

RESEARCH

Open Access

Accelerometer-assisted 802.11 rate adaptation on mobile WiFi access

Yu-Jen Lai¹, Wei-Hao Kuo¹, Wan-Ting Chiu¹ and Hung-Yu Wei^{2*}

Abstract

The expansion of 802.11 AP deployment provides opportunistic Wi-Fi access in underground mass rapid transit (MRT) system. However, such vehicular network faces the challenge of limited time for the MS on the train to connect the base station at the station. Therefore, to maximize the throughput within these several tens of seconds becomes crucial to the network. To achieve this goal, we propose accelerometer-assisted rate adaptation (AARA), a mechanism utilizes the out-of-band information of a train's acceleration to improve the conventional rate adaptation scheme. AARA consists of two parts: First, AARA divides a train's movement into four phases and performs real-time estimation on the train's current movement phase. Second, AARA employs the estimation results to enhance the bit-rate selection during each phase. We conduct experiments on two different Taipei MRT systems: high-capacity MRT and medium-capacity MRT. The experimental results show that the average throughput of AARA outperforms that of the conventional scheme in different scenarios. In addition, we also design a mechanism of power saving with the aid of the movement estimation.

Keywords: Rate adaptation, IEEE 802.11, Sensor-assisted communications, Vehicular network, Network experiment, Public transportation system

Introduction

As the public transportation system rapidly develops, demands to access the Internet in transportation systems are increasing over time. Immediate information access to the Internet is especially important and valuable for transportation system commuters in metropolises. Due to the development in wireless technology and the expansion of WLAN deployment, this type of vehicular network becomes attainable nowadays. Consider the scenario of Figure 1, base station (BS) is located at a roadside spot like a train station, a bus stop, or an intersection to provide Wi-Fi service to mobile station (MS) on vehicles. However, due to limited transmission coverage, wireless signal disappears after the vehicle departs from the station. Therefore, selecting appropriate transmission rates to employ is vital to maximize the transmission throughput when the train stops at the station.

In the mass rapid transit (MRT) system, the channel quality varies greatly when the train arrives at and departs from the station, but is relatively stable when the train stops at the station. Most of the conventional rate adaptation schemes suffer from the degradation of transmission performance since the signal environment changes radically when trains passing by the station. To solve this problem, it is conceivable to employ different rate adaptation schemes according to the phases of the train's movement.

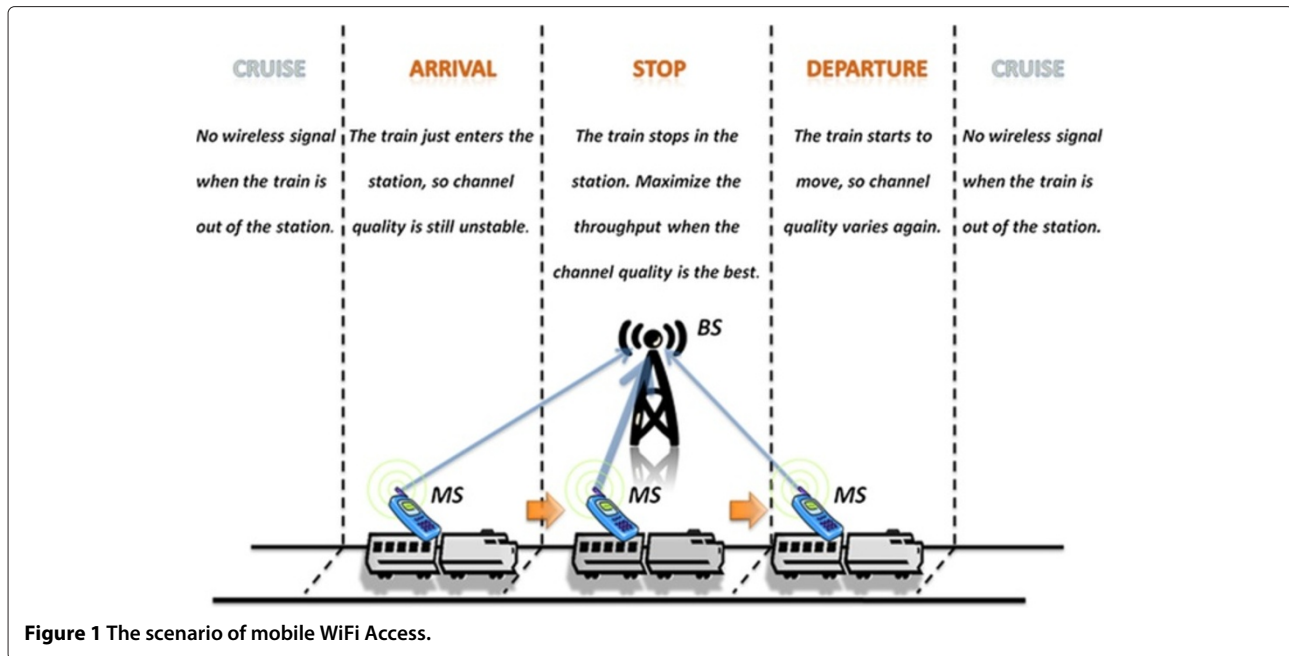
In addition, since Wi-Fi has fairly high power consumption when the NIC is activated, we should take energy consumption in mobile devices into consideration. As the train leaves the station, the MS on the train is outside of the Wi-Fi coverage. If we can detect the train's departure, we will be able to switch the wireless device to sleep mode in order to conserve energy. Moreover, we should also recognize when the train arrives at the next station and reactivate the wireless device to relish Internet service again.

Therefore, the first challenge is to accurately estimate the movement of the train. Intuitively, we would attempt to employ the metric of signal strength to estimate the train's distance from the station. Nevertheless,

*Correspondence: hywei@cc.ee.ntu.edu.tw

²Department of Electrical Engineering, Graduate Institute of Communication Engineering, Intel-NTU Connected Context Computing Center, National Taiwan University, Taipei 106, Taiwan

Full list of author information is available at the end of the article



this method requires extra energy to monitor the wireless channel along the whole path, even when it is out of signal coverage, i.e., the train is cruising between stations. Thus, in this article, we have designed a context-aware mechanism assisted by out-of-band acceleration information to estimate the movement of the train. We also propose a rate adaptation scheme assisted by the estimation result of the train's movement in mobile WiFi devices. The experimental results demonstrated the feasibility of the proposed scheme. In our current prototyping, the accelerometer is attached to the mobile device. As acceleration information has been shown useful to assist wireless communications, we could seek to obtain acceleration information with various methods. For example, for radio system built-in for train vehicles, the acceleration information might possibly be obtained from train control system. For radio system on motorcycles, the acceleration information might be obtained from its control panel. Accelerometer could be built-in internally (e.g. smartphones have built-in accelerometer) or connected externally (e.g. connected via USB port). We foresee this acceleration-assisted wireless communication technology to be applied in different forms.

The rest of this article is organized as follows. Section "Related work" discusses the related work. Section "Design overview" gives an overview of the system design. Section "Preliminary experiments" presents the preliminary data collection and observation from the preliminary experiments. Section "System design" introduces the detailed design of accelerometer-assisted rate adaptation (AARA). Section "Experimental evaluation" presents the comparison of the evaluation results of AARA and

SampleRate on Taipei MRT. Section "Power saving" compares the energy consumptions of AARA and SampleRate. Section "Discussion" discusses the applicability of the proposed scheme. Section "Conclusion" concludes the article.

Related work

Movement estimation

GPS has been utilized to estimate the moving speed or people's behavior. Zheng et al. [1] analyze GPS logs with graph-based post-processing algorithm to identify transportation modes. Patterson et al. [2] combine GPS with a learned Bayesian model to infer the transportation modes. Reddy et al. [3] solve the same issue by dealing with GPS and acceleration information based on a discrete Hidden Markov Model. However, instead of estimating the vehicle's departure and arrival, these articles tend to focus on recognizing human behaviors. More than that, using the information of GPS location requires the user to keep the records of the BS's GPS location in advance, and the GPS signal cannot be received in underground scenarios.

