IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 2, 2020, accepted September 12, 2020, date of publication September 18, 2020,
date of current version September 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3024707

End-to-End TCP Congestion Control for Mobile Applications

HAICHANG HUANG, ZHIYANG SUN, AND XIN WANG, (Senior Member, IEEE)
Department of Communication Science and Engineering, Fudan University, Shanghai 200082, China

Corresponding author: Xin Wang (xwang11@fudan.edu.cn)

This work was supported in part by the CERNET Innovation Project under Grant NGII20180117, and in part by the Innovation Program of Shanghai Municipal Science and Technology Commission under Grant 17510710400.

ABSTRACT This paper addresses the re-engineering of congestion control for TCP applications over networks with coupled wireless links. Using queueing delay as a congestion measure, we show that optimal TCP congestion control can be achieved by developing window-control oriented implicit primal-dual solvers for intended network utility maximization problem. Capitalizing on such an idea, we prove the existence of scalable, easy-to-deploy, yet optimal end-to-end congestion control schemes for networks with wireless links, given that the wireless access point appropriately schedules packet transmissions. A class of so-called QUIC-TCP congestion control algorithms are developed. Relying on a Lyapunov method, we rigorously establish the global convergence/stability of the proposed QUIC-TCP to optimal equilibrium in the network fluid model. Numerical results corroborate the merits of the proposed schemes in IPv6-based Internet environments.

INDEX TERMS Congestion control, wireless-link scheduling, convex optimization, network fluid model, Lyapunov method.

I. INTRODUCTION

The Internet TCP (transmission control protocol) is well suited for extensibility and scalability since it does not need network configuration information. This protocol performs end-to-end congestion control schemes [2], where sources of data transfers monitor their own connections and adjust their transmission window sizes to avoid overloading the network without explicit signals fed back from network routers. Most existing data traffic over Internet is carried by TCP-based protocols (such as HTTP, SMTP and FTP). After years of development, although the network has expanded greatly in size, load, and connectivity (by more than six orders of magnitude), these protocols still function well [3]. However, plenty of recent works showed that TCP protocols poorly perform in (IPv6 based) Internet with lossy wireless links and large bandwidth-delay products [4], [5]. As the continuously growing Internet contains more and more wireless links, the current TCP schemes need to be re-designed or re-optimized for emerging mobile applications.

To this end, a Lagrange duality based network fluid model was proposed in [6]. Leveraging optimization tools, a network utility maximization (NUM) paradigm was put forth to analyze and design cross-layer schemes and network protocols [7]. In such a framework, congestion control is modeled as a source rate controller, and the queue length of the link is used as the Lagrange multiplier. Based on this approach, many gradient-type NUM solutions were developed. Although a direct source-rate control may be suitable for delay-/bandwidth-sensitive non-TCP audio/video applications, it is difficult to be operated in a TCP window-control manner. Furthermore, such rate controllers require the network routers to pass message of the queue lengths, thereby undermining the scalability of TCP. Due to the latter implementation and scalability issues, it is hardly possible to apply NUM schemes for TCP applications.

On the other hand, an engineering method is also used to strengthen the transmission control protocol for large-scale (wireless) networks, see, e.g., TCP-Vegas, DFTCP, STCP, XJTCP, Cubic, Sprout, Verus, ABC [2], [8]–[20]. In order to improve the efficiency of the TCP congestion control, a TCP congestion control algorithm called TCP BBRv2 was proposed based on the Kleinrock congestion model. Since this algorithm does not rely on packet loss to estimate the end-to-end bandwidth, it is insensitive to the packet loss rate. However, when TCP BBRv2 occupies a large amount of

The associate editor coordinating the review of this manuscript and approving it for publication was Rentao Gu.

bandwidth, it does not work well and the fairness is relatively low. Some recent works developed solutions to improve the performance [21]–[23]. In addition, artificial intelligence and machine learning approaches were recently adopted to develop TCP congestion control schemes [24]–[27]. Simulations and experiments were employed to validate those new schemes. Yet, they somewhat lack systematic design methodologies and theoretical performance guarantees. With queueing delays playing the roles of congestion measures as well as Lagrange multipliers (for intended optimization problems), congestion control algorithms in the Mo-Walrand scheme and FAST-TCP were developed in a systematic manner [4], [28]. Stability of these schemes was demonstrated either analytically or empirically for *wired* networks.

We can observe that both NUM solutions and enhanced TCP schemes could be regarded as (explicit or implicit) primal-dual solvers to relevant NUM problems. In optimization theory, Lagrange multipliers are introduced as auxiliary dual variables to help solve constrained problems. In NUM solutions, queue lengths play the role of Lagrange dual variables [7]; whereas queueing delays are mapped into these dual variables in TCP schemes [4], [28]. These ''dual variables'' also serve as the congestion measures for flow sources. Without the participation of network routers, the round-trip delay (i.e., the total queueing delay) of the packet can be directly calculated at the source node. Hence, it appears that queueing delays can be a better choice as congestion measure. Confined to the TCP design space, we need to execute a window-based congestion control algorithm at the source based on such a measure. With the ''primal'' source rates and ''dual'' queueing delays becoming functions of window sizes, end-to-end window adjustments equivalently turn into *implicit* primal-dual updates. The optimal TCP congestion control then amounts to development of *window-control oriented* implicit primal-dual solvers for intended optimization problem.

Capitalizing on this idea, we prove the existence of *scalable, easy-to-deploy, yet optimal* congestion control solutions for mobile TCP applications. To this end, we reveal that a joint design of TCP congestion control (at transport layer) and wireless-link scheduling (at link layer) is required. In particular, a queueing-delay based ''MaxWeight''-type scheduling needs to be operated at the access point of wireless links. With such a scheduler, the wireless link capacities, if coupled, can be coupled in a ''good'' way, making the optimal end-to-end TCP congestion control feasible. Besides, we generalize the Mo-Walrand scheme [28] to develop a class of QueueIng-Control (QUIC) TCP congestion control algorithms. We show that the proposed QUIC-TCP algorithms together with the queueing-delay based MaxWeight-scheduler amount to window-control oriented implicit primal-dual solvers for the intended network optimization problem. Leveraging the Lyapunov method, we rigorously establish the global convergence of the proposed schemes to an optimal equilibrium in the network fluid model.

Notice that the present work is an extension of [29, Chapter 2], where we only considered the case that there exists a smooth capacity region for the wireless links (i.e., the case considered in Sec. IV-B). In this paper, we extend the results to the general cases that we may have a box capacity region or a convex piece-wise linear capacity region (see Sec. IV-A and Sec. IV-C). In current wireless standards, wireless links are usually given dedicated TDMA/FDMA channels and only a few levels of rate and/or power adaptations are adopted, leading to the box capacity region or convex piece-wise linear region. Hence, our generalizations here are both of theoretic values and of practical significance. The proposed approach can serve as a stepping stone to advance the theory for general cross-layer optimization of Internet protocols.

The remainder of the paper is presented as follows. Section II delineates underlying optimization, prior works, the gap and challenges. Section III develops the proposed joint TCP congestion control and wireless-link scheduling scheme, while Section IV establishes the global convergence/stability of the proposed scheme to optimal network equilibrium. Section V contains simulation results to validate the merits of QUIC-TCP. Section VI concludes the paper.

II. PRELIMINARIES

Consider a network with a wired backbone and a wireless access point (AP) for mobile devices. The set of data links $L = L_f \cup L_w$ in the network consists of a wired link set L_f and a wireless link set L_w. While any wired link $l \in L_f$ has a constant capacity c_l, the capacities for wireless links need to be carefully modeled.

Let $\boldsymbol{r} := \{r_l, \forall l \in L_w\}$, where r_l represents the capacity of wireless link l. In the existing wireless standards [30], error correction coding, rate and power adaptation, and hybrid ARQ are generally used at the link layer to ensure reliable data transmission. As a result, the capacities r_l of wireless links are (possibly) coupled and dynamic, due to random fading and broadcast nature of the wireless medium. For each fading realization $\boldsymbol{h}$, denote by $\mathcal{R}(\boldsymbol{h})$ the (closed and convex) achievable rate region for wireless links. With $\mathbb{E}_{\boldsymbol{h}}$ denoting the expectation over fading distribution, then we have on average $\boldsymbol{r} \in \mathbb{E}_{\boldsymbol{h}}[\mathcal{R}(\boldsymbol{h})] := \bar{\mathcal{R}}$.

The network traffic consists of a set of unicast flows from source $s \in S$. Toward its destination, each flow s may travel over a number of wired or wireless links. Denote by $L(s) \subseteq L$ the set of links which flow s goes through, and $S(l) \subseteq S$ the set of flows carried by link l. Denote by x_s the transmission rate of source s. Given a weight vector $\boldsymbol{p} := \{p_s, \forall s\}$, we assign a weighted proportionally-fair utility function $p_s \log x_s$ per flow [4].[1] Then we wish to solve the following

[1] It was shown that p_s is the number of packets maintained by flow s in the buffers along its route at the equilibrium [4], [28]. Hence, the value of p_s can be determined as the target backlog in the network put by flow s.

optimization problem:

$$\begin{aligned} &\max \sum_{s\in S} p_s \log x_s \\ &\text{s. t. (C1):} \quad \sum_{s\in S(l)} x_s \le c_l, \quad \forall l \in L_f \\ &\qquad\text{(C2):} \quad \sum_{s\in S(l)} x_s \le r_l, \quad \forall l \in L_w \\ &\qquad\text{(C3):} \quad \boldsymbol{r} \in \bar{\mathcal{R}}, \quad x_s \ge 0, \ \forall s \in S. \end{aligned} \tag{1}$$

The utility maximization in (1) is of interest since its solution can lead to a fair and efficient equilibrium which does not penalize the flows with large propagation delays [28].

Problem (1) can be readily solved by the NUM paradigm; e.g., a number of queue-length based NUM solutions were available in [7], [31], [32]. On the other hand, plenty of heuristics based TCP enhancements for such a last-hop wireless network were proposed [12], [33], [34]. As will be elaborated, there exists a significant gap between those solutions and desired optimal schemes for mobile TCP applications.

A. NUM SCHEMES

For (1), the primal optimization variables include vector $\boldsymbol{x} := \{x_s, \forall s\}$ of source rates and vector $\boldsymbol{r}$ of wireless-link capacities. Denote by $\boldsymbol{\lambda} := \{\lambda_l, \forall l\}$ the Lagrange multipliers for the link capacity constraints; and define $\lambda^s := \sum_{l\in L(s)} \lambda_l$. Since problem (1) is convex, the globally optimal $\{\boldsymbol{x}^*, \boldsymbol{r}^*\}$ and $\boldsymbol{\lambda}^*$ must satisfy the following Karush-Kuhn-Tucker (KKT) conditions [35]:

$$\frac{p_s}{x_s^*} = \lambda^{s*} := \sum_{l\in L(s)} \lambda_l^*, \quad \forall s \in S \tag{2}$$

$$\lambda_l^* \left(c_l - \sum_{s\in S(l)} x_s^* \right) = 0, \quad \forall l \in L_f \tag{3}$$

$$\sum_{s\in S(l)} x_s^* \le c_l, \quad \forall l \in L_f \tag{4}$$

$$\lambda_l^* \left(r_l^* - \sum_{s\in S(l)} x_s^* \right) = 0, \quad \forall l \in L_w \tag{5}$$

$$\sum_{s\in S(l)} x_s^* \le r_l^*, \quad \forall l \in L_w \tag{6}$$

$$\boldsymbol{r}^* \in \arg\max_{\boldsymbol{r}\in\bar{\mathcal{R}}} \sum_{l\in L_w} \lambda_l^* r_l, \quad \boldsymbol{x}^* \ge 0, \ \ \boldsymbol{\lambda}^* \ge 0. \tag{7}$$

In existing NUM schemes [31], [36], [37], queue lengths of links are used to play the role of scaled Lagrange multipliers, $Q_l \equiv V\lambda_l$. The source-rate controllers, wireless-link scheduler and the queue evolutions are seamlessly ''glued together'' as the decomposition of a sub-gradient type primal-dual iteration to solve the problem. Specifically, with rate control $x_s(t) = Vp_s / \sum_{l\in L(s)} Q_l(t)$ and scheduling policy $\boldsymbol{r}(t) = \arg\max_{\boldsymbol{r}\in\mathcal{R}(\boldsymbol{h}(t))} \sum_{l\in L_w} Q_l(t) r_l$, the queue evolution

$$Q_l(t+1) = \left[Q_l(t) + [\sum_{s\in S(l)} x_s(t) - \{c_l \text{ or } r_l(t)\}] \right]^+$$

then entails a stochastic sub-gradient descent iteration for λ_l to solve (1) with a stepsize $1/V$. Drawing from the stochastic optimization tools [37], asymptotic convergence of these schemes to optimal $\{\boldsymbol{x}^*, \boldsymbol{r}^*, \boldsymbol{\lambda}^*\}$ can be established.

B. WINDOW-BASED TCP ENHANCEMENTS

In TCP congestion control, flow sources adjust their transmission window sizes based on their own (local) congestion observations. Upon receiving an acknowledgement (ACK) packet, the source transmits a new packet when the window size is unchanged; or, it transmits out bulk traffic in bursts when the window size is changed, to maintain that the number of packets in-fly is the same as the current window size. This window-based operation is the key for implementability and reliability of TCP in asynchronous manner and in the presence of feedback delays.

