Research Article

Protecting Download Traffic from Upload Traffic over Asymmetric Wireless Links

Hwangnam Kim ¹, **Woonghee Lee** ¹, **Hyunsoon Kim**,²
Hwantae Kim,³ and **Jaehyeuk Michael Yang**⁴

¹*School of Electrical Engineering, Korea University, Republic of Korea*

²*TmaxSoft Inc., Seongnam-si, Gyeonggi-do 13613, Republic of Korea*

³*Kukka, Inc., Republic of Korea*

⁴*QCT CE Model Technologies, QualComm, Inc., Republic of Korea*

Correspondence should be addressed to Hwangnam Kim; hncim@korea.ac.kr

Received 5 February 2018; Revised 23 May 2018; Accepted 19 June 2018; Published 12 July 2018

Academic Editor: Vicente Casares-Giner

Copyright © 2018 Hwangnam Kim et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many varied mobile device networks have been developed with the advancement of communication and network technologies. Cellular data networks are currently the most widely used, and the number of cellular network subscriptions has increased steadily. Most recent wireless access technologies employ asymmetric uplinks and downlinks because mobile subscribers usually download contents from the Internet. Therefore, most cellular network service providers allocate more bandwidth to downlinks than uplinks for mobile subscribers. However, this asymmetry can have unexpected influence on network performance, particularly TCP performance. When the uplink interface is congested, TCP ACK packets are delayed by TCP data packets on the uplink, causing considerable TCP retransmissions on the downlink channel. Thus, downlink bandwidth cannot be fully utilized, which results in significantly degraded downlink throughput. To resolve this problem, this paper proposes a feedback scheme, network traffic chunk regulator (NCR). We analyzed the aforementioned problem through the empirical study, and we designed and implemented NCR based on the analysis. NCR adaptively controls TCP according to the degree of link usage asymmetry. We evaluate NCR performance through simulations and experiments with real devices. We verify that the proposed scheme allows the downlink traffic to not interfere with the aggressive uplink traffic. Thus, NCR increases total link utilization and aggregated throughput significantly, without imposing additional overhead on base or mobile stations.

1. Introduction

Emerging mobile services and systems allow one or more computing elements, such as applications or control programs, to control one or more physical entities, such as sensors or actuators, and generally assume a networked control system between them. Recent system development trends are tending towards scalable architectures that can be adapted to any application or system rather than systems dedicated to specific purposes. Thus, the systems can be applied to many fields, such as smart grids [1], health care [2], surveillance [3], hazard detection [4], criminal tracking [5], and any system based on networked control, such as cyber-physical systems (CPSs) [6]. The wide application range of mobile services or systems raises concerns that the network infrastructure needs

to be suitable for each mobile service or system type. For example, mobile surveillance systems using drones require wireless interface on physical entities (drones) and access points that carry data from drones to control system [3]. Also, traditional surveillance systems, usually based on closed-circuit televisions (CCTVs), are moving toward wireless internet interfaces [7]. In contrast, mobile systems for smart grids require physical entities to periodically record power usage of each location. However, the physical entities are not required to move around, making wired network easily fit into the system [8]. Reasons to employ wireless networks for the network infrastructure to construct such systems, rather than wired or ad hoc sensor networks, are twofold. One is that many mobile services require proper mobility of physical entities to reduce installation complexity. The other is that

most systems require reliable and qualified communication [9], which is not easily handled by low power ad hoc sensor networks.

On the other hand, emerging networking technologies pose new transmission control protocol (TCP) challenges in terms of performance, requiring analysis, and solutions. TCP enables network management and reliable communication between computing and physical systems. World-wide interoperability for microwave access (WiMAX) and long term evolution (LTE) are widely used cellular packet data networks [10–13]. They have network asymmetry with respect to network performance to satisfy demanding mobile subscribers by allocating them more downlink than uplink bandwidth. This is because mobile subscribers generally download content from the internet.

However, poor downlink throughput has been reported on mobile handsets when downlink and uplink TCP data transfers are simultaneously activated. In such mixed TCP case, TCP ACK packets are delayed by TCP data packets on the uplink, causing considerable downlink TCP retransmissions, as detailed in Section 2. We attempted various TCP configurations, such as TCP window size, delay ACK, selective ACK, buffer size, and TCP autotuning to address this problem, but no setting choices were successful. Some studies have addressed TCP upload or download throughput degradation in asymmetric links, including ADSL, cable, and satellite [14–19]. However, the proposed solutions generally impose additional overhead of maintaining separate packet queues [15–24], modifying TCP protocols or parameters [25, 26], or tracking per-flow traffic statistics [18] on access points (i.e., base stations) or mobile stations. Thus, these solutions are either inappropriate or not sufficiently satisfactory to be applied to emerging wireless technologies.

To address these asymmetric characteristics and their negative effect on TCP throughput, we propose an adaptive TCP regulator called network traffic chunk regulator (NCR). The contributions of this paper can be summarized as follows.

- (i) We show that asymmetry link capacity can have unexpected influence on network performance, particularly TCP performance, and empirically confirm downlink TCP traffic interference from uplink traffic with unexpected throughput degradation.
- (ii) We design and implement NCR to adaptively control TCP according to the degree of asymmetry in link usage and discuss the design for the feedback value based on the traffic volumes for each flow, class, and device.
- (iii) We evaluate NCR through simulations and empirical experiments using real devices, verifying that NCR successfully regulates uplink and downlink traffic to protect downlink performance, significantly increasing aggregated throughput.
- (iv) We show that NCR only requires mobile node management without extra load on managing packet queues in network systems.

Current WLANs commonly use contention based resource management, which is inappropriate to provide

real-time stable communication between control elements and physical entities. Thus, WLAN is not considered as a target network for the solution, and we only consider quality manageable access networks, such as LTE and WiMAX. However, since we cannot access cellular network and driver internal architectures, we emulate the systems with WLAN devices enabled with interface cards.

The remainder of this paper is organized as follows. We present the problem statement based on specific empirical results in Section 2, and then we devise the feedback scheme in Section 3. We explain an extensive set of simulation studies and empirical experimentation in Section 4. Then, we summarize the related work in Section 5. We conclude the paper with Section 6.

2. Problem Statement

TCP is adaptive to the available network bandwidth in a wide variety of networking environments. However, emerging networking technologies pose new challenges for TCP in terms of performance, requiring analysis and solutions. CDMA evolution data only (1xEV-DO), 3GPP high speed packet data access (HSPA), WiMAX, and LTE are widely used cellular packet data networks [10–13]. They have network asymmetry with respect to network performance [27, 28]. To satisfy demanding mobile subscribers, networks allocate more downlink bandwidth than uplink bandwidth because mobile subscribers generally download content from the internet.