Rate adaptation

Typically, packet transmission's statistics of success and failure are the indicators of channel conditions, which are used by wireless LAN devices to adjust the bit-rate. Auto rate fallback (ARF) is the first rate adaptation scheme designed for WaveLan II devices. Lacage et al. design adaptive auto rate fallback (AARF) [4] to further improve ARF. After a fixed number of consecutive successful transmissions, both ARF and AARF increase their bit-rate and then decrease it after some transmission failures

occur. The main difference between them is that AARF doubles the number of consecutive successful transmissions needed to switch to a higher bit-rate when transmission failure occurs. Onoe [5] is a credit-based algorithm and has been implemented in Linux MadWifi driver. It computes the credit of each bit-rate per second to decide whether to increase, decrease, or remain the bit-rate. When the retransmission times are higher than a certain level, it will then decrease the credit of the given bit-rate. RRAA [6] monitors the short term loss ratio (during a window) in a given bit-rate and compares the loss ratio with two pre-defined thresholds called P_{ORI} and P_{MTL} . When the loss ratio exceeds P_{MTL} , the bit-rate decreases; while the loss ratio is below P_{ORI} , the bit-rate increases. The idea of RRAA is to switch to a bit-rate only when it can supply a higher throughput than the original bit-rate. SampleRate is the default rate adaptation algorithm in MadWifi driver [7]. The idea of SampleRate is to choose the bit-rate scheme with the shortest expected transmission time to transmit. To achieve this, it maintains a table of the average successful transmission time of each bit-rate, and sends probing packets periodically to update the table. For probing efficiency, SampleRate blocks (not to probe) the bit-rates which have consecutive transmission failures.

Context-aware wireless transmission

Most of the current rate adaptation mechanisms are designed for wireless networks in general scenarios. Different from the previous rate adaptation schemes, the proposed rate adaptation mechanism is aware of the context assisted by out-of-band acceleration information designed for wireless access on mobile environment. CARS [8] is focused on rate adaptation for vehicular networks. With GPS devices, CARS obtains context information like vehicle speed and distance to the receiver. Rahmati and Zhong [9] also consider employing context information to optimize the energy by switching wireless interface between Wi-Fi and GSM. Joshi and Kulkarni's [10] work compares the rate adaptation schemes implemented in MadWifi, and address a rate adaptation scheme based on history record of the relation between SNR and throughput. Santhapuri et al. [11] discuss the application of out-of-band context information from sensors to assist wireless communication design. Note that we choose context information from accelerometers rather than GPS locations, as it incurs several problems while employing information of GPS locations. (1) Underground metro system fails to access GPS signal. (2) Problems like weak signal and long GPS acquisition/re-acquisition time occur in urban settings with tall building. (3) Prior knowledge of the access point locations is needed. (4) GPS consumes more than ten times of power than that of accelerometer.

Design overview

This section describes our system's design overview. The main contribution we are making is to boost the throughput with the aid of a more suitable rate adaptation strategy for the scenario of "drive-thru with a stop", as shown in Figure 1.

The drive-thru scenario, brought up by Ott and Kutscher [12], refers to a scenario in which the vehicle drives through a roadside unit and access it in a very short time. Many related works adopt different methods to increase the throughput in such scenario [13-17]. On the basis of this concept, we observe that the scenario of drive-thru with a stop should be even more critical as it supplies a higher connection opportunity when the vehicle stops near a roadside unit. For example, if there is a municipal plan to deploy Wi-Fi AP on a specific road for building a vehicular network, it would be more reasonable to deploy APs at an intersection, where most vehicles might stop for a while. Similarly, the most sensible place to deploy APs in the public transportation system is in the station, where the train also stops for a short period of time.

Even so, the connection time is still limited. In our testbed, the Taipei MRT system, it takes about 40-50 s for a train to approach to, stop at, and depart from a station. Due to the changing channel quality along the path, we need a more suitable rate adaptation scheme specifically designed for the drive-thru with a stop scenario.

In this article, we propose a novel rate adaptation scheme, AARA, assisted by accelerometer that mainly designed for transportation systems. As illustrated in Figure 2, the mechanism of AARA is composed of two parts: the estimation of movement phases and the scheme for rate adaptation.

Accelerometer-assisted rate adaptation measures the 3-axis acceleration of the MS, and transforms the data into two indicators: Moving-Acc (M-Acc) and Vibrational-Acc (V-Acc). These indicators describe the acceleration of a train's movement and vibration, respectively. The phase estimation is acquired from a simple integrated decision function. The system passes the estimation result to the MAC layer to facilitate the bit-rate selection as well as the power saving function. The proposed rate adaptation scheme, assisted by accelerometer, is an enhancement to the SampleRate algorithm, which is the default bit-rate selection mechanism in MadWifi driver. Moreover, the phase estimation is achieved in real-time and the system is implemented and tested in the real-world experiments.

Preliminary experiments

Experiment setup

Before introducing our system design in detail, we summarize the characteristics of our experimental testbed, the equipments, the platform, and the methodology in this section. The experimental results presented in this

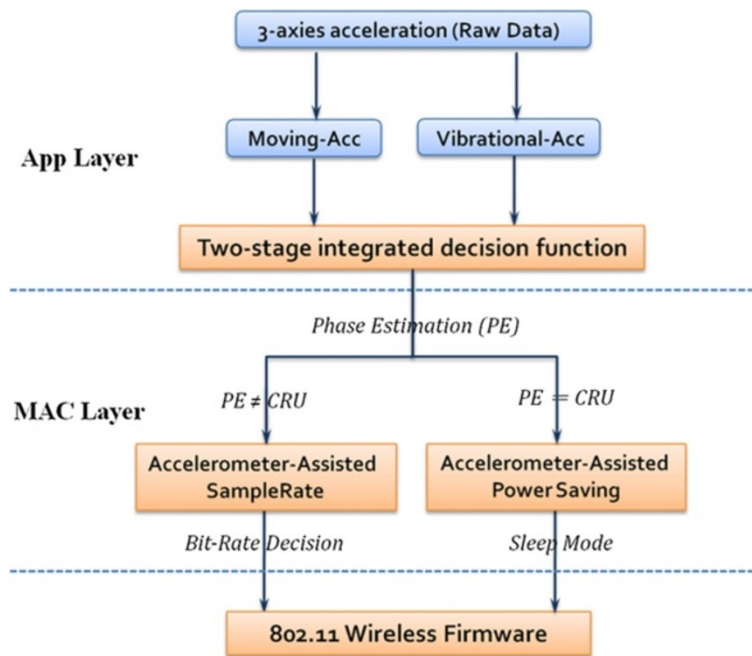


Figure 2 The system architecture of AARA.

article are obtained by conducting real-world experiments in outdoor public transportation system (Taipei MRT).

Methodology

We conduct the experiments with three laptops operating on Linux. We choose MadWifi [18] as the driver of the wireless adaptor since it is one of the most advanced open-source WLAN driver available for Linux. The detailed parameters of our experiments are presented in Table 1.

The scenario of our system is shown in Figure 1. We place two laptops on the train and one on the platform as the access point. One of the laptops on the train acted as an Iperf client [19], taking charge of continuous

transmission to the access point situated on the platform which acted as Iperf server. The other laptop on the train, on the other hand, acted as a sniffer running in the monitor mode and using Wireshark [20] to sniff and log the information of the transmission packets. The information of packets' bit rates are extracted from the Radiotap header [21]. Meanwhile, the client on the train also runs the accelerometer, attached via USB, to measure its 3-axis acceleration. The raw data of 3-axis acceleration is collected and processed into two types of information to estimate the train's movement phases. Next, we use the estimation result to improve the transmission strategy accordingly. We further describe our design in Section "System design".

Table 1 Experimental parameters

Parameters	Values
Computer	Toshiba Satellite M200
Operating system	Linux Kernel 2.6.28-18
Wireless adaptor	DWL-AG660
Wireless driver	MadWifi v0.9.4
Protocol	802.11 b/g
Frequency	2.4 GHz
Transmission power	63 mW
Traffic generator	Iperf - UDP
Payload size	1470 Mbytes
CBR rate	54 Mbps
Accelerometer	Hitachi H48C

Transportation system characteristics

Since our system aims to be widely workable in different train stations and transportation systems, we conduct our experiments on both the High-Capacity MRT and the Medium-Capacity MRT, each with its own movement patterns. There are two types of MRT systems in Taipei MRT, the High-Capacity MRT and Medium-Capacity MRT [22]. These two systems have different rolling stocks, control modes (automatically driven or driver driven), platforms, tracks, and wheels. The time for the train to speed up and to slow down in the station are also different in the two MRT systems. For example, the time the train takes from entering the station to stopping at the platform is about 12 s in Medium-Capacity MRT, and 18 s in High-Capacity MRT, as shown in Figure 3. This difference is