In the standard TCP, an additive-increase-multiplicative-decrease (AIMD) window adjustment scheme based on packet loss is adopted. In this scheme, it is generally believed that the only reason for packet loss is network congestion. This leads to poor performance of TCP over wireless links. As a remedy, some heuristic window adjustment schemes were proposed to increase and decrease window sizes more appropriately [5], [9]–[12], [38]. Yet, none of these schemes is guaranteed to solve an optimization problem such as (1).

It is increasingly acknowledged that TCP relying on packet loss as congestion measure is not stable and efficient, especially for wireless networks. This motivates a number of TCP schemes that employ queueing delays as congestion measure to implement congestion control [8], [11], [12]. In order to achieve the desired optimization, congestion control algorithms in Mo-walrand scheme and Fast-TCP [4], [28] were proposed by using queueing delays as Lagrange multipliers. Those algorithms were analyzed in the network fluid model for *wired* networks (i.e., $L_w = \emptyset$ in (1)). Stability/optimality of the resultant schemes were either proven analytically, or validated by simulations/experiments.

C. THE GAP

In general, both the existing NUM optimization solutions and the enhanced TCP schemes can be regarded as primal-dual solvers for NUM problem (1). However, *queue lengths* act as (scaled) Lagrange multipliers in the former, whereas *queueing delays* play such a role in the latter. Dynamics of both queue lengths and queueing delays have the right scaling with the link capacities; yet, they are by no means equivalent measures. In addition, the primal updates in the NUM are directly performed by source-rate controllers, while they are implicitly affected by the window controllers in TCP designs. There is clearly a gap to address.

The NUM paradigm originated from the reverse engineering of TCP, as its first application was to show that TCP congestion control indeed solves a NUM problem [6]. This paradigm is expected to guide re-engineering the Internet protocols. However, the source-rate controllers proposed in NUM schemes [7] are not confined to the design space of

TCP congestion control. Such controllers need explicit message passing of aggregate queue lengths $\sum_{l\in L(s)} Q_l$ from the network routers, thereby undermining the protocol scalability. In addition, because the mapping from the TCP window size w_s to the source rate x_s may be complicated, the proposed rate control scheme may be difficult to be performed by TCP. Consequently, these issues preclude operating the relevant NUM schemes for TCP applications.

Unlilke queue lengths, the aggregate queueing delay can be locally estimated at the flow source without explicit feedback from the network [3], [4], thereby preserving the scalability of TCP. Based on this congestion measure, window-based congestion controllers in Mo-Walrand scheme and FAST-TCP [4], [28] were developed for the wired networks. In optimization theory, Lagrange multipliers are introduced as auxiliary dual variables to help solve constrained problems. In network optimization (1), mapping these mathematical variables to physical measures is non-unique. From scalability and implementation considerations, it appears that queueing delays can be a better mapping choice for Lagrange multipliers than queue lengths. Relying on this mapping, we can develop *non-standard* and *implicit* primal-dual solvers to perform end-to-end window adjustment in TCP congestion control. To this end, we next generalize Mo-Walrand approach to develop TCP window control oriented network optimization schemes for mobile applications.

III. ALGORITHM DEVELOPMENT

Adopting aggregate queueing delay as the congestion measure at the flow source, we develop the following TCP congestion control and wireless-link scheduling schemes.

Congestion Control: Just like FAST-TCP, our proposed congestion control method at the transport layer is divided into four processes, namely, estimation, window control, data control, and burstiness control [4]. Upon receiving an in-order ACK packet, estimation component at flow source s calculates the current round-trip time (RTT) and updates local values of $AvgRTT_s$ and $BaseRTT_s$, where $BaseRTT_s$ is given by the minimum RTT observed so far to approximate the propagation (plus processing) delay, and average RTT $AvgRTT_s$ is updated according to the current RTT_s as:

$$AvgRTT_s \leftarrow \frac{255}{256} \times AvgRTT_s + \frac{1}{256} \times RTT_s. \tag{8}$$

Based on $AvgRTT_s$ and $BaseRTT_s$, window control component adjusts the transmission window size w_s:

$$w_s \leftarrow w_s - \kappa \frac{BaseRTT_s}{AvgRTT_s} w_s^{-2\rho+1} \left(w_s - \frac{BaseRTT_s}{AvgRTT_s} w_s - p_s\right) \tag{9}$$

where κ is a positive stepsize, and $\rho \in [0, 1]$ is a constant parameter. Then data control component decides which packets to be transmitted, whereas burstiness control component decides when to transmit [4]. When an ACK packet is lost or not received in order, the slow start or fast recovery algorithm in current TCP will be used to handle transmission time-outs or duplicate ACKs.

Wireless-Link Scheduling: The AP coordinates slotted transmission over wireless links. At slot n, it reads the queue length $QueLEN_l[n]$ and computes the average rate $AveR_l[n]$ using a similar low-pass filter as with (8) per wireless link l. Then the scheduler performs a policy so that

$$\boldsymbol{r}^*[n] = \arg \max_{\boldsymbol{r}\in\mathcal{R}(\boldsymbol{h}[n])} \sum\nolimits_{l\in L_w} \frac{QueLEN_l[n]}{AveR_l[n]} r_l. \tag{10}$$

In particular, for a time-division downlink or uplink, let $R_l[n]$ be the maximum rate which link l can achieve at slot n, $\forall l \in L_w$. Then the strategy (10) is just to select the link with the highest $\frac{QueLEN_l[n]}{AveR_l[n]} R_l[n]$ for transmissions per slot. This scheduler in fact performs a modified largest-weighted-delay-first (M-LWDF) strategy [39].

Stemmed from a cross-layer design, the proposed solutions could be deployed with the current layered network infrastructure without difficulty. The congestion control (9) does not violate the distributed end-to-end mechanism of TCP, whereas the scheduler (10) can be readily modified from existing scheduling scheme at wireless AP. Working together, these two schemes can also approach the optimal solution of the problem (1), as will be shown in the sequel.

A. FLUID MODEL OF NETWORK

We assume a network fluid model in which data packets can be divided indefinitely [28]. Let $\boldsymbol{w} := \{w_s, \forall s\}$ and $\boldsymbol{d} := \{d_s, \forall s\}$, where w_s denotes the window size, and d_s denotes the fixed round-trip propagation (plus processing) delay for flow $s \in S$. Let $\boldsymbol{q} := \{q_l, \forall l\}$ collect the round-trip queueing delays q_l for links, $\forall l \in L$; and define the aggregate queueing delay $q^s := \sum_{l\in L(s)} q_l$ along the routes for flow s. Together with source rates x_s, we have the following relationships:

$$x_s(d_s + q^s) = w_s, \quad \forall s \in S \tag{11}$$

$$q_l\left(c_l - \sum_{s\in S(l)} x_s\right) = 0, \quad \forall l \in L_f \tag{12}$$

$$\sum_{s\in S(l)} x_s \le c_l, \quad \forall l \in L_f \tag{13}$$

$$q_l\left(r_l - \sum_{s\in S(l)} x_s\right) = 0, \quad \forall l \in L_w \tag{14}$$

$$\sum_{s\in S(l)} x_s \le r_l, \quad \forall l \in L_w \tag{15}$$

$$\boldsymbol{r} \in \bar{\mathcal{R}}, \ \boldsymbol{x} \ge 0, \quad \boldsymbol{q} \ge 0 \tag{16}$$

where (11) is due to the fact that the flow rate x_s is equal to the window size w_s divided by the total RTT d_s+q^s of flow s; (13) and (15) are due to link capacity constraints; (12) and (14) are due to the the following reasons: because the data packet can be divided indefinitely (i.e., infinitely small), when the total rate through the link l is less than its capacity c_l or r_l, the queue size (and queueing delay) on this link is zero.

Assume that the AP can accurately maintain the values of average rates $AveR_l[n]$ per wireless link. It follows from the Little's law that the average queueing delay is equal to

average queue length divided by average rate. In the scheduling policy (10), the "instantaneous" delay, given by the ratio of current queue length $QueLEN_l[n]$ and the average rate $AveR_l[n]$, is used as the weight for user rate r_l. By taking appropriate time scaling, a fluid limit argument can be applied such that the "stochastic" delay $\frac{QueLEN_l[n]}{AveR_l[n]}$ can take the place of the average delay q_l in the so-called fluid model [31]. The scheduling policy (10) then in turn amounts to a average queueing-delay based one; i.e., it maximizes the sum of delay-capacity products with all links per $\boldsymbol{h}$:

$$\boldsymbol{r}^*(\boldsymbol{h}) \in \arg\max_{\boldsymbol{r}\in\mathcal{R}(\boldsymbol{h})} \sum_{l\in L_w} q_l r_l.$$

Based on this scheduler, we then have:

$$\boldsymbol{r}^* := \mathbb{E}_{\boldsymbol{h}}\left[\boldsymbol{r}^*(\boldsymbol{h})\right] \in \arg\max_{\boldsymbol{r}\in\bar{\mathcal{R}}} \sum_{l\in L_w} q_l r_l. \tag{17}$$

Also assume that $BaseRTT_s$ and $AvgRTT_s$ can precisely approximate the propagation delay d_s and total RTT $\bar{d}_s := d_s + q^s$. The window update (9) then amounts to:

$$dw_s = -\kappa \frac{d_s}{\bar{d}_s} w_s^{-2\rho+1}(w_s - d_s\frac{w_s}{\bar{d}_s} - p_s).$$

Noting $x_s = w_s/\bar{d}_s$ from (11), define $v_s := w_s - x_s d_s - p_s$ and $\boldsymbol{v} := \{v_s, \forall s\}$. The window adjustments indeed correspond to an ordinary differential equation per flow s in the fluid model:

$$\frac{d}{dt}w_s(t) = -\kappa\frac{d_s}{\bar{d}_s} w_s^{-2\rho+1} v_s, \quad \forall s. \tag{18}$$

Our TCP congestion control scheme then amounts to adjusting the window size w_s with (18), which in turn controls the source-rate vector $\boldsymbol{x}$, wireless-link capacity vector $\boldsymbol{r}$, and queueing delay vector $\boldsymbol{q}$ via (11)–(17). Notice that the window update (18) in fact controls the queueing of the flows; hence, we name the scheme QUIC-TCP with $0 \le \rho \le 1$. With $\rho = 1$, the algorithm becomes the Mo-Walrand's scheme [28]: $\frac{d}{dt}w_s(t) = -\kappa\frac{d_s}{\bar{d}_s}\frac{v_s}{w_s}$. For $\rho = 1/2$, it corresponds to $\frac{d}{dt}w_s(t) = -\kappa\frac{d_s}{\bar{d}_s}v_s$, which resembles FAST-TCP [4]: $\frac{d}{dt}w_s(t) = -\kappa v_s$.

The proposed scheduler (17) is in fact a MaxWeight-type one [7], [32], [36]; yet, instead of queue lengths, here queueing delays are used as the weights for transmission rates over links. This queueing-delay based scheduling (17) is readily implied by the KKT condition (7), due to the fact that queueing delays play the role of Lagrange multipliers in the proposed schemes. Interestingly, we will show that such a scheduling policy and window control strategy (18) are actually "glued" together to construct a primal-dual solver for (1). Note that the proposed MaxWeight-scheduler (17) is easy to implement for the one-hop wireless networks under consideration, where an AP exists to compute the weights for all its links and make the scheduling decision.

B. THE MAPPING $F : \boldsymbol{w} \to (\boldsymbol{x}, \boldsymbol{q})$

The relationships (11)–(17) actually determine a mapping from the window sizes to the flow rates and queueing delays: $F : \boldsymbol{w} \to (\boldsymbol{x}, \boldsymbol{q})$ in TCP congestion control. To analyze the proposed schemes, it is important to specify this mapping. To this end, consider a convex optimization problem:

$$\max_{\boldsymbol{x},\boldsymbol{r}} \sum_{s\in S}(w_s \log x_s - d_s x_s), \quad \text{s. t. (C1)–(C3) of (1).} \tag{19}$$

Careful examination can tell that (11)–(17) are equivalent to the KKT conditions for the problem (19) [cf. (2–(7)], with $\boldsymbol{q}$ acting as Lagrange multipliers for the constraints (C1) and (C2). Hence, the mapping F can be determined by the solutions of (19) and its dual problem.

As with [28], we define that a link is a "bottleneck" if the sum-rate over this link matches its capacity. Denote by $B(\boldsymbol{w})$ the set of bottleneck links for a given $\boldsymbol{w}$. In addition, $\boldsymbol{w}$ is called an "interior point" if there exists an $\epsilon > 0$ so that $B(\boldsymbol{w}')$ remains unchanged, $\forall \boldsymbol{w}' \in N_\epsilon(\boldsymbol{w})$ neighborhood of $\boldsymbol{w}$. All other $\boldsymbol{w}$ are called "boundary points". Capitalizing on that the mapping F can be determined by the solutions of (19), we then prove the following properties of $\boldsymbol{w} \to (\boldsymbol{x}, \boldsymbol{q})$ in the Appendix A:

1) The source rate vector $\boldsymbol{x}(\boldsymbol{w})$ is a continuous function of $\boldsymbol{w}$, and this function is differentiable except at the boundary points.
2) The source rate x_s is a non-decreasing function of its own w_s if all other $w_{s'}$, $\forall s' \neq s$, remain fixed; in addition, when $\boldsymbol{x}(\boldsymbol{w})$ is differentiable, we must have $0 \le \frac{\partial x_s}{\partial w_s} \le \frac{1}{d_s}$, $\forall s$.
3) The aggregate queueing delay vector $\boldsymbol{q}^s(\boldsymbol{w}) := \{q^s(\boldsymbol{w}), \forall s\}$ is a continuous function of $\boldsymbol{w}$, and it is differentiable except at the boundary points; when $\boldsymbol{q}^s(\boldsymbol{w})$ is differentiable, we have $0 \le \frac{\partial q^s}{\partial w_s} \le \frac{1}{x_s}$, $\forall s$.