However, poor downlink throughput has been reported on mobile handsets for simultaneous downlink and uplink activations. This is because TCP ACK packets are delayed by TCP data packets on the uplink, causing significant TCP retransmissions on the downlink. Poor downlink throughput can also cause control message losses from computing (cyber) elements, breaking real-time physical entity control.

To visualize this problem, we constructed a testbed to emulate an asymmetric access network, configuring two laptops as mobile and base stations, respectively. To emulate cellular networks, we used two IEEE 802.11 wireless interface cards for each station. We configured the routing table to employ one interface card as the downlink channel and the other as the uplink channel. We enabled the ad hoc interface mode for the base station laptop. The specific details were as follows:

- (i) Base station: Fujitsu Lifebook A6010.
- (ii) Mobile station: Fujitsu Lifebook A6010.
- (iii) Uplink interface: Intel 3945 ABG with ipw3945 driver.
- (iv) Downlink interface: 3Com Wireless 11 a/b/g PC card with madwifi driver.
- (v) Operating system: Ubuntu 8.04 LTS with Linux kernel 2.4.24.6.

As for routing table modifications, ipw3945(eth1) was set as the higher uplink routing table entry on the mobile station. On the contrary, madwifi(ath0) was set as the higher downlink routing table entry on the base station. We modified

the routing tables to separate uplink and downlink traffic between the mobile and base stations. However, this meant address resolution protocol (ARP) messages were not sent and received correctly. Therefore, we also modified the ARP table by inserting internet protocol (IP) address and medium access control (MAC) address tuples into the ARP table statically. To ensure network asymmetry, we also set network bandwidth to 54 and 1 Mb/s for the downlink and uplink interfaces, respectively.

We emulated the cellular system using WLAN because we had limited access to manipulate cellular devices and software. We employed IEEE 802.11 as the testbed access network. Thus, the channel access mechanism and bandwidth allocation scheme of the testbed may differ from actual cellular networks. Generally, cellular networks guarantee contention-free channel access and bandwidth allocation for each station. On the contrary, IEEE 802.11 networks provide contention based channel access and bandwidth allocation. However, since the testbed included only a single base and mobile station, the configuration ensured contention-free channel access and bandwidth allocation using IEEE 802.11. Therefore, there was negligible difference between the testbed and cellular networks in terms of network bandwidth, and the testbed was able to successfully emulate network asymmetry.

We measured TCP throughput using the FileZilla [29] FTP application and the *Iperf* tool. *Iperf* is the widely used network testing tool [30]. In principle, FTP and *Iperf* conduct TCP data communication without limiting the data rate. Downlink and uplink data rates were approximately 14 and 1 Mb/s, respectively, for the experiment.

2.1. Single Downlink Throughput. Figure 1 shows the throughput of downlink TCP traffic generated with FileZilla and *Iperf*. Average downlink throughput was approximately 14 Mb/s for both FileZilla and *Iperf*, but FTP throughput was slightly less and more stable than *Iperf*. FTP works in user mode and uses the read system call to access its data from the kernel application buffer, which slows it slightly, resulting in more stable read speed.

2.2. Mixed Downlink and Uplink Throughput. Figure 2 shows throughput for simultaneous downlink and uplink TCP flows. Downlink TCP traffic was activated through the whole experiment, and then uplink traffic was started. There is sudden downlink TCP throughput degradation when uplink TCP traffic commences, reducing from 14 Mb/s to 80 Kb/s, whereas uplink TCP throughput was 1 Mb/s. TCP downlink traffic can fully use the allocated link bandwidth provided TCP ACK packet flow to the sender is timely. However, when TCP uplink traffic commenced, the smaller uplink bandwidth (compared to the downlink) was quickly used up. This caused uplink congestion, and the TCP downlink was unable to receive TCP ACK packets timely enough. Consequently, it could exploit the whole downlink bandwidth, and downlink throughput was degraded. There is some fluctuation in *Iperf* throughput after TCP uplink traffic terminates. This is because the *Iperf* TCP congestion window is not large enough to saturate the downlink once it reduces due to interference with uplink TCP data traffic.

Thus, we verified that uplink and downlink flow interfere with each other such that TCP flow cannot fully use the available downlink bandwidth. Therefore, to resolve the problem, we need to protect downlink TCP flow from uplink TCP flow (or other flows). The problem will become more severe as physical cyber systems generate more uplink traffic from physical entities.

3. Network Traffic Chunk Regulator

This section introduces the proposed network traffic chunk regulator (NCR) to regulate downlink and uplink traffic at the uplink interface of a mobile device. We first present a brief feedback control scheme and explain how to design the feedback value based on traffic volume for each flow, class (uplink or downlink), and device.

3.1. Feedback Control Scheme for Regulation. The NCR uses a feedback value to quantify uplink and downlink traffic volumes. We propose a feedback control scheme that delivers the feedback value to TCP senders in the local stations (uplink traffic) or remote stations (downlink traffic), which allows TCP senders to adjust their sending rates according to the feedback value. Figure 3 shows the overall block diagram of NCR. For every interval, NCR calculates the per-flow, class wide, and device wide feedback values specified in Section 3.2. When a packet arrives at the NCR module, it is identified as downlink or uplink traffic. Then, NCR produces a random number with value uniformly distributed over the feasible range of feedback values, normalized to [0,1], and compares the random value with the feedback value. If the number is less than the feedback value, NCR sets the explicit congestion notification (ECN) bit in the IP header and delivers the packet to the intended sender. Thus, the probability of setting the ECN bit increases as the feedback value increases. TCP senders reduce their congestion window size upon receiving a packet with the ECN bit set or increase the window size if the ECN bit is unset.

Figure 4 shows a conceptual feedback architecture for the proposed NCR. NCR is located between the IP and device drivers. NCR calculates the feedback value using the algorithm from Section 3.2 and then sends the feedback value to each TCP receiver.

3.2. Feedback Value for Regulation. The NCR monitors device wide, class (uplink or downlink) wide, and per-flow traffic and calculates the feedback value for each network or device time interval, T_{net} or T_{dev} , respectively. Since the feedback value is monitored and calculated at given intervals, the computation is also executed at the corresponding intervals. The feedback value, ρ_i , for flow i consists of three components:

$$\rho_i[t] = \rho_f^i[t] + \rho_c[t] + \rho_d[t], \quad (1)$$

where

- (i) ρ_f^i is the difference between the desired and measured per-flow traffic;

FIGURE 1: One-way TCP downlink throughput.

FIGURE 2: Combined TCP downlink and uplink throughput.

FIGURE 3: The overall block diagram of NCR.

FIGURE 4: The feedback architecture of NCR.

