RoFi: Rotation-Aware WiFi Channel Feedback

Yongsen Ma, Gang Zhou, *Senior Member, IEEE*, Shan Lin, and Haiming Chen, *Member, IEEE*

***Abstract*—Multiple-input multiple-output (MIMO) provides high throughput for WiFi networks, but it also leads to high overhead due to channel state information (CSI) feedback. Based on experiment measurements, this paper shows that MIMO has different feedback requirements when the receiver is rotating compared with when the receiver is in other mobility scenarios. Experiments of four popular Android games show that device rotation accounts for around 50% of the running time for these games, which implies that rotation-awareness could improve WiFi efficiency significantly for these games. We propose rotation-aware WiFi (RoFi) channel feedback to eliminate unnecessary CSI feedback while maintaining high throughput. We show the failure of existing mobility-aware methods, including CSI similarity, time-of-flight (ToF), and compression noise, in distinguishing the mobility status of rotation and mobile. RoFi calculates power delay profile (PDP) similarity for rotation detection and performs feedback compression and rate selection accordingly. To deal with false rotation detection and status transition between rotation and static, RoFi uses the power of the strongest path, which is calculated from PDP, to further refine CSI feedback when necessary. The RoFi design is compatible with legacy 802.11 protocols and is easy to be deployed on existing WiFi systems. Evaluation results show that RoFi reduces 25%–40% overhead with negligible signal-to-noise ratio decrease in rotation scenarios. RoFi also consumes 29%–69% less energy compared with state-of-the-art feedback compression and rate selection algorithms.**

***Index Terms*—Channel state information, correlation coefficient, energy efficiency, Internet of Things, multipath channels, multiple-input multiple-output (MIMO), wireless LAN, wireless networks.**

I. Introduction

WiFi HAS a very rapid growth with the increasing popularity of wireless devices and the growing demands of wireless data traffic. Multiple-input multiple-output (MIMO) is one of the key technologies for WiFi to achieve high throughput. Specifically, 802.11n employs single-user MIMO to improve the receiving signal-to-noise ratio (SNR) and data rates [1]. 802.11ac uses multiuser MIMO, which allows transmitting multiple packets concurrently to different receivers, to further improve throughput [2]. Both 802.11n and 802.11ac employ transmit beamforming to improve SNR by concentrating radio energy on the targeted receivers. Furthermore, MIMO provides channel state information (CSI) per subcarrier, which is used for combating multipath and frequency-selective fading effects, to accurately predict packet delivery ratio (PDR) and select the best transmission strategies [3], [4].

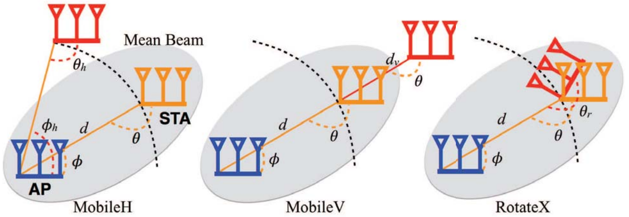


Fig. 1. Transmit beamforming is impacted differently when the STA is in different mobility scenarios. ϕ is the AoD, θ is the AoA, and d is the distance from AP to STA. For MobileH, ϕ is changed to ϕ_h and θ is changed to θ_h. For MobileV, d is changed to d_v. For RotateX, θ is changed to θ_r. The STA remains in the mean beam for RotateX, but not for MobileH and MobileV.

However, CSI introduces high measurement and feedback overhead for WiFi, especially for mobile and handheld devices. The WiFi access point (AP) needs CSI measurement and feedback to calculate the beamforming matrix and select the best transmission strategies. The transmission time for data packets is dramatically sacrificed for sending CSI and control packets, since the size of CSI grows rapidly as the number of antennas and channel width increase. Multiuser MIMO has even higher overhead since it needs higher frequency of CSI measurements and feedback to deal with interuser interference [5]. Moreover, the WiFi station (STA) consumes much energy for sending CSI feedback to the AP. The CSI feedback overhead accounts up to 91% of the total energy consumption of WiFi receivers.[1] Thus it is crucial to eliminate unnecessary CSI feedback, especially for mobile and handheld devices, because they are typically battery powered.

For WiFi networks with transmit beamforming enabled, the AP needs to steer the signal to the direction of the STA, so it has different feedback requirements if the STA is in different mobility scenarios. For instance, the AP does not need frequent CSI feedback for the STA that is only rotating, such as a mobile device running games that only require device rotation. As shown in Fig. 1, the distance and angle of departure (AoD) between the AP and STA do not change if the STA is rotating along the *x*-axis [marked as RotateX, shown

Manuscript received March 22, 2017; revised June 6, 2017; accepted June 13, 2017. Date of publication June 23, 2017; date of current version October 9, 2017. This work was supported in part by the U.S. National Science Foundation under Grant CNS-1253506 (CAREER) and Grant CNS-1553273 and in part by the Startup Research Fund of Ningbo University. *(Corresponding author: Yongsen Ma.)*

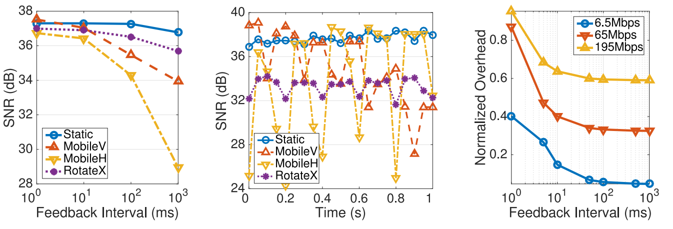
Y. Ma and G. Zhou are with the Department of Computer Science, College of William and Mary, Williamsburg, VA 23187-8795 USA (e-mail: yma@cs.wm.edu).

S. Lin is with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794-2350 USA.

H. Chen was with the Department of Computer Science, College of William and Mary, Williamsburg, VA 23187-8795 USA. He is now with the Department of Computer Science, Ningbo University, Ningbo, China.

Digital Object Identifier 10.1109/JIOT.2017.2719621

[1] The result is calculated by energy consumption measurements of the Intel 5300 WiFi chipset with data packet of 1500 bytes. The calculation and parameter settings are shown in (12) in Section III-D.



(a) (b) (c)

Fig. 2. SNR results of the STA show different feedback requirements when the STA is in different mobility scenarios. (a) For feedback interval of 1000 ms, the SNR decrease is 8 dB for MobileH and 1.2 dB for RotateX. (b) RotateX has more stable SNR variations than MobileH and MobileV. (c) Normalized overhead is reduced by 85%–94% for feedback interval from 1 to 100 ms.

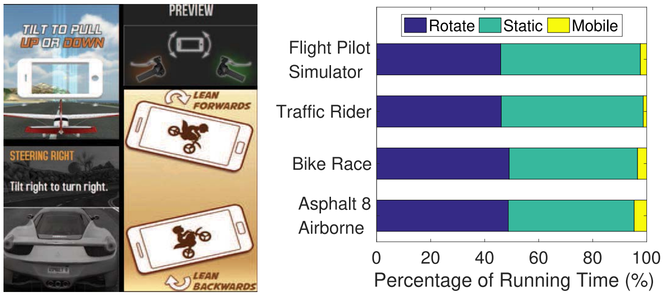
TABLE I
WIRELESS SYSTEMS THAT REQUIRE DEVICE ROTATION

Devices/Applications	Examples
Wireless cameras	Netgear Arlo [15]; Homeboy [16]; Logitech Circle [17]
Home and industrial robots	iRobot [18]; Double Robotics [19]; Dyson 360 Eye [20]
Smartphone and tablet applications; racing and simulation games	Flight Pilot Simulator [6]; Traffic Rider [7]; Asphalt 8 Airborne [8]; Bike Race [9]
Wireless Virtual Reality headsets and handheld controllers	Rivvr [21]; DisplayLink [22]; HTC VIVE [23]
Wireless drones, remote controllers/monitors, and first person view headsets	DJI [24]; Parrot [25]; Yuneec [26]

in Fig. 4(b)], but either one changes if the STA is moving vertically (MobileV) or horizontally (MobileH) to the circle around the AP. The AP has very different CSI feedback requirements when the STA is rotating compared with when it is moving or static.

If the STA sends CSI feedback only when it is needed, the CSI feedback overhead can be significantly reduced while maintaining high throughput. Fig. 2(a) shows SNR results of the STA with different feedback intervals in different mobility scenarios. For RotateX, the AP is able to tolerate long feedback intervals with negligible SNR decrease for the STA. Besides, the STA has more stable SNR variations when it is rotating than when it is moving, as shown in Fig. 2(b). If the STA is rotating, the normalized overhead, which is computed as the ratio of transmission time for control packets to the total transmission time, can be reduced by 85%–94% by using feedback interval of 100 ms, as shown in Fig. 2(c). Therefore, different feedback intervals and transmission strategies should be used if the STA is in different mobility scenarios.

There are many mobile and Internet of Things systems that require wireless connections and device rotation at the same time, as shown in Table I. For example, some wireless cameras need to rotate to get a better view angle. Home and industrial robots need rotation for certain tasks. Wireless virtual reality devices sometimes require the user to rotate the headset or handheld controller. Wireless drones sometimes rotate because of in-device or remote control commands; remote controllers/monitors of drones also rotate in some cases. We run four racing and simulation games [6]–[9] on an Android smartphone and show the percentage of the running time of different mobility types in Fig. 3(b). The total running time for each game is about 20 min. The mobility status of the smartphone is detected by the geomagnetic field sensor and accelerometer every 5 ms. For each game, the device is in the rotation state for about 50% of the running time. Thus, it is necessary to distinguish whether the device is rotating in the running time, considering different CSI feedback requirements in different mobility scenarios.



