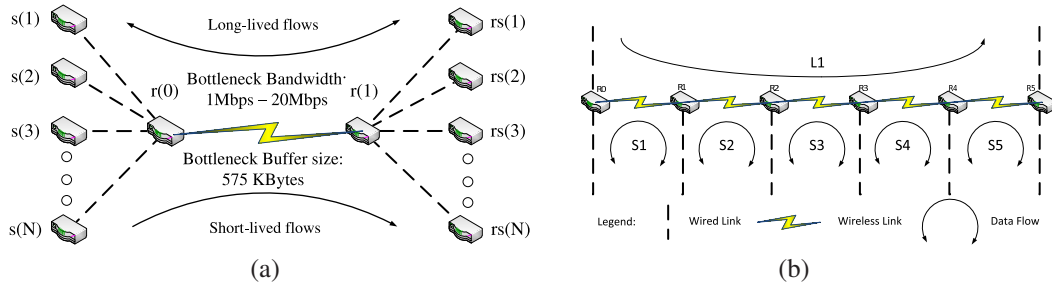


Fig. 2. An illustration of the (a) dumbbell, and (b) parking-lot topology used in our simulations.

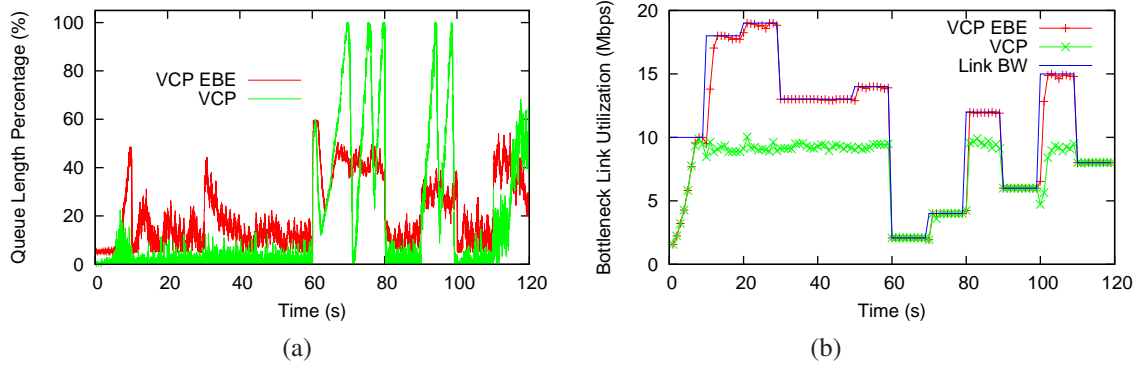


Fig. 3. A performance comparison of (a) the persistent queue size, and (b) the utilization of the bottleneck link over the dumbbell topology for VCP and VCP-EBE.

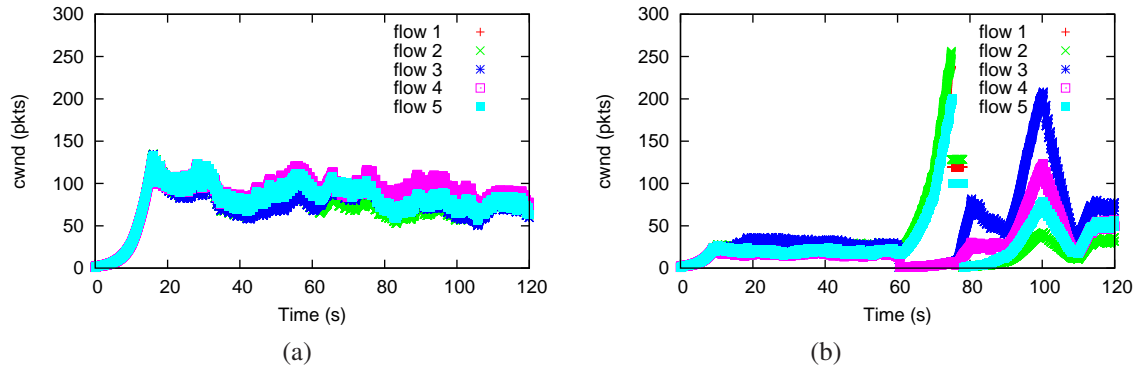


Fig. 4. A five-flow performance comparison of the *cwnd* of the bottleneck link of the dumbbell topology for (a) VCP-EBE and (b) VCP.