- (ii) ρ_c is the class (downlink or uplink) wide difference between desired and measured class wide traffic;
- (iii) ρ_d is the device wide difference between target and measured aggregate traffic load at the uplink interface in the mobile station.

The first two terms in (1) are updated at intervals of T_{dev} , while the last one is updated at intervals of T_{net} .

3.2.1. Per-Flow Feedback Value. To estimate per-flow feedback value, we first assign a weight, φ_i , to each flow, i , which is used to determine traffic for the flow. After determining each flow usage, if there is spare bandwidth in output link, the bandwidth is redistributed to each flow according to φ_i . In the fair condition, all φ_i are equal. Let $R_i(t)$, $i = 1, \dots, n$, be the flow i and $R_i^{in}(t)$, $i = 1, \dots, n$, be the cumulative (cumulative means we regulate flows by overall traffic volume rather than instantaneous traffic rate) input volume for flow i entering the NCR until the t -th interval of T_{dev} . Similarly, let $R_i^{out}(t)$, $i = 1, \dots, n$, be the cumulative output volume for flow i exiting the NCR. In this situation, if $\varphi_i \geq \varphi_j$, then data transmission in time interval $[s+1, t]$ is

$$\frac{R_i^{out}(t) - R_i^{out}(s)}{\varphi_i} \geq \frac{R_j^{out}(t) - R_j^{out}(s)}{\varphi_j}. \quad (2)$$

Therefore,

$$\varphi_j (R_i^{out}(t) - R_i^{out}(s)) \geq \varphi_i (R_j^{out}(t) - R_j^{out}(s)), \quad (3)$$

and summing up for all j , yields

$$\sum_{j=1}^n \varphi_j (R_i^{out}(t) - R_i^{out}(s)) \geq \varphi_i \sum_{j=1}^n (R_j^{out}(t) - R_j^{out}(s)). \quad (4)$$

Since

$$\sum_{j=1}^n (R_j^{out}(t) - R_j^{out}(s)) = c(t-s)T_{dev}, \quad (5)$$

where c is the link capacity and $C(t) = \sum_{i=1}^t c(i)T_{dev}$ is the cumulative capacity in the interval $[1, t]$,

$$R_i^{out}(t) - R_i^{out}(s) \geq \frac{\varphi_i}{\sum_{j=1}^n \varphi_j} c(t-s)T_{dev}. \quad (6)$$

Let $s+1$ be the beginning time slot for the last busy period (busy period means the duration that the uplink interface is continuously used) for flow i up to time t . Then,

$$\begin{aligned} R_i^{out}(t) &\geq R_i^{in}(s) + \frac{\varphi_i}{\sum_{j=1}^n \varphi_j} c(t-s)T_{dev}, \\ &\geq \min_{0 \leq s \leq t} \left[R_i^{in}(s) + \frac{\varphi_i}{\sum_{j=1}^n \varphi_j} c(t-s)T_{dev} \right], \end{aligned} \quad (7)$$

where $R_i^{out}(s) = R_i^{in}(s)$ since all input traffic is transmitted to the output before a new busy period commences. Therefore, each flow i should have at least

$$f_i(s, t) = \frac{\varphi_i}{\sum_{j=1}^n \varphi_j} c(t-s)T_{dev}, \quad (8)$$

during the interval $[s, t]$.

Thus, we can define $\rho_f^i(s, t)$ as

$$\rho_f^i(s, t) = \kappa_f \left(\frac{[R_i^{out}(s, t) - f_i(s, t)]}{R_i^{out}(s, t)} \right), \quad (9)$$

where κ_f is a smoothing factor and $R_i^{out}[t]$ represents the measured traffic of flow i . Equation (9) can be simplified to provide the output volume for flow i for a single time interval,

$$\rho_f^i(t) = \rho_f^i(t-1, t) = \kappa_f \left(\frac{[r_i^{out}(t) - f_i(t)]}{r_i^{out}(t)} \right), \quad (10)$$

where $f_i(t) = f_i(t-1, t) = (\varphi_i / \sum_{j=1}^n \varphi_j) c(t - (t-1))T_{dev}$ and $r_i^{out} = R_i^{out}(t) - R_i^{out}(t-1)$.

3.2.2. *Class Wide Feedback Value.* Since each flow i is assigned weight φ_i , the weight for the aggregate uplink traffic is

$$\varphi_{ul} = \sum_{i \in M_{ul}} \varphi_i, \quad (11)$$

where M_{ul} is the set of all uplink flows. Similarly,

$$\varphi_{dl} = \sum_{i \in M_{dl}} \varphi_i, \quad (12)$$

where M_{dl} is the set of downlink flows.

To prioritize downlink traffic, we apply the constraint $\varphi_{dl} > \varphi_{ul}$, which imposes the additional constraint on each $\varphi_i (\in M_{dl})$. Let $R_{dl}^{in}(t)$ and $R_{ul}^{in}(t)$ be the cumulative input volume for aggregate downlink and uplink traffic, respectively, until the t -th interval of T_{dev} . Similarly, let $R_{dl}^{out}(t)$ and $R_{ul}^{out}(t)$ be the cumulative output volume for aggregate downlink and uplink traffic, respectively. Using the equal procedure of deriving ρ_f^i ,

$$R_{dl}^{out}(t) - R_{dl}^{out}(s) \geq \frac{\varphi_{dl}}{\varphi_{dl} + \varphi_{ul}} c (t - s) T_{dev}. \quad (13)$$

Let $s+1$ be the beginning slot of the last busy period of the aggregate downlink input up to time t . Then, since $R_{dl}^{out}(s) = R_{dl}^{in}(s)$,

$$\begin{aligned} R_{dl}^{out}(t) &\geq R_{dl}^{in}(s) + \frac{\varphi_{dl}}{\varphi_{dl} + \varphi_{ul}} c (t - s) T_{dev}, \\ &\geq \min_{0 \leq s \leq t} \left[R_{dl}^{in}(s) + \frac{\varphi_{dl}}{\varphi_{dl} + \varphi_{ul}} c (t - s) T_{dev} \right]. \end{aligned} \quad (14)$$

Therefore, the aggregate downlink traffic should have at least volume

$$f_{dl}(s, t) = \frac{\varphi_{dl}}{\varphi_{dl} + \varphi_{ul}} c (t - s) T_{dev}, \quad (15)$$

during interval $[s, t]$. Thus, $\rho_c(s, t)$ for aggregate downlink traffic is

$$\rho_c(s, t) = \kappa_c \left(\frac{[R_{dl}^{out}(s, t) - f_{dl}(s, t)]}{R_{dl}^{out}(s, t)} \right), \quad (16)$$

where κ_c is a smoothing factor and $R_{dl}^{out}[t]$ is the measured volume for aggregate downlink traffic. Hence, the feedback value for a single time interval is

$$\rho_c^i(t) = \rho_c^i(t-1, t) = \kappa_c \left(\frac{[r_{dl}^{out}(t) - f_{dl}(t)]}{r_{dl}^{out}(t)} \right), \quad (17)$$

where $f_{dl}(t) = (\varphi_{dl}/(\varphi_{dl} + \varphi_{ul}))cT_{dev}$ and $r_{dl}^{out} = R_{dl}^{out}(t) - R_{dl}^{out}(t-1)$.