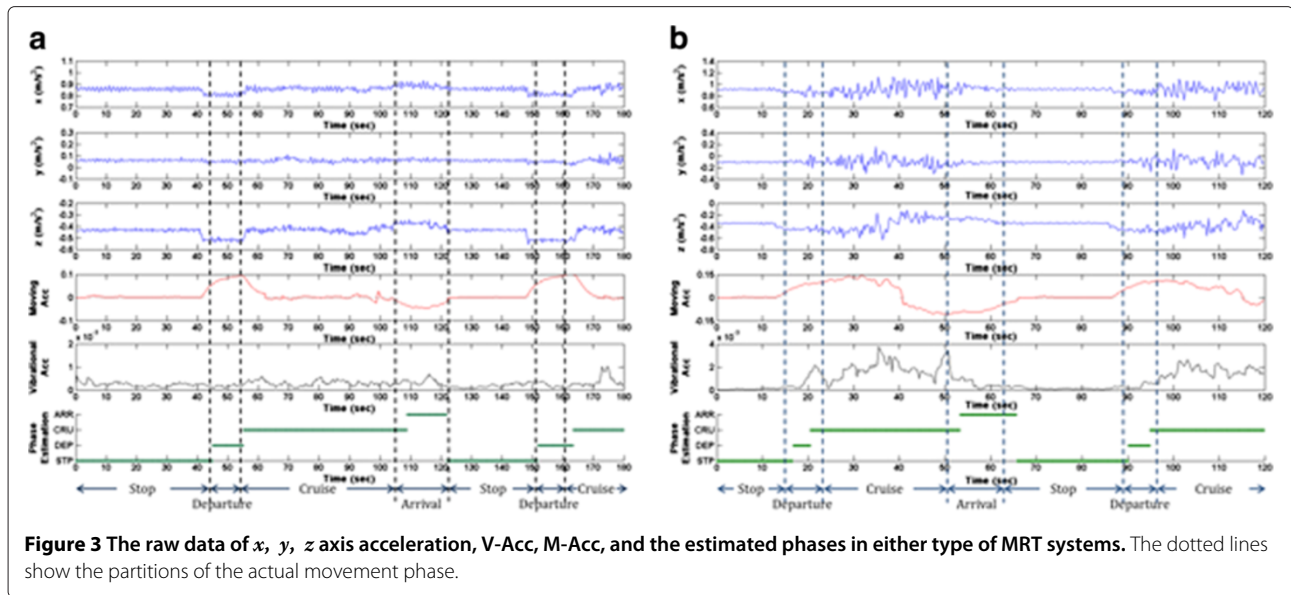


Figure 3 The raw data of x , y , z axis acceleration, V-Acc, M-Acc, and the estimated phases in either type of MRT systems. The dotted lines show the partitions of the actual movement phase.

due to the platform lengths and the numbers of car in two kinds of MRT systems. To understand more about the moving patterns in these two MRT systems, we conduct experiments to collect data in advance. For simplicity, we abbreviate High-Capacity MRT and Medium-Capacity MRT as HC-MRT and MC-MRT, respectively, in the rest of this article.

We conduct our experiments on Wenshan-Neihu Line for MC-MRT, on Xiaonanmen Line and Nangang Line for HC-MRT, as shown in Figure 4. We carry the client and sniffer nodes on the train moving from Taipei Zoo station (start) to Xinhai station (end), and the server node situated in Wanfang Community station (middle) is in between. When the train arrives at Wanfang Community station (middle), the node starts to transmit and lasts for a while until the train departs from the station. The experiments conducted in HC-MRT have similar experimental settings.

Preliminary results

Prior to estimating the movement of the train with accelerometers, we must perform preliminary experiments to investigate the acceleration properties of MRTs. We conduct the following experiments: tilting the accelerometer in different angles, setting the client at the side or the middle of the train, collecting raw data in different stations, and perform the experiments in different types of MRT systems. After analyzing the experimental data, we conclude that the most critical factors are the orientation of the accelerometer and the MRT system type.

In this article, we categorize the train's movement into four phases according to its behavior: Stop (STP), Departure (DEP), Cruise (CRU), and Arrival (ARR). STP

phase refers to the duration the train stops at the station. The DEP phase is the time the train starts to move and then accelerates. In CRU phase, the train moves between the stations at a relatively steady speed. When the train approaches to the station and slows down, it enters the ARR phase. This also represents the process a passenger gets on the train, stays on the train, and gets off the train. This process has a total of five phases:

$$\text{STP(get on)} \rightarrow \text{DEP} \rightarrow \text{CRU} \rightarrow \text{ARR} \rightarrow \text{STP(get off)}$$

Figure 5 shows the raw data of 3-axis acceleration when the train passes through a station in MC-MRT and HC-MRT, respectively. Note that our goal is to estimate a train's movement regardless of the position and orientation of users' accelerometers. Thus, we collect the information of acceleration when the accelerometer is set to head-orientated and tilted orientation. In this article, head-orientated refers to when the x -direction of the accelerometer points towards the first car, the y -direction towards the right side of the train when the user faces the first car, and the z -direction towards the floor. The tilted orientation refers to when the accelerometer is randomly placed. For instance, we put the accelerometer in our bags, or hold it at different orientations.

As shown in Figure 5a,b, we observe that the vibration of acceleration is more acute in MC-MRT. The vibration level is lowest in STP phase, and when the train starts to move (DEP phase), the vibration level immediately elevates. The amplitude of vibration reaches its maximum in CRU phase, and decreases again when the train approaches to the station (ARR phase). The vibration of acceleration turns stable when the train fully stops



Figure 4 The experimental topologies of the two MRT systems.

at the station. The experimental results in both the head-orientated scenario and the tilted scenario are shown to be similar in the MC-MRT.

On the other hand, in Figure 5c,d, the vibration level in HC-MRT is lower than it is in MC-MRT. However, in Figure 5c, we notice that there is a rise in acceleration in the x -axis during DEP phase (the red circle) and a decline when the train approaches to the station (the green rectangle). These rise and decline are actually the acceleration generated from the train's speeding up and slowing down. We can see the acceleration in x -axis increases when the train starts to move, and decreases to about 0 when the

train enters CRU phase, which means the train is moving with a steady speed. When the train approaches to the station, the acceleration in the x -axis drops again in the negative region (starting from around 100 s) because the train's direction of acceleration is now in reverse. Finally, the acceleration is back to around 0 when the train fully stops in the station. Though the acceleration pattern in ARR phase is not as obvious as that in DEP phase, it is still feasible to exploit the information to estimate the movement phases.

Figure 5d is the measured result of acceleration in HC-MRT system when the accelerometer is tilted. As shown