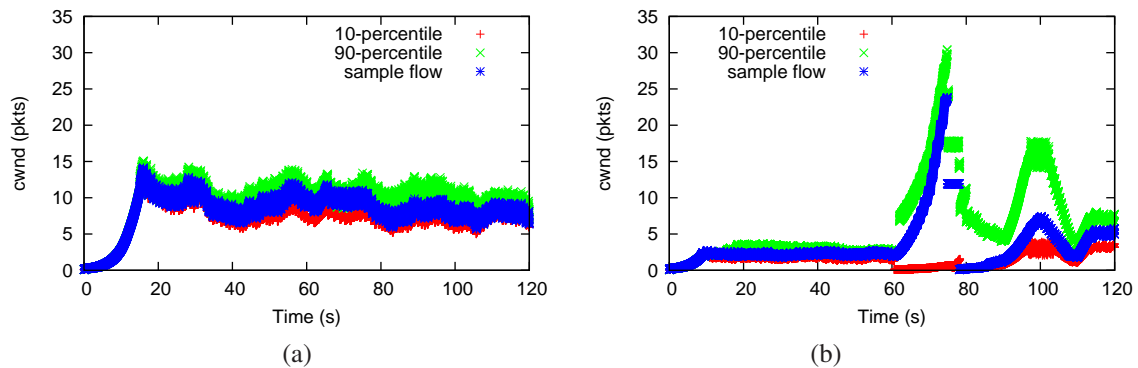


Fig. 5. A fifty-flow statistical performance comparison of the *cwnd* of the bottleneck link of the dumbbell topology for (a) VCP-EBE and (b) VCP.

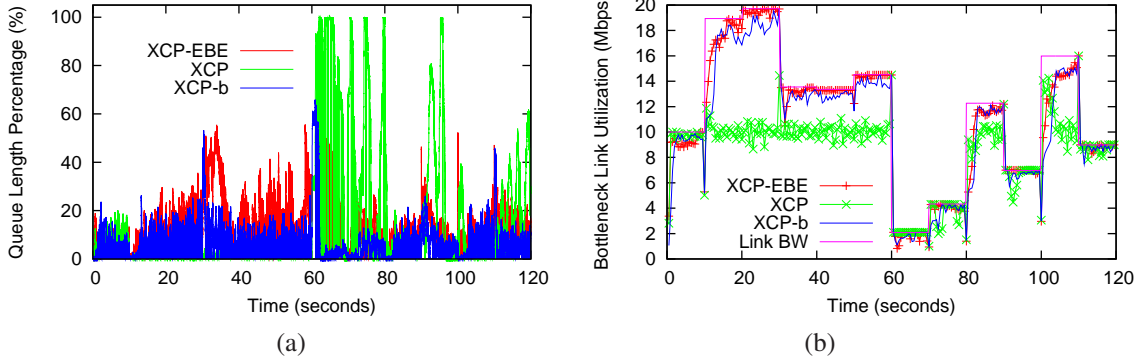


Fig. 6. A performance comparison of (a) the persistent queue size, and (b) the utilization of the bottleneck link over the dumbbell topology for XCP, XCP-b, and XCP-EBE.

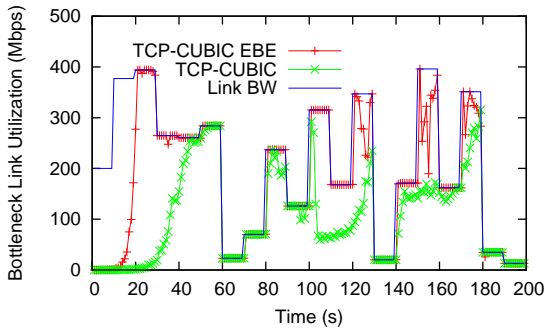


Fig. 7. A performance comparison of the bottleneck link utilization over the dumbbell topology for CUBIC and CUBIC-EBE.

a similar behavior as that of XCP-EBE. However, XCP-EBE typically achieves a higher utilization and has a faster response than XCP-b as the bandwidth varies.

3) *Performance Comparison of CUBIC and CUBIC-EBE:* Fig. 7 compares the bottleneck link bandwidth utilization achieved by CUBIC and CUBIC-EBE. CUBIC fails to acquire the increased bandwidth and converges very slowly because the flows are asynchronously generated in the region of 0 sec to 100 sec. However, it should be noted that although CUBIC performs poorly when the bandwidth increases, the utilization rate is still better than that of XCP or VCP due to the aggressive probing of the available bandwidth. In contrast to CUBIC, CUBIC-EBE responds very quickly to the change of link bandwidth and achieves a nearly 100% utilization rate during the whole simulation period. Furthermore, CUBIC-EBE takes one third of the time CUBIC takes to converge. We could also observe oscillations when the link bandwidth is not changing, which is mainly caused by the buildup of the RTT value and the aggressiveness of EBE Maintainer in EBE module.