Similarly, $\rho_c(s, t)$ and $\rho_c(t)$ for aggregate uplink traffic are

$$\begin{aligned} \rho_c(s, t) &= \kappa_c \left(\frac{[R_{ul}^{out}(s, t) - f_{ul}(s, t)]}{R_{ul}^{out}(s, t)} \right), \\ \rho_c^i(t) &= \kappa_c \left(\frac{[r_{ul}^{out}(t) - f_{ul}(t)]}{r_{ul}^{out}(t)} \right), \end{aligned} \quad (18)$$

where $R_{ul}^{out}(t)$ is the measured aggregate uplink traffic,

$$f_{ul}(t) = \frac{\varphi_{ul}}{\varphi_{ul} + \varphi_{dl}} c T_{dev}, \quad (19)$$

and

$$r_{ul}^{out} = R_{ul}^{out}(t) - R_{ul}^{out}(t-1). \quad (20)$$

3.2.3. *Device Wide Feedback Value.* Let $R(t)$ be device wide cumulative traffic until the t -th interval of T_{net} , $R^{in}(t)$ be the cumulative input volume for aggregate device traffic and $R^{out}(t)$ be the cumulative output volume for aggregate device traffic. Then,

$$R^{out}(t) - R^{out}(s) \geq c (t - s) T_{net}. \quad (21)$$

where c is the link capacity. Thus,

$$\begin{aligned} R^{out}(t) &\geq R^{in}(s) + c (t - s) T_{net}, \\ &\geq \min_{0 \leq s \leq t} [R^{in}(s) + c (t - s) T_{net}], \end{aligned} \quad (22)$$

since $R^{out}(s) = R^{in}(s)$. Therefore, we can define the device wide feedback value as

$$\rho_d(s, t) = R(s, t) - c (t - s) T_{net}. \quad (23)$$

Hence, the feedback value for a single time interval is

$$\rho_d(t) = R(t) - c T_{net}. \quad (24)$$

Consider the effect of a buffer, which can accommodate burst traffic, and feedback value error in calculating $\rho_d(t)$. Since the network interface device usually employs the buffer to accommodate burst traffic, (24) can be extended to

$$\rho_d(t) = R(t) - c T_{net} + \gamma (q(t) - q_{room}), \quad (25)$$

where $q(t)$ is the queue length at the t -th interval of T_{net} , q_{room} is the desired queue length, and $\gamma (> 0)$ is a control gain. However, the feedback value may include error relative to the correct value, which will accumulate over time. Thus, (25) can be expressed as

$$\begin{aligned} \rho_d(t) &= R(t) - c T_{net} + \gamma (q(t) - q_{room}) \\ &+ \beta \sum_{k=0}^{t-1} ((R(k) - c T_{net}) + \gamma (q(k) - q_{room})), \end{aligned} \quad (26)$$

where the last term is the result of summing errors incurred by (25) over all intervals up to t , and $\beta (> 0)$ is a control gain.

Finally, the feedback value with control gain $\alpha (> 0)$ can be expressed as

$$\begin{aligned} \rho_d(t) &= \alpha \left(R(t) - c T_{net} + \gamma (q(t) - q_{room}) \right. \\ &\left. + \beta \sum_{k=0}^{t-1} ((R(k) - c T_{net}) + \gamma (q(k) - q_{room})) \right). \end{aligned} \quad (27)$$

This final feedback value can be used to let traffic senders adjust their transmission rate together with ρ_f and ρ_c . The term $(R(k) - cT_{net})$ in (27) can be approximated as

$$R(k) - cT_{net} = q(k) - q(k-1). \quad (28)$$

With (28), the feedback value from (27) can be more easily estimated using the queue length,

$$\rho_d(t) = \alpha \left[(\gamma + \beta + 1) q(t) - q(t-1) - \gamma q_{room} + \beta \gamma \sum_{k=0}^{t-1} (q(k) - q_{room}) \right]. \quad (29)$$

4. Performance Evaluation

This section details the NCR performance evaluation using simulations and experiment studies. We implemented the proposed NCR as a light interface layer between the IP and MAC layers at the uplink interface.

4.1. Simulation Studies. For the simulation, we employed *ns-2*, *ns-3*, and *OPNET* simulators since each simulates a different cellular network type. We employed *ns-2* to emulate general access networks with separated uplink and downlink, enabling multiple interfaces for a station and using each interface as an uplink or downlink. We also simulated cellular access networks with *OPNET* and *ns-3*. *OPNET* has an elaborate WiMAX module, whereas *ns-3* has a precise LTE module.

Figure 5 shows the network topology we used for the simulation and empirical studies. One downlink and one uplink channel were activated at the mobile station communicating with its peer downlink and uplink server. The downlink channel receives TCP data packets via the downlink interface from the downlink interface of the base station, i.e., its connection peer. In contrast, the downlink channel sends TCP ACK packets via the uplink interface to the uplink interface of the base station, i.e., its connection peer. The uplink channel sends TCP data packets through the uplink interface and receives TCP ACK packets from the downlink interface.

The control parameters of the NCR scheme, α , β , γ , κ_f , and κ_c , were set to 0.03, 0.03, 0.05, 1.0, and 1.0, respectively. Update intervals, T_{net} and T_{dev} , were set to 10 and 10 ms, respectively. Recall that T_{dev} is the update interval to calculate per-flow and class wide feedback value and T_{net} is the update interval to computer device wide feedback value. These parameters were determined from a preevaluation simulation series, a common method for parameter tuning in many research fields. The target for the uplink interface queue was set to 50 packets.

4.1.1. Simulation Study with IEEE 802.11. We implemented NCR in the *ns-2* simulation and subsequently improved the simulation by enabling multiple interfaces in IEEE 802.11-based WLANs and employing each interface as separate

uplink and downlink channels. To ensure network bandwidth asymmetry, we set the downlink interface at 11 Mb/s and uplink interface at 1 Mb/s for both the mobile station and its associated AP. Downlink and uplink data rates were approximately 3.5 and 1 Mb/s, respectively.