Property 1) was proven for wired networks in [28, Claims 3 and 4]. A similar proof can be used to prove it for last-hop wireless networks here, provided that the AP performs queueing-delay based scheduling. Properties 2) and 3) are new. Property 2) can be derived by examining the partial first derivative $\partial x_s/\partial w_s$, whereas property 3) is a consequence of property 2) and equation (11).

Property 2) states that a source s could increase its rate x_s by increasing its transmission window size w_s, as all other $w_{s'}$, $\forall s' \neq s$, remain fixed. The marginal increase $\partial x_s/\partial w_s$ is at most $1/d_s$ which occurs when its aggregate queueing delay $q^s = 0$ [cf. (11)], i.e., the network is under-loaded. Note that although the aggregate queueing delay vector $\boldsymbol{q}^s$ is uniquely determined for a given $\boldsymbol{w}$, the queueing delay vector $\boldsymbol{q}$ is not necessarily unique, unless the number of independent bottleneck links is equal to the number of sources such that $\boldsymbol{q}$ can be solved from $\boldsymbol{q}^s$. It follows from property 3) that aggregate queueing delay q^s increases as w_s increases, and its marginal increase cannot exceed $1/x_s$. These properties are useful for the analysis in the sequel.

C. OPTIMAL NETWORK EQUILIBRIUM

At equilibrium of window-size adjustment (18), we clearly have $v_s = 0$, $\forall s$. Let $\boldsymbol{w}^*$ stand for the window-size vector at this equilibrium. Correspondingly, $\boldsymbol{x}^*$, $\boldsymbol{r}^*$ and $\boldsymbol{q}^*$ stand

for the source-rate vector, wireless-link capacity vector, and queueing-delay vector yielded by $\boldsymbol{w}^*$.

Theorem 1: For the proposed joint design, we have a unique window-size vector $\boldsymbol{w}^*$ to make $\boldsymbol{v} = 0$; and the rate vector $\boldsymbol{x}^* = \boldsymbol{x}(\boldsymbol{w}^*)$ is an optimal solution to problem (1).

Proof: See Appendix B. □

The sketches of the proof for Theorem 1 (as well as Lemma 2, Lemma 3, and Theorem 3) appeared in [29, Chapter 2]; we include them in the Appendix to make this paper self-contained. Theorem 1 demonstrates that by forcing $v_s = 0, \forall s$, with the proposed QUIC-TCP window control, one can approach the optimal source rate vector $\boldsymbol{x}^*$ for (1). In other words, the absolute value of v_s can indicate the distance from optimality per flow source s. Recall that the value of v_s can be obtained with only local observations ($BaseRTT_s$ and $AvgRTT_s$); hence, $v_s = 0$ can serve as a decoupled optimality criterion for the flow source s. Together with the link-scheduling policy (17), window control (18) constructs a negative feedback system towards $\boldsymbol{w}^*$. For a large w_s with $v_s = w_s - x_s d_s - p_s > 0$, we have a negative dw_s/dt to enforce decreasing of the window size which in turn decreases v_s towards zero; and vice verse. The extra factor $\frac{d_s}{\bar{d}_s} w_s^{-2\rho+1}$ in (18) is to ensure the convergence of such a system to the desired equilibrium.

Relying on only local observations without support of routers, the window control (18) and link-scheduling policy (17) entail *implicit primal-dual* updates of $\{\boldsymbol{x}, \boldsymbol{r}, \boldsymbol{q}\}$ to solve (1). This is different from the gradient-type *explicit primal-dual* solutions in the NUM schemes [31], [32], which are hard to implement by the TCP. On the other hand, different from the heuristic TCP enhancements without analytical performance guarantees [12], [33], [34], we next establish convergence of the proposed QUIC-TCP to optimal equilibrium via a Lyapunov method.

IV. PERFORMANCE ANALYSIS

To show the global convergence of (18), we first characterize the capacity region $\bar{\mathcal{R}}$ for wireless links. Rewrite the problem (1) into a matrix form:

$$\begin{aligned} &\max \sum_{s\in S} p_s \log x_s \\ &\text{s. t. } \boldsymbol{A}_f \boldsymbol{x} \le \boldsymbol{c}, \quad \boldsymbol{A}_w \boldsymbol{x} \le \boldsymbol{r}, \\ &\qquad \boldsymbol{r} \in \bar{\mathcal{R}}, \quad \boldsymbol{x} \ge 0 \end{aligned} \tag{20}$$

where $\boldsymbol{c} := \{c_l, \forall l \in L_f\}$ and the routing matrix $\boldsymbol{A} := [\boldsymbol{A}_f^T, \boldsymbol{A}_w^T]^T$ with its (l,s)th entry $A_{ls} = 1$ if $s \in S(l)$ and $A_{ls} = 0$ otherwise.

For the closed and convex capacity region $\bar{\mathcal{R}}$, we can establish that:

Lemma 1: We can delineate the capacity region as $\bar{\mathcal{R}} = \{\boldsymbol{r} \ge 0 \mid f(\boldsymbol{r}) \le 0\}$ where $f(\boldsymbol{r})$ is a convex function.

Proof: Given any convex $\bar{\mathcal{R}}$, we can always define an indicator-alike function:

$$f(\boldsymbol{r}) = \begin{cases} 0, & \text{if } \boldsymbol{r} \in \bar{\mathcal{R}}, \\ \infty, & \text{otherwise.} \end{cases}$$

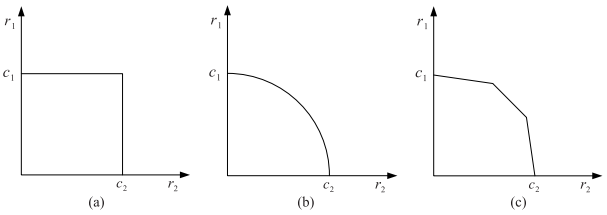


FIGURE 1. Typical capacity regions.

Since $\bar{\mathcal{R}}$ is convex, we can show that this $f(\boldsymbol{r})$ is a convex function. Actually such a function is not unique in many cases. Here since $\bar{\mathcal{R}}$ is closed, we can also define

$$f(\boldsymbol{r}) = \text{dist}(\boldsymbol{r}, \bar{\mathcal{R}})$$

where $\text{dist}(\boldsymbol{r}, \bar{\mathcal{R}})$ calculates the distance between $\boldsymbol{r}$ and the region $\bar{\mathcal{R}}$ [35, p. 103]. It can be shown that this $f(\boldsymbol{r})$ is continuous and convex. □

There are possibly many choices of $f(\boldsymbol{r})$. What we are interested in are the ones that are continuous and differentiable. For most of the capacity regions with wireless links, such functions with the desired properties can be readily found. Some typical capacity regions are shown in Fig. 1. If every wireless link is given a dedicated channel with capacity c_l by the AP using e.g., TDMA or FDMA, the capacity region is a box region in Fig. 1(a), for which we have:

$$f(\boldsymbol{r}) = ||\boldsymbol{C}^{-1}\boldsymbol{r}||_\infty - 1 = \left(\sum\nolimits_l |r_l/c_l|^\infty\right)^{\frac{1}{\infty}} - 1$$

where $||\cdot||_\infty$ denotes the (convex) infinity norm of vector. For the smooth capacity region in Fig. 1(b), some differentiable $f(\boldsymbol{r})$ can be defined; while a convex piece-wise linear $f(\boldsymbol{r})$ can be found for the region in Fig. 1(c).

A. DEDICATED CHANNEL CASE

We first show the convergence of QUIC-TCP (18) for the case that each wireless link is given a dedicated TDMA/FDMA channel with capacity c_l by the AP; see Fig. 1(a). In this case, wireless links can be actually treated as "wired" links since they have decoupled and constant ergodic capacities; i.e., we have $r_l \equiv c_l$ in (20), $\forall l \in L_w$, regardless of the scheduling scheme in use. The problem (20) then reduces to

$$\sum\nolimits_{s\in S} p_s \log x_s, \quad \text{s. t. } \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{c}, \quad \boldsymbol{x} \ge 0. \tag{21}$$

This is the same problem that was considered for wired networks in [4], [6], [8], [28]. As the wireless-link scheduling becomes irrelevant, the existence and uniqueness of optimal window vector $\boldsymbol{w}^*$ in Theorem 1 holds under the scheduler (17) or any other schedulers. For this simple case, we prove the convergence of QUIC-TCP algorithms (18) to $\boldsymbol{w}^*$ in the following theorem:

Theorem 2: When each wireless link is given a dedicated channel, the window adjustment scheme (18) with $\rho \in (0, 1]$ can globally converge to the equilibrium point $\boldsymbol{w}^*$, and the corresponding flow rates converge to $\boldsymbol{x}^*$.

Proof: Denote by B the set of bottleneck links for a given $\boldsymbol{w}$, $\boldsymbol{A}_B$ the sub-matrix of $\boldsymbol{A}$ obtained by keeping only

the rows that correspond to bottleneck links, and $\boldsymbol{q}_B$ and $\boldsymbol{c}_B$ the corresponding sub-vector of $\boldsymbol{q}$ and sub-vector of $\boldsymbol{c}$ for bottleneck links.

Construct the matrices $\boldsymbol{W} := \text{diag}(\boldsymbol{w})$, $\boldsymbol{X} := \text{diag}(\boldsymbol{x})$, $\boldsymbol{D} := \text{diag}(\boldsymbol{d})$, and $\bar{\boldsymbol{D}} := \text{diag}(\bar{\boldsymbol{d}})$ with $\bar{\boldsymbol{d}} := \{\bar{d}_s, \forall s\}$. Recall that non-bottleneck links should have zero queueing delays. Using the latter diagonal matrices, we re-express (11) as

$$\boldsymbol{X}(\boldsymbol{A}_B^T\boldsymbol{q}_B + \boldsymbol{d}) = \boldsymbol{w}. \tag{22}$$

For an interior point $\boldsymbol{w}$, both $\boldsymbol{x}(\boldsymbol{w})$ and $\boldsymbol{q}^s(\boldsymbol{w})$ are differentiable. Notice that $\boldsymbol{A}_B^T\boldsymbol{q}_B + \boldsymbol{d} = \bar{\boldsymbol{d}}$. Denote by $\boldsymbol{J}_{\boldsymbol{a}|\boldsymbol{b}} := \{\frac{\partial a_i}{\partial b_j}, \forall i, j\}$ the Jacobian matrix of vector $\boldsymbol{a}$ over vector $\boldsymbol{b}$. Differentiating both sides of (22) with respect to $\boldsymbol{w}$, we have:

$$\bar{\boldsymbol{D}}\boldsymbol{J}_{\boldsymbol{x}|\boldsymbol{w}} + \boldsymbol{X}\boldsymbol{A}_B^T\boldsymbol{J}_{\boldsymbol{q}_B|\boldsymbol{w}} = \boldsymbol{I} \tag{23}$$

where $\boldsymbol{I}$ denotes identity matrix. Multiplying both sides of (23) by $\boldsymbol{A}_B\bar{\boldsymbol{D}}^{-1}$, we further have:

$$\boldsymbol{A}_B\boldsymbol{J}_{\boldsymbol{x}|\boldsymbol{w}} + \boldsymbol{A}_B\bar{\boldsymbol{D}}^{-1}\boldsymbol{X}\boldsymbol{A}_B^T\boldsymbol{J}_{\boldsymbol{q}_B|\boldsymbol{w}} = \boldsymbol{A}_B\bar{\boldsymbol{D}}^{-1}. \tag{24}$$

But since it holds for the bottleneck links that $\boldsymbol{A}_B\boldsymbol{x} = \boldsymbol{c}_B$, we also have $\boldsymbol{A}_B\boldsymbol{J}_{\boldsymbol{x}|\boldsymbol{w}} = 0$. Substituting this into (24), we can obtain: $\boldsymbol{J}_{\boldsymbol{q}_B|\boldsymbol{w}} = (\boldsymbol{A}_B\bar{\boldsymbol{D}}^{-1}\boldsymbol{X}\boldsymbol{A}_B^T)^{-1}\boldsymbol{A}_B\bar{\boldsymbol{D}}^{-1}$. Substituting this term into (23) yields:

$$\boldsymbol{J}_{\boldsymbol{x}|\boldsymbol{w}} = \bar{\boldsymbol{D}}^{-1}(\boldsymbol{I} - \boldsymbol{X}\boldsymbol{A}_B^T(\boldsymbol{A}_B\bar{\boldsymbol{D}}^{-1}\boldsymbol{X}\boldsymbol{A}_B^T)^{-1}\boldsymbol{A}_B\bar{\boldsymbol{D}}^{-1}). \tag{25}$$

Using the convenient notation

$$\boldsymbol{M} := \boldsymbol{A}_B^T(\boldsymbol{A}_B\bar{\boldsymbol{D}}^{-1}\boldsymbol{X}\boldsymbol{A}_B^T)^{-1}\boldsymbol{A}_B,$$

we have $\boldsymbol{J}_{\boldsymbol{x}|\boldsymbol{w}} = \bar{\boldsymbol{D}}^{-1}(\boldsymbol{I} - \boldsymbol{X}\boldsymbol{M}\bar{\boldsymbol{D}}^{-1})$. Clearly, here $\boldsymbol{M}$ is positive semi-definite.