4) *Performance Comparison of FTP File Transmission:* In this scenario, we have 5 FTP flows running in the dumbbell topology. For XCP and VCP, each FTP flow transfers a 50 MB data file. Flows start randomly within the interval of [0, 300] msec. The average completion time of VCP, VCP-EBE, XCP, and XCP-EBE averaged over 3 experiments are 289 sec, 198 sec, 270 sec, and 212 sec, respectively. It is obvious that with

EBE the congestion control protocols could still maintain a high utilization rate even when the link bandwidth is not fixed. The latter leads to a significantly shorter completion time for file transfer. For CUBIC, each FTP flow transfers a 100 MB data file. Flows start randomly within the interval of [0, 200] msec. The average completion time of CUBIC and CUBIC-EBE averaged over 3 experiments are 211 sec and 185 sec, respectively. CUBIC-EBE is able to perceive the link bandwidth variations and maintain a high utilization rate while the original CUBIC fails to do so. Therefore, the completion time is about 15% shorter for CUBIC-EBE than CUBIC.

B. Parking-Lot Topology

Fig. 2(b) shows the parking-lot topology used in our study. There are 6 nodes in our parking-lot topology. The link bandwidth for each link varies independently according to a uniform distribution throughout the simulation time. The lowest capacity link is originally set to be the link between $R2$ and $R3$. If not specified below, simulation settings are the same as those of the dumbbell topology. There are two types of FTP flows. The first type of flows are aggregated end-to-end FTP flows moving in the forward direction from left to the right and the other type of flows are local FTP flows moving in both directions. For XCP and VCP, there are 30 flows of the former type and 10 flows of the latter type. For CUBIC, there are 5 flows of the former type and 2 flows of the latter type. For XCP and VCP, all flows begin at a random start time between 0 msec to 300 msec. The initial bandwidth of the link between $R2$ and $R3$ is set to 10 Mbps while it is set to 20 Mbps for other bottleneck links. For CUBIC, all flows begin at a random start time between 0 sec to 100 sec. The initial bandwidth of the link between $R2$ and $R3$ is set to 200 Mbps while it is set to 400 Mbps for the other bottleneck links.

1) *Performance Comparison of VCP and VCP-EBE:* Fig. 8(a) compares the persistent queue size of the bottleneck link measured in the cases of VCP and VCP-EBE. For VCP, there are two sharp spikes at around 30 sec and 100 sec. These spikes represent the ability of VCP to drain the queue quickly in response to a bandwidth increase. However it should be noted that these quick responses are observed only when the bandwidth value is no greater than the fixed bandwidth value

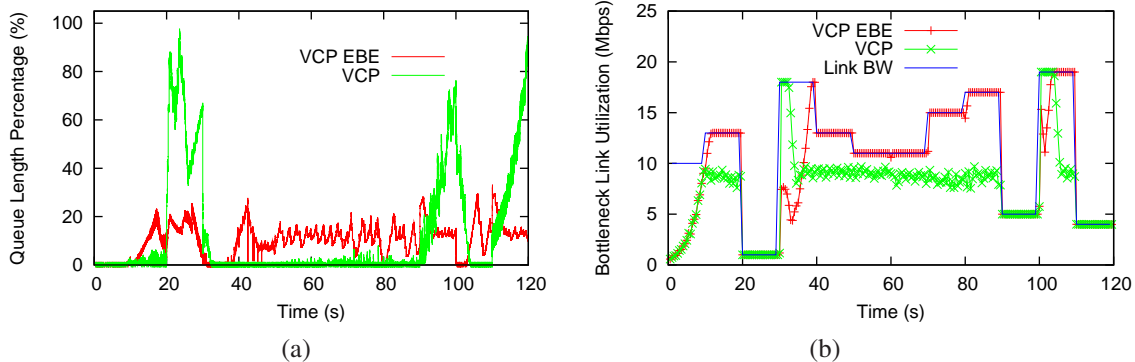


Fig. 8. A performance comparison of (a) the persistent queue size, and (b) the utilization of the bottleneck link over the parking-lot topology for VCP and VCP-EBE. The bottleneck link bandwidths vary from 1 to 40 Mbps.

of VCP. From 70 sec to 80 sec, the bandwidth increases twice with values beyond the fixed bandwidth value of VCP. However, the persistent queue size of VCP remains still. For VCP-EBE, the persistent queue size is maintained approximately at 20% of the buffer size throughout the simulation period except a sharp bandwidth decrease at around 20 sec. Since VCP-EBE attempts at maintaining the queue size at a certain level, queue size variations also representing bandwidth variations are not as conspicuous as in the case of VCP. Nonetheless, queue size variations are observed and VCP-EBE controls the queue size better than VCP. Due to the introduction of the local flows in the parking-lot topology, the average queue sizes of VCP-EBE and VCP are higher than those in the dumbbell topology.