The first simulation study with *ns-2* used the following simulation scenario: (i) downlink traffic was active in the entire simulation period [0, 60] s; (ii) uplink traffic was started at 20 s and stopped at 40 s precisely. Figure 6 shows that throughput dynamics between downlink and uplink flow are strongly consistent with the testbed evaluation (Figures 1 and 2). Downlink flow is severely reduced in the presence of uplink flow, with uplink and downlink flows 765 and 870 Kb/s during the period [20, 40] s, whereas downlink flow for periods [0, 20] s and [40, 60] s (before and after uplink flow was activated, respectively) reached 3.29 and 3.35 Mb/s, respectively. When we enabled NCR, downlink flow was protected, reaching 3.29, 3.08, and 3.35 Mb/s for periods [0, 20] s (downlink only), [20, 40] s (uplink flow 164 Kb/s), and [40, 60] s (downlink only), respectively.

As the second simulation study, we evaluated the NCR effectiveness for various uplink and downlink flows, aggregate downlink and uplink throughputs, and average end-to-end delay using the following simulation scenarios: (i) the total simulation time was 60 s; (ii) uplink and downlink flow were active throughout the entire simulation; (iii) the number of downlink flows was varying from 1 to 3, and the number of uplink flows was changing from 1 to 5; (iv) 10 simulation runs were conducted for each simulation configuration. Table 1 compares indicative IEEE 802.11-based WLAN (normal WLAN) and NCR enabled WLAN cases. Downlink and uplink throughput values in the table are aggregate throughput over all the relevant flows, averaged over the 10 simulation runs. The outcomes are consistent with the previous simulation study: uplink flow severely degrades downlink flow, regardless of the number of uplink or downlink flows for normal WLAN conditions, but NCR effectively protects downlink flow.

Figure 7 shows that NCR also positively influences overall network throughput and end-to-end delay. As shown in the figure, the aggregate network throughput for NCR was 2.58 times that for the normal WLAN case (Figure 7(a)). Also, average end-to-end delay for NCR was 6.85 times less than that for the normal WLAN case (Figure 7(b)).

4.1.2. Simulation Study with WiMAX. We implemented NCR in the WiMAX module of the *OPNET* simulator. To verify that the proposed NCR can efficiently manage the network for a long period of time, we conducted a simulation with much longer traffic period. The specific simulation settings were as follows: (i) the downlink station received data from the server during the period [10, 600] s; (ii) the uplink station started sending data to the server at 200 s and stopped at 400 s precisely. We allocated 25% capacity to uplink and the remaining 75% to downlink to ensure network asymmetry and data rate similar to reality. We used the WiMAX characteristics of Vividwireless [31, 32]. Downlink and uplink data rates were approximately 4.5 and 1.5 Mb/s, respectively.

FIGURE 5: Network topology for simulation and experiment studies.

FIGURE 6: TCP throughput comparison between downlink and uplink flow.

FIGURE 7: Aggregate throughput and end-to-end delay for differently configured networks.

TABLE 1: Average uplink (UL) and downlink (DL) throughput for differently configured networks.

# of flows		IEEE 802.11 WLAN		NCR enabled WLAN	
DL	UL	DL (Mb/s)	UL (Mb/s)	DL (Mb/s)	UL (Mb/s)
1	1	0.754	0.771	3.078	0.164
1	2	0.432	0.843	2.988	0.292
1	3	0.391	0.848	2.892	0.386
1	4	0.385	0.810	2.773	0.461
1	5	0.366	0.840	2.594	0.521
2	1	1.347	0.707	5.665	0.105
2	2	1.046	0.771	5.357	0.204
2	3	0.963	0.786	4.884	0.284
2	4	1.199	0.740	4.567	0.333
2	5	0.999	0.778	4.330	0.376
3	1	3.366	0.479	5.799	0.068
3	2	2.929	0.554	5.769	0.137
3	3	2.617	0.598	5.685	0.186
3	4	2.558	0.601	5.439	0.240
3	5	2.148	0.710	5.417	0.266

Figure 8 shows throughput results. Similar throughput was achieved for the normal and NCR enabled WiMAX networks during the period [10, 200] s (no uplink traffic), 4.28 and 4.33 Mb/s, respectively. However, during [200, 400] s (active uplink traffic), normal WiMAX network performance was degraded to 1.52 Mb/s, whereas the NCR enabled network achieved 3.17 Mb/s.

Figure 9 shows TCP round trip times (RTT). As shown in the figure, the RTT of downlink traffic was reduced approximately 47% in NCR enabled WiMAX network. A notable point of the results is that RTT of the uplink traffic was also reduced (28%) when we enabled NCR. Since NCR usually dropped uplink packets and hence could shorten the uplink interface queue, downlink and uplink RTTs were both reduced.

Figure 10 shows aggregate throughput for normal and NCR enabled WiMAX networks. NCR allowed more efficient use of the limited wireless network capacity, with aggregated throughput increasing approximately 28% compared to the normal WiMAX network.

4.1.3. Simulation Study with LTE. We performed the simulation study on LTE network which is the most advanced network with *ns-3*. We employed *ns-3* LTE module of LENA project [33]. We used the same scenario as Section 4.1.2 except that downlink traffic was active [1, 50] s and uplink traffic was active [13, 39] s. We assigned 25 resource blocks to the uplink and 50 blocks to the downlink to ensure network asymmetry and data rate similar to reality. We used Verizon wireless network characteristics from [34]. Downlink and uplink data rates were approximately 30 and 15 Mb/s, respectively.

Figure 11 compares normal and NCR enabled network throughputs. NCR successfully regulated uplink and downlink traffic to protect mobile device downlink performance. Average downlink throughputs were 16.99 and 30.02 Mb/s

during the period [13, 39] s for normal and NCR enabled LTE networks, respectively.

Figure 12 compares aggregated throughputs of normal and NCR enabled LTE network. Similar to the previous simulation results, NCR achieved significantly more efficient link usage. Average aggregated throughputs were 31.36 and 36.19 Mb/s during the period [13, 39] s for normal and NCR enabled LTE networks, respectively. The normal LTE aggregated throughput does not show the sink behavior evident for the previously considered wireless networks. This is because LTE cells provide considerably more uplink bandwidth since many mobile phone applications require duplex communication, e.g., Skype and p2p torrent services. However, NCR still shows performance enhancement because the asymmetric bandwidth problem remains not fully resolved.

Thus, the simulations confirm that the proposed NCR can successfully protect downlink traffic in WiMAX, LTE, and general asymmetric network architectures.