For the window control (18), construct a function

$$Y(\boldsymbol{w}) = \frac{1}{2}\sum_{s\in S}\left(\frac{v_s}{(w_s)^\rho}\right)^2. \tag{26}$$

For an interior point $\boldsymbol{w}$, we have:

$$\begin{aligned}
&\frac{d}{dt}Y(\boldsymbol{w}(t)) \\
&= \sum_s\left(\frac{\partial Y}{\partial w_s}\frac{dw_s(t)}{dt}\right) \\
&= \sum_s\left[\left(\sum_{s'}\left(\frac{v_{s'}}{w_{s'}^\rho}\frac{\partial(v_{s'}/w_{s'}^\rho)}{\partial w_s}\right)\right)\frac{dw_s(t)}{dt}\right] \\
&= -\kappa\boldsymbol{v}^T\boldsymbol{W}^{-\rho}[\boldsymbol{W}^{-\rho}\boldsymbol{J}_{\boldsymbol{v}|\boldsymbol{w}} - \rho\boldsymbol{W}^{-\rho-1}\boldsymbol{V}]\boldsymbol{D}\bar{\boldsymbol{D}}^{-1}\boldsymbol{W}^{-2\rho+1}\boldsymbol{v} \\
&= -\kappa\boldsymbol{v}^T[\boldsymbol{W}^{-2\rho}(\boldsymbol{I} - \boldsymbol{D}\boldsymbol{J}_{\boldsymbol{x}|\boldsymbol{w}}) - \rho\boldsymbol{W}^{-2\rho-1} \\
&\quad\times(\boldsymbol{W} - \boldsymbol{D}\bar{\boldsymbol{D}}^{-1}\boldsymbol{W} - \boldsymbol{P})]\boldsymbol{D}\bar{\boldsymbol{D}}^{-1}\boldsymbol{W}^{-2\rho+1}\boldsymbol{v} \\
&= -\kappa\boldsymbol{v}^T[\boldsymbol{W}^{-2\rho}(\boldsymbol{I} - \boldsymbol{D}\bar{\boldsymbol{D}}^{-1}(\boldsymbol{I} - \boldsymbol{X}\boldsymbol{M}\bar{\boldsymbol{D}}^{-1})) - \rho\boldsymbol{W}^{-2\rho-1} \\
&\quad\times(\boldsymbol{W} - \boldsymbol{D}\bar{\boldsymbol{D}}^{-1}\boldsymbol{W} - \boldsymbol{P})]\boldsymbol{D}\bar{\boldsymbol{D}}^{-1}\boldsymbol{W}^{-2\rho+1}\boldsymbol{v} \\
&= -\kappa\boldsymbol{v}^T[(1-\rho)\boldsymbol{W}^{-2\rho}(\boldsymbol{I} - \boldsymbol{D}\bar{\boldsymbol{D}}^{-1})\boldsymbol{D}\bar{\boldsymbol{D}}^{-1}\boldsymbol{W}^{-2\rho+1} \\
&\quad+ \rho\boldsymbol{W}^{-2\rho-1}\boldsymbol{P}\boldsymbol{D}\bar{\boldsymbol{D}}^{-1}\boldsymbol{W}^{-2\rho+1} \\
&\quad+ \boldsymbol{W}^{-2\rho+1}\boldsymbol{D}\bar{\boldsymbol{D}}^{-2}\boldsymbol{M}\bar{\boldsymbol{D}}^{-2}\boldsymbol{D}\boldsymbol{W}^{-2\rho+1}]\boldsymbol{v}
\end{aligned} \tag{27}$$

where we define diagonal matrices $\boldsymbol{V} := \text{diag}(\boldsymbol{v})$ and $\boldsymbol{P} := \text{diag}(\boldsymbol{p})$ for convenience. Recall that $\boldsymbol{M}$ is positive semi-definite, $(\boldsymbol{I} - \boldsymbol{D}\bar{\boldsymbol{D}}^{-1})$ is diagonal with nonnegative entries, all $\boldsymbol{W}$, $\boldsymbol{D}$, $\bar{\boldsymbol{D}}$ and $\boldsymbol{P}$ are also diagonal with positive diagonal entries. It then readily follows that the whole matrix inside the square bracket of (27) is positive definite, $\forall\rho \in (0, 1]$. This implies that $dY(\boldsymbol{w}(t))/dt < 0$; in other words, $Y(\boldsymbol{w}(t))$ is strictly decreasing in t, at all interior points unless $\boldsymbol{v} = 0$.

At the boundary points, for any arbitrary direction d [28, Corollary 3], the right-hand directional derivative of $\boldsymbol{x}(\boldsymbol{w})$ can be always easily defined; so we can extend the definition of $\boldsymbol{J}_{\boldsymbol{x}|\boldsymbol{w}}$ as a function of direction d. Using this extended definition, we can mimic the proof of [28, Theorem 5] to show that $Y(\boldsymbol{w}(t))$ is also strictly decreasing in t at boundary points unless $\boldsymbol{v} = 0$.

Now it is proven that $Y(\boldsymbol{w}(t))$ is a Lyapunov function, which is nonnegative and unbounded, and has globally negative time derivative. As a result, the unique equilibrium $\boldsymbol{v} = 0$ of the system (18) is globally stable asymptotically [40]; this implies that (18) globally converges to its unique equilibrium $\boldsymbol{w}^*$, and the corresponding flow rates converge to $\boldsymbol{x}^*$. □

Theorem 2 holds for the class of QUIC-TCP schemes (18), $\forall\rho \in (0, 1]$. It is actually a generalization of [28, Theorem 5], which proves the convergence of Mo-Walrand scheme, i.e., (18) with $\rho = 1$. An intriguing case is the QUIC-TCP (18) with $\rho = 0$, leading to $\frac{d}{dt}w_s(t) = -\kappa\frac{d_s}{\bar{d}_s}\frac{v_s}{w_s}$, $\forall s$. Relying on a Lyapunov function $Y(\boldsymbol{w}) = (1/2)\sum_{s\in S}(v_s)^2$, we can show that $\frac{d}{dt}Y(t) = -\kappa\boldsymbol{v}^T[(\boldsymbol{I} - \boldsymbol{D}\bar{\boldsymbol{D}}^{-1})\boldsymbol{D}\boldsymbol{X} + \boldsymbol{D}\bar{\boldsymbol{D}}^{-1}\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}\bar{\boldsymbol{D}}^{-1}\boldsymbol{D}]\boldsymbol{v}$. In this case, the matrix inside the bracket is positive semi-definite but it is not always guaranteed to be positive definite; the equilibrium $\boldsymbol{v} = 0$ is then stable yet not asymptotically stable.

B. SMOOTH CAPACITY REGION CASE

From an information-theoretic viewpoint, downlink and uplink with dedicated TDMA/FDMA channels are often strictly sub-optimal. In wireless standards, the high-speed data transmissions are actually carried out over a physical channel shared by all logical wireless links and an adaptive scheduling algorithm is performed by the AP. In this case, the capacity region $\bar{\mathcal{R}}$ is no longer a box region and the capacities of wireless links are coupled. We now consider a smooth capacity region as in Fig. 1(b). For such a region, a strictly convex and twice differentiable $f(\boldsymbol{r})$ can be found to delineate the boundary of $\bar{\mathcal{R}}$.

In this case, AP needs to preform the scheduler (17) for transmissions over coupled wireless links. Let $\boldsymbol{q}_w := \{q_l, \forall l \in L_w\}$. With ∇f denoting the gradient of f, we can prove the following lemma for the scheduling policy (17):

Lemma 2: With a strictly convex and twice differentiable $f(\boldsymbol{r})$, we have $\boldsymbol{r}^* = \arg\max_{\boldsymbol{r}\in\bar{\mathcal{R}}}\sum_{l\in L_w}q_l r_l$ iff $f(\boldsymbol{r}^*) = 0$ and $\theta\nabla f(\boldsymbol{r}^*) = \boldsymbol{q}_w$ for a certain constant $\theta > 0$.

Proof: See Appendix C. □

Lemma 2 states that achieved $\boldsymbol{r}^*$ by the scheduler (17) indeed resides on the boundary of $\bar{\mathcal{R}}$ (i.e., $f(\boldsymbol{r}^*) = 0$) and

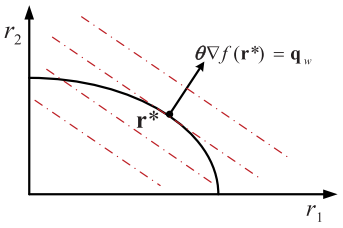


FIGURE 2. Geometric interpretation of Lemma 2.

the gradient of f at this point has the same direction with the queueing-delay vector $\boldsymbol{q}_w$ (i.e., $\theta\nabla f(\boldsymbol{r}^*) = \boldsymbol{q}_w$); see an illustration in Fig. 2.

Based on Lemma 2, it can be also shown that

Lemma 3: With the scheduling policy (17), the Jacobian matrix $\boldsymbol{J}_{\boldsymbol{r}^*|\boldsymbol{q}_w}$ is positive definite.

Proof: See Appendix D. □

Building on Lemma 3, it is ready to establish that:

Theorem 3: For the capacity region $\bar{\mathcal{R}} = \{\boldsymbol{r} \mid \boldsymbol{r} \geq 0,\ f(\boldsymbol{r}) \leq 0\}$ with a strictly convex and twice differentiable $f(\boldsymbol{r})$, the window adjustment (18), $\forall \rho \in (0, 1]$ together with the scheduling plicy (17) can converge to the unique equilibrium $\boldsymbol{w}^*$, and the corresponding flow rates converge to $\boldsymbol{x}^*$.

Proof: See Appendix E. □

Theorem 3 shows the existence of optimal end-to-end window-based schemes for network congestion control over coupled wireless links. This becomes possible with the help of the queueing-delay based scheduler (17) performed at the wireless AP. With this scheduler, the wireless links are coupled in a nice manner so that the Jacobian matrix $\boldsymbol{J}_{\boldsymbol{r}^*|\boldsymbol{q}_w}$ is positive definite, leading to global stability of QUIC-TCP.

C. DISCUSSIONS ON OTHER CASES

When only a few levels of rate and/or power adaptations are supported by the wireless standards, the capacity region $\bar{\mathcal{R}}$ may have a piece-wise linear boundary as in Fig. 1(c). Such a region is defined by the intersection of the non-negative orthant and a number of half-spaces:

$$\boldsymbol{a}_i^T\boldsymbol{r} - b_i \leq 0, \quad i = 1, \ldots, I, \tag{28}$$

where $\boldsymbol{a}_i$ is the (L_w-dimensional) normal vector and the scalar b_i determines the offset from the origin for the ith half-space. The piece-wise linear boundary of $\bar{\mathcal{R}}$ (i.e., $f(\boldsymbol{r}) = 0$) is then dictated by the hyperplanes $\{\boldsymbol{r} \mid \boldsymbol{a}_i^T\boldsymbol{r} - b_i = 0\}$. Suppose that there exist N corner points for this boundary surface, where a corner point $\boldsymbol{z}_n$ is the intersection of L_w hyperplanes with the half-spaces (28). For the nth corner point, let the set $\mathcal{A}(n)$ collect its associated L_w normal vectors $\boldsymbol{a}_i$, and $\mathcal{C}(\mathcal{A}(n))$ denote the conic hull of all the vectors $\boldsymbol{a}_i \in \mathcal{A}(n)$.

Claim 1: For a piece-wise linear $f(\boldsymbol{r})$, the optimal rate vector $\boldsymbol{r}^* = \arg\max_{\boldsymbol{r}\in\bar{\mathcal{R}}}\ \boldsymbol{q}_w^T\boldsymbol{r}$ is given by

$$\boldsymbol{r}^* = \boldsymbol{z}_n, \quad \text{if} \quad \boldsymbol{q}_w \in \mathcal{C}(\mathcal{A}(n)).$$

Proof: For a corner point $\boldsymbol{z}_n$, we must have:

$$f(\boldsymbol{r}) \geq f(\boldsymbol{z}_n) + \boldsymbol{a}_i^T(\boldsymbol{r} - \boldsymbol{z}_n), \quad \forall \boldsymbol{a}_i \in \mathcal{A}(n)$$

since $f(\boldsymbol{r})$ is convex. This implies that for all $\boldsymbol{r} \in \bar{\mathcal{R}}$.

$$\boldsymbol{a}_i^T(\boldsymbol{z}_n - \boldsymbol{r}) \geq f(\boldsymbol{z}_n) - f(\boldsymbol{r}) \geq 0, \quad \forall \boldsymbol{a}_i \in \mathcal{A}(n);$$

where the last inequality holds due to $f(\boldsymbol{z}_n) = 0$ and $f(\boldsymbol{r}) \leq 0$, $\forall \boldsymbol{r} \in \bar{\mathcal{R}}$.