Fig. 8(b) compares the achieved bottleneck link utilization of VCP and VCP-EBE. VCP fails to achieve a high utilization when the actual link bandwidth is higher than 10 Mbps. However, from 30 sec to 40 sec and from 100 sec to 110 sec, VCP catches up with the bandwidth increase for about 8 sec, and then it drops to a low level. The reason is that before the bandwidth increases, the actual bandwidth stays much lower than 10 Mbps which causes the queue to build up to a certain level. Upon an increase of the actual bandwidth, the queue drainage contributes to a high bandwidth utilization. Therefore, VCP is able to enhance its sending rate. However, when the persistent queue size is finally lowered to a small value, VCP cannot obtain the actual link bandwidth and as such utilization degrades. Notably, upon two sharp bandwidth increases at 30 sec and 100 sec, VCP temporarily outperforms VCP-EBE due to a significant queue build up. Beyond that and in contrast to the performance of VCP, VCP-EBE demonstrates near 100% utilization during the whole simulation period illustrating good sensitivity and responsiveness to bandwidth variations.

2) *Performance Comparison of XCP, XCP-b, and XCP-EBE*: In this section, we compare the performance of XCP, XCP-b, and XCP-EBE. The utilizations and queue sizes are observed over five consecutive links of the parking-lot topology.

While Fig. 9(a) compares persistent queue sizes for the three variants of XCP, Fig. 9(b) compares the link bandwidth utilization achieved by them. In [4] and when evaluating performance in a parking-lot topology, the authors show that XCP achieves a low utilization for those links traversed before

the bottleneck link. The reason is that those links are throttled at the bottleneck link, yet XCP still keeps shuffling the bandwidth from local flows to long-distance flows. Therefore, XCP achieves a low utilization at those links preceding the middle link, then a higher utilization at the middle link, and finally maintains utilization at the same level in the links passed the middle link. However in our simulation, the link bandwidth for each link varies independent of other links, meaning that the link with the lowest capacity could be changed during the simulation period. It is shown in Fig. 9(b) that the utilization rate of XCP cannot follow the same pattern as that in [4]. The reason for the spike in the middle link is that the bandwidth value is initially set to 10 Mbps while it is set to 20 Mbps for other links. Further, the amplitude of bandwidth variation in the middle link is higher than that of the other links. Therefore and when the bandwidth of the middle link is larger than 10 Mbps, utilization significantly drops due to protocol's inability to observe a bandwidth increase beyond its original set value. It should be noted that the utilization of XCP-b is only higher than that of XCP in the middle link. The reason is that with a much larger queue size build up during the overall simulation and as illustrated in Fig. 9(a), XCP could still respond well by observing the queue draining and growing. On the other hand, XCP-b maintains its queue size at a very small level which leads to inefficiency in response to bandwidth variations. XCP-EBE also maintains its queue size at a certain level although the value is much larger than that of XCP-b. This gives XCP-EBE the advantage of efficiently detecting bandwidth variations while maintaining a queue size much smaller than that of XCP. It is shown in Fig. 9(b) that the utilization of XCP-EBE is better than that of both XCP-b and XCP during the simulation period.

3) *Performance Comparison of CUBIC and CUBIC-EBE*: Fig. 10 compares the bottleneck link bandwidth utilization achieved by CUBIC and CUBIC-EBE. As demonstrated in the figure, CUBIC fails to converge during the simulation period. Moreover, CUBIC fails to acquire the available bandwidth when the link bandwidth increases. On the other hand, CUBIC-EBE converges in approximately 60 sec. Furthermore, CUBIC-EBE responds very quickly to the changes of link bandwidth and achieves a high utilization close to 100% during