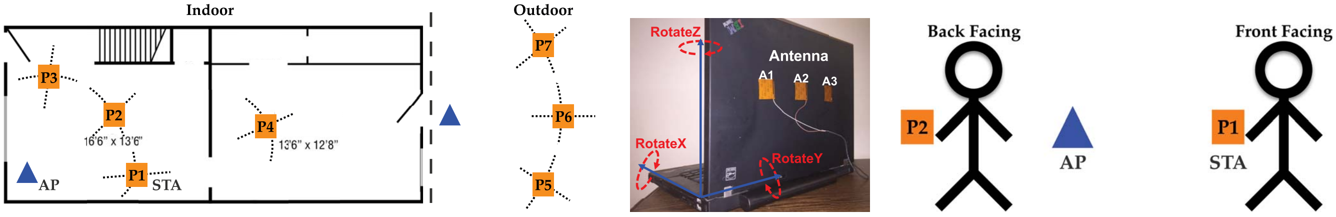
(a) (b)

Fig. 3. Some game applications need users to rotate the device. (a) Games that require device rotation. Top-left: flight pilot simulator [6]; top-right: traffic rider [7]; bottom-left: asphalt 8 airborne [8]; and bottom-right: bike race [9]. (b) Percentage of running time of each mobility type for each game.

Existing mobility-aware metrics, such as CSI similarity [4], [10], time-of-flight (ToF) [10]–[13], and compression noise [14], cannot distinguish rotation from other mobility scenarios. CSI similarity and ToF are used for mobility-aware rate selection in [10]. Experiments show no significant difference for CSI similarity in rotation and mobile scenarios. ToF results are also similar when the STA remains static, rotates locally, or moves horizontally to the circle around the AP, since the distance between the AP and STA does not change for these three scenarios. Compression noise is used to adjust feedback compression levels for mobile and static scenarios in [14], but experiments show indistinguishable compression noise results for rotation and mobile scenarios. For these three metrics, the AP still needs per-packet CSI feedback if the STA is rotating. Therefore, rotation detection is needed to eliminate unnecessary CSI feedback. The challenge is how to detect STA rotation just based on CSI and how to give efficient CSI feedback in different mobility scenarios.

We propose rotation-aware WiFi (RoFi) channel feedback to eliminate unnecessary CSI feedback by addressing this challenge. RoFi uses power delay profile (PDP) similarity to distinguish device rotation from other mobile scenarios. The STA sends CSI to the AP with the proper feedback interval according to the mobility detection result. The STA calculates the power of the strongest path (PSP) from PDP to refine CSI feedback when the STA is detected in the status of rotation and static. The AP calculates the beamforming matrix and selects the data rate based on the most recent CSI feedback. In summary, we make the following contributions.

1) We conduct CSI measurements and show that the AP has different CSI feedback requirements when the STA is in the mobility status of rotation, mobile, or static.
2) We show the failure of CSI similarity, ToF, and compression noise, in distinguishing rotation from other



(a) (b) (c)

Fig. 4. Experiment setup for CSI measurements in different mobility scenarios. (a) Indoor and outdoor deployments. (b) Rotation directions. (c) STA placements with or without human blocking.

mobility scenarios. Therefore, we propose PDP similarity to detect the mobility status of the STA by just using CSI.

3) We present rotation-aware CSI feedback, which reduces unnecessary CSI feedback with negligible SNR decrease, to improve the performance and efficiency of WiFi STAs.

The RoFi design does not need frame format modifications and is compatible with legacy 802.11 protocols. RoFi is evaluated with CSI traces collected in different mobility scenarios. Performance metrics, including overhead, throughput, and energy consumption, are used to compare RoFi with state-of-the-art feedback compression and rate selection algorithms. For fixed data rates, RoFi reduces 7%–38% feedback overhead in different mobility scenarios, and the maximum SNR decrease introduced by RoFi is lower than 1 dB. RoFi also provides up to 52% higher throughput and 48% lower energy consumption. In rotation scenarios, with rate selection enabled, RoFi has up to 22% higher throughput and 47% less energy consumption than existing rate selection algorithms that do not use CSI.

The rest of this paper is organized as follows. Section II gives the motivation of RoFi with SNR measurements in different mobility scenarios. Section III presents the RoFi design, including rotation detection and rotation-aware CSI feedback. Evaluation results of overhead, throughput, and energy consumption are shown in Section IV. Section V summaries related works, and Section VI concludes this paper.

II. Motivation

This section presents experiment measurements to analyze receiving SNR of the STA in different mobility scenarios. We found that rotation needs to be separately addressed to eliminate unnecessary CSI feedback.

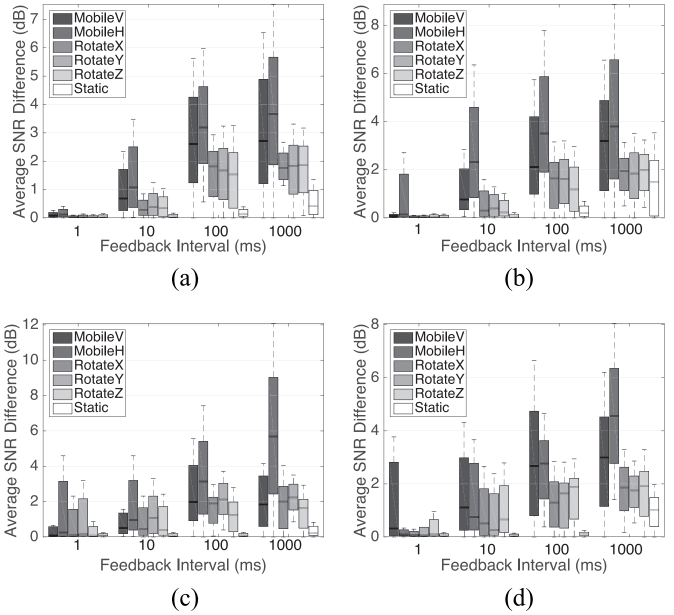
A. Experimental Setup

We conduct CSI measurements using Intel WiFi Link 5300 and 802.11n CSI tool [27] in various real-world scenarios. Deployment locations of the AP and STA are shown in Fig. 4(a). Indoor and outdoor experiments are conducted separately, and there is only one AP and one STA at the same time. At each STA position, i.e., P1–P7, the STA moves vertically (MobileV) or horizontally (MobileH) to the circle around the AP, with the speed of about 1.2 m/s. The STA rotates along x/y/z-axis (RotateX/Y/Z), as shown in Fig. 4(b), or remains static (Static). The rotation speed for RotateX/Y/Z is about 180 degrees per second. Mobile stands for either MobileV or MobileH, and Rotate represents either RotateX, RotateY, or RotateZ. For each mobility scenario, CSI is measured with or without human blocking, as shown in Fig. 4(c). CSI measurements for each scenario at each position are repeated for at least 20 times.

The WiFi AP and STA operate at 5 GHz, and the channel width is 20 MHz. The AP has three external antennas. The STA has three internal antennas spaced 3-in apart, which can be installed on smartphones and tablets, as shown in Fig. 4(b). The transmitting power of the AP is fixed at 17 dBm, and there are no other interference sources. The AP continuously sends packets to the STA, which collects CSI measurements about every 0.5 ms. Each received packet has a preamble that contains training symbols for calculating the transmitted signal X. When the STA receives the packet, it gets the corresponding received signal Y. The STA calculates the feedback CSI H_f for each subcarrier by the MIMO channel model $Y = H_fX + N$, where N is the noise signal. Note that 802.11n CSI tool [27] only provides CSI values of 30 subcarriers even though a 20-MHz WiFi channel has 52 subcarriers [1], [2], [5], [28]. H_f is sent back to the AP to calculate the beamforming matrix Q for transmitting data packets.

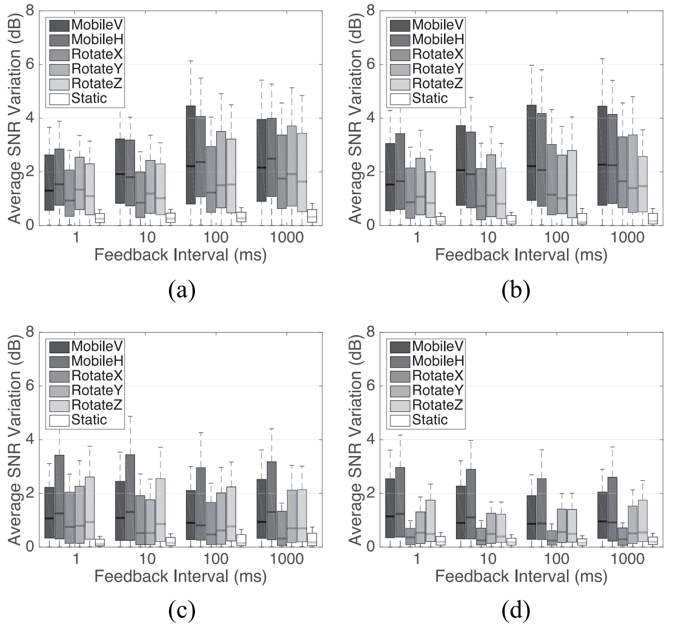
For a data packet, the transmitted signal is QX instead of X. The AP calculates Q as a function of H_f to map X to different spatial streams, so that it can steer the radio signal to the target receivers. In zero forcing beamforming (ZFBF) [14], [29], which is widely used for both single- and multiuser beamforming, the beamforming matrix is $Q = H_f^*(H_fH_f^*)^{-1}$, where $(\cdot)^*$ is the conjugate transpose operation. Now the channel model for data packet transmission is $Y = H_dQX + N$, where H_d is the CSI matrix measured by the data packet. Note that there is a time interval between H_f and H_d. After receiving Y, the STA uses minimum mean square error (MMSE) [3], [4], [29] to decode the received signal. The SNR for the kth subcarrier of the jth spatial stream is $\mathrm{snr}_{k,j} = 1/Y_{jj} - 1$, where $Y = (H_k^*H_k+I_S)^{-1}$, $H_k = H_dQ$ is the effective CSI of subcarriers k for the ZFBF transmitter, and I_S is a $S \times S$ identity matrix with $S = \min(N_t, N_r)$ as the maximum number of streams supported by the MIMO channel [3], [4]. The difference between H_d and H_f introduces beamforming errors to ZFBF and influences the receiving SNR for the STA. The receiving SNR at time t with feedback interval δ is

$$\mathrm{snr}(t, \delta) = db\Big(\sum \mathrm{snr}_{k,j}/\sqrt{S}\Big) \quad (1)$$