Now if $\boldsymbol{q}_w \in \mathcal{C}(\mathcal{A}(n))$, we have $\boldsymbol{q}_w = \sum_{i:\boldsymbol{a}_i\in\mathcal{A}(n)} \theta_i\boldsymbol{a}_i$ for some $\theta_i \geq 0$ by the definition of conic hull [35]. Therefore, it follows that

$$\begin{aligned}\boldsymbol{q}_w^T(\boldsymbol{z}_n - \boldsymbol{r}) &= \sum_{i:\boldsymbol{a}_i\in\mathcal{A}(n)} \theta_i\boldsymbol{a}_i^T(\boldsymbol{z}_n - \boldsymbol{r}) \\ &\geq \sum_{i:\boldsymbol{a}_i\in\mathcal{A}(n)} \theta_i(f(\boldsymbol{z}_n) - f(\boldsymbol{r})) \geq 0.\end{aligned}$$

In other words, $\boldsymbol{z}_n = \arg\max_{\boldsymbol{r}\in\bar{\mathcal{R}}}\ \boldsymbol{q}_w^T\boldsymbol{r}$. □

Note that a corner point $\boldsymbol{z}_n$ can be also the intersection of $L_w - F$ hyperplanes with the half-spaces (28) and F facets of the non-negative orthant. In this case, we let $\mathcal{A}(n)$ collect the associated $L_w - F$ normal vectors $\boldsymbol{a}_i$ and their projects on the F facets of the non-negative orthant. Then it can be similarly shown that Claim 1 still holds.

Given a wireless capacity region with a piece-wise linear boundary as in Fig. 1(c), clearly the optimal $\boldsymbol{r}^*$ is always attained at one of the corner points for any non-negative queueing delay vector $\boldsymbol{q}_w$. The Lyapunov approach in the proof of Theorem 3 cannot be generalized to this case since $\boldsymbol{r}^*$ is non-differentiable with respect to $\boldsymbol{q}_w$ at all the points $\boldsymbol{q}_w = \boldsymbol{a}_i$. We argue that convergence of the proposed schemes (18) is also expected in this case as follows. Note that for a normal vector $\boldsymbol{a}_i := \{a_{i,l}, l = 1, \ldots, L_w\}$, the portion of the linear function $\boldsymbol{a}_i^T\boldsymbol{r} - b_i$ in the non-negative orthant coincides with that of a l_1 norm: $\sum_l(a_{i,l}r_l) - b_i$. With a small ϵ, we replace the linear function $\boldsymbol{a}_i^T\boldsymbol{r} - b_i$ using a $l_{1+\epsilon}$ norm: $\left(\sum_l(a_{i,l}r_l)^{1+\epsilon}\right)^{\frac{1}{1+\epsilon}} - b_i$. Imagine that the capacity region $\bar{\mathcal{R}}$ becomes the intersection of non-negative orthant and

$$\left(\sum\nolimits_l(a_{i,l}r_l)^{1+\epsilon}\right)^{\frac{1}{1+\epsilon}} - b_i \leq 0, \quad i = 1, \ldots, I. \tag{29}$$

The corresponding function $f^{(\epsilon)}(\boldsymbol{r})$ for $\bar{\mathcal{R}}$ then become piece-wise $l_{1+\epsilon}$ norm of $\boldsymbol{r}$. Since the spaces defined by (29) are convex, as the intersection of these convex spaces the region $\bar{\mathcal{R}}$ is still convex. Furthermore, smooth the intersection points of (29) such that $f^{(\epsilon)}(\boldsymbol{r})$ has a continuous gradient vector. This is possible due to the convexity of $\bar{\mathcal{R}}$. Then the original piece-wise linear boundary of $\bar{\mathcal{R}}$ is smoothed such that the corresponding $f^{(\epsilon)}(\boldsymbol{r})$ becomes strictly convex and twice differentiable. Consequently, the convergence of (18) readily follows from Theorem 3. As a limiting case of $f^{(\epsilon)}(\boldsymbol{r})$ with $\epsilon \to 0$, we then expect that the convergence of (18) follows also for a piece-wise linear $f(\boldsymbol{r})$. To corroborate this argument, simulation results in Section V-A will be provided to show the convergence of (18) in this case.

For all cases that the capacity region $\bar{\mathcal{R}}$ is a combination of those in Fig. 1(a)–(c), convergence of proposed schemes can be argued by combining the proofs for individual cases.

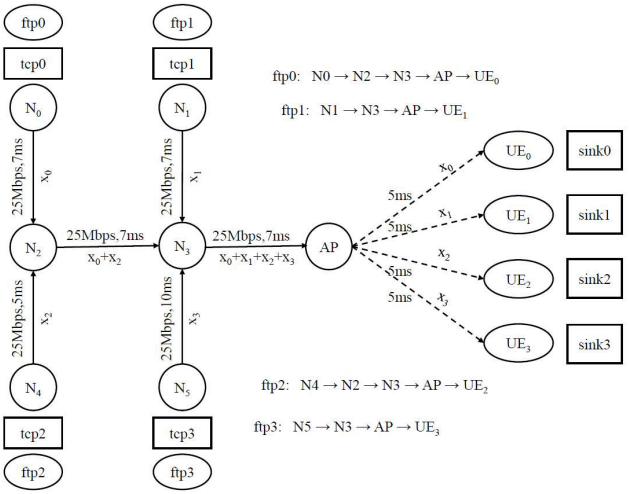
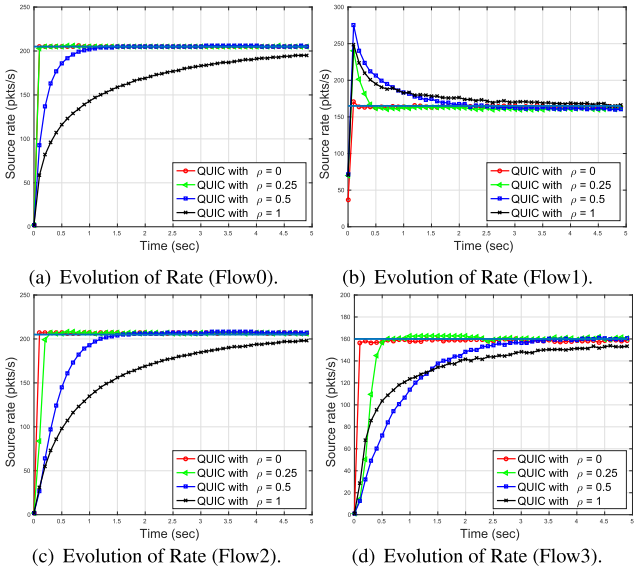


FIGURE 3. Network topology.



(a) Evolution of Rate (Flow0). (b) Evolution of Rate (Flow1).

(c) Evolution of Rate (Flow2). (d) Evolution of Rate (Flow3).

FIGURE 4. Evolutions of the source rates.

V. SIMULATION RESULTS

A. MATLAB SIMULATIONS

First, we conduct Matlab simulations to assess the performance of the proposed schemes in the fluid model. The considered network has six fixed computer nodes (N_0–N_5), a base station (BS) and four wireless user equipments (UEs). As shown in Fig. 3, the BS is the access point for the wireless UEs. The capacities and delays of all links are marked. The BS transmits to users over a shared downlink with a 5 MHz bandwidth, and the average received signal-to-noise ratios (SNRs) for users (UE_0–UE_3) are 11, 17, 12, and 15 dB, respectively. For transmissions over wireless links, the BS supports five adaptive modulation and coding (AMC) modes. The independent fading processes for wireless links follow a Rayleigh random process. The mobility-induced Doppler spread f_d for each UE is 20 Hz. The AMC and wireless channel modeling follow [41].

Four TCP connections are set up to carry four FTP streams, where the packet size of each FTP stream is 200 bytes. We implement the proposed window control scheme at the source nodes, and employ the scheduling policy (17) at the BS. In the simulations, the algorithm (9) is used to update the window size every 20 milliseconds, where the step size $\kappa = 0.05$. Local information d_s, $\bar{d}_s$ and v_s are assumed available at each flow source s without feedback delay. When ρ is set to $0, 0.25, 0.5, 1$, respectively, the source rate evolution processes of the four FTP streams with the proposed algorithms are shown in Fig. 4. The optimal source rate obtained by standard sub-gradient iterative solution (1) is also provided to serve as a benchmark. Evidently, all QUIC-TCP algorithms can converge to the optimal equilibrium point, thereby validating the correctness and effectiveness of the proposed schemes.

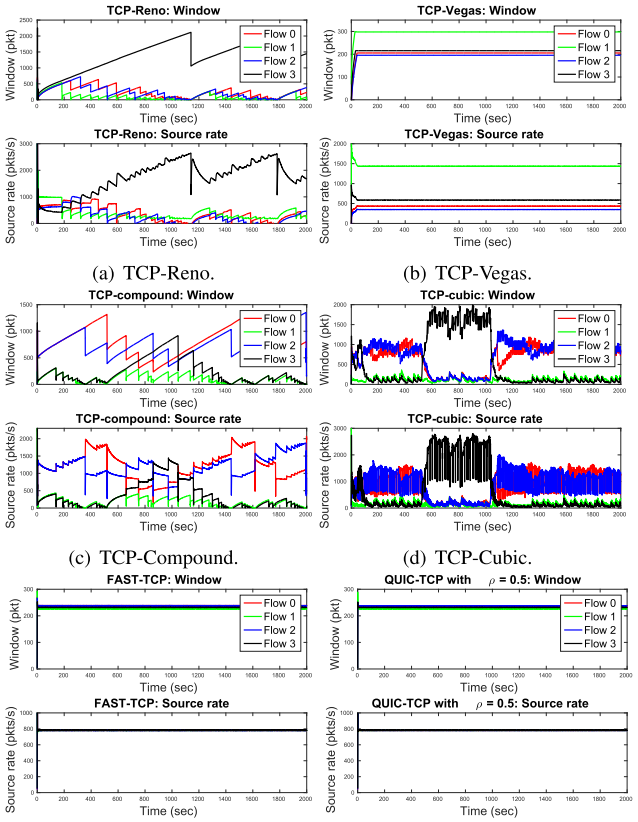
B. NS-2 SIMULATIONS

Consider the network in Fig. 3 again. We rely on ns-2 simulator to assess the performance of the proposed schemes in IPv6 based Internet environment. We modify the ns-2 module of the existing FAST-TCP in accordance with the algorithm (9) to generate our QUIC-TCP proxy. The wireless network simulates an IEEE 802.16 WiMax network, and its bandwidth is 10 MHz. The average SNRs for users (UE_0–UE_3) are 12, 19, 13, and 16 dB, respectively. The wireless link is affected by Rayleigh fading, with a Doppler frequency of 128 Hz. Eight different convolutional turbo coded quadrature amplitude modulations (QAMs) are used at the BS as the AMC modes for downlink transmission [41].

Data packets are transmitted to UEs over the slotted downlink, and the duration of the time slot is 10 ms. The four FTP flows are *asynchronous*: ftp0 starts at 0 sec and ends at 2000 sec; ftp1 starts at 0 sec and ends at 2000 sec; ftp2 starts at 0 sec and ends at 2000 sec; and ftp3 starts at 0 sec and ends at 2000 sec. The packet size for FTP flows is 200 bytes. Buffer size at fixed nodes and the BS is 2000 packets. The target queue size p_s of each flow is set to 200 packets. The BS implements the scheduling policy (10). Taking into account the feedback delay, we run TCP-Reno [11], TCP-Vegas [8], TCP-Cubic [19], TCP-Compound [18], FAST-TCP [4], and the proposed QUIC-TCP ($\rho = 0.5$). The window update step size for FAST-TCP and QUIC-TCP is set as $\kappa = 1.5$. Fig. 5 shows the evolution processes of the window sizes and source rates with different TCP schemes. According to (9), when the congestion window in QUIC-TCP is far from the equilibrium point, it will be adjusted sharply, and the update speed will slow down near the equilibrium. It is clear that the proposed QUIC-TCP algorithm can quickly and steadily converge to the equilibrium point, and the achieved average rates of flows ftp0, ftp1, ftp2, and ftp3 are 780.75, 781.46, 781.11, and 781.36 packets/s, respectively. Although the global convergence of FAST-TCP has not been proven, its convergence is also shown, with achieved average flow rates given by 780.86, 781.26, 781.06, and

781.46 packets/s, just a bit smaller than those with QUIC-TCP. TCP-Compound uses estimated queuing delay to measure congestion; if the queuing delay is small, it assumes that there is no congestion on the link and rapidly increases its rate. TCP-Compound maintains two congestion windows: a regular AIMD window, and a delay-based window. The final actual sliding window size is the sum of these two windows. The AIMD window is increased in the same way as TCP-Reno. If the delay is small, the delay-based window will increase rapidly to improve network utilization. Once heavy queueing is experienced, the delay window will gradually decrease to compensate for the increased AIMD window. As a result, the average flow rates are 1216.97, 184.54, 1186.04, and 357.48 packets/s, respectively. Compared with QUIC-TCP, its total throughput loses 6.2%. TCP-Cubic uses a cubic function as the growth function of the congestion window. The growth of the congestion window is no longer related to RTT, but only depends on the time interval since the last congestion and the maximum window value when the congestion last occurred. At the beginning, the congestion window grows quickly. When approaching the symmetric center of cubic function, the growth rate becomes gentle, avoiding sudden increase in traffic and causing packet loss. While it is close to symmetric center, the congestion window no longer increases. After moving away from the symmetric center, the congestion window continues to grow. When packet loss occurs, the congestion window is multiplicatively reduced, then the aforementioned window growth process is resumed. The resultant average flow rates are 1008.06, 177.17, 1045.91, and 742.42 packets/s; the total throughput is lost by 5.1% compared to QUIC-TCP. The convergence speed of TCP-Vegas is very slow. Here, the lower and upper limits of the window values are set to 200 and 203 packets. After comparing the size of the current queue with the latter two thresholds, TCP-Vegas decreases or increases the size of the congestion window accordingly. Even if it is far from the equilibrium point, its window-size adjustment also changes linearly, leading to a slow convergence speed. As a result, the average flow rates are 436.48, 1453.28, 351.09, and 589.47 packets/s, respectively. Compared with QUIC-TCP, its total throughput loses 10.4%. Unlike other TCPs, TCP-Reno uses packet loss as congestion measure, and implements the AIMD algorithm to adjust the window-size value. At the beginning, the algorithm linearly increases the window size to detect the effective bandwidth until the packet is lost. After packet loss, it reduces the window size by half to avoid congestion. Because of the AIMD scheme, the TCP-Reno algorithm exhibits ''sawtooth'' oscillations during the window adjustment process. The average flow rates are 356.78, 345.91, 265.31, and 1718.54 packets/s; the total throughput is lost by 16.3% compared to QUIC-TCP. It can be also seen that all TCP schemes except FAST-TCP and QUIC-TCP allocate less resources to certain flows, resulting in unfairness among flows.