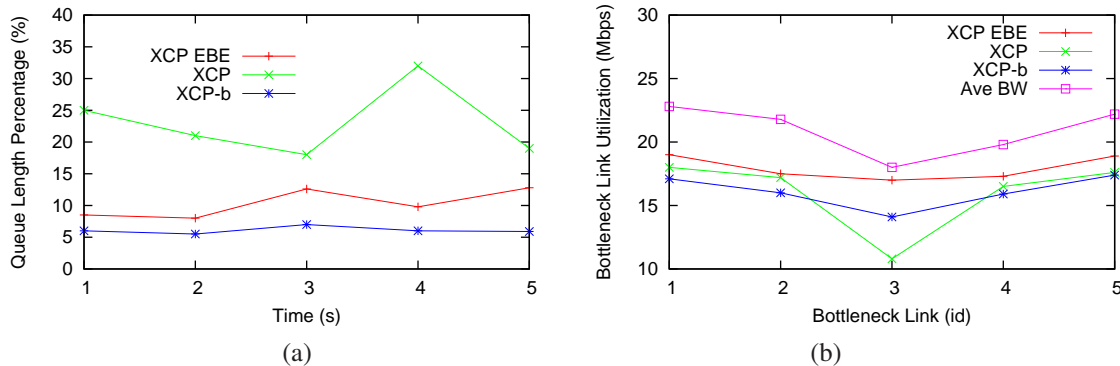


Fig. 9. A performance comparison of (a) the average persistent queue sizes, and (b) the utilizations of five consecutive links over the parking-lot topology for XCP, XCP-b, and XCP-EBE. The bottleneck link bandwidths vary from 1 to 40 Mbps.

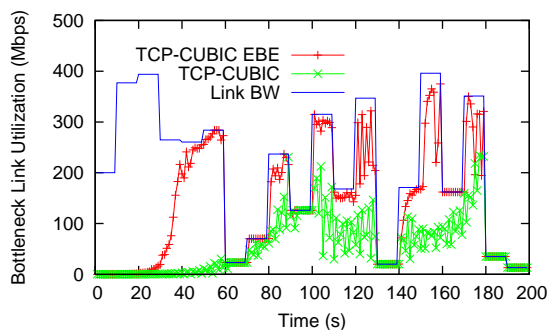


Fig. 10. A performance comparison of the bottleneck link utilization over the parking-lot topology for CUBIC and CUBIC-EBE. The bottleneck link bandwidths vary from 1 to 400 Mbps.

the simulation period. We also observe oscillations when the link bandwidth is not changing which is mainly caused by the build up of the RTT value and the aggressiveness of EBE Maintainer in the EBE module. In addition, convergence in the parking-lot topology is significantly slower than that in dumbbell topology due to the introduction of the local flows.

At the end of this section, we would like to comment on the convergence performance of EBE for scenarios in which multiple flows join or leave a bottleneck link at different times. We note that the target application scenario of EBE is wireless networks where physical link capacity changes occur. Since available link capacity changes due to flow joins and leaves would not cause sudden queue length oscillation, EBE does not offer significant performance improvements. In such scenarios, the convergence performance of EBE is comparable to that of standard approaches without EBE in terms of bandwidth utilization. More specifically, the use of EBE slightly improves the speed of convergence at the cost of slightly higher oscillations. Further, standard approaches outperform EBE in terms of queue dynamics in such scenarios since EBE attempts to maintain the queue length at a certain level.

C. ANOVA Analysis

The one-way analysis of variance (ANOVA) can be used to determine whether there is any significant difference between

the means of two groups of data. However, in the case of having more than two groups of data, one-way ANOVA can only tell whether there are at least two different groups of data, but not which specific groups were significantly different from each other. In such cases, post-hoc tests, e.g., Tukey test, can be performed to compare every two groups.

In order to perform statistical tests, we run 30 sets of experiments with VCP, VCP-EBE, XCP, XCP-b, XCP-EBE, TCP CUBIC, and TCP CUBIC-EBE. The settings of the experiments are the same as those presented in Section IV-A. We compare the average bandwidth utilization and the average Jain's fairness index [32]. The results are analyzed utilizing one-way ANOVA followed by a Tukey Post Hoc test with a significance level of 0.01. We use IBM SPSS Statistics 21 tool to perform ANOVA analysis and Tukey test. It turns out that EBE does make significant performance improvements in comparison with its non-EBE alternatives. In all tests, ANOVA analysis and Tukey Post Hoc test verify that the EBE performance improvements are significant at a significance level of 0.01. Note that in the case of VCP and TCP, Tukey Post Hoc test is not performed since we only have two variants to compare. The output of SPSS contains a *descriptives* table providing useful descriptive statistics, including mean, standard deviation, and standard error of the data set, followed by an *ANOVA* table showing the analysis results of ANOVA. The value in the last column of the ANOVA table and the fourth column of the Multiple Comparisons table *Sig.*, i.e. *P*, shows whether there is a statistically significant difference between group means. Taking XCP as an example, we report the ANOVA analysis results in Fig. 11 and 12. For the reported results, the significance level is set to 0.01. Thus, there is a significant difference between the means of data sets if the output value of ANOVA *P* is less than 0.01. It is observed in all of our tests that there is a statistically significant difference between EBE and non-EBE alternatives in terms of average bandwidth utilization and fairness. Since the reported means of EBE are larger than those of non-EBE, we make the argument that EBE is statistically better than the non-EBE alternatives. The only exception is a case in which XCP-b achieves better fairness than XCP-EBE as shown in Fig. 12. However, the output value of Tukey Post Hoc test *P* is 0.237 in that