(a) (b) (c) (d)

Fig. 5. Rotate has smaller SNR differences than MobileH and MobileV. The average SNR difference of rotate is less than 2 dB for feedback interval of 100 ms, while MobileV and MobileH have much larger SNR differences. (a) Indoor, front facing. (b) Indoor, back facing. (c) Outdoor, front facing. (d) Outdoor, back facing.



(a) (b) (c) (d)

Fig. 6. Rotate has more stable SNR variations than MobileH and MobileV. The average SNR variation of rotate is about 0.5–1 dB lower than that of mobile for different feedback interval settings in different scenarios. (a) Indoor, front facing. (b) Indoor, back facing. (c) Outdoor, front facing. (d) Outdoor, back facing.

where δ is the time interval between H_d and H_f, and $\sqrt{S}$ is the scaling factor [27].

B. Measurement Results

Next, we show SNR results in terms of feedback interval and time in different mobility scenarios.

1) SNR Versus Feedback Interval: Fig. 2(a) shows SNR results with different feedback intervals in different mobility scenarios. *Rotate has much smaller SNR differences when using long feedback intervals compared with mobile*. For MobileV, SNR with feedback interval of 1 ms is about 3 dB higher than that of 10 ms. MobileH has 8 dB lower SNR when feedback interval is changed from 1 to 1000 ms. For rotate, there is no significant SNR difference for feedback intervals less than 100 ms. To quantify the impact of feedback intervals in different scenarios, we define SNR difference as

$$\mathrm{snrdiff}(t, \delta) = \mathrm{snr}(t, 0) - \mathrm{snr}(t, \delta) \tag{2}$$

where $\mathrm{snr}(t, \delta)$ is the SNR at time t with feedback interval δ and is calculated by (1). Here $\mathrm{snr}(t, 0)$ represents the optimal SNR at time t without feedback delay, which means that H_f and H_d are measured at the same time, i.e., $H_f = H_d$.

Fig. 5 shows the average SNR difference for different mobility scenarios. The average SNR difference for rotate is less than 2.1 dB. For a certain feedback interval, rotate has much smaller SNR differences than mobile. Thus, if the STA is rotating, it should choose a long feedback interval, e.g., 100 ms, to reduce CSI feedback overhead. In other words, rotation-awareness could significantly reduce feedback overhead with negligible SNR decrease.

2) SNR Versus Time: Fig. 2(b) shows SNR variations over time for different mobility scenarios. *Rotate has more stable and predictable SNR variations compared with mobile*. Both MobileV and MobileH have random and large SNR variations. At 0.6 s, for example, the next SNR after 50 ms changes 7 and 9 dB, respectively, for MobileV and MobileH. However, SNR variations are within 1 and 2 dB for static and RotateX, respectively. To quantify statistical results of SNR variations for all mobility scenarios, we define SNR variation as

$$\mathrm{snrvari}(t, \delta) = |\mathrm{snr}(t + \Delta t, \delta) - \mathrm{snr}(t, \delta)| \tag{3}$$

where $\mathrm{snr}(t, \delta)$ is the SNR at time t with feedback interval δ, and Δt is the time interval between two SNR measurements.

Statistical results of SNR variations of different mobility scenarios are shown in Fig. 6. The measurement interval Δt is 50 ms. For indoor, the average SNR variation of rotate is about 0.5–1 dB lower than that of mobile. For example, the average SNR variation of RotateX is 1 dB lower than that of mobile for feedback interval of 100 ms. For outdoor, SNR variations of both rotate and mobile are smaller than that of indoor. The average SNR variation for mobile and rotate slightly increases for indoor but remains almost the same for outdoor, as the feedback interval increases from 1 to 1000 ms.

To sum up, rotate has smaller SNR differences and SNR variations than mobile. Thus, WiFi should select different CSI feedback and transmission strategies for rotate and other mobility scenarios so as to improve the performance and efficiency of WiFi STAs. For this purpose, we are motivated to propose RoFi channel feedback.

III. RoFi Design

This section presents RoFi design and how it can be used to optimize feedback compression and rate selection.

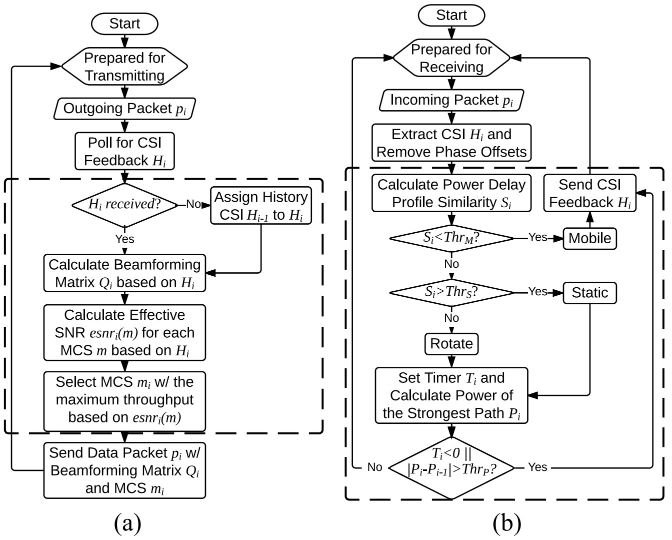

Start
Prepared for Transmitting
Outgoing Packet p_i
Poll for CSI Feedback H_i
H_i received?
No
Assign History CSI H_{i-1} to H_i
Yes
Calculate Beamforming Matrix Q_i based on H_i
Calculate Effective SNR $esnr_i(m)$ for each MCS m based on H_i
Select MCS m_i w/ the maximum throughput based on $esnr_i(m)$
Send Data Packet p_i w/ Beamforming Matrix Q_i and MCS m_i
(a)
Start
Prepared for Receiving
Incoming Packet p_i
Extract CSI H_i and Remove Phase Offsets
Calculate Power Delay Profile Similarity S_i
Send CSI Feedback H_i
S_i<Thr_M?
Yes
Mobile
No
S_i>Thr_S?
Yes
Static
No
Rotate
Set Timer T_i and Calculate Power of the Strongest Path P_i
T_i<0 \|\| \|P_i-P_{i-1}\|>Thr_P?
No
Yes
(b)

Fig. 7. RoFi design with added components in dashed rectangles. (a) AP calculates beamforming matrix and selects MCS by the latest CSI feedback. (b) STA detects mobility types by PDP similarity. For mobile, it sends CSI feedback for each packet. For rotate or static, if the time interval since last CSI feedback is greater than 50 or 100 ms, or the change of PSP is larger than Thr$_P$, the STA sends CSI feedback. Otherwise, there is no CSI feedback.

A. RoFi Overview

The overview of RoFi design for the AP and STA is shown in Fig. 7. When the AP has an outgoing data packet p_i for the STA, it first notifies the STA to measure the current CSI H_i and then polls CSI feedback from the STA. If the AP does not receive the CSI feedback, it assigns history CSI H_{i-1} as the current CSI H_i. The AP calculates beamforming matrix Q_i and effective SNR $\text{esnr}_i(m)$ for each modulation and coding scheme (MCS) index m based on H_i. The AP selects the MCS index m_i with the maximum throughput based on $\text{esnr}_i(m)$. Finally, the AP sends the data packet to the STA using beamforming matrix Q_i and MCS index m_i.

The STA extracts CSI H_i from the CSI measurement packet. Based on H_i, the STA calculates PDP similarity S_i to detect whether the STA is in the status of mobile, rotate, or static. If it is mobile, the STA sends CSI feedback to the AP for each data packet. If it is rotate or static, the STA calculates the PSP P_i based on PDP $h_i(t)$. The STA only sends CSI feedback when the change of PSP is larger than a threshold Thr$_P$, or the time interval since the previous CSI feedback is greater than 50 and 100 ms for rotate and static, respectively.

B. Rotation Detection

1) Existing Methods: There are three mobility-aware methods using CSI similarity [4], [10], [30], compression noise [14], and ToF [10]–[13]. However, we found that none of these three methods is able to tell whether the STA is in the status of rotate, as shown in Fig. 8.