We also test the TCP performance under different link scheduling schemes. In addition to the queueing delay based



(a) TCP-Reno. (b) TCP-Vegas.

(c) TCP-Compound. (d) TCP-Cubic.

(e) FAST-TCP. (f) QUIC-TCP.

FIGURE 5. Window size and source rate evolutions.

scheme (10), we simulate three other schedulers: a classic Round-robin scheduler, a max-SNR scheduler which simply selects the wireless link with highest SNR per slot n, and the queue-length based scheduler in prior NUM solutions [32]. Assuming that all data flows run from 0 seconds to 2000 seconds, Fig. 6 shows the performance of TCP-Reno, TCP-Vegas, TCP-Compound, TCP-Cubic, FAST-TCP and QUIC-TCP ($\rho = 0.5$) under different link scheduling algorithms. In addition to the total throughput $\sum_s x_s$, we use the celebrated Jain's index to gauge fairness [42]: $F = \frac{(\sum_s x_s)^2}{|S|\sum_s (x_s)^2}$ ($|S|$ represents the number of data streams). Because there are four data flows in this experiment ($|S| = 4$), this index ranges from 0.25 (least unfair) to 1 (most fair). For comparison, we also provide the results $\sum_s p_s \log x_s$ (i.e., objective of optimization problem (1)) achieved by different schemes. According to Theorem 1, QUIC-TCP only generates optimal $\boldsymbol{x}^*$ for (1) under the scheduling policy (10). This can be shown from Fig. 6 (top), in which QUIC-TCP using this scheduling algorithm generates the highest $\sum_s p_s \log x_s$, thereby achieving a good balance between the total throughput $\sum_s x_s$ and the flow fairness. As shown in Fig. 6 (middle), the total throughput of QUIC-TCP with scheduling policy (10) is 3124.68 packets/s, which is 9.3% larger than the throughput (2856.75 packets/s) under the Round-robin scheduler and is close to the those under max-SNR and queue-length based schedulers (3313.89 and 3215.84 packets/s).

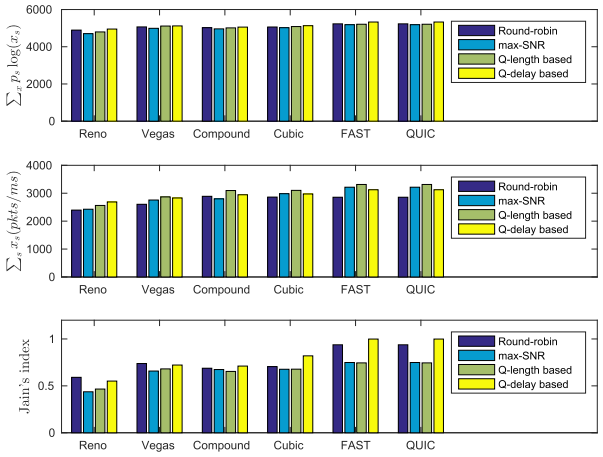
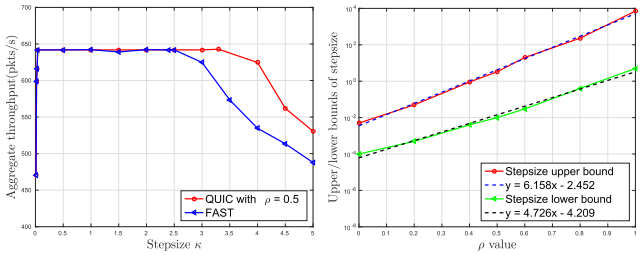


FIGURE 6. TCP performance under different schedulers.

On the other hand, as shown in Fig. 6 (bottom), the Jain's fairness index under the scheduler (10) is $F = 0.999$, which is significantly higher than the fairness (0.749 and 0.745) under the maximum SNR and queue-length based schedulers, and is even better than that (0.938) under the Round-robin scheduler. Under different scheduling policies, the performance of FAST-TCP and QUIC-TCP is similar, while the performance of other TCP drops significantly. For example, under the queueing-delay based scheduler (10), the Jain's index with TCP-Cubic is 0.820 (0.179 less); while Jain's indices with TCP-Compound and TCP-Vegas are 0.771 and 0.722, which are reduced by 0.288 and 0.277 respectively compared with TCP-QUIC. Performance degradation for TCP-Reno is even larger. Under the queueing-delay based scheduler, its Jain's index is 0.551 (0.448 less).

To further compare the FAST- and QUIC-TCP, Fig. 7 (a) shows the aggregate throughputs with them for the asynchronous FTP flow case, when different stepsizes are adopted. It is found that performance of both FAST- and QUIC-TCP is not so sensitive to stepsize; both work well for a large range of stepsizes. The aggregate throughput with FAST-TCP is essentially unaffected for stepsizes within $[0.01, 2.5]$; whereas QUIC-TCP with $\rho = 0.5$ has a larger stepsize range $[0.01, 3.3]$ for unchanged throughput. When the stepsize is larger than 2.5 for FAST-TCP, large oscillations in source rates (and window sizes) occur, leading to throughput degradation. The source-rate oscillations with QUIC-TCP are reduced for large stepsizes such that the decrease in total throughput is slower than the FAST-TCP. Compared with FAST-TCP algorithm $\frac{d}{dt}w_s(t) = -\kappa v_s$, the window update $\frac{d}{dt}w_s(t) = -\kappa \frac{d_s}{\bar{d}_s} v_s$ in QUIC-TCP with $\rho = 0.5$ has an extra $d_s/\bar{d}_s$ factor. In Lyapunov analysis, this factor helps establish the global stability of the QUIC-TCP. In simulations, it turns out to also help stabilize the window-size and source-rate updates, especially for large stepsizes. It is worth mentioning that the ns-2 module of QUIC-TCP here is simply modified from the module of FAST-TCP. Re-designing QUIC-TCP according to its specifications may have better performance.



(a) FAST vs QUIC with $\rho = 0.5$. (b) QUIC with different ρ

FIGURE 7. FAST-TCP and QUIC-TCP performance for different stepsizes.

Besides $\rho = 0.5$, QUIC-TCP with other $\rho \in [0, 1]$ works as well. Under queueing-delay based scheduler (10), Fig. 7 (b) shows the lower and upper bounds of the "good" stepsizes for QUIC-TCP with different ρ, where the aggregate throughputs do not degrade. The Matlab simulations in Fig. 4 implies that given the same stepsize, QUIC-TCP converges quicker for smaller ρ. For Internet with large bandwidth-delay product, the flow source needs maintain a pretty large window size w_s. In the QUIC-TCP window update $\frac{d}{dt}w_s(t) = -\kappa \frac{d_s}{\bar{d}_s} w_s^{-2\rho+1} v_s$, a small stepsize κ is thus required for $\rho < 0.5$ to prevent too drastic window-size changes, while a large κ is needed for $\rho > 0.5$ to avoid too slow window updates. Fig. 7 (b) also shows the least-square-error fittings of (logarithm of) lower- and upper-bounds to a linear model $y = ax + b$. It is seen that the slope of the upper-bounds is larger than that of the lower-bounds. This implies that QUIC-TCP with larger ρ has a larger good-stepsize ranges; e.g., QUIC-TCP with $\rho = 1.0$ can have a range across three magnitude orders. Overall, QUIC-TCP with a ρ slightly larger than 0.5 (e.g., $\rho \approx 0.55$) could serve best, i.e., having a large range of "normal" stepsizes (0.02–7.1) without performance degradation.

VI. CONCLUDING REMARKS

We proved the existence of optimal window-based TCP congestion control schemes for networks with wireless links. A class of QUIC-TCP schemes were proposed. Leveraging the Lyapunov method, we established that QUIC-TCP can globally converge to the optimal network equilibrium with the help of a queueing-delay based MaxWeight scheduler at wireless AP. For simplicity, we assumed a last-hop wireless network. Generalization of our approach to networks with distributed ad-hoc wireless links is possible, e.g., along with the works in [43], [44].

The utility function in (1) is the $(\boldsymbol{p}, 1)$-proportionally fair function – a special case of the $(\boldsymbol{p}, \alpha)$-proportionally fair functions [28]. The proposed QUIC-TCP algorithms can be generalized to achieve the $(\boldsymbol{p}, \alpha)$-proportional fairness which includes the max-min fairness as $\alpha \to \infty$. Morever, it would be meaningful and valuable to evaluate the proposed schemes in a real testbed. Establishing a state-of-the-art testbed and testing our schemes in such a testbed will be an interesting direction to pursue in our future work.

The proposed approach can serve as a stepping stone in the theory of applying optimization tools to develop and analyze readily deployed network schemes over the Internet infrastructure. The framework has far-reaching implications and enables more general cross-layer design of Internet protocols for mobile applications. For instance, here we assumed single-path routing and unicast TCP flows. It will be also interesting to explore generalization of our approach to networks with multipath source routing in emerging (e.g., multi-homing) applications, and to TCP-friendly multicast congestion control for real-time audio/video non-TCP flows.

APPENDIX

A. PROOF FOR PROPERTIES OF THE MAPPING $F : w \to (x, q)$

Proof of Property 1: Observe that the mapping $F : \boldsymbol{w} \to \boldsymbol{x}$ can be in fact defined by the unique solution of (19). Following the similar lines proving [29, Claims 3], we take a sequence $\boldsymbol{w}^{(n)}$ such that $\boldsymbol{w}^{(n)} \to \boldsymbol{w}$ and let $\boldsymbol{x}^{(n)} = F(\boldsymbol{w}^{(n)})$. By the compactness of the constraint set in (19), we can have a subsequence $\boldsymbol{w}^{(n_k)} \to \boldsymbol{w}$ so that the corresponding $\boldsymbol{x}^{(n_k)}$ converges to, say $\vec{\boldsymbol{x}}$.

Define $f(\boldsymbol{w}, \boldsymbol{x}) := \sum_{s\in S}(w_s \log x_s - d_s x_s)$. By the optimality of $\boldsymbol{x}^{(n_k)}$, we have $f(\boldsymbol{w}^{(n_k)}, \boldsymbol{x}^{(n_k)}) \geq f(\boldsymbol{w}^{(n_k)}, \boldsymbol{x})$ for all other feasible $\boldsymbol{x}$ of (19). Upon taking the limit, it follows from the continuity of function f that $f(\boldsymbol{w}, \vec{\boldsymbol{x}}) \geq f(\boldsymbol{w}, \boldsymbol{x})$ for all feasible $\boldsymbol{x}$. This implies exactly $\vec{\boldsymbol{x}} = F(\boldsymbol{w})$. Therefore, the function $\boldsymbol{x}(\boldsymbol{w}) := F(\boldsymbol{w})$ is continuous.

Mimicking the similar lines in the proof of [29, Claims 4]Mo00, we can further prove function $\boldsymbol{x}(\boldsymbol{w})$ is differentiable except at the boundary points.

Proof of Property 2: Suppose without loss of generality that we have two window size vector $\boldsymbol{w}$ and $\boldsymbol{w}'$, where only the sth entry $w_s > w'_s$ and all other $w_{s'} = w'_{s'}$, $\forall s' \neq s$. Let $\boldsymbol{x} := \boldsymbol{x}(\boldsymbol{w})$ and $\boldsymbol{x}' := \boldsymbol{x}(\boldsymbol{w}')$ be the corresponding rate vectors for $\boldsymbol{w}$ and $\boldsymbol{w}'$, respectively. Since $\boldsymbol{x}$ and $\boldsymbol{x}'$ are given by the solution of (19), we clearly have:

$$f(\boldsymbol{w}, \boldsymbol{x}) \geq f(\boldsymbol{w}, \boldsymbol{x}'), \quad \text{and} \quad f(\boldsymbol{w}', \boldsymbol{x}) \leq f(\boldsymbol{w}', \boldsymbol{x}').$$

Subtracting both sizes of the two inequalities then yield:

$$(w_s - w'_s)\log x_s \geq (w_s - w'_s)\log x'_s.$$

Given $w_s > w'_s$, it follows that $\log x_s \geq \log x'_s$, or simply, $x_s \geq x'_s$. This proves that x_s is non-decreasing in its own window size w_s if all $w_{s'}$, $\forall s' \neq s$, remain fixed;

When $\boldsymbol{x}(\boldsymbol{w})$ is differentiable, we can mimic the proof of Theorems 2 and 3 to establish that the Jacobian matrix

$$\boldsymbol{J}_{\boldsymbol{x}|\boldsymbol{w}} = \bar{\boldsymbol{D}}^{-1}(\boldsymbol{I} - \boldsymbol{X}\boldsymbol{M}\bar{\boldsymbol{D}}^{-1})$$

where the matrix $\boldsymbol{M}$ (and thus $\boldsymbol{X}\boldsymbol{M}\bar{\boldsymbol{D}}^{-1}$) is positive semi-definite [cf. (25)]. Since the positive semi-definite $\boldsymbol{X}\boldsymbol{M}\bar{\boldsymbol{D}}^{-1}$ has non-negative diagonal entries, it readily follows that $\frac{\partial x_s}{\partial w_s} \leq \frac{1}{d_s}$, $\forall s$. With $\frac{\partial x_s}{\partial w_s} \geq 0$ inferred by that x_s is non-decreasing in its window size w_s when other $w_{s'}$, $\forall s' \neq s$, remain fixed, Property 2 is proven.