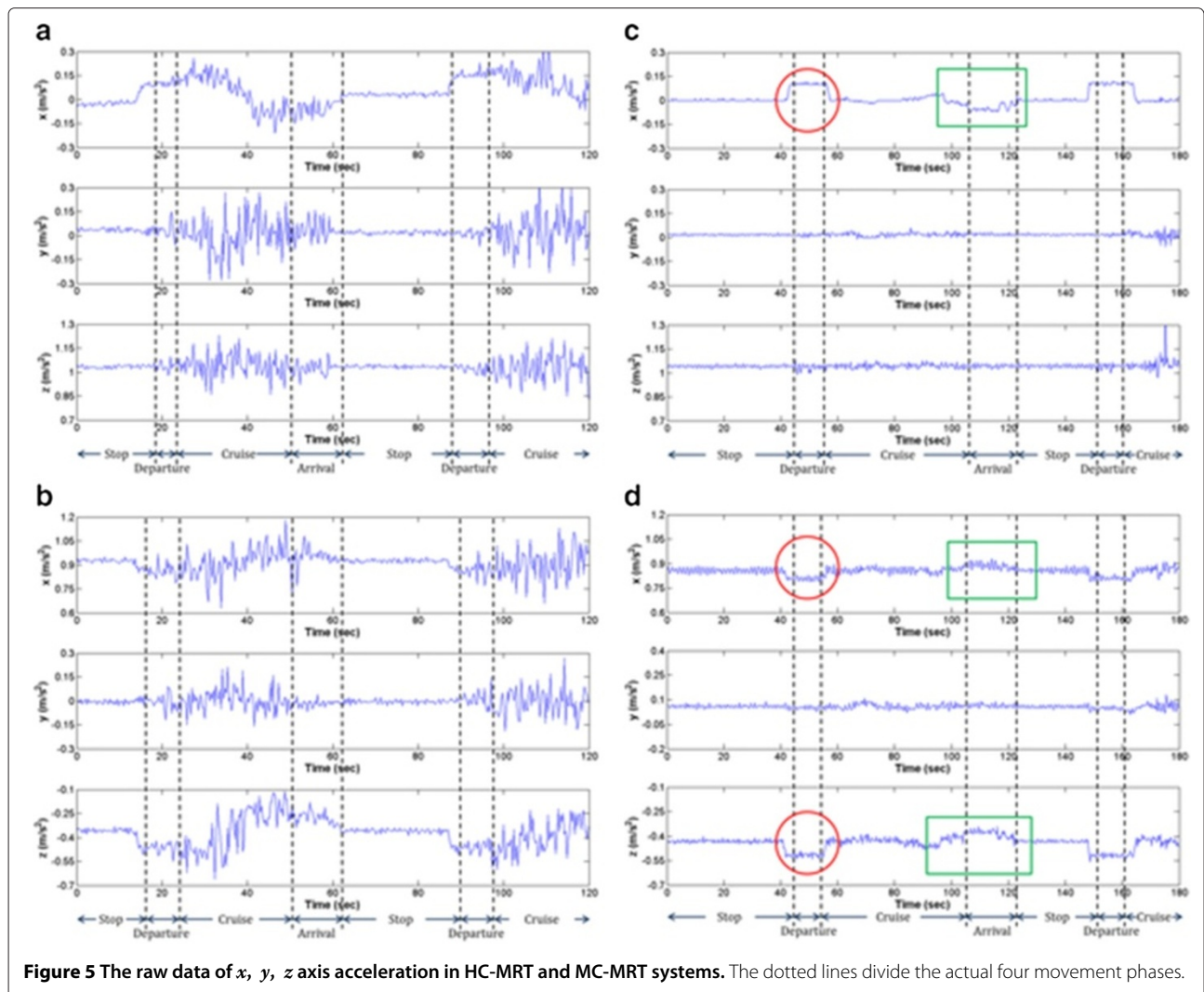


Figure 5 The raw data of x , y , z axis acceleration in HC-MRT and MC-MRT systems. The dotted lines divide the actual four movement phases.

in the circled areas of Figure 5d, although the acceleration patterns are less noticeable, they can still be observed from x -axis and z -axis. The main reason is that acceleration information scatters into two or three axes. This results in a mitigation of the signal intensity. We can also observe that the “rise” during DEP phase in Figure 5c becomes a “decline” in Figure 5d because we have arbitrarily set the orientation of accelerometer. The direction of x -axis of the accelerometer might be opposite to the train’s moving direction. These observations infer that we can not estimate the phases solely by relying on information from one specific axis.

From the observation above, we conceive that the information from the acceleration should be classified into two parts: vibration and movement. These information reside in the high frequency and low frequency of the acceleration, respectively. In the next section, we illustrate the design of our estimation scheme and how it extracts

information from both parts to enhance the estimation accuracy.

System design

In this section, we describe the system design and the implementation of AARA. The details of the two components, movement phase estimation and rate adaptation scheme, are respectively introduced in the following two subsections.

Movement phase estimation by accelerometer

Our goal is to estimate in real-time which movement phase the train is in: STP, DEP, CRU, or ARR. The estimation mechanism has the following properties:

- robust against the accelerometer’s orientation and position
- previous off-line training not required

- user's knowledge about the environments not required

The proposed scheme designed for both HC-MRT and MC-MRT excluded the need of user's knowledge about the environments. In other words, the users neither need to input the information about which type of MRT system they are taking nor make the accelerometer's orientation pointing to a specific direction. However, as described previously, we notice that different transportation systems have different moving patterns. It is hard to estimate the train's movement simply based on the raw data or any single feature. Other than that, as shown in Section "Preliminary results", the measured acceleration is composed of two parts: high frequency and low frequency. The high frequency indicates the short-term vibration of the train, whereas the low frequency shows its long-term movement.

In HC-MRT system, the train's vibration level is relatively low, but the train's speeding up and slowing down patterns are more obvious. By contrast, in MC-MRT system, the train's vibration level is high, but the train's speeding up and slowing down patterns are less obvious. In order to estimate the movement phases based on these two characteristics that apply to different MRT systems, we define those two indicators of acceleration as M-Acc and V-Acc. M-Acc depicts the train's movement, and V-Acc represents the vibration of the train.

The two indicators are defined as follows:

$$\text{Vibrational-Acc} = \text{Var}(\|\hat{A}\| - \|\vec{G}\|),$$

$$\text{Moving-Acc} = \|\text{LP-Filter}(\vec{M}_{\text{est}})\| \times \text{sgn}(\vec{M}_{\text{est}} \cdot \vec{M}_{\text{DEP}}),$$

where \hat{A} is the original measurement of the 3-axis accelerations, \vec{G} is the gravitation, $\vec{M}_{\text{est}} = \hat{A} - \vec{G} - \vec{D}_{\text{off}}$ is the estimated moving acceleration of the train, \vec{D}_{off} is the bias offset due to the imperfection of the accelerometer, and \vec{M}_{DEP} is the estimated moving acceleration recorded in the DEP phase for directional reference.

V-Acc & M-Acc

We define V-Acc as an indicator of the vibration level, and thus we simply compute the variance of the measure's length minus the gravitation's length. The length of a vector is calculated as $\sqrt{x^2 + y^2 + z^2}$. Note that we set $\|\vec{G}\|$ as 1 since the value of gravitation is typically $1 \times g$. The size of a sliding window for computing the variance of the difference between the measurement and gravitation is set to be 20 data points. This means we calculate 20 sets of $(\|\hat{A}\| - \|\vec{G}\|)$, and then calculate their variance. Since the sampling rate of accelerometer is about 0.25 s, the latency of estimation is tolerable. In addition, the size of sliding window is also feasible to estimate the acceleration of a train's vibration.

As for M-Acc, we use Use Use Figure 6 to illustrate the relationship between the acceleration components. We denote the acceleration of a train's speeding up and slowing down as \vec{M} . The vector \vec{M} varies over time, and is actually the moving acceleration that we want to predict by M-Acc. In STP phase, the acceleration of the train's movement should be zero vector ($\vec{M} = \vec{0}$). Our measurement, \hat{A} , however, is not zero vector due to gravitation and the bias offset generated by the imperfection of the accelerometer. We firstly record $(\vec{G} + \vec{D}_{\text{off}})$ and calibrate \hat{A} to be zero vector by eliminating $(\vec{G} + \vec{D}_{\text{off}})$, as shown in Figure 6a.

As the train starts to move, \vec{M} increases. In ideal case, the measurement \hat{A} should be $\vec{M} + (\vec{G} + \vec{D}_{\text{off}})$, so that we can extract \vec{M} by eliminating $(\vec{G} + \vec{D}_{\text{off}})$ from \hat{A} . However, the measurement \hat{A} is easily corrupted, as shown in Figure 6b. Thus, after eliminating $(\vec{G} + \vec{D}_{\text{off}})$, the corrupted \hat{A} will be \vec{M}_{est} but not \vec{M} . We calculate the length of \vec{M}_{est} and apply the EWMA (exponentially weighted moving average) with $\alpha = 0.075$ as a LP-Filter to prevent the abrupt burst of the measurement. Though it is enough to predict the length of \vec{M} , its direction is unknown, so we are not able to distinguish whether the moving acceleration is caused by speeding up or slowing down. Therefore, we record \vec{M}_{est} in DEP phase as \vec{M}_{DEP} , which represents the directional reference of the train's movement. And then, we compute the inner-product between \vec{M}_{est} and \vec{M}_{DEP} . If the sign of $(\vec{M}_{\text{est}} \cdot \vec{M}_{\text{DEP}})$ is positive, it means that the angle between \vec{M}_{est} and the direction of train's movement is less than 90° (as shown in Figure 6b, the α angle is less than 90° ; thus, the direction of \vec{M}_{est} and the direction of train's movement are in the similar directions).

On the other hand, if the sign of $(\vec{M}_{\text{est}} \cdot \vec{M}_{\text{DEP}})$ is negative, the angle of the measurement and the direction of the train's movement is greater than 90° (as shown in Figure 6c, the β angle is over 90° , which means the measured \vec{M}_{est} and the direction of train's movement are in opposing directions).