Descriptives				
Bandwidth Utilization (Mbps)				
	N	Mean	Std. Deviation	Std. Error
XCP	30	7.384123529430856	.030016957852316	-.005480321641213
XCP-B	30	9.892779745886358	.031711562547082	-.005789712713673
XCP-EBE	30	10.002485510593097	.029159135611505	-.005323705444006
Total	90	9.093129595303436	1.216423465886428	-.12822291715938

ANOVA					
Bandwidth Utilization (Mbps)					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	131.612	2	65.806	71608.850	.000
Within Groups	.080	87	.001		
Total	131.692	89			

Post Hoc Tests

Multiple Comparisons

Dependent Variable: V3
Tukey HSD

(I)	(J)	Mean Difference (I-J)	Std. Error	Sig.	99% Confidence Interval	
					Lower Bound	Upper Bound
XCP	XCP-b	-2.508656216455603*	.007827155219771	.000	-2.532072322100718	-2.485240110810488
	XCP-EBE	-2.618361981162341*	.007827155219771	.000	-2.641778086807456	-2.594945875517226
XCP-b	XCP	2.508656216455603*	.007827155219771	.000	2.485240110810488	2.532072322100718
	XCP-b	-.109705764708839*	.007827155219771	.000	-.133121870351954	-.086289659061724
XCP-	XCP	2.618361981162341*	.007827155219771	.000	2.594945875517226	2.641778086807456
	EBE	-.109705764708839*	.007827155219771	.000	-.086289659061724	-.133121870351954

*. The mean difference is significant at the 0.01 level.

Fig. 11. The ANOVA analysis results of XCP Bandwidth Utilization.

Descriptives				
Jain Fairness Index				
	N	Mean	Std. Deviation	Std. Error
XCP	30	.973706965160460	.002756002356490	.000503174886477
XCP-b	30	.995883034358211	.002637449261602	.000481530151700
XCP-EBE	30	.994804562148393	.002248120407205	.000410448753086
Total	90	.988131520555688	.010572948158792	.001114486592245

ANOVA					
Jain Fairness Index					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	.009	2	.005	717.885	.000
Within Groups	.001	87	.000		
Total	.010	89			

Post Hoc Tests

Multiple Comparisons

Dependent Variable: Jain Fairness Index
Tukey HSD

(I)	(J)	Mean Difference (I-J)	Std. Error	Sig.	99% Confidence Interval	
					Lower Bound	Upper Bound
XCP	XCP-b	-.022176069197850*	.000680062841557	.000	-.024150745921922	-.020201392473779
	XCP-EBE	-.021097596989033*	.000680062841557	.000	-.023072273712104	-.0191922920263962
XCP-b	XCP	.022176069197850*	.000680062841557	.000	.020201392473779	.024150745921922
	XCP-EBE	.001078472203917	.000680062841557	.237	-.000896204514354	.003053148833988
XCP-	XCP	.021097596989033*	.000680062841557	.000	.019122920263962	.023072273712104
	EBE	-.001078472203917	.000680062841557	.237	-.003053148833988	.000896204514354

*. The mean difference is significant at the 0.01 level.

Fig. 12. The ANOVA analysis results of XCP Fairness.

case showing the difference between the two alternatives is insignificant.

D. Effectivity Discussion

In this subsection, we provide an effectivity discussion on the use of EBE. In the case of router-assisted protocols, EBE relies on queue length variations to predict link capacity. Since sudden queue length oscillations are not caused by flow joins and leaves, EBE does not offer significant performance improvements in such cases. Thus, non-EBE approaches can outperform EBE approaches since EBE attempts to maintain the queue length at a certain level. Furthermore, due to the

use of the EBE maintainer, the performance of EBE is loosely coupled with the queue size. While EBE functions properly when utilizing very small buffers, such cases do not represent the best use case of EBE because it will have difficulty detecting bandwidth increases in such cases.

In the case of TCP CUBIC, EBE maps delay variations to bandwidth variations and provides a more accurate available bandwidth estimation than that of CUBIC. However, bandwidth and delay variations can be relatively small in low delay and low bandwidth networks yielding insufficient delay variations. In such scenarios, the use of EBE does not significantly improve the performance of TCP CUBIC. Accordingly, we argue that the best use case of EBE for TCP CUBIC and other similar TCP variants is in high delay and high bandwidth networks.