CSI similarity is calculated as

$$CS_i = \frac{\sum_{k=1}^{N_s}\left(h_i(k)-\overline{h_i}\right)\left(h_{i-1}(k)-\overline{h_{i-1}}\right)}{\sqrt{\sum_{k=1}^{N_s}\left(h_i(k)-\overline{h_i}\right)^2}\sqrt{\sum_{k=1}^{N_s}\left(h_{i-1}(k)-\overline{h_{i-1}}\right)^2}} \qquad (4)$$

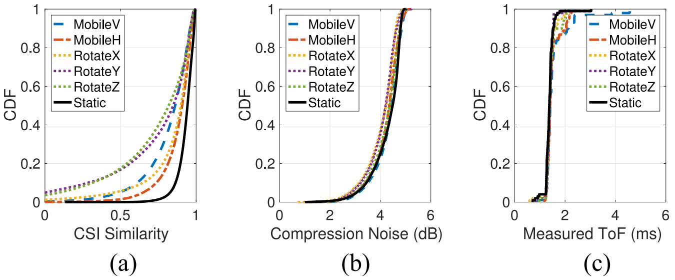


Fig. 8. Neither CSI similarity nor compression noise is able to distinguish whether the STA is in the status of rotation or mobile. (a) CSI similarity. (b) Compression noise. (c) Measured ToF.

where $h_i(k)$ is the CSI magnitude of the kth subcarrier, and $\overline{h_i}$ is the average CSI magnitude across N_s subcarriers of the ith packet [4], [10], [30]. CSI similarity can detect static, but it can hardly distinguish rotate from mobile, as shown in Fig. 8(a).

Compression noise is defined as

$$CN_i = \sum_{k=1}^{K}\left|(H_i(k) - H_{i-1}(k))(H_i(k) - H_{i-1}(k))^*\right| \qquad (5)$$

where $H_i(k)$ is the CSI value of the kth subcarrier of the ith packet [14]. Static, mobile, and rotate show indistinguishable compression noise results, as shown in Fig. 8(b).

The measured ToF tof_m between the data and ACK packet is given by

$$\text{tof}_m = 2 * \text{tof}_a + t_{\text{SIFS}} + t_{\text{ACK}} \qquad (6)$$

where tof_a is the propagation time of the radio signal, t_{SIFS} is the short interframe space (SIFS) time between the data and ACK packet, and t_{ACK} is the transmission time for the ACK packet [12], [13]. tof_m is measured by the elapsed time from the departure time of the data packet to the arrival time of the ACK packet. The detail of how to measure tof_m can be found in [12] and [13]. Fig. 8(c) shows that the measured ToF is not able to distinguish rotate from either static or MobileH.

2) Proposed Method: We propose PDP similarity to detect the mobility status of the STA. Since the AoD and distance (shown in Fig. 1) between the AP and STA remain unchanged for rotate while either one changes for MobileV and MobileH, rotate and mobile should have different multipath fading results. PDP characterizes multipath channel dynamics of MIMO channels, so PDP similarity provides better rotation detection results than CSI similarity, ToF, and compression noise.

PDP is the time-domain transformation of channel frequency response by applying inverse fast Fourier transformation on the frequency-domain CSI [31], [32]. The corresponding PDP of CSI $H(f)$ is $h(t) = \sum_{k=1}^{K}\alpha_k\delta(t-\tau_k)$, where K is the number of paths, α_k and τ_k are the attenuation and delay for path k, respectively. $\delta(\cdot)$ is the delta function. The norm of $h(t)$, $\|h(t)\|_2$, represents the signal strength of each path along which the transmitted signal arrives at the receiver with different time delays. Let $f_i(k) = \|\alpha_k\delta(t-\tau_k)\|_2$ be the signal strength of the kth path of the PDP derived from the ith

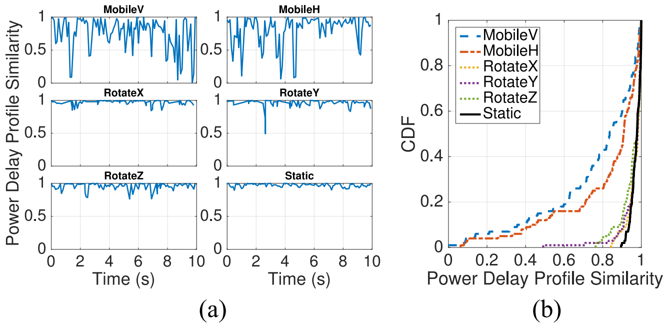


Fig. 9. PDP similarity of different mobility traces. 90% PDP similarity are larger than 0.9 for rotate and 0.95 for static, while 60% are smaller than 0.9 for mobile. (a) One example of PDP similarity. (b) CDF of PDP similarity.

packet, then the PDP similarity between the ith and $(i-1)$th packet is

$$S_i = \frac{\sum_{k=1}^{K}\left(f_i(k)-\overline{f_i}\right)\left(f_{i-1}(k)-\overline{f_{i-1}}\right)}{\sqrt{\sum_{k=1}^{K}\left(f_i(k)-\overline{f_i}\right)^2}\sqrt{\sum_{k=1}^{K}\left(f_{i-1}(k)-\overline{f_{i-1}}\right)^2}} \qquad (7)$$

where $\overline{f_i}$ is the average PDP norm of the ith packet.

Fig. 9 shows the CDF of PDP similarity in different mobility scenarios. The time interval between two adjacent packets is 100 ms. The PDP similarity for MobileV and MobileH is much lower than that of rotate and static. This means that the multipath channel of mobile is less stable than that of rotate and static. We hence use different thresholds of PDP similarity to distinguish mobile, rotate, and static. Since 90% of PDP similarity are larger than 0.9 for rotate and 0.95 for static, while 60% are smaller than 0.9 for mobile, we use the threshold setting of $\text{Thr}_S = 0.95$ and $\text{Thr}_M = 0.9$. If the PDP similarity S_i is greater than the threshold Thr_S, the STA is detected as static; if S_i is smaller than Thr_M, the STA is detected as mobile; otherwise the STA is detected as rotate.

C. Rotation-Aware Channel Feedback

The STA determines the CSI feedback interval based on the rotation detection result. For mobile, the STA sends CSI feedback for each packet. For rotate and static, the feedback interval is 50 and 100 ms, respectively. If the rotation detection result is changed, the STA resets the feedback timer T_i to 50 or 100 ms. Otherwise, the STA checks the feedback timer T_i. If $T_i > 0$, the STA changes to receiving state without sending CSI feedback; otherwise the STA sends CSI feedback and transforms to receiving state. The reason for selecting feedback interval of 50 and 100 ms is that it has a good tradeoff of feedback overhead and SNR. As shown in Fig. 2(c), the normalized overhead is significantly reduced using feedback interval of 50 ms, but it does not change much when the feedback interval is larger than 50 ms. The average SNR decrease for rotate is less than 2 dB by choosing feedback interval 50 ms. For static, the average SNR decrease is less than 1 dB for feedback interval of 100 ms, as shown in Fig. 5.

The AP calculates the beamforming matrix Q_i and selects the MCS index m_i using CSI feedback H_i before sending packet p_i. If no CSI feedback for packet p_i is received, the AP uses history CSI H_{i-1} as the current CSI H_i. In this paper, ZFBF is used as the beamforming algorithm, i.e., $Q_i = H_i^*(H_iH_i^*)^{-1}$. The AP calculates effective SNR (eSNR) [3], [4] for each MCS index using H_i, and selects the MCS m_i with the maximum achievable throughput by solving

$$\begin{aligned} &\arg\max_{m} \quad \text{pdr}_i(m) * \text{rate}(m) \\ &\text{subject to} \quad \text{pdr}_i(m) > \text{Thr}_{\text{pdr}}(m), \quad 0 \le m \le m\text{Max} \end{aligned} \qquad (8)$$

where $\text{pdr}_i(m)$ is the PDR using MCS m calculated before transmitting packet p_i, $\text{Thr}_{\text{pdr}}(m)$ is the corresponding PDR threshold, $\text{rate}(m)$ is the theoretical data rate of MCS m, and mMax is the maximum MCS index. For a 20-MHz MIMO channel with three transmitting antennas ($N_t = 3$) and three receiving antennas ($N_r = 3$), the maximum MCS is $m\text{Max} = 23$. The AP predicts $\text{pdr}_i(m)$ based on the eSNR threshold Thr_{esnr}, above which $\text{pdr}_i(m)$ is larger than Thr_{pdr}, i.e., $\text{pdr}_i(m) > \text{Thr}_{\text{pdr}}$ if $\text{esnr}_i(m) > \text{Thr}_{\text{esnr}}$, for each MCS index m. After calculating the beamforming matrix Q_i and selecting the MCS index m_i, the AP sends the data packet to the STA using Q_i and m_i.

The threshold-based rotation detection algorithm sometimes classifies mobile as rotate or static, since PDP similarity of mobile could be greater than 0.9 in some cases, as shown in Fig. 9. Consequently, the STA does not send CSI feedback, while it is needed for the AP. Furthermore, rotate has small SNR differences and stable SNR variations only during the rotation process but not at the beginning or end of rotation, in which cases CSI feedback is still needed for rotate. Thus, the STA needs to send CSI feedback to the AP when necessary if the STA is detected as rotate or static.