Auto-calibration

As previously stated, we calibrate \hat{A} by removing $(\vec{G} + \vec{D}_{\text{off}})$, when the train is in STP phase. We also record \vec{M}_{est} in DEP phase as \vec{M}_{DEP} , as a directional reference of the train's movement. AARA performs the auto-calibration in both STP and DEP phases to prevent the propagation of estimation error. In Figure 6d, the orientation of the accelerometer is changed. The vectors of the gravitation and the bias offset in the new vector space are also changed to be \vec{G}' and \vec{D}'_{off} , respectively. If we make the measurement \hat{A} minus the same $(\vec{G} + \vec{D}_{\text{off}})$ as before, we derive a wrong \vec{M}_{est} , as shown in Figure 6d. Therefore, AARA calibrates the $(\vec{G} + \vec{D}_{\text{off}})$ in STP phase and the \vec{M}_{DEP} in DEP phase automatically. In other words, the unexpected estimation error will not propagate because the

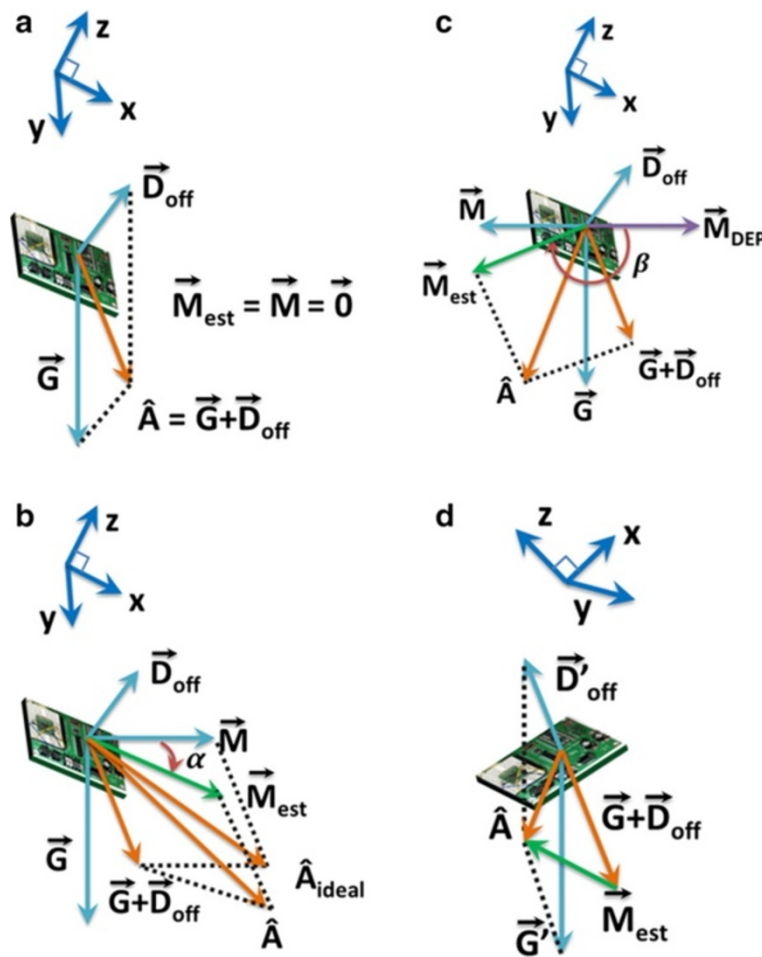


Figure 6 Acceleration components of the accelerometer in different phases.

calibration of the estimator will be reset when the train stops at the next station.

Figure 3a,b show the two indicators when a train passes through the stations in either MRT system.

Two-stage integrated decision function

After processing the raw data, the information of movement and vibration are now much clear to estimate the phases of the train’s movement. In our design, we utilize a two-stage estimation method to determine the movement phase.

Before introducing the two-stage integrated decision function, we present the behavior of the acceleration in either MRT system. From Figure 3a, we see that the pattern of the M-Acc is neat in HC-MRT system because the train runs more smoothly. In STP phase, the M-Acc is stable. In DEP phase, the M-Acc starts to increase and reaches to the maximum as it changes to CRU phase. In CRU phase, the M-Acc is stable at around 0 most of the time, only with a few sharp fluctuations. In ARR phase, the

M-Acc gradually decreases in the negative region and then increases. The result shows the M-Acc has the high correlation with the speed changes of the train in HC-MRT system.

However, in Figure 3b, it is observed that the use of M-Acc alone is incapable to precisely estimate the phases in MC-MRT system. For example, due to the noise caused by the more acute vibration of the car, the M-Acc in the CRU phase is not as stable as that in the HC-MRT system. Hence, these unpredictable shakings of M-Acc in the CRU phase will cause the estimation mechanism misjudge the shakings as the signal of deceleration of the train, and thus misestimate the ARR phase too early. The V-Acc is helpful in this problem. It is clear that the V-Acc is higher when the train is running. We set thresholds of the V-Acc according to it’s average value for assisting in the estimation. While the V-Acc is high, the train must be between the stations and not entering the station yet.

The proposed two-stage integrated decision function is able to estimate the movement phases precisely. The first

stage is to estimate which type of the train the user is on, the HC-MRT or the MC-MRT, according to the level of V-Acc. Note that the vibration level is very low in STP phase in both of the MRT systems, so we are unable to distinguish which MRT system it is when the passenger just gets on-board. Fortunately, since the pattern of M-Acc from STP phase to DEP phase, and from DEP phase to CRU phase, are very similar in both MRT systems, AARA can determine the time the train enters the DEP and CRU phase, respectively even when AARA does not yet know which type of MRT the passenger is taking. After the train enters CRU phase, the level of V-Acc is considerably

different in HC-MRT and MC-MRT, and thus it is able to distinguish the type of MRT system then.

The next stage is to estimate the ARR and STP phases in both MRT systems. The decision function mainly utilize the M-Acc to predict the phases while V-Acc is used to improve estimation accuracy. Since the level of M-Acc and V-Acc are different in the two MRT systems, we set different thresholds to estimate the ARR phase. We present the details and the parameters of the two-stage decision function in Figure 7 and Table 2. In Table 2, T denotes different thresholds in each decision expression, k denotes the multiples of the thresholds, and $t_1 - t_7$ refer to the

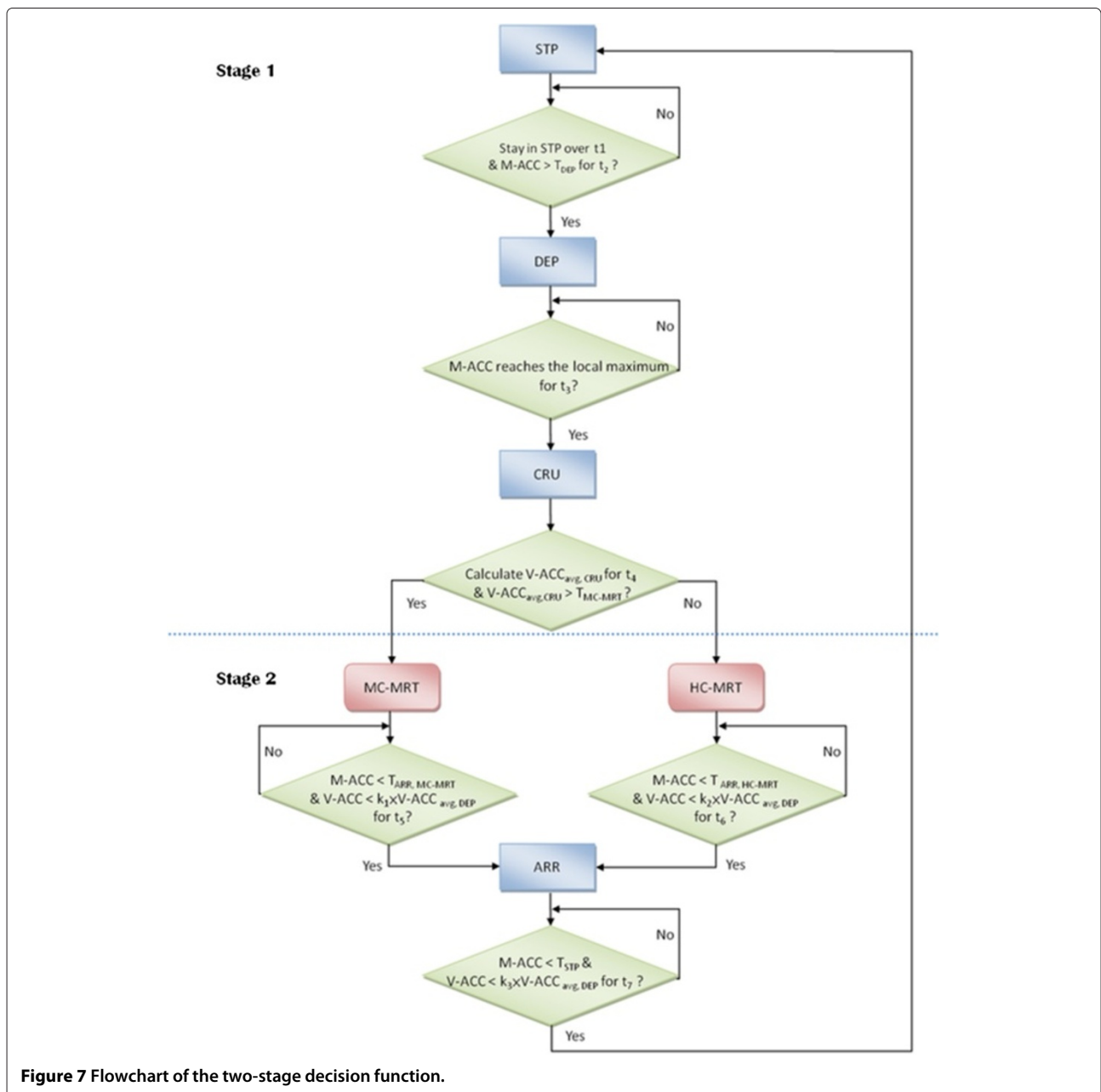


Figure 7 Flowchart of the two-stage decision function.

time duration for the decision expression to stay True so that the decision expression goes to Yes. Take the top decision expression in Figure 7 for example, the decision expression shows that it switches from STP phase to DEP phase only when the train is estimated to stay in STP over t_1 (10 s) and when M-Acc is greater than T_{DEP} for t_2 (3 s).