4.2. Empirical Study. For the empirical study, we employed the testbed from Section 2 and implemented NCR in the mobile station. Figure 13 shows the overall NCR software architecture and data flows at the Linux kernel function level. As discussed before, NCR can be implemented as a light interface layer between the IP and MAC layer. Therefore, to implement the layer, we created a virtual device driver and made hooking in `dev_hard_start_xmit()` function to perform packet queuing from IP layer to MAC layer (device driver). Then, we implemented NCR in the virtual device driver.

Figure 14 shows empirical results from experiments using FileZilla and *Iperf* as traffic generators. Downlink and uplink data rates were approximately 14 and 1 Mb/s, respectively. We employed the same evaluation scenarios described in Section 2. Cellular networks cannot protect downlink traffic

FIGURE 8: Comparison of WiMAX network throughputs.

FIGURE 9: Comparison of round trip times (RTT) between normal and NCR enabled WiMAX networks.

from uplink traffic interference as discussed in Section 2. However, NCR secured downlink flow from uplink interference, with average 10.7 and 11.6 Mb/s for FileZilla and *Iperf* through [60, 120] s (uplink and downlink traffic simultaneously enabled). Also, during the period [60, 120] s, average aggregated throughputs were 1.24 and 11.0 Mb/s for normal and NCR enabled devices with FileZilla traffic, respectively. Similarly, during that period, average aggregated throughputs were 1.28 and 11.7 Mb/s for normal and NCR enabled devices with *Iperf* traffic, respectively. Substantially similar results were obtained when we used different combinations of uplink and downlink traffic and thus are omitted for brevity.

5. Related Work

Several previous studies have considered TCP performance degradation in asymmetric networks, but they have usually focused on wired or satellite networks. These solutions cannot be directly applied to cellular or Wi-Fi networks since mobile devices have limited computing power compared to wired

stations. Hence, the solutions cannot be fully implemented within mobile devices. This section summarizes existing solutions for digital subscriber networks and relevant solutions to enforce fairness between the uplink and the downlink stations in WLANs.

5.1. Networked Control System Solutions. Recent trends in networked control and/or filtering systems emphasize network delay effects on the whole system. Rajesh et al. presented simple performance issues of TCP/IP data streams when cyber control was interconnected with physical entities over the network [35]. Zhang et al. surveyed sampling rate, network delay, packet dropping, and communication error effects on networked control [36].

We have not attempted to compare the proposed NCR and these studies. The reasons are threefold: (i) they did not consider asymmetric uplink and downlink capacity assignments in cellular wireless network infrastructures from the mobile station view point; (ii) they simply focused on analytical modeling results under network operation assumptions;

FIGURE 10: Aggregated throughput of normal and NCR enabled WiMAX networks.

and (iii) NCR has already shown it can protect downlink traffic from uplink traffic interference, and hence no further enhancement in downlink usage is expected. The proposed NCR scheme can work with any current or future quality of service (QoS) schemes for mobile services or systems because NCR provides isolated uplink and downlink capacities for them. Thus, schemes using NCR can focus on service fairness or differentiation on top of guaranteed network performance.

5.2. Direct Solutions in Computer Networks. Some previous studies considered uplink or downlink TCP throughput degradation for asymmetric links, such as ADSL, cable, and satellite links. Solutions have mainly proposed maintaining separate TCP data and ACK packet queues [18, 19, 37]. TCP parameters have also been employed for the solution in [38], and queuing delay analysis has been exploited in [18]. Ho et al. proposed a TCP segment delay based solution to solve specific asymmetric link problems [38]. Xu et al. proposed preemptive ACK queueing to completely decouple queues and ensure ACKs no longer experience queueing delay by prioritizing ACK above TCP data [37]. Park et al. proposed ACKs-first variable-size queueing (AFVQ) to reduce excessive queueing delay for TCP ACK packets where TCP performance was degraded due to the asymmetric link [18]. They implemented the proposed AFVQ on a residential ADSL based testbed and showed that the proposed system achieved balanced TCP throughput improvements in both directions. Podlesny and Williamson proposed an asymmetric queueing mechanism to maximize the use of the bottleneck link in residential networks [19]. They showed that the proposed solution was easily deployed and configured in residential networks.

Szilagy and Vulkan proposed an evolved Node B (eNB) based solution called TcpAsym, by packing an additional entity [20]. TcpAsym controls each user equipment (UE) that might contain multiple TCP flows. The entity monitors downlink traffic flow to detect abnormal behavior due to excessive uplink data flow. When asymmetric behavior is identified, it either decreases the uplink traffic congestion

window size or delays uplink ACK transmission or uplink data reception. Priya and Murugan proposed a dual queue approach similar to multiqueue approaches in previous studies but applied optimization schemes for queue size and packet scheduling on each queue [21].

5.3. Indirect Solutions Based on Computer Network Fairness. Several previous studies focused on fairness between downlink and uplink flows in Wi-Fi networks. The main approaches are classified into TCP, MAC, and queue solutions.

As for TCP solutions [25, 26], they allowed an AP to directly handle TCP layer information, such as TCP congestion window size. Zhang et al. constrained TCP sending rate of each uplink mobile node in order to make uplink and downlink flows share bandwidth fairly [25]. Jian et al. proposed another solution that makes an AP directly control TCP traffic whenever the AP detects unfairness between uplink and downlink TCP flows [26].

Mahmoodi et al. [39] and Wang [40] exploited MAC protocol parameters to enforce the fairness among stations. Mahmoodi et al. proposed an orthogonal frequency-division multiple access (OFDMA) system solution to not only achieve optimal TCP throughput but also allocate resources fairly between downlink and uplink [39]. Wang proposed a cross layer adaptive algorithm that dynamically controls the AP minimum contention window size according to the number of users and channel conditions [40]. Nayak et al. analyzed IEEE 802.11ac operated wireless LAN behavior with downlink multiuser MIMO enabled at the access point and showed significant performance degradation [24]. They proposed a method to resolve the degradation, achieving the theoretical TCP traffic limit for the network.

Various solutions have been proposed based on queue management and rate control schemes [17, 41–43]. Hayashi et al. proposed a buffer management scheme that deals with the transmission buffer of the AP to solve unfairness, where the AP removes redundant ACKs in the transmission buffer [41]. Wu et al. implemented a per-flow scheduling scheme with separate TCP data and TCP ACK queues at AP level to achieve per-flow fairness [42]. Huang et al. suggested a queue allocation boundary for the number of ACKs to limit ACK packet greedy behavior that can impair coexisting TCP flow fairness [43]. Xia et al. proposed a dual queue management scheme that maintains different queues for uplink and downlink flows to achieve both fairness and high utilization [17]. Shaik et al. proposed a system that utilizes the hybrid coordination function controlled channel access (HCCA) scheduler for 802.11e Wireless LANs [44]. In contrast to previous studies, the proposed system focused on relative fairness differentiated by required QoS for each service type.