To further refine the aforementioned CSI feedback design, we here define the PSP as $P_i = \max f_i(k),\ 1 \le k \le K$, where $f_i(k)$ is the signal strength of the kth multipath component from the PDP norm of the ith packet, and K is the total number of multipath components. Fig. 10(a) shows one example of PSP and SNR difference for different mobility traces. For rotate, *there is a negative relation between PSP and SNR difference*: if PSP remains stable, SNR difference is very low; if PSP decreases a lot, SNR difference increases accordingly.

PSP also has a close relation with SNR variation since the strongest path contributes the most to the receiving SNR. Rotate has stable SNR variations, and it should have less

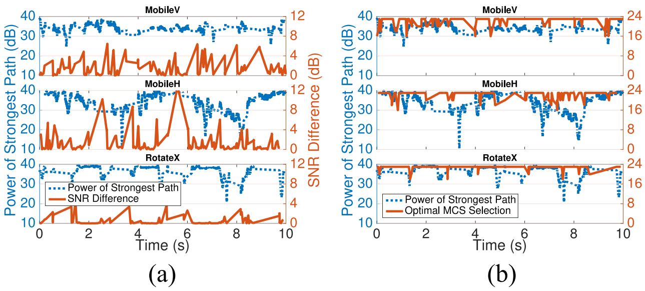


Fig. 10. PSP is a good indicator of SNR difference and the optimal MCS selection for rotate traces, but not for mobile traces. (a) PSP and SNR difference. (b) PSP and the optimal MCS.

frequent rate selection correspondingly. Fig. 10(b) shows the relation between PSP and the optimal MCS selection, which assumes that the AP knows the PDR of each MCS at any time and selects the MCS with the maximum throughput. For rotate, *there is a positive relation between PSP and the optimal MCS selection*: when PSP is at a high level, the optimal MCS selection stays at 23; when PSP drops a lot, it leads to a lower MCS selection.

Based on these two observations, we use PSP to refine CSI feedback when the STA is detected as rotate or static. If the PSP change between two adjacent packets is larger than the threshold Thr_P, the STA sends CSI feedback to the AP. PSP is used only if the STA is detected as rotate or static, and it does not work for mobile. Different from rotate that keeps the STA in the main beam, mobile changes either the distance or AoD from the AP to STA. For mobile, there are many variations for SNR difference and the optimal MCS selection even when PSP remains stable, as shown in Fig. 10. PSP is not the major factor influencing SNR difference or SNR variation for mobile. Therefore, both PDP similarity and PSP are needed so that CSI feedback is sent only when it is needed.

D. Overhead Analysis

Next, we present overhead analysis of the RoFi design to explore potential performance improvements on throughput and energy consumption. Normalized overhead is defined as

$$\tau = t_c/(t_c + t_d) \tag{9}$$

where t_c is the transmission time for control packets and t_d for data packets. The AP selects the MCS index m, each with a theoretical data rate $\text{rate}(m)$, for each data packet. So $t_d = \sum_{i=1}^{N}(\text{size}(p_i)/\text{rate}(m_i))$, where N is the number of data packets and $\text{size}(p_i)$ is the size of data packet p_i. t_c is calculated as

$$t_c = \sum\nolimits_{i=1}^{N}\left(\frac{\text{size}(\text{ctr}_i) + \text{size}(\text{csi}_i)}{\text{rate}(0)} + n * \text{SIFS}\right) + \sum_{j=1}^{M}\left(\frac{\text{size}(\text{pro}_j)}{\text{rate}(m_j)} + \text{SIFS}\right)$$

where $\text{size}(\text{ctr}_i)$ is the size of control packets, $\text{size}(\text{csi}_i)$ is the size of CSI, n is the number of SIFSs for data packet p_i, $\text{size}(\text{pro}_j)$ is the size of the jth probing packet, and M is the number of probing packets. CSI and control packets, including null data packet announcement, null data packet, poll, and ACK, are always transmitted using the lowest data rate, i.e., $\text{rate}(m_i)|_{m_i=0} = \text{rate}(0)$. The size of CSI is $\text{size}(\text{csi}_i) = N_t * N_r * N_s * \text{bits}(\text{csi}) + \text{size}(hdr)$, where bits(csi) is the number of bits used for each CSI entry and $\text{size}(hdr)$ is the size of packet header. The normalized overhead is significantly reduced when using long feedback intervals, as shown in Fig. 2(c). RoFi eliminates unnecessary CSI feedback, so the number of CSI packets is much smaller and the normalized overhead is significantly reduced.

The STA spends much less time for CSI and control packets by using RoFi, so it has more time for transmitting data packets to achieve higher throughput, which is calculated by

$$tpt = \sum_{i=1}^{N'} \text{size}(p_i)/(t_c + t_d) \tag{10}$$

where N' is the number of received packets. Using long feedback intervals introduces only small SNR decrease if the STA is rotating, as shown in Fig. 5 in Section II. The number of received packets for RoFi is not significantly influenced. RoFi has much smaller t_c, so it provides higher throughput.

RoFi also improves energy efficiency for the STA by sending less CSI packets. Energy efficiency of the STA is evaluated by energy consumption per data bit

$$eb = \frac{\sum_{i=1}^{N}(er(0) * \text{size}(\text{ctr}_i) + et(0) * \text{size}(\text{csi}_i))}{\sum_{i=1}^{N'} \text{size}(p_i)} + \frac{\sum_{i=1}^{N} er(m_i) * \text{size}(p_i) + \sum_{j=1}^{M} er(m_j) * \text{size}(\text{pro}_j)}{\sum_{i=1}^{N'} \text{size}(p_i)}. \tag{11}$$

where $et(m)$ and $er(m)$ stand for energy consumption per bit for transmitting and receiving, respectively, as using MCS index m [33], [34]. For the Intel 5300 WiFi chipset with $et(0) = 90$ nJ/bit and $er(23) = 11$ nJ/bit [33], $\text{size}(p_i) = 1500$ bytes, and $\text{size}(\text{csi}_i) = 1872$ bytes, the percentage of energy consumption of CSI feedback is about

$$e_{\text{csi}} = 90 * 1872 * 8/(90 * 1872 * 8 + 11 * 1500 * 8) = 91\%. \tag{12}$$

RoFi reduces the number of CSI packets $\sum_{i=1}^{N} \text{size}(\text{csi}_i)$ to increase the transmission time for data packets. Besides, $et(m_i)|_{m_i=0}$ for CSI packets is much larger than $er(m_i)$ for data packets [33], [34], so RoFi remarkably improves the energy efficiency of the STA.

IV. Evaluation

This section shows evaluation results of overhead, throughput, and energy consumption of RoFi compared with state-of-the-art feedback compression and rate selection algorithms.

A. Evaluation Methodology

The performance of RoFi is evaluated using CSI measurement traces as illustrated in Section II. Three performance metrics, including overhead, throughput, and energy consumption (9)–(11), are quantified in different mobility scenarios. Energy consumption parameters, $et(m)$ and $er(m)$ [used in (11)], for the Intel 5300 WiFi chipset are from [33]. The channel width is 20 MHz, and the MCS index m can be selected from 0 to 23 with the data rate ranging from 6.5 to 195 Mb/s [37]. The size of data packets is 1500 bytes. The AP uses ZFBF [14], [29] for transmit beamforming and the STA uses the MMSE receiver [3], [4], [29]. The transmitting power is fixed at 17 dBm. We compare RoFi with state-of-the-art methods, as shown in Table II, by CSI traces in four mobility scenarios: 1) mobile; 2) static; 3) rotate; and 4) gaming. The gaming scenario contains the mobility traces

TABLE II
EXISTING METHODS TO COMPARE WITH

Feedback Compression (Fig. 11-13)	**Rate Selection** (Fig. 14)
CSI Similarity, CoNEXT'14 [10]	SNR-based, SIGCOMM'06 [36]
Compression Noise, MobiCom'13 [14]	PDR-based, Linux Minstrel [37]
Full Feedback, 802.11ac [2]	eSNR-based, SIGCOMM'10 [3]

of four games [6]–[9] shown in Fig. 3. For the gaming scenario, the ratio of rotate, static, and mobile traces is about 47%, 49%, and 4%, respectively.

1) Existing Feedback Compression Methods to Compare With: We compare RoFi with three feedback compression methods: 1) CSI similarity [4], [10], [30]; 2) compression noise [14]; and 3) full feedback [2]. ToF measured by off-the-shelf WiFi chipsets has very low accuracy and it provides much worse rotation detection results than CSI similarity and compression noise, so we omit the evaluation of ToF due to space constraints.

CSI similarity, which is calculated by (4), is used to detect the mobility status of the STA. The STA sends CSI feedback for each packet if it is moving; otherwise it sends CSI feedback every 100 ms. Compression noise, which is calculated by (5), is used to calculate the SNR decrease caused by feedback compression. The AP polls for CSI feedback only if the SNR decrease is large enough to reduce the current data rate. Note that compression noise is defined in three domains: 1) time; 2) frequency; and 3) quantization, in [14]. We only use compression noise in the time domain since the 802.11n CSI tool [27] provides noncompressed CSI neither in frequency nor quantization domain. The number of subcarriers is $N_s = 30$ and the number of bits for each CSI entry is $\text{bits}(\text{csi}) = 16$. There is also a full feedback scheme that requires the STA to send CSI feedback for each data packet.