We pick two lines from HC-MRT and one line from MC-MRT yielding a total of 11 stations to verify the accuracy of the estimation mechanism. The experimental results are presented in the following section.

Rate adaptation scheme

In this section, we describe how we exploit the phase estimation to enhance the rate adaptation in AARA. We implement AARA by modifying SampleRate bit rate selection algorithm [7]. As described in Section “Related work”, SampleRate is the default rate control algorithm of the latest version of MadWifi driver [18]. Its performance has been extensively tested and verified. Since it is the most widely used rate adaptation scheme, we integrate it into our phase estimation to boost the transmission performance.

The concept of our rate adaptation scheme is simple and novel. Since traditional rate adaptation algorithms are designed for general purposes, it has no auxiliary information about the channel quality in general situations. They rely on certain measurements, such as packet error rate or signal strength, to recognize the channel’s quality and can only be a passive adaptor to the channel. By contrast, since we know that the channel condition is different when a train passes through a station in each phase, AARA identifies the train’s current movement phases to proactively adopt suitable transmission strategies. In other words, the proposed mechanism is a two-tier design. We use the information of the train’s movement phases estimated by accelerometer to deal with long-term channel variation, and make use of the rate adaptation scheme, such as SampleRate algorithm, to handle the channel variation in short-term time.

Table 2 Parameters of the two-stage decision function

Parameters	Values	Parameters	Values (s)
T_{MC-MRT}	0.001	t_1	10
T_{DEP}	0.01	t_2	3
$T_{ARR, MC-MRT}$	-0.02	t_3	1
$T_{ARR, HC-MRT}$	-0.06	t_4	10
T_{STP}	0.02	t_5	2
k_1	3	t_6	2
k_2	2	t_7	1
k_3	3		

The SampleRate algorithm maintains a table of average successful transmission time of each rate scheme, and it chooses the bit rate scheme with the shortest expected transmission time to transmit, as described in Section “Related work”. In ARR phase, when the train is just approaching to the station, there is no history record yet, so SampleRate chooses the highest rate to transmit, e.g. 54 Mbps in 802.11g. However, the channel is poor due to the large distance at this instance, and thus the transmission success rate is low. AARA knows that the channel quality is unstable when the train enters the station, so the MS should adopt a more robust bit rate. Therefore, instead of adopting from the high bit-rates as SampleRate does, AARA adopts from low bit-rates and uses a higher probing rate to adapt to the changing channel. We choose the parameter of probing rate in ARR phase r_0 in Figure 8 as 10%, which is the default probing rate in SampleRate algorithm.

When the train stops, the wireless channel turns stable, so the MS should immediately switch to higher rates to maximize the throughput. However, due to the consecutive failure of probing high rate in the previous phase, SampleRate still considers the channel quality poor, thereby using the lower rate to transmit. AARA clears the stale record and probes from the highest rate as it enters STP phase. Besides, once the MS reaches to its optimal bit-rate, the extra probing packets should be reduced, so the MS can stay using the optimal bit-rate most of the time. Hence, we decrease the probing rate in STP phase (r_1 is set to be 1% in our parameter setting). When the train starts moving, the channel quality degrades promptly as the train speeds up, so the MS should sense the unstable environment and automatically adapt to the

AARA Procedure

1. Sense the change of the movement phases
2. **if** *phase* = *Arrival* **then**
3. Activate the wireless device
4. Clear history record
5. Adopt from Lowest Bit Rate
6. Probing Rate = r_0
7. **else if** *phase* = *Stop* **then**
8. Clean history record
9. Adopt from Highest Bit Rate
10. Probing Rate = r_1
11. **else if** *phase* = *Departure* **then**
12. Probing Rate = r_0
13. **else if** *phase* = *Cruise* **then**
14. Turn to Sleep mode
15. **end if**

Figure 8 The procedure of AARA.

channel again. Figure 8 illustrates the pseudo code of AARA's procedure.

Experimental evaluation

We test the transmission performance of AARA and SampleRate in Taipei MRT's HC-MRT and MC-MRT systems under the same conditions. The following experimental result shows that AARA can accurately estimate the movement and improve the throughput.

Movement estimation of the trains

In order to adopt different rate adaptation strategies to every movement phase, the most critical issue is to accurately estimate the movement phases. We employ two metrics in the evaluation: recall and delay. Recall is defined as the portion of correct categories that are estimated, and is represented by Estimated-Y/Real-Y. For example, in Figure 3a, there are 67 data points in DEP phase, however, the estimator only estimate 60 data points to be in DEP phase during the real DEP phase; thus, the recall of the estimation of DEP phase is 89.6%. On the other hand, we also measure the estimation delay time, which is a metric more directly related to the user. We define a positive value of delay time to be when the beginning of the estimative phase is later than its actual time. By contrast, a negative delay denotes that the beginning of the estimative phase is earlier than it really is.

The experimental results for the estimative mechanism are summarized in Tables 3 and 4. We show the recall and delay in each phase in the two types of MRT systems. Note that the result shown in the two tables agrees with the result shown in Figure 3a,b.

From Tables 3 and 4, we observe that the total recall in the two MRT systems are 94.2 and 89.4%. The recall is lower in MC-MRT because the train's vibration is severer, causing the M-Acc more disorder than that in HC-MRT. Thus, this effect caused by severer vibration degrades the performance of estimation in MC-MRT. We can see that most of the recall is over 74%, with only one exception. In the DEP phase of MC-MRT, the recall is 55.8%. This is due to some inevitable estimation delay generated by the estimator when using EWMA to smooth M-Acc. Since the duration of DEP phase is relatively short, only 8.37 and

7.19 s in Figure 3b, the effect of delay is more obvious. Actually, from the user's view, there is only 2.04 s delay in estimation of DEP phase in MC-MRT, similar to the delays in other phases.

Rate adaptation in MC-MRT and HC-MRT

In this section, we present the result of the proposed rate adaption scheme assisted by the movement phases estimation. In Figure 9, we show the throughput and bit-rate selection of AARA and SampleRate, respectively, during the four movement phases in HC-MRT. The same experiments are also conducted in MC-MRT, as shown in Figure 10. We plot the figures of AARA and SampleRate in different figures because the trains' behavior might not totally match in each run. Hence, the duration of the movement phases is also slightly different in the experiments conducted.

First, we observe that AARA achieves significantly higher throughput than that of SampleRate during the four phases. In ARR phase, AARA adopts from the lowest bit-rate because the MS should use a robust bit-rate when the train just enters the station. However, SampleRate chooses the bit-rate with the shortest expected transmission time to adopt, i.e., 54 Mbps in 802.11g. As described in Section "System design", we know that the distance between the MS on the train and the BS in the station is still far, hence, the channel quality is poor. These packets in higher bit-rate mostly fail to transmit. This not only wastes the time to send packets that are unable to be received, but also hinders the bit-rate to increase in the latter part of the ARR phase. In Figure 9, we see that the throughput of AARA begins to increase from -5 s, while SampleRate maintains the same throughput level when the train is close to the station (still in ARR phase). This is because the default SampleRate will block the bit-rates with consecutive transmission failures. Since consecutive failures often occur when SampleRate tries the bit-rate from the highest one, it cannot switch to a higher bit-rate even the channel quality is better, i.e., the train is closer to the station but not yet fully stops.