Finally, we would like to note that the use of EBE is complementary to congestion control protocols. EBE does not change the control algorithm of such protocols and hence does not affect their behavior significantly.

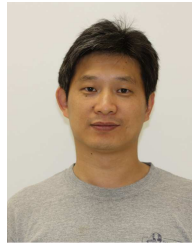
V. CONCLUSION

In this paper, we proposed EBE an EKF-based bandwidth estimation module in order to provide an accurate measurement of the real-time link bandwidth necessary to stabilize the operation of congestion control protocols in wireless networks. We discussed how our proposed protocol-agnostic bandwidth estimation module differs from the existing bandwidth estimation methods. Rather than directly measuring link capacity, our module used the persistent queue size and maximum congestion window size to estimate link capacity. Moreover, we showed that our module could help a number of congestion control protocols achieve a high utilization when link bandwidth increases while the same protocols failed to perform the same way without the aid of our module. We also evaluated the performance of our module after implementing it in NS2. Through simulation studies, we demonstrated how the use of EBE could eliminate the oscillatory behavior of a variety of congestion control protocols in wireless networks, thereby significantly improving the performance of original protocols.

REFERENCES

- [1] L. Xu, K. Harfoush, and I. Rhee, "Binary Increase Congestion Control (BIC) for Fast Long-Distance Networks," in *Proc. Infocom 2004*, 2004.
- [2] I. Rhee and L. Xu, "CUBIC: A New TCP-Friendly High-Speed TCP Variant," in *Proc. PFLDNet 2005*, Feb. 2005.
- [3] S. Floyd, "HighSpeed TCP for Large Congestion Windows," Aug. 2002.
- [4] D. Katabi, M. Handley, and C. Rohrs, "Congestion Control for High Bandwidth-Delay Product Networks," in *Proc. ACM SIGCOMM 2002*, Aug. 2002.
- [5] Y. Xia, L. Subramanian, I. Stoica, and S. Kalyanaram, "One More Bit Is Enough," in *Proc. ACM SIGCOMM 2005*, Aug. 2005.
- [6] X. Li and H. Yousefi'zadeh, "MPCP: Multi Packet Congestion-control Protocol," *ACM Computer Communications Review*, Oct. 2009.
- [7] S. Ekelin, M. Nilsson, E. Hartikainen, A. Johnsson, J.-E. Mangs, B. Melander, and M. Bjorkman, "Real-Time Measurement of End-to-End Available Bandwidth using Kalman Filtering," in *Proc. IEEE/IFIP NOMS 2006*, Apr. 2006, pp. 73 – 84.
- [8] J. Liebeherr, M. Fidler, and S. Valae, "A System-Theoretic Approach to Bandwidth Estimation," *Networking, IEEE/ACM Transactions on*, vol. 18, no. 4, p. 1040, Aug 2010.