2) Existing Rate Selection Methods to Compare With: We compare RoFi with rate selection algorithms based on PDR [36], SNR [35], and eSNR [3], [4]. These rate selection algorithms select the MCS by solving the same problem in (8), but measure or predict $\text{pdr}_i(m)$ differently. The PDR-based algorithm measures $\text{pdr}_i(m)$ by probing packets. For probing packets using MCS index m, PDR is calculated by $\text{pdr}_i(m) = \alpha * \text{pdr}_{t-1}(m) + (1 - \alpha) * \text{pdr}_t(m)$, where $\text{pdr}_t(m)$ is the PDR measured during the most recent time window and $\text{pdr}_{t-1}(m)$ for the previous time window, and α is the averaging weight. It is the default rate selection algorithm for Linux WiFi driver, wherein the time window length is 50 ms and the averaging weight α is 0.125 [36], [38].

The SNR-based algorithm predicts $\text{pdr}_i(m)$ based on the SNR threshold $\text{Thr}_{\text{snr}}(m)$ for each MCS index m, i.e., $\text{pdr}_i(m) > \text{Thr}_{\text{pdr}}(m)$ if $\text{snr}_i(m) > \text{Thr}_{\text{snr}}(m)$ [35]. The eSNR-based algorithm uses eSNR to predict $\text{pdr}_i(m)$, which is the same as RoFi, for each packet p_i [3], [4]. Unlike RoFi, the eSNR-based algorithm requires CSI feedback before transmitting each data packet p_i. To avoid unnecessary CSI feedback, the eSNR-based rate selection uses CSI similarity to detect the mobility status of the STA. If the CSI similarity is greater than 0.9, the STA sends CSI feedback for each packet; otherwise it sends CSI feedback every 100 ms. Both PDR- and eSNR-based rate selections require sending probing packets. There is also an optimal rate selection algorithm. It assumes that the AP knows CSI and PDR for each MCS index at any time and selects the MCS with the highest throughput. Results of the PDR-based algorithm are from real-world measurements, and other rate selection algorithms are calculated from CSI traces.

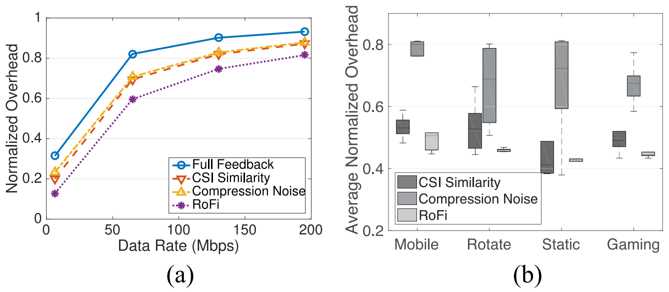


Fig. 11. Normalized overhead. (a) Using fixed data rates for rotate. (b) Statistical results for data rate of 65 Mb/s in different mobility scenarios. The average normalized overhead for full feedback is fixed at 0.82 for 65 Mb/s.

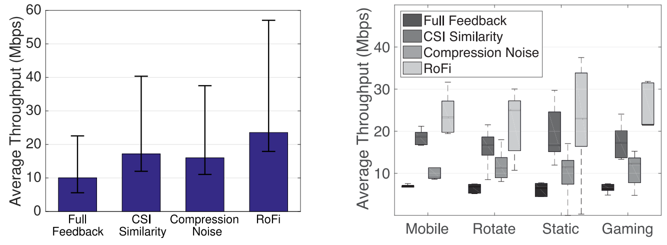
B. Performance Results of Feedback Compression

We first compare RoFi with existing feedback compression schemes. Results show that RoFi has lower overhead and energy consumption and higher throughput in different mobility scenarios.

1) Overhead: Fig. 11(a) shows the normalized overhead, as defined in (9), using fixed data rates. It is evaluated from the RotateX trace measured at P6 [shown in Fig. 4(a)]. Both CSI similarity and compression noise have much higher overhead than RoFi. At data rate of 6.5 Mb/s, the normalized overhead of RoFi is 0.12, which is only 60% of that of CSI similarity and compression noise. At higher data rates, the normalized overhead of RoFi is 75% of that of CSI similarity and compression noise. In other words, RoFi reduces the transmission time CSI packets by 25%–40%. At the same time, there is no obvious SNR difference between RoFi, CSI similarity, compression noise, and full feedback. The maximum SNR decrease of RoFi is lower than 1 dB.

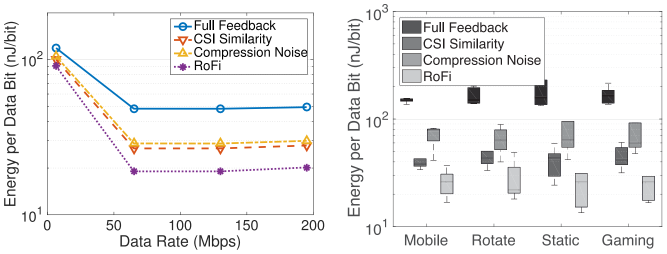
Statistical results of the average normalized overhead for each mobility scenario are shown in Fig. 11(b). For rotate, the normalized overhead of RoFi is 89% and 63% of that of CSI similarity and compression noise, respectively. RoFi also reduces overhead when the STA is not rotating. The normalized overhead of RoFi is 63% and 60% of that of compression noise for mobile and static, respectively. For gaming traces, the normalized overhead of RoFi is 93% and 62% of that of CSI similarity and compression noise. RoFi and CSI similarity have comparable overhead for mobile, static, and gaming scenarios. The average normalized overhead of full feedback is 0.82 for data rate 65 Mb/s for all mobility scenarios.

2) Throughput: Fig. 12(a) shows throughput, as defined in (10), for the RotateX trace using fixed data rates. RoFi eliminates unnecessary CSI feedback with negligible SNR decrease, so it provides higher throughput. Full feedback has



(a) (b)

Fig. 12. Average throughput. (a) Using fixed data rates for rotate. (b) Statistical results for data rate of 65 Mb/s in different mobility scenarios.



(a) (b)

Fig. 13. Energy consumption. (a) Using fixed data rates for rotate. (b) Statistical results for data rate of 65 Mb/s in different mobility scenarios.

the lowest throughput because of sending CSI feedback for each data packet. The throughput of CSI similarity, compression noise, and RoFi is 70%, 60%, and 140%, respectively, higher than that of full feedback. Fig. 12(b) shows statistical throughput results for all traces. For rotate, the throughput of RoFi is 1.52× and 2.16× of that of CSI similarity and compression noise. RoFi has 21%, 43%, and 35% higher throughput than CSI similarity for mobile, static, and gaming, respectively. RoFi introduces smaller SNR decrease as CSI similarity, so it still provides higher throughput, even though RoFi has higher normalized overhead for static traces as shown in Fig. 11(b).

3) Energy Consumption: Fig. 13(a) shows energy consumption, as defined in (11), for the RotateX trace with fixed data rates. At data rate of 6.5 Mb/s, energy consumption is almost the same for all feedback compression methods. For data rates of greater than 50 Mb/s, energy consumption is about 20 nJ/bit for RoFi, 30 nJ/bit for CSI similarity and compression noise, and 49 nJ/bit for full feedback. Fig. 13(b) shows statistical results of energy consumption for different mobility scenarios. For rotate, the energy consumption of RoFi is 48% and 66% lower than that of CSI similarity and compression noise. RoFi consumes less energy by sending less CSI packets for the STA. For mobile, energy consumption of RoFi is 24 nJ/bit, which is 45% and 53% lower than that of CSI similarity and compression noise, respectively. For static, RoFi consumes 29% and 69% less energy than CSI similarity and compression noise, respectively. The energy consumption results of gaming are similar to that of static.

C. Performance Results of Rate Selection

Next, we show performance results of RoFi and existing rate selection algorithms based on SNR, PDR, and eSNR. Results show that RoFi has higher throughput and lower energy consumption in rotate and static scenarios.

1) Throughput: Fig. 14(a) shows statistical results of throughput for different mobility scenarios. The throughput of eSNR-based rate selection is the lowest in all mobility scenarios, since it needs extensive CSI measurements and feedback. For mobile and static, RoFi has lower throughput than the SNR-based algorithm, since RoFi has much higher normalized overhead as shown in Fig. 14(b). For rotate and gaming, RoFi has 8% and 22% higher throughput than PDR- and SNR-based algorithms, respectively. The reason is that RoFi is able to select much higher data rates with high PDR to send more data packets during the same transmission time. For static, the average throughput of RoFi is slightly lower than SNR- and PDR-based algorithms. For gaming traces, RoFi has slightly higher throughput than SNR- and PDR-based rate selections.

2) Overhead: The results of normalized overhead are shown in Fig. 14(b). SNR-based rate selection has the lowest normalized overhead in all mobility scenarios, since it does not need CSI feedback or probing packets. The PDR-based algorithm has higher overhead than SNR-based rate selection due to probing packets. The eSNR-based algorithm has the highest overhead since it requires extensive CSI measurements and feedback. The normalized overhead of RoFi is greater than that of SNR-based rate selection, but it is much lower than that of eSNR-based rate selection, in all mobility scenarios. The normalized overhead of PDR- and eSNR-based rate selections is stable across different mobility traces.

3) Energy Consumption: Fig. 14(c) shows the results of energy consumption in different mobility scenarios. For mobile and static, the energy consumption of eSNR-based rate selection is similar to that of RoFi. For static, the energy consumption of RoFi is 25% and 37% lower than that of SNR- and PDR-based algorithms, respectively. For rotate, the energy consumption of RoFi is 47%, 31%, and 15% lower than that of SNR-, PDR-, and eSNR-based algorithms, respectively. For gaming traces, RoFi consumes 43%, 25%, and 17% less energy than SNR-, PDR-, and eSNR-based algorithms, respectively.