Proof of Property 3: From (11), we have $\boldsymbol{q}^s = w_s/x_s - d_s$, $\forall s$. It thus simply follows from Property 1 that $\boldsymbol{q}^s(\boldsymbol{w})$ is continuous and differentiable except at the boundary points.

When $\boldsymbol{q}^s(\boldsymbol{w})$ is differentiable, we can show that the Jacobian matrix $\boldsymbol{J}_{\boldsymbol{q}^s|\boldsymbol{w}} = \boldsymbol{M}\bar{\boldsymbol{D}}^{-1}$, where the matrix $\boldsymbol{M}$ is positive semi-definite; see the proof of Theorems 2 and 3. This implies that $\boldsymbol{J}_{\boldsymbol{q}^s|\boldsymbol{w}}$ is positive semi-definite, and thus all its diagonal entries $\frac{\partial q^s}{\partial w_s} \geq 0$. Using $\boldsymbol{J}_{\boldsymbol{q}^s|\boldsymbol{w}}$, we have the matrix $\boldsymbol{J}_{\boldsymbol{x}|\boldsymbol{w}} = \bar{\boldsymbol{D}}^{-1}(\boldsymbol{I} - \boldsymbol{X}\boldsymbol{J}_{\boldsymbol{q}^s|\boldsymbol{w}})$. Therefore, $\frac{\partial x_s}{\partial w_s} = \frac{1}{d_s} - \frac{x_s}{d_s}\frac{\partial q^s}{\partial w_s}$, $\forall s$. Since $\frac{\partial x_s}{\partial w_s} \geq 0$, it readily follows $\frac{\partial q^s}{\partial w_s} \leq \frac{1}{x_s}$, $\forall s$. The proof is complete.

B. PROOF OF THEOREM 1

At the equilibrium $\boldsymbol{v} = 0$, we have $w^*_s - x^*_s d_s - p_s = 0$; and it is implied by (11) that $w^*_s = x^*_s(d_s + q^{s*})$. Hence, we readily have $p_s = x^*_s q^{s*}$, $\forall s$. Now, if we set $\boldsymbol{\lambda}^* \equiv \boldsymbol{q}^*$, the latter becomes precisely the KKT condition (2). In addition, given this equivalence mapping, the relationships (12)–(17) for $\{\boldsymbol{x}^*, \bar{\boldsymbol{r}}^*, \boldsymbol{q}^*\}$ turn into the KKT conditions (3)–(7). This proves that $(\boldsymbol{x}^*, \bar{\boldsymbol{r}}^*)$ and $\boldsymbol{q}^*$ in fact consist of the optimal solutions of (1) and its dual problem. Moreover, it is easy to prove the existence and uniqueness of the optimal $\boldsymbol{x}^*$ for (1). As the mapping $\boldsymbol{w}^* \to \boldsymbol{x}^*$ is also unique, the existence and uniqueness of the optimal window-size vector $\boldsymbol{w}^*$ follow.

C. PROOF OF LEMMA 2

As $f(\boldsymbol{r})$ is strictly convex, it follows that

$$f(\boldsymbol{r}^*) + [\nabla f(\boldsymbol{r}^*)]^T(\boldsymbol{r} - \boldsymbol{r}^*) < f(\boldsymbol{r}), \quad \forall \boldsymbol{r} \neq \boldsymbol{r}^*.$$

We in turn have

$$f(\boldsymbol{r}^*) - f(\boldsymbol{r}) < [\nabla f(\boldsymbol{r}^*)]^T(\boldsymbol{r}^* - \boldsymbol{r}), \quad \forall \boldsymbol{r} \neq \boldsymbol{r}^*.$$

Suppose that $f(\boldsymbol{r}^*) = 0$ and $\theta\nabla f(\boldsymbol{r}^*) = \boldsymbol{q}_w$. As $f(\boldsymbol{r}) \leq 0 = f(\boldsymbol{r}^*)$, $\forall \boldsymbol{r} \in \bar{\mathcal{R}}$, we then have

$$\boldsymbol{q}_w^T(\boldsymbol{r}^* - \boldsymbol{r}) > \theta(f(\boldsymbol{r}^*) - f(\boldsymbol{r})) \geq 0, \quad \forall \boldsymbol{r} \neq \boldsymbol{r}^*,\ \boldsymbol{r} \in \bar{\mathcal{R}}$$

It readily follows that

$$\boldsymbol{r}^* = \arg\max_{\boldsymbol{r}\in\bar{\mathcal{R}}} \sum_{l\in L_w} q_l r_l.$$

Now, for a strictly convex and twice differentiable $f(\boldsymbol{r})$, its gradient $\nabla f(\boldsymbol{r})$ can take any value in the non-negative orthant. As a result, we can always find a $\boldsymbol{r}^*$ which satisfies $f(\boldsymbol{r}^*) = 0$ and $\theta\nabla f(\boldsymbol{r}^*) = \boldsymbol{q}_w$ for any $\boldsymbol{q}_w \geq 0$. Hence, the converse is also proven.

D. PROOF OF LEMMA 3

Differentiating both sides of $f(\boldsymbol{r}^*) = 0$ yields

$$[\nabla f(\boldsymbol{r}^*)]^T \boldsymbol{J}_{\boldsymbol{r}^*|\boldsymbol{q}_w} = 0. \tag{30}$$

Noting that θ in $\theta\nabla f(\boldsymbol{r}^*) = \boldsymbol{q}_w$ is actually a function $\theta(\boldsymbol{r}^*)$ of $\boldsymbol{r}^*$, we also have:

$$\nabla f(\boldsymbol{r}^*)\left(\left[\frac{\partial r_1^*}{\partial q_l}, \cdots, \frac{\partial r_{L_w}^*}{\partial q_l}\right]\nabla\theta(\boldsymbol{r}^*)\right) + \theta\nabla^2 f(\boldsymbol{r}^*)\begin{bmatrix}\frac{\partial r_1^*}{\partial q_l}\\ \vdots \\ \frac{\partial r_{L_w}^*}{\partial q_l}\end{bmatrix} = \boldsymbol{e}_l, \quad \forall l \in L_w$$

where $\boldsymbol{e}_l$ denotes the vector $[0, \cdots, 1, \cdots, 0]^T$ with only its lth entry equal to one. In the matrix form, we have

$$\nabla f(\boldsymbol{r}^*)\boldsymbol{J}^T_{\boldsymbol{r}^*|\boldsymbol{q}_w}\nabla\theta(\boldsymbol{r}^*) + \theta\nabla^2 f(\boldsymbol{r}^*)\boldsymbol{J}_{\boldsymbol{r}^*|\boldsymbol{q}_w} = \boldsymbol{I}. \tag{31}$$

But since $\nabla f(\boldsymbol{r}^*)\boldsymbol{J}^T_{\boldsymbol{r}^*|\boldsymbol{q}_w} = ([\nabla f(\boldsymbol{r}^*)]^T\boldsymbol{J}_{\boldsymbol{r}^*|\boldsymbol{q}_w})^T = 0$, we readily have $\theta\nabla^2 f(\boldsymbol{r}^*)\boldsymbol{J}_{\boldsymbol{r}^*|\boldsymbol{q}_w} = \boldsymbol{I}$ and thus

$$\boldsymbol{J}_{\boldsymbol{r}^*|\boldsymbol{q}_w} = (1/\theta)[\nabla^2 f(\boldsymbol{r}^*)]^{-1}.$$

With $\nabla^2 f(\boldsymbol{r}^*)$ positive definite, it follows that $\boldsymbol{J}_{\boldsymbol{r}^*|\boldsymbol{q}_w}$ is too.

E. PROOF OF THEOREM 3

Note that all wireless links should be utilized for some TCP flows, since links not in use can be simply removed from the logical link set. Given a $\boldsymbol{w}$, the set of bottleneck links then either does not contain any wireless links or contains all wireless links when we have a smooth capacity region. If the bottleneck set contains only part of the wireless links, we have some $q_l > 0$ and some $q_l = 0$ for $l \in L_w$. According to the queueing-delay based scheduling strategy (17), the links with zero queueing delays must have zero capacity $r_l^* = 0$ for this smooth capacity region case. This means that the rates x_s for the flows traveling over these links must be zero. However, this is impossible for the following reasons: For a given $\boldsymbol{w}$, the source rate vector must be the solution of (19); however, if $x_s = 0$, the rate vector will make the objective function of (19) become $-\infty$, which clearly cannot solve (19).

Suppose that $\boldsymbol{w}$ is an interior point. If the set of bottleneck links does not include any wireless links, the derivation of (27) in the proof of Theorem 2 is clearly still valid; i.e., $dY(\boldsymbol{w}(t))/dt < 0$ unless $\boldsymbol{v} = 0$.

Given a bottleneck set containing all wireless links, we partition the bottleneck-only routing matrix $\boldsymbol{A}_B$ and queueing delay vector $\boldsymbol{q}_B$ into two parts:

$$\boldsymbol{A}_B = \begin{bmatrix}\boldsymbol{A}_{f,B}\\ \boldsymbol{A}_w\end{bmatrix}, \qquad \boldsymbol{q}_B = \begin{bmatrix}\boldsymbol{q}_{f,B}\\ \boldsymbol{q}_w\end{bmatrix}$$

where subscripts $_{f,B}$ and $_w$ denote the parts related to the wired and wireless bottleneck links, respectively.

Following the lines from (22) to (24), we have

$$\boldsymbol{A}_B\boldsymbol{J}_{\boldsymbol{x}|\boldsymbol{w}} + \boldsymbol{A}_B\bar{\boldsymbol{D}}^{-1}\boldsymbol{X}\boldsymbol{A}_B^T\boldsymbol{J}_{\boldsymbol{q}_B|\boldsymbol{w}} = \boldsymbol{A}_B\bar{\boldsymbol{D}}^{-1}. \tag{32}$$

For the wired bottleneck links, it holds $\boldsymbol{A}_{f,B}\boldsymbol{x} = \boldsymbol{c}_B$, and thus $\boldsymbol{A}_{f,B}\boldsymbol{J}_{\boldsymbol{x}|\boldsymbol{w}} = 0$. For wireless bottleneck links, we have: $\boldsymbol{A}_w\boldsymbol{x} = \boldsymbol{r}^*$. This implies:

$$\boldsymbol{A}_w\boldsymbol{J}_{\boldsymbol{x}|\boldsymbol{w}} = [0 \;\; \boldsymbol{J}_{\boldsymbol{r}^*|\boldsymbol{q}_w}]\boldsymbol{J}_{\boldsymbol{q}_B|\boldsymbol{w}}.$$

Then overall we have

$$\boldsymbol{A}_B\boldsymbol{J}_{\boldsymbol{x}|\boldsymbol{w}} = \begin{bmatrix}\boldsymbol{A}_{f,B}\\ \boldsymbol{A}_w\end{bmatrix}\boldsymbol{J}_{\boldsymbol{x}|\boldsymbol{w}} = \begin{bmatrix}0 & 0\\ 0 & \boldsymbol{J}_{\boldsymbol{r}^*|\boldsymbol{q}_w}\end{bmatrix}\boldsymbol{J}_{\boldsymbol{q}_B|\boldsymbol{w}} := \boldsymbol{N}\boldsymbol{J}_{\boldsymbol{q}_B|\boldsymbol{w}}$$

It then follows from (32) that

$$(\boldsymbol{N} + \boldsymbol{A}_B\bar{\boldsymbol{D}}^{-1}\boldsymbol{X}\boldsymbol{A}_B^T)\boldsymbol{J}_{\boldsymbol{q}_B|\boldsymbol{w}} = \boldsymbol{A}_B\bar{\boldsymbol{D}}^{-1}.$$

Since $\boldsymbol{J}_{\boldsymbol{r}^*|\boldsymbol{q}_w}$ is positive definite by Lemma 3, we have a positive semi-definite $\boldsymbol{N}$; thus, $\boldsymbol{N} + \boldsymbol{A}_B\bar{\boldsymbol{D}}^{-1}\boldsymbol{X}\boldsymbol{A}_B^T$ is positive definite. It follows that

$$\boldsymbol{J}_{\boldsymbol{q}_B|\boldsymbol{w}} = (\boldsymbol{N} + \boldsymbol{A}_B\bar{\boldsymbol{D}}^{-1}\boldsymbol{X}\boldsymbol{A}_B^T)^{-1}\boldsymbol{A}_B\bar{\boldsymbol{D}}^{-1}.$$

Substituting the latter into (23), we further obtain:

$$\boldsymbol{J}_{\boldsymbol{x}|\boldsymbol{w}} = \bar{\boldsymbol{D}}^{-1}(\boldsymbol{I} - \boldsymbol{X}\boldsymbol{A}_B^T(\boldsymbol{N} + \boldsymbol{A}_B\bar{\boldsymbol{D}}^{-1}\boldsymbol{X}\boldsymbol{A}_B^T)^{-1}\boldsymbol{A}_B\bar{\boldsymbol{D}}^{-1}).$$

Define the matrix

$$\boldsymbol{M}' := \boldsymbol{A}_B^T(\boldsymbol{N} + \boldsymbol{A}_B\bar{\boldsymbol{D}}^{-1}\boldsymbol{X}\boldsymbol{A}_B^T)^{-1}\boldsymbol{A}_B \tag{33}$$

and use the function $Y(\boldsymbol{w})$ in (26). We can again show that the time derivative $dY(\boldsymbol{w}(t))/dt$ is given by (27) with $\boldsymbol{M}$ replaced by $\boldsymbol{M}'$. Since $\boldsymbol{M}'$ is positive semi-definite as with $\boldsymbol{M}$, the same result holds $\forall\rho \in (0, 1]$; i.e., $dY(\boldsymbol{w}(t))/dt < 0$ at all interior points unless $\boldsymbol{v} = 0$.