Finally, cross-layer solutions consider TCP and MAC control parameters simultaneously to address problems arising from TCP and MAC interactions [45].

5.4. Remark. The proposed NCR differs from all the above solutions in that (i) it does not maintain separate queues for

FIGURE 11: Comparison of normal and NCR enabled LTE network throughputs.

FIGURE 12: Aggregated throughput of normal and NCR enabled LTE networks.

TCP data and TCP ACK packets, (ii) it does not directly manipulate TCP parameters or MAC parameters, (iii) it does not rely on active queue management, (iv) it does not depend on any function installed onto a base station, and (v) it directly considers cellular network technologies.

Additionally, the proposed NCR solution also has several advantages in that (i) it directly addresses asymmetric uplink and downlink capacity assignments and protects downlink traffic from uplink traffic interference, (ii) when multiple heterogeneous network interfaces are enabled in mobile devices or systems, NCR can isolate downlink from uplink traffic and thus enable any QoS scheme to be implemented on top of multiple networks, and (iii) NCR can differentiate

multiple flows according to service quality requirements by changing the weight (φ_i) for each flow.

6. Conclusion

This paper showed that downlink TCP traffic suffers from performance degradation due to asymmetric links in current cellular networks. We showed experimentally that TCP ACK packets were delayed on the uplink when uplink traffic overruns the available bandwidth, causing inefficient use of the available downlink bandwidth. We proposed the network traffic chunk regulator (NCR) feedback control scheme regulating downlink and uplink traffic in preference of downlink traffic in the congested uplink interface. We implemented NCR for various simulations and also on a Linux platform equipped with two physical wireless interface cards. Simulation and experimental performance evaluations consistently verified that NCR protected downlink TCP traffic from uplink TCP traffic interference. Thus, the proposed NCR scheme can effectively secure downlink traffic against aggressive uplink traffic, significantly increasing total link utilization, and aggregated throughput. Furthermore, NCR does not impose additional overhead on base or mobile stations.

This study focused on resolving significant downlink degradation due to relatively small uplink traffic volumes in asymmetric links. The proposed NCR scheme regulates downlink and uplink traffic at the mobile device's uplink interface, which could degrade uplink traffic volume. However, downlink performance improvement is much larger than the potential uplink degradation. Hence, in principle, NCR would increase total link utilization and aggregated throughput, and this was verified by simulation and experiments. For cases where uplink traffic should be preserved, the user can ensure uplink network quality by simply decreasing

FIGURE 13: Linux kernel architecture and function level flows.

FIGURE 14: Throughput comparison of downlink and uplink for the real tested device.

the NCR feedback value. In addition to the cases, for crowded places with a lot of social activity, the uplink traffic may exceed the downlink traffic, since many people in the places often share videos of events using social networking applications. Thus, to take into account such various situations, we plan to conduct a research on dynamically adjusting control parameters of NCR in future work.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon reasonable request and with permission of funders.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Authors' Contributions

The first four authors contributed equally to this work.

Acknowledgments

This research was supported in part by Unmanned Vehicles Advanced Core Technology Research and Development Program through the Unmanned Vehicle Advanced

Research Center (UVARC) funded by the Ministry of Science, ICT and Future Planning, Republic of Korea (NRF-2016M1B3A1A01937599), and in part by “Human Resources program in Energy Technology” of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) granted financial resource from the Ministry of Trade, Industry & Energy, Republic of Korea (no. 20174030201820).