D. Energy Impact of PDP Similarity Calculation

RoFi needs to calculate PDP similarity which may introduce computation overhead for MIMO receivers. In this section, we investigate the energy impact of PDR similarity calculation. We run different CSI feedback schemes, including full feedback, CSI similarity, compression noise, and RoFi, using CSI traces collected in different scenarios. At the same time, we measure the Energy Impact of the simulation process by the Linux command `top`. Energy Impact measures per-process power consumption by CPU usage and wakeup frequency, and it has no physical unit [39]. Fig. 15 shows energy impact of four CSI feedback schemes in running time. RoFi has slightly higher energy impact than full feedback, which does not need calculations to determine when to send CSI feedback. The average energy impact as running all CSI traces is summarized in

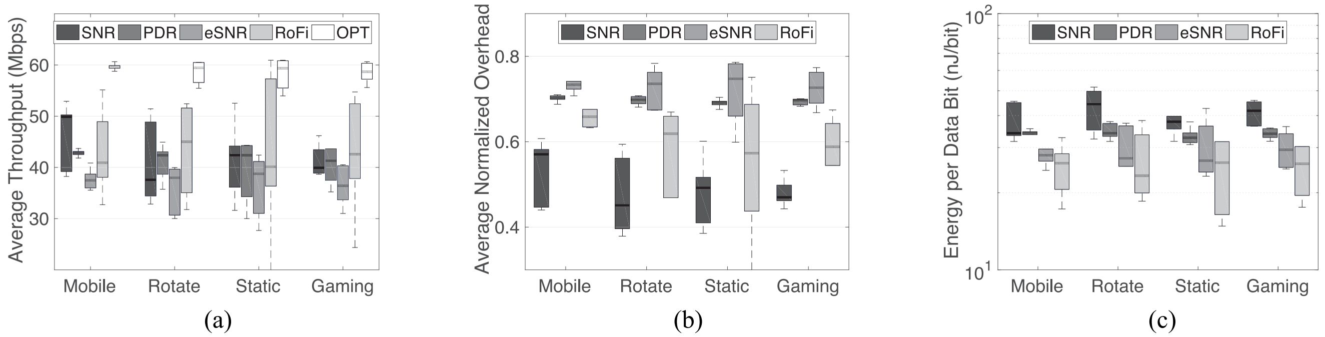


Fig. 14. Performance results of different rate selection algorithms in different mobility scenarios. Average (a) throughput, (b) normalized overhead, and (c) energy consumption.

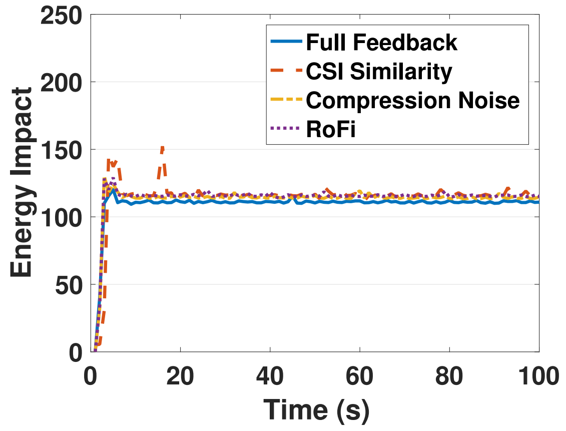


Fig. 15. Energy impact of different CSI feedback schemes.

TABLE III
AVERAGE ENERGY IMPACT

	Energy Impact
Full Feedback	106.45 (100.00%)
CSI Similarity	114.29 (107.37%)
Compression Noise	113.79 (106.90%)
RoFi	112.69 (105.86%)

Table III. Compared with full feedback, RoFi only introduces 5.86% extra energy impact. Besides, RoFi has slightly less energy impact than CSI similarity and compression noise.

V. RELATED WORK

A. CSI Feedback Compression

The 802.11 protocol allows feedback compression by sharing the same CSI for multiple packets or subcarriers, or representing each CSI value with less bits of data [1], [2], [14]. For example, Intel 5300 only reports CSI for 30 subcarriers with each entry represented by 16 bits [27], while the default CSI requires 32 bits each for 52 subcarriers for a 20-MHz channel. Different quantization techniques [40] can be used to reduce the size of the CSI matrix. CSI-SF [41] predicts multistream CSI values using CSI of single-stream packets to reduce CSI sampling overhead. AFC [14] adaptively selects compression levels based on the SNR decrease caused by compression noise. But it does not distinguish whether the receiver is rotating or moving and requires per-packet feedback for both cases. Thus, it fails to eliminate unnecessary CSI feedback if the STA is rotating. RoFi provides CSI feedback only when it is needed by rotation-aware channel feedback.

B. MIMO Rate Selection

There are many works on WiFi rate selection, where the data rate is determined by channel width, antenna selection, code rate, and modulation scheme. Each data rate selection has the maximum rate and the corresponding PDR it can delivery. The problem is how to select the rate index satisfying certain requirements, such as high throughput, low delay, low energy consumption, etc. A simple yet effective solution is to predict the PDR based on per-packet SNR and the PDR-SNR curve [35]. For MIMO, the SNR-based algorithm performs poorly since the PDR-SNR model is not accurate due to frequency-selective fading effects. eSNR [3], [4] accurately predicts PDR using CSI, instead of per-packet SNR, and provides high throughput for MIMO networks. But it needs to measure and exchange CSI continuously, introducing huge measurement and feedback overhead. The Linux WiFi driver uses PDR-based rate selection that measures PDR by probing packets every 50 ms [36], [38]. The PDR-based algorithm has high probing overhead. It is not suitable for mobile environments since the MIMO channel changes quickly during the 50-ms measurement period.

C. Mobility-Aware WiFi Protocols

Sensors are used to enhance WiFi protocols by providing movement information [42], but it only provides boolean movement hints and requires modifications of WiFi frame formats and protocols. CSI similarity is used to enable mobility-aware rate selection in [10]. The aforementioned mobility-aware methods are not able to distinguish whether the STA is in the status of rotation or mobile. CSI provides detailed information of attenuation and phase shifts [43], [44] to calculate angle of arrival (AoA) and ToF in decimeter-level accuracy [45], [46]. AoA and ToF can be used to detect rotation, but it requires extensive CSI measurements from multiple packets and APs [45] or scanning of all available frequency bands [31], [46]. ToF can also be measured by the time interval between data and ACK packets using off-the-shelf WiFi chipsets [10]–[13], but the accuracy cannot be guaranteed at nanosecond level, which makes it hard to distinguish whether the STA is rotating. BeamAdapt [47] brings beamforming to mobile devices, and performance considering device rotation is studied. Unlike RoFi considering the STA as the receiver, BeamAdapt uses the STA as the transmitter, and it does not consider the accuracy and overhead of CSI feedback.

VI. Conclusion

We show the failure of existing mobility-aware methods, including CSI similarity, ToF, and compression noise, in distinguishing rotation from other mobility scenarios. We propose RoFi channel feedback to eliminate unnecessary CSI feedback while maintaining high SNR in different mobility scenarios. RoFi uses PDP similarity to detect the mobility status of the STA by just using CSI. The STA provides CSI feedback only when it is needed based on rotation detection results. At the same time, RoFi uses the PSP, which is calculated from PDP, to refine CSI feedback when the STA is detected in the status of rotation or static. RoFi brings rotation-awareness to WiFi and helps the AP select the best data rate accurately without extensive CSI measurements and feedback. RoFi significantly improves the performance and efficiency of WiFi STAs in different mobility scenarios by reducing unnecessary CSI feedback with negligible SNR decrease.

References

[1] *Enhancements for Higher Throughput*, IEEE Standard 802.11n-2009, 2009.

[2] *Enhancements for Very High Throughput for Operation in Bands Below 6 GHz*, IEEE Standard 802.11ac-2013, 2013.

[3] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, “Predictable 802.11 packet delivery from wireless channel measurements,” in *Proc. ACM SIGCOMM*, New Delhi, India, 2010, pp. 159–170.

[4] D. Halperin, “Simplifying the configuration of 802.11 wireless networks with effective SNR,” Ph.D. dissertation, Dept. Comput. Sci. Eng., Univ. Washington, Seattle, WA, USA, 2012.

[5] M. Gast, *802.11ac: A Survival Guide*. Sebastopol, CA, USA: O’Reilly Media, 2013.

[6] *Flight Pilot Simulator 3D Free*. Accessed on Dec. 31, 2016. [Online]. Available: https://goo.gl/2PU2Jq

[7] *Traffic Rider*. Accessed on Dec. 31, 2016. [Online]. Available: https://goo.gl/uVRbke

[8] *Asphalt 8: Airborne*. Accessed on Dec. 31, 2016. [Online]. Available: https://goo.gl/Bvms7m

[9] *Bike Race Free Motorcycle Game*. Accessed on Dec. 31, 2016. [Online]. Available: https://goo.gl/2R7riz

[10] L. Sun, S. Sen, and D. Koutsonikolas, “Bringing mobility-awareness to WLANs using PHY layer information,” in *Proc. ACM CoNEXT*, Sydney, NSW, Australia, 2014, pp. 53–66.

[11] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim, “Sail: Single access point-based indoor localization,” in *Proc. ACM MobiSys*, 2014, pp. 315–328.