Using the same arguments for boundary points in the proof of Theorem 2, we conclude that $Y(\boldsymbol{w})$ is a Lyapunov function, which has a negative time derivative globally for the system (18). The theorem readily follows.

ACKNOWLEDGMENT

This article was presented in part by the Proceedings of GLOBECOM.

REFERENCES

[1] X. Wang, Z. Li, and N. Gao, "Joint congestion control and wireless-link scheduling for mobile TCP applications," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Houston, TX, USA, Dec. 2011, pp. 1–6.
[2] M. U. Daigavhane and M. D. Chawhan, "Congestion control algorithm for TCP in wireless network," in *Proc. 4th Int. Conf. Recent Adv. Inf. Technol. (RAIT)*, Dhanbad, India, Mar. 2018, pp. 1–4.
[3] S. Ahmad and M. J. Arshad, "Enhancing fast TCP's performance using single TCP connection for parallel traffic flows to prevent Head-of-Line blocking," *IEEE Access*, vol. 7, pp. 148152–148162, 2019.
[4] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, "FAST TCP: Motivation, architecture, algorithms, performance," *IEEE/ACM Trans. Netw.*, vol. 14, no. 6, pp. 1246–1259, Dec. 2006.
[5] C. Goswami and R. Shahane, "Transport control protocol (TCP) enhancement over wireless environment: Issues and challenges," in *Proc. Int. Conf. Inventive Comput. Informat. (ICICI)*, Coimbatore, India, Nov. 2017, pp. 742–749.
[6] S. H. Low, "A duality model of TCP and queue management algorithms," *IEEE/ACM Trans. Netw.*, vol. 11, no. 4, pp. 525–536, Aug. 2003.
[7] M. Chiang, S. H. Low, R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition," *Proc. IEEE*, vol. 95, no. 1, pp. 255–312, Jan. 2007.
[8] C. A. Talay, F. A. Trinidad, D. R. Rodriguez Herlein, M. Luz Almada, C. N. Gonzalez, and L. A. Marrone, "Analysis of the performance of TCP vegas and its relationship with alpha and beta parameters in a wireless links network and burst errors," in *Proc. Congreso Argentino de Ciencias de la Informática y Desarrollos de Investigación (CACIDI)*, Buenos Aires, AR, USA, Nov. 2018, pp. 1–6.
[9] X. Zhang, N. Gu, J. Su, and K. Ren, "DFTCP: A TCP-friendly delay-based high-speed TCP variant," in *Proc. Int. Conf. Netw. Netw. Appl. (NaNA)*, Hakodate, Japan, Jul. 2016, pp. 273–278.

[10] H. L. Gururaj and B. Ramesh, "An efficient switching TCP (STCP) approach to avoid congestion in ad-hoc networks," in *Proc. IEEE Int. Advance Comput. Conf. (IACC)*, Banglore, India, Jun. 2015, pp. 191–195.

[11] S. U. Shenoy, M. S. Kumari, U. K. K. Shenoy, and N. Anusha, "Performance analysis of different TCP variants in wireless ad hoc networks," in *Proc. I-SMAC*, Palladam, India, 2017, pp. 891–894.

[12] K. Xu, Y. Tian, and N. Ansari, "TCP-jersey for wireless IP communications," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 4, pp. 747–756, May 2004.

[13] M. J. Mbyamm K. and J. Zhang, "Improved implementation of TCP-vegas method in interchanges of satellite links," in *Proc. 5th Int. Conf. Comput. Sci. Netw. Technol. (ICCSNT)*, Changchun, China, Dec. 2016, pp. 529–534.

[14] P. Dong, K. Gao, J. Xie, W. Tang, N. Xiong, and A. V. Vasilakos, "Receiver-side TCP countermeasure in cellular networks," *Sensors*, vol. 19, no. 12, p. 2791, Jun. 2019.

[15] J. Gomez, E. Kfoury, J. Crichigno, E. Bou-Harb, and G. Srivastava, "A performance evaluation of TCP BBRv2 alpha," in *Proc. 43rd Int. Conf. Telecommun. Signal Process. (TSP)*, Milan, Italy, Jul. 2020, pp. 309–312.

[16] K. Winstein, A. Sivaraman, and H. Balakrishnan, "Stochastic forecasts achieve high throughput and low delay over cellular networks," in *Proc. NSDI*, Lombard, IL, USA, Apr. 2013, pp. 459–471.

[17] Y. Zaki, T. Pötsch, J. Chen, L. Subramanian, and C. Görg, "Adaptive congestion control for unpredictable cellular networks," in *Proc. ACM Conf. Special Interest Group Data Commun. (SIGCOMM)*, New York, NY, USA, 2015, pp. 509–522.

[18] S. Manjunath and G. Raina, "Stability and performance of compound TCP with a proportional integral queue policy," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 5, pp. 2139–2155, Sep. 2019.

[19] M. Ahmad, A. Ahmad, S. Jabbar, M. Asif, M. A. Habib, S. H. Ahmed, and S. C. Shah, "TCP CUBIC: A transport protocol for improving the performance of TCP in long distance high bandwidth cyber-physical systems," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Kansas City, MO, USA, May 2018, pp. 1–6.

[20] P. Goyal, M. Alizadeh, and H. Balakrishnan, "Rethinking congestion control for cellular networks," in *Proc. 16th ACM Workshop Hot Topics Netw.*, New York, NY, USA, Nov. 2017, pp. 29–35.

[21] Y.-J. Song, G.-H. Kim, and Y.-Z. Cho, "Enhanced loss-recovery mechanism for BBR to improve inter-protocol fairness," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Jeju Island, South Korea, Oct. 2019, pp. 1181–1183.

[22] C. A. Grazia, M. Klapez, and M. Casoni, "BBRp: Improving TCP BBR performance over WLAN," *IEEE Access*, vol. 8, pp. 43344–43354, 2020.

[23] K. Sasaki, M. Hanai, K. Miyazawa, A. Kobayashi, N. Oda, and S. Yamaguchi, "TCP fairness among modern TCP congestion control algorithms including TCP BBR," in *Proc. IEEE 7th Int. Conf. Cloud Netw. (CloudNet)*, Tokyo, Japan, Oct. 2018, pp. 1–4.

[24] K. Han, J. Y. Lee, and B. C. Kim, "Machine-learning based loss discrimination algorithm for wireless TCP congestion control," in *Proc. Int. Conf. Electron., Inf., Commun. (ICEIC)*, Auckland, New Zealand, Jan. 2019, pp. 1–2.

[25] N. Li, Z. Deng, Q. Zhu, and Q. Du, "AdaBoost-TCP: A machine learning-based congestion control method for satellite networks," in *Proc. IEEE 19th Int. Conf. Commun. Technol. (ICCT)*, Xi'an, China, Oct. 2019, pp. 1126–1129.

[26] K. Xiao, S. Mao, and J. K. Tugnait, "TCP-drinc: Smart congestion control based on deep reinforcement learning," *IEEE Access*, vol. 7, pp. 11892–11904, 2019.

[27] I. L. Afonin, A. V. Gorelik, S. S. Muratchaev, A. S. Volkov, and E. K. Morozov, "Development of an adaptive TCP algorithm based on machine learning in telecommunication networks," in *Proc. Syst. Signal Synchronization, Generating Process. Telecommun. (SYNCHROINFO)*, Moscow, Russia, Jul. 2019, pp. 1–5.

[28] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 556–567, Oct. 2000.

[29] X. Wang, *Scheduling Congestion Control for Wireless Internet*. New York, NY, USA: Springer, 2014.

[30] *3rd Generation Partnership Project, Technical Specification Group Radio Access Network; Physical Layer Aspects for Evolved Universal Terrestrial Radio Access (UTRA)*, 3GPP Standard TR 25.814 v. 7.0.0, 2006.

[31] A. L. Stolyar, "Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm," *Queueing Syst.*, vol. 50, no. 4, pp. 401–457, 2005.

[32] X. Lin and N. Shroff, "The impact of imperfect scheduling on cross-layer rate control in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 14, no. 2, pp. 302–315, Apr. 2006.

[33] D. Ron, J.-H. Bang, and J.-R. Lee, "Fair resource allocation in wireless networks," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Chicago, IL, USA, Aug. 2018, pp. 1–2.

[34] M. Shivaranjani, R. Shanju, M. J. A. Jude, and V. C. Diniesh, "Analysis of TCP's micro level behaviour in wireless multi-hop environment," in *Proc. Int. Conf. Comput. Commun. Informat. (ICCCI)*, Coimbatore, India, Jan. 2016, pp. 1–6.

[35] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[36] P. K. Taksande, P. Chaporkar, P. Jha, and A. Karandikar, "Proportional fairness through dual connectivity in heterogeneous networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Seoul, South Korea, May 2020, pp. 1–6.

[37] X. Wang, G. B. Giannakis, and A. G. Marques, "A unified approach to QoS-guaranteed scheduling for channel-adaptive wireless networks," *Proc. IEEE*, vol. 95, no. 12, pp. 2410–2431, Dec. 2007.

[38] D. S. Meitis, D. S. Vasiliev, A. Abilov, and I. Kaysina, "Comparison of TCP congestion control algorithms in network coded relaying scheme," in *Proc. Int. Siberian Conf. Control Commun. (SIBCON)*, Tomsk, Russia, Apr. 2019, pp. 1–4.

[39] H. Haci and A. Abdelbari, "A novel scheduling scheme for ultra-dense networks," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Istanbul, Turkey, Jun. 2019, pp. 1–6.

[40] H. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[41] Q. Liu, X. Wang, and G. B. Giannakis, "A cross-layer scheduling algorithm with QoS support in wireless networks," *IEEE Trans. Veh. Technol.*, vol. 55, no. 3, pp. 839–847, May 2006.

[42] C. Guo, M. Sheng, X. Wang, and Y. Zhang, "Throughput maximization with short-and long-term Jain's index guarantees in OFDMA systems," in *Proc. IEEE 24th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, London, U.K., Sep. 2013, pp. 1518–1522.

[43] X. Wang, Z. Li, and J. Wu, "Joint TCP congestion control and CSMA scheduling without message passing," *IEEE Trans. Wireless Commun.*, vol. 12, no. 12, pp. 6194–6204, Dec. 2013.

[44] X. Wang and Z. Li, "Joint TCP congestion control and wireless-link scheduling for wireless Internet applications," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Xi'an, China, Aug. 2013, pp. 752–757.

HAICHANG HUANG received the B.Sc. degree from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2018. He is currently pursuing the M.Sc. degree with the Department of Communication Science and Engineering, Fudan University, Shanghai, China. His research interests include wireless communication and TCP congestion control.

ZHIYANG SUN received the B.Sc. degree from Fudan University, Shanghai, China, in 1997, and the M.Eng. degree from the Jiangxi University of Finance and Economics, Jiangxi, China, in 2012. He is currently pursuing the D.Eng. degree with the Department of Communication Science and Engineering, Fudan University. His research interests include computer networks and wireless resource allocation.

XIN WANG (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from Fudan University, Shanghai, China, in 1997 and 2000, respectively, and the Ph.D. degree from Auburn University, Auburn, AL, USA, in 2004, all in electrical engineering.

From September 2004 to August 2006, he was a Postdoctoral Research Associate with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA. In August 2006, he joined the Department of Electrical Engineering, Florida Atlantic University, Boca Raton, FL, USA, as an Assistant Professor, then was promoted to a tenured Associate Professor, in 2010. He is currently a Distinguished Professor and the Chair of the Department of Communication Science and Engineering, Fudan University. His research interests include stochastic network optimization, energy-efficient communications, cross-layer design, and signal processing for communications. He has served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, as an Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and as an Associate Editor for the IEEE SIGNAL PROCESSING LETTERS. He also serves as a Senior Area Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and as an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. He is an IEEE Distinguished Lecturer of the Vehicular Technology Society.