Next, we see the STP phase, the most crucial phase that dominates the overall throughput. In the beginning of STP phase, AARA adjusts to the highest bit-rate because it

Table 3 Results of movement estimation in high-capacity MRT

Phase	Recall (%)	Delay (s)
STP	100	-0.32
DEP	89.6	1.22
CRR	94.5	1.20
ARR	74.6	4.22
Total	94.2	-

Table 4 Results of movement estimation in medium-capacity MRT

Phase	Recall (%)	Delay (s)
STP	92.9	2.98
DEP	55.8	2.04
CRR	100	-2.33
ARR	76.2	3.00
Total	89.4	-

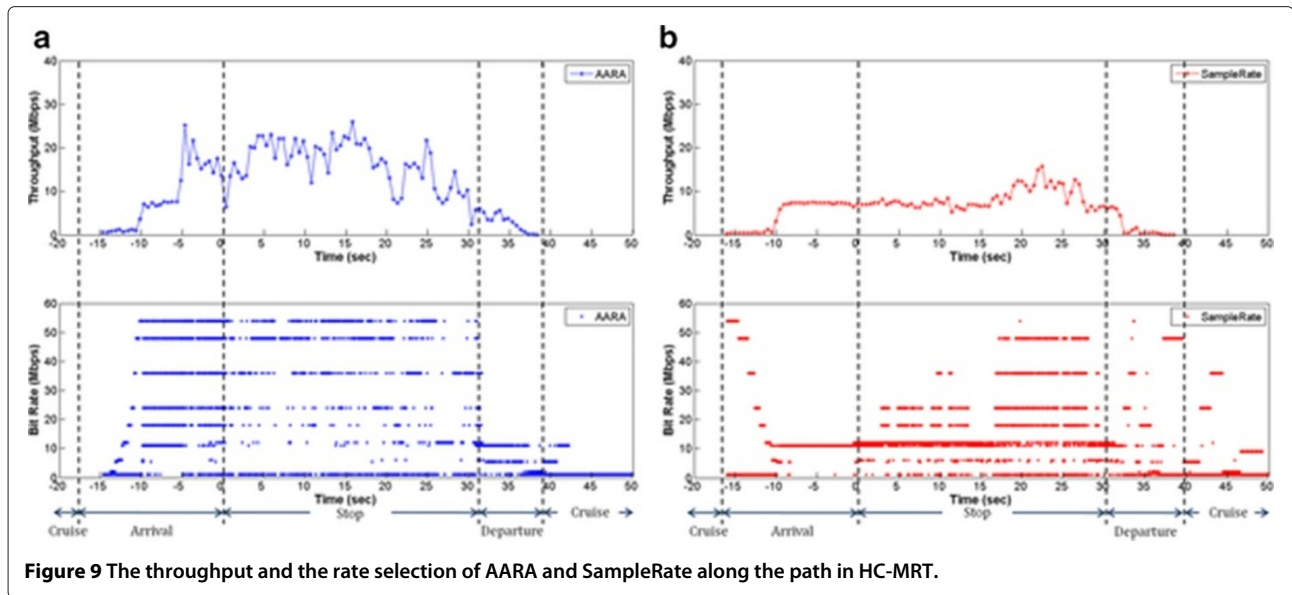


Figure 9 The throughput and the rate selection of AARA and SampleRate along the path in HC-MRT.

removes the history transmission statistics, which records the outdated channel condition in the previous phase. On the other hand, SampleRate has no extra information to know that the history record cannot reflect the channel condition anymore. Instead, SampleRate considers that the channel quality is as bad as before; thus, it uses a relatively lower bit-rate, which then downgrades the throughput. SampleRate keeps using lower rates, mostly 11 Mbps, for about 15s in both Figures 9b and 10b. Although SampleRate starts to adopt higher rates and increase the throughput from 20s, it is unable to fully utilize the STP phase as the train is about to depart.

During the DEP phase, the performances of AARA and SampleRate are similar. In fact, SampleRate is a sensitive

rate adaptation scheme, because it is conservative to raise its bit-rate, yet it is susceptible to drop off its bit-rate. In the DEP phase, the channel quality degrades quickly. SampleRate with default parameter is able to handle the degradation of channel quality because it can quickly decrease the bit-rate. Thus, we adjust the probing rate back to default value 10%. Besides, AARA shuts down the wireless device to enter sleep mode for the sake of conserving power during the CRU phase. We analyze the performance of power-saving in Section “Power saving”.

To further investigate what causes the difference between the performances of AARA and SampleRate, we also analyze the ratio of transmitted packet in each bit-rate of all four phases. In Use Figure 11, darker marked colors

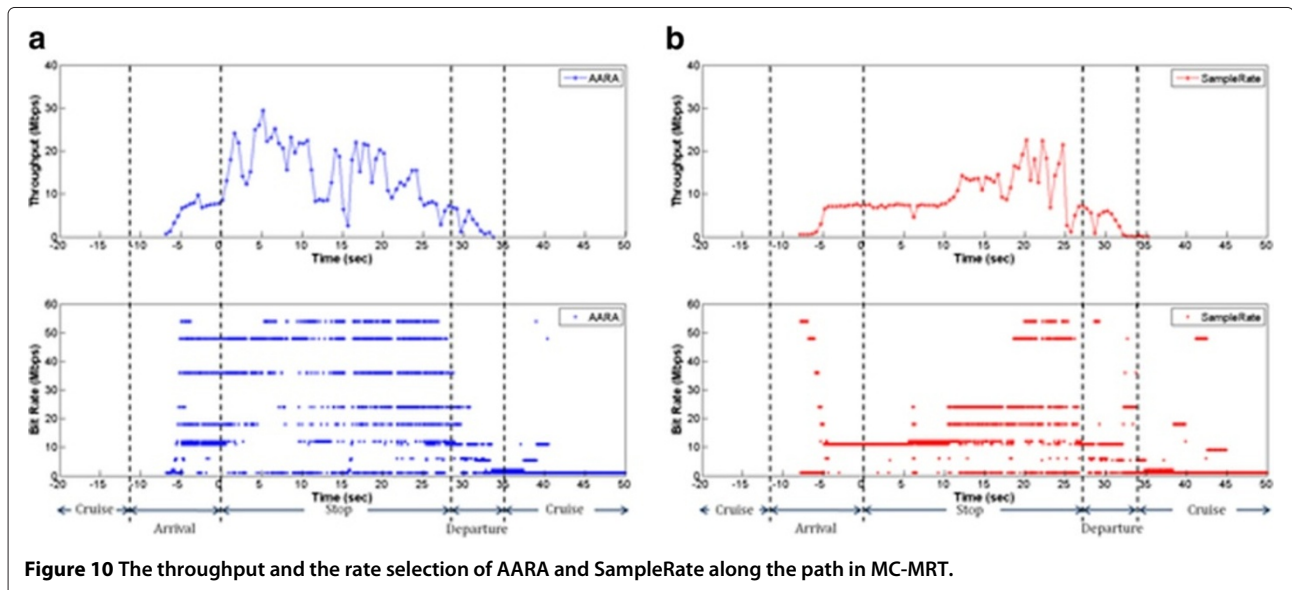
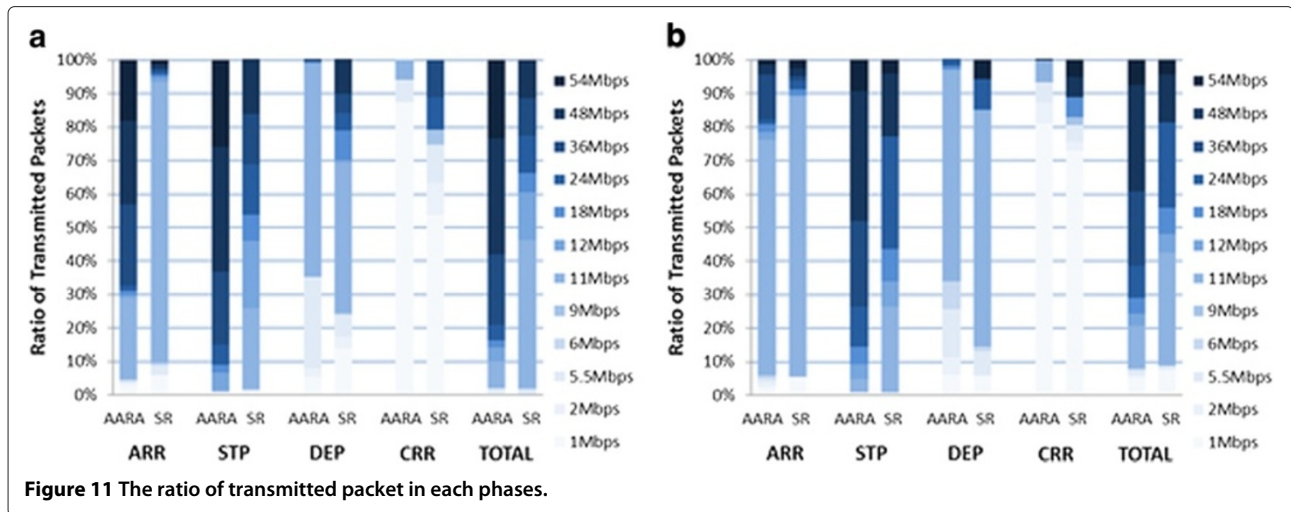