[12] A. Marcaletti, M. Rea, D. Giustiniano, V. Lenders, and A. Fakhreddine, “Filtering noisy 802.11 time-of-flight ranging measurements,” in *Proc. ACM CoNEXT*, Sydney, NSW, Australia, 2014, pp. 13–20.

[13] D. Giustiniano, T. Bourchas, M. Bednarek, and V. Lenders, “Deep inspection of the noise in WiFi time-of-flight echo techniques,” in *Proc. ACM MSWiM*, Cancún, Mexico, 2015, pp. 5–12.

[14] X. Xie, X. Zhang, and K. Sundaresan, “Adaptive feedback compression for MIMO networks,” in *Proc. ACM MobiCom*, 2013, pp. 477–488.

[15] *Netgear Arlo*. Accessed on Jun. 1, 2017. [Online]. Available: https://www.arlo.com

[16] *Homeboy*. Accessed on Jun. 1, 2017. [Online]. Available: https://www.homeboy.com

[17] *Logitech Circle*. Accessed on Jun. 1, 2017. [Online]. Available: https://www.logitech.com/en-us/product/circle

[18] *iRobot*. Accessed on Jun. 1, 2017. [Online]. Available: http://www.irobot.com

[19] *Double Robotics*. Accessed on Jun. 1, 2017. [Online]. Available: http://www.doublerobotics.com

[20] *Dyson 360 Eye*. Accessed on Jun. 1, 2017. [Online]. Available: http://www.dyson.com/vacuum-cleaners/robot/dyson-360-eye.aspx

[21] *Rivvr Wireless VR*. Accessed on Jun. 1, 2017. [Online]. Available: https://www.rivvr.com

[22] *DisplayLink Wireless VR*. Accessed on Jun. 1, 2017. [Online]. Available: http://www.displaylink.com/vr

[23] *HTC VIVE Wireless VR*. Accessed on Jun. 1, 2017. [Online]. Available: https://www.vive.com

[24] *DJI*. Accessed on Jun. 1, 2017. [Online]. Available: http://www.dji.com

[25] *Parrot*. Accessed on Jun. 1, 2017. [Online]. Available: https://www.parrot.com

[26] *Yuneec*. Accessed on Jun. 1, 2017. [Online]. Available: http://us.yuneec.com

[27] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, “Tool release: Gathering 802.11n traces with channel state information,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, pp. 53–53, 2011.

[28] E. Perahia and R. Stacey, *Next Generation Wireless LANs: 802.11n and 802.11ac*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[29] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, “802.11 with multiple antennas for dummies,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 1, pp. 19–25, 2010.

[30] S. Byeon *et al.*, “MoFA: Mobility-aware frame aggregation in Wi-Fi,” in *Proc. ACM CoNEXT*, Sydney, NSW, Australia, 2014, pp. 41–52.

[31] Y. Xie, Z. Li, and M. Li, “Precise power delay profiling with commodity WiFi,” in *Proc. ACM MobiCom*, Paris, France, 2015, pp. 53–64.

[32] S. Sen, J. Lee, K.-H. Kim, and P. Congdon, “Avoiding multipath to revive inbuilding WiFi localization,” in *Proc. ACM MobiSys*, Taipei, Taiwan, 2013, pp. 249–262.

[33] D. Halperin, B. Greenstein, A. Sheth, and D. Wetherall, “Demystifying 802.11n power consumption,” in *Proc. USENIX HotPower*, Vancouver, BC, Canada, 2010, Art. no. 1.

[34] S. K. Saha, P. Deshpande, P. P. Inamdar, R. K. Sheshadri, and D. Koutsonikolas, “Power-throughput tradeoffs of 802.11n/ac in smartphones,” in *Proc. IEEE INFOCOM*, Hong Kong, 2015, pp. 100–108.

[35] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, “Measurement-based models of delivery and interference in static wireless networks,” in *Proc. ACM SIGCOMM*, Pisa, Italy, 2006, pp. 51–62.

[36] *Minstrel Rate Control Algorithm*. Accessed on Dec. 31, 2016. [Online]. Available: https://goo.gl/T3AFV2

[37] *IEEE 802.11n-2009 Data Rates*. Accessed on Dec. 31, 2016. [Online]. Available: https://en.wikipedia.org/wiki/IEEE_802.11n-2009#Data_rates

[38] *The 802.11 Subsystems for Kernel Developers*. Accessed on Dec. 31, 2016. [Online]. Available: https://www.kernel.org/doc/htmldocs/80211/index.html

[39] *What Does the OS X Activity Monitor’s ‘Energy Impact’ Actually Measure?* Accessed on Dec. 31, 2016. [Online]. Available: https://goo.gl/H2yHmz

[40] Q. Wang *et al.*, “Comparison of quantization techniques for downlink multi-user MIMO channels with limited feedback,” *IEEE Wireless Commun. Lett.*, vol. 3, no. 2, pp. 165–168, Apr. 2014.

[41] R. Crepaldi, J. Lee, R. Etkin, S.-J. Lee, and R. Kravets, “CSI-SF: Estimating wireless channel state using CSI sampling & fusion,” in *Proc. IEEE INFOCOM*, Orlando, FL, USA, 2012, pp. 154–162.

[42] L. Ravindranath, C. Newport, H. Balakrishnan, and S. Madden, “Improving wireless network performance using sensor hints,” in *Proc. USENIX NSDI*, Boston, MA, USA, 2011, pp. 281–294.

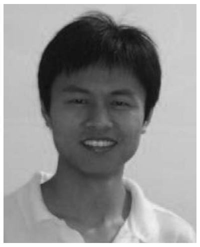
[43] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[44] J. Gjengset, J. Xiong, G. McPhillips, and K. Jamieson, “Phaser: Enabling phased array signal processing on commodity WiFi access points,” in *Proc. ACM MobiCom*, 2014, pp. 153–164.

[45] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, “SpotFi: Decimeter level localization using WiFi,” in *Proc. ACM SIGCOMM*, London, U.K., 2015, pp. 269–282.

[46] D. Vasisht, S. Kumar, and D. Katabi, “Decimeter-level localization with a single WiFi access point,” in *Proc. USENIX NSDI*, 2016, pp. 165–178.

[47] H. Yu, L. Zhong, A. Sabharwal, and D. Kao, “Beamforming on mobile devices: A first study,” in *Proc. ACM MobiCom*, Las Vegas, NV, USA, 2011, pp. 265–276.



Yongsen Ma received the B.S. degree in control science and engineering from Shandong University, Jinan, China, and the M.S. degree in control science and engineering from Shanghai Jiao Tong University, Shanghai, China and is currently pursuing the Ph.D. degree at the Department of Computer Science, College of William and Mary, Williamsburg, VA, USA.

He is currently a member of the LENS Research Group, where he is advised by Dr. G. Zhou. He was a Research Assistant with Intel, Shanghai, China. His current research interests include wireless networking, ubiquitous sensing, and mobile systems.

Gang Zhou (GS'06–M'07–SM'13) received the Ph.D. degree from the University of Virginia, Charlottesville, VA, USA, in 2007, under the supervision of Prof. J. A. Stankovic.

He is an Associate Professor with the Computer Science Department, College of William and Mary, Williamsburg, VA, USA, where he served as the Graduate Program Director from 2015 to 2017. He has authored or co-authored over 80 papers in the areas of ubiquitous and mobile computing, body sensor networks, smart healthcare, Internet of Things, and wireless communication and networking with over 6000 Google Scholar citations. He also has 15 papers, each of which, has been cited over 100 times since 2004.

Dr. Zhou was a recipient of an award for his outstanding service to the IEEE Instrumentation and Measurement Society in 2008, the Best Paper Award of IEEE ICNP 2010, the NSF CAREER Award in 2013, and the 2015 Plumeri Award for Faculty Excellence. He serves on the Journal Editorial Board of the IEEE INTERNET OF THINGS JOURNAL, *Elsevier Computer Networks*, and *Elsevier Smart Health*. He served as an NSF, NIH, and GENI Proposal Review Panelists multiple times. He is a Senior Member of the ACM.

Shan Lin received the Ph.D. degree in computer science from the University of Virginia, Charlottesville, VA, USA.

He is an Assistant Professor with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY, USA. His current research interests include networked systems, cyber-physical systems, Internet of Things, wireless network protocols, medical systems and devices, smart buildings, and smart transportation systems.

Dr. Lin was a recipient of the NSF CAREER Award in 2016.

Haiming Chen (M'15) received the B.Eng. and M.Eng. degrees in computer engineering from Tianjin University, Tianjin, China, in 2003 and 2006, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2010.

He is currently an Associate Professor with the Department of Computer Science, Ningbo University, Ningbo, China. From 2010 to 2017, he was a Research Assistant Professor with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing. He has authored or co-authored over 40 papers in the areas of wireless, *ad hoc*, sensor networks, and networked embedded computing systems.

Dr. Chen was a recipient of the Outstanding Faculty Award from the Chinese Academy of Sciences in 2011 and 2016, the Third Prize of the Beijing Municipal Science and Technology Award in 2013, and the Best Paper of CCF CWSN 2015. He has served as a TPC member for several international conferences such as IEEE ICPADS 2014–2016 and EAI COLLABORATECOM 2015–2016 and a Reviewer for several international journals such as *ACM Transactions on Sensor Networks*, the IEEE INTERNET OF THINGS JOURNAL, *Elsevier Computer Networks*, and *Ad Hoc Networks*. He is a member of the ACM and China Computer Federation.