- [9] M. Liu, M. Claypool, and R. Kinicki, "WBest: A bandwidth estimation tool for IEEE 802.11 wireless networks," in *IEEE LCN 2008*, 2008, p. 374.
- [10] X. Liu, K. Ravindran, and D. Loguinov, "A Stochastic Foundation of Available Bandwidth Estimation: Multi-Hop Analysis," *Networking, IEEE/ACM Transactions on*, vol. 16, no. 1, p. 130, Aug. 2008.
- [11] A. Cabellos-Aparicio, F. Garcia, and J. Domingo-Pascual, "A Novel Available Bandwidth Estimation and Tracking Algorithm," in *NOMS Workshops IEEE*, 2008, p. 87.
- [12] S. Keshav, "A control-theoretic approach to flow control," in *Proc. ACM SIGCOMM*, 1991.
- [13] J. Bolot, "End-to-end packet delay and loss behavior in the internet," in *Proc. ACM SIGCOMM*, 1993.
- [14] R. Carter and M. Crovella, "Measuring bottleneck link speed in packet-switched networks," *Performance Evaluation*, Oct. 1996.
- [15] C. Dovrolis, P. Ramanathan, and D. Moore, "Packet-dispersion techniques and a capacity-estimation methodology," *IEEE/ACM Trans. on Networking*, vol. 12, no. 6, 2004.
- [16] F. Abrantes and M. Ricardo, "XCP for shared-Access Multi-Rate Media," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 3, pp. 27 – 38, July 2006.
- [17] Y. Su and T. Gross, "WXCP: Explicit Congestion Control for Wireless Multi-hop Networks," in *Proc. IWQoS 2005*, June 2005, pp. 313 – 326, available at <http://www.springerlink.com/content/xqvhwdmtj0bym6jfl/>.
- [18] S. Rangwala, A. Jindal, K. Jang, and K. Psounis, "Understanding Congestion Control in Multi-hop Wireless Mesh Networks," in *Proc. ACM SIGMOBILE 2008*, 2008, pp. 291 – 302.
- [19] K. Chen, K. Nahrstedt, and N. Vaidya, "The Utility of Explicit Rate-Based Flow Control in Mobile Ad Hoc Networks," in *Proc. IEEE WCNC 2004*, vol. 3, Mar. 2004, pp. 1921 – 1926.
- [20] J. Pu and M. Hamdi, "Enhancements on Router-Assisted Congestion Control for Wireless Networks," *Wireless Communications, IEEE Transactions on*, vol. 7, no. 6, pp. 2253 – 2260, June 2008.
- [21] K. Lakshminarayanan, V. Padmanabhan, and J. Padhye, "Bandwidth estimation in broadband access networks," in *Proc. of IMC*, 2004.
- [22] A. Johnsson, B. Melander, and M. B. Orkman, "Bandwidth measurement in wireless networks," in *Proc. Mediterranean Ad Hoc Networking Workshop*, 2005.
- [23] H. Lee, V. Hall, K. Yum, K. Kim, and E. Kim, "bandwidth estimation in wireless lans for multimedia streaming services," *Advances in Multimedia*, January 2007.
- [24] C. Casetti, M. Gerla, S. Mascolo, M. Sanadidi, and R. Wang, "TCP Westwood: end-to-end congestion control for wired/wireless networks," *Wireless Networks (WINET)*, vol. 8, no. 5, pp. 467–479, Sept. 2002.
- [25] L. Brakmo and L. Peterson, "TCP Vegas: end to end congestion avoidance on a global internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, pp. 1465 – 1480, Oct. 1995.
- [26] P. Maybeck, "The Kalman Filter: An Introduction to Concepts," 1990.
- [27] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved Particle Filter for Nonlinear Problems," in *Proc. Inst. Elect. Eng., Radar, Sonar, Navig*, 1999.
- [28] N. Stuckey, J. Vasquez, S. Graham, and P. Maybeck, "Stochastic Control of Computer Networks," *IET Control Theory and Applications*, vol. 6, no. 3, pp. 403–411, February 2012.
- [29] N. Stuckey, "Stochastic estimation and control of queues within a computer network." Master's thesis, AIR FORCE INSTITUTE OF TECHNOLOGY, Wright-Patterson Air Force Base, Ohio, Dec. 2007.
- [30] X. Li and H. Yousefi'zadeh, "A Performance Evaluation of Modern Feedback-Based and Unfriendly-Region Congestion Control Solutions," in *Proc. IEEE MILCOM 2010*, 2010.
- [31] X. Jiang and C. Dovrolis, "Passive Estimation of TCP Round-Trip Times," *ACM Computer Communications Review (CCR)*, vol. 32, no. 3, pp. 75–88, July 2002.
- [32] R. Jain, D. Chiu, and W. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems," *DEC Research Report*, no. TR-301, 1984.



Xiaolong Li is currently a Research Specialist at the Department of EECS at UC, Irvine. He received his MS degree from the department of Computer Science and Engineering at the University of Notre Dame in 2006, and his Ph.D. from the department of EECS at the University of California, Irvine in 2009. His research interests are in the area of wireless congestion control and wireless routing. Xiaolong is a member of IEEE.



Hodayoun Yousefi'zadeh is an Associate Adjunct Professor at the Department of EECS at UC, Irvine. He also holds a Consulting Chief Technologist position at the Boeing Company. In the recent past, he was the CTO of TierFleet, a Senior Technical and Business Manager at Procom Technology, and a Technical Consultant at NEC Electronics. He is the inventor of several US patents, has published more than sixty scholarly reviewed articles, and authored more than twenty design articles associated with deployed industry products. Dr. Yousefi'zadeh is with the editorial board of IEEE Trans. Wireless Communications and Journal of Communications Networks. Previously, he served as an editor of IEEE Communications Letters, an editor of IEEE Wireless Communications Magazine, the lead guest editor of IEEE JSTSP the issue of April 2008, and the track chair as well as the TPC memembr of various IEEE and ACM conferences. He was the founding Chairperson of systems' management workgroup of the Storage Networking Industry Association, a member of the scientific advisory board of Integrated Media Services Center at the University of Southern of California (USC), and a member of American Management Association. He received the Ph.D. degree from the Dept. of EE-Systems at USC in 1997.