Figure 10 The throughput and the rate selection of AARA and SampleRate along the path in MC-MRT.



represent higher bit-rates. We see that AARA adopts more “higher rates” than SampleRate does during the most critical period, the STP phase. This is the main reason why the throughput of AARA outperforms that of SampleRate. Note that, in ARR phase, AARA uses much more higher bit-rates in HC-MRT compared to MC-MRT system. As mentioned earlier, the duration of ARR phase is shorter in MC-MRT, so the time AARA can use higher bit-rate is also shorter. Hence, the advantage of employing the bit-rate from the lowest rate in ARR phase is less noticeable in MC-MRT. Nevertheless, AARA still sends more packets in higher bit-rate in MC-MRT. In DEP phase, since the channel qualities degrade rapidly, both AARA and SampleRate tend to employ lower bit-rates. SampleRate uses more higher bit-rate in CRR phase, but the transmitted packets are mostly useless. This is because the number of packets in this phase is extremely few and the packets cannot be received due to the long distance. We also compute the ratio of transmitted packets in all the phases. The result shows that AARA transmits more packets in high bit-rate when the channel quality is good, and thereby AARA achieves higher throughput than SampleRate does.

Next, we discuss the latency issue in AARA’s phase estimation. The estimation of the phases is sequential. For instance, when in the ARR phase, the only two next possibilities are either to switch to the STP phase or to stay in the ARR phase. Because the most important phase to the overall throughput is STP phase, we focus on the estimation of STP phase start point. If the estimated start point is delayed, AARA will clear the history record at a later point. For example, if AARA estimates the train stopping in the station at a time that is already 5 s later than it really is, then AARA clears the history record 5 s later. This means we will waste 5 s without taking the opportunity to use higher bit-rates. However, such loss will not cause disaster as it only degrades the

throughput gain. The overall throughput of AARA is still superior to SampleRate, which has no extra context information.

Effect of channel quality

So far, we focus on the idea that employing higher bit-rate to bring higher throughput when the channel quality is good. We also need to study that what if the channel quality is not sufficient to employ high bit-rate even when the channel quality is already the best. In this section, we evaluate the effect of channel quality as well. We design three scenarios of the experiments where the MS is placed at different positions in the car while keeping the BS in the same position on the platform. A total of three different MS-to-BS distances are measured (near, medium, far), as shown in Figure 12. The distance of MS and BS correlates with the channel quality. The setting of the experiment of medium distance is the same as the previous subsection, so Figure 12c,d are the same as Figure 9a,b. We put the figures here again for easier comparison.

In the far scenario, because the transmission failure is too serious due to the worse channel quality, SampleRate does not try higher bit-rate at all; while AARA still tries higher bit-rate in STP phase. The main reason is that AARA cleans the stale transmission record at the beginning of STP phase, and also disable the blocking mechanism of bit-rate with consecutive failures. Though the channel quality is worse than that in the setting of medium distance, it is possible to send packets successfully in higher bit-rate sometimes. However, the sensitive property of SampleRate almost completely abandons the opportunities. On the other hand, in the near scenario, the overall throughput of AARA is naturally superior to that in the medium scenario. SampleRate in near scenario outperforms itself in medium scenario as well. The result is

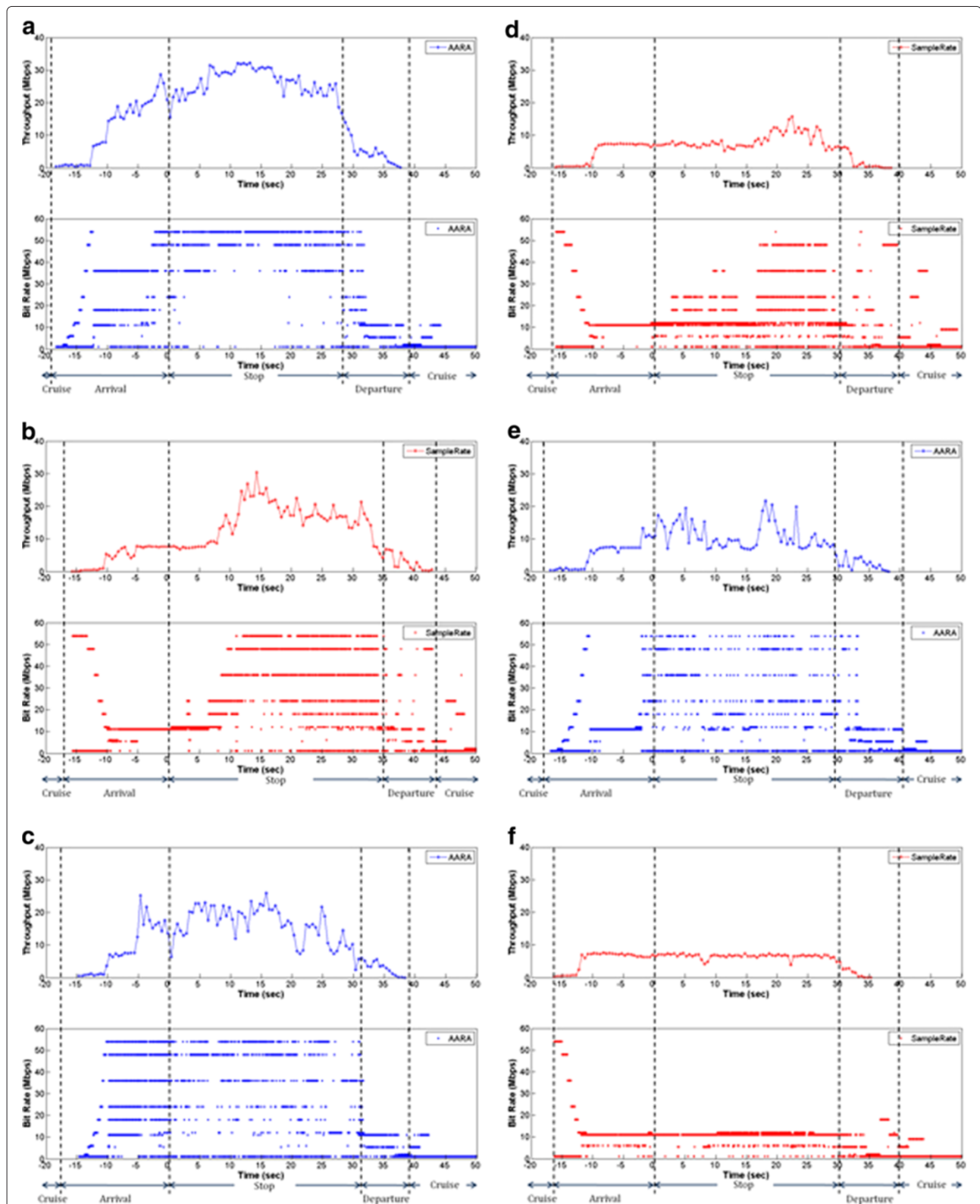
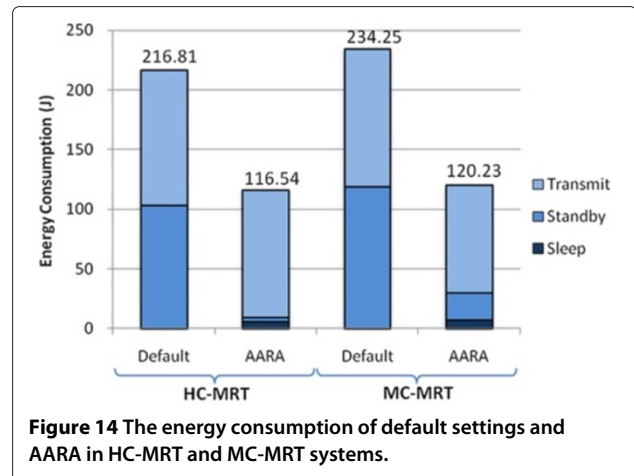