References

- [1] K. Kim, H. Kim, J. Jung, and H. Kim, “AFAR: A robust and delay-constrained communication framework for smart grid applications,” *Computer Networks*, vol. 91, pp. 1–25, 2015.
- [2] S. Jabbar, F. Ullah, S. Khalid, M. Khan, and K. Han, “Semantic interoperability in heterogeneous IoT infrastructure for health-care,” *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 9731806, 10 pages, 2017.
- [3] D. Afolabi, K. L. Man, H.-N. Liang et al., “A WSN approach to unmanned aerial surveillance of traffic anomalies: Some challenges and potential solutions,” in *Proceedings of the 2013 11th IEEE East-West Design and Test Symposium, EWDTS 2013*, pp. 1–4, September 2013.
- [4] J. Guo, I. Chen, J. J. Tsai, and H. Al-Hamadi, “Trust-based IoT participatory sensing for hazard detection and response,” in *Proceedings of the International Conference on Service-Oriented Computing*, Lecture Notes in Computer Science, pp. 79–84, Springer International Publishing, 2016.
- [5] S. Wang, “The design and implementation of a smartphone and Bluetooth-based criminal tracking system,” 2013.
- [6] M. Karaköse and H. Yetiş, “A cyberphysical system based mass-customization approach with integration of industry 4.0 and smart city,” *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 1058081, 9 pages, 2017.
- [7] J. Hwang and H. Yoe, “Study of the ubiquitous hog farm system using wireless sensor networks for environmental monitoring and facilities control,” *Sensors*, vol. 10, no. 12, pp. 10752–10777, 2010.
- [8] M. Bae, K. Kim, and H. Kim, “Preserving privacy and efficiency in data communication and aggregation for AMI network,” *Journal of Network and Computer Applications*, vol. 59, pp. 333–344, 2016.
- [9] M. Elattar and J. Jasperneite, “Using LTE as an access network for internet-based cyber-physical systems,” in *Proceedings of the 2015 11th IEEE World Conference on Factory Communication Systems, WFCS 2015*, esp, May 2015.
- [10] 3GPP, “3GPP Specifications,” 2002, <http://www.3gpp.com>.
- [11] 3GPP2, “3GPP2 Specifications,” 2001, <http://www.3gpp2.com>.
- [12] S. Sesia, I. Toufik, and M. Baker, *LTE: the UMTS long term evolution*, Wiley Online Library, 2009.
- [13] J. G. Andrews, A. Ghosh, and R. Muhamed, *Fundamentals of WiMAX: understanding broadband wireless networking*, Pearson Education, 2007.
- [14] I. T. Ming-Chit, D. Jinsong, and W. Wang, “Improving TCP performance over asymmetric networks,” *Computer Communication Review*, vol. 30, no. 3, pp. 45–54, 2000.
- [15] F. Louati, C. Barakat, and W. Dabbous, “Handling Two-Way TCP Traffic in Asymmetric Networks,” in *Proceedings of the 7th IEEE International Conference on High Speed Networks and Multimedia Communications*, Z. Mammeri and P. Lorenz, Eds., vol. 3079 of *Lecture Notes in Computer Science*, pp. 233–243, Springer, 2004.
- [16] W. Al-Khatib and K. Gunavathi, “A New Approach to Improve TCP Performance over Asymmetric Networks,” *International Journal of Electronics and Electrical Engineering*, vol. 7, p. 71, 2006.
- [17] Q. Xia, X. Jin, and M. Hamdi, “Dual queue management for improving TCP performance in multi-rate infrastructure WLANs,” in *Proceedings of the IEEE International Conference on Communications, ICC 2008*, pp. 2531–2535, chn, May 2008.
- [18] J. Park, D. Park, S. Hong, and J. Park, “Preventing TCP performance interference on asymmetric links using ACKs-first variable-size queuing,” *Computer Communications*, vol. 34, no. 6, pp. 730–742, 2011.
- [19] M. Podlesny and C. Williamson, “Improving TCP performance in residential broadband networks: A simple and deployable approach,” *SIGCOMM Computer Communication Review*, vol. 42, pp. 62–68, 2012.
- [20] P. Szilagyí and C. Vulkan, “Two-way TCP performance issues and solutions in asymmetric LTE radio access,” in *Proceedings of the 2014 25th IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communication, IEEE PIMRC 2014*, pp. 1603–1608, usa, September 2014.
- [21] S. Sathya Priya and K. Murugan, “Enhancing TCP Fairness in Wireless Networks using Dual Queue Approach with Optimal Queue Selection,” *Wireless Personal Communications*, vol. 83, no. 2, pp. 1359–1372, 2015.
- [22] C. Ahn and S. Chung, “Enhancing WLAN Performance with Rate Adaptation Considering Hidden Node Effect,” *International Journal of Distributed Sensor Networks*, vol. 11, no. 2, p. 940505, 2015.
- [23] W. Kim, “Adaptive Resource Scheduling for Dual Connectivity in Heterogeneous IoT Cellular Networks,” *International Journal of Distributed Sensor Networks*, vol. 2016, Article ID 6036952, 12 pages, 2016.
- [24] P. Nayak, M. Garetto, and E. W. Knightly, “Multi-user downlink with single-user uplink can starve TCP,” in *Proceedings of the 2017 IEEE Conference on Computer Communications, INFOCOM 2017*, usa, May 2017.
- [25] L. Zhang, P. Sénac, E. Lochin, and M. Diaz, “A cross-layer architecture to improve mobile host rate performance and to solve unfairness problem in WLANs,” *Telecommunication Systems*, vol. 53, no. 3, pp. 343–356, 2013.
- [26] Z. Jian, G. Tianhao, L. Jilong, and Y. Mingwu, “Adaptive flow control proxy for enhancing downlink TCP throughput in infrastructure WLANs,” in *Proceedings of the 14th International Symposium on Communications and Information Technologies, ISCIT 2014*, pp. 56–60, kor, September 2014.
- [27] J.-S. Kim, J.-K. Kim, and J.-H. Kim, “Advanced handover scheme considering downlink and uplink service traffic in asymmetric channel,” *Computer Networks*, vol. 89, pp. 1–13, 2015.
- [28] E. Gelenbe and Z. Kazhmagambetova, “Cognitive packet network for bilateral asymmetric connections,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 3, pp. 1717–1725, 2014.
- [29] T. Kosse, FileZilla. filezilla-project.org, since 2001.
- [30] W. Lee and H. Kim, “Channel Quality Estimation for Improving Awareness of Communication Situation in the 2.4 GHz ISM Band,” *IEEE Transactions on Mobile Computing*, 2018.
- [31] B. Winterford, “Vividwireless reveals Perth WiMAX network speed,” IT news, 2010.
- [32] J. Hutchison, “Updated: vividwireless users reveal network speed,” Computer World, 2010.

- [33] N. Baldo, M. Requena-Esteso, M. Miozzo, and R. Kwan, "An open source model for the simulation of LTE handover scenarios and algorithms in ns-3," in *Proceedings of the 16th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM 2013*, pp. 289–298, esp, November 2013.
- [34] L. La, "4G LTE showdown: How fast is your carrier?" CNET news, 2014.
- [35] R. Rajesh, M. Lakshmanan, and V. Noor Mohammed, "Implementation of Networked Control Systems using TCP/IP," *International Journal of Computer Applications*, vol. 18, no. 2, pp. 1–5, 2011.
- [36] X.-M. Zhang, Q.-L. Han, and X. Yu, "Survey on Recent Advances in Networked Control Systems," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1740–1752, 2016.
- [37] Y. Xu, F. Ren, S. Xu, C. Lin, and S. K. Das, "Investigating the interacting two-way TCP connections over 3GPP lte networks," in *Proceedings of the 15th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM 2012*, pp. 39–46, cyp, October 2012.
- [38] C.-Y. Ho, C.-Y. Ho, and J.-T. Wang, "Performance improvement of delay-based TCPs in asymmetric networks," *IEEE Communications Letters*, vol. 15, no. 3, pp. 355–357, 2011.
- [39] T. Mahmoodi, V. Friderikos, and H. Aghvami, "Using traffic asymmetry to enhance TCP performance," *Computer Networks*, vol. 57, no. 1, pp. 317–329, 2013.
- [40] C. Wang, "Achieving per-flow and weighted fairness for uplink and downlink in IEEE 802.11 WLANs," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, 9 pages, 2012.
- [41] K. Hayashi, S. Shioda, N. Komuro, S. Sakata, and T. Murase, "Adaptive pushout: A buffer management scheme to improve TCP fairness in wireless LANs," in *Proceedings of the IEEE 75th Vehicular Technology Conference, VTC Spring 2012*, jpn, June 2012.
- [42] Y. Wu, Z. Niu, and J. Zheng, "Study of the TCP upstream/downstream unfairness issue with per-flow queuing over infrastructure-mode WLANs," *Wireless Communications and Mobile Computing*, vol. 5, no. 4, pp. 459–471, 2005.
- [43] J. Huang, J. Wang, and J. Ye, "A buffer management algorithm for improving up/down transmission congestion protocol fairness in IEEE 802.11 wireless local area networks," *International Journal of Communication Systems*, vol. 27, pp. 2228–2240, 2012.
- [44] M. S. Shaik, A. K. Bhattacharjee, D. Hemadrasa, and R. Kar, "An Integrated Approach to Enhance TCP Fairness in HCCA Scheduler for 802.11e Wireless LANs," *Journal of Advances in Information Technology*, vol. 4, no. 1, pp. 51–57, 2013.
- [45] E.-C. Park, D.-Y. Kim, H. Kim, and C.-H. Choi, "A cross-layer approach for per-station fairness in TCP over WLANs," *IEEE Transactions on Mobile Computing*, vol. 7, no. 7, pp. 898–911, 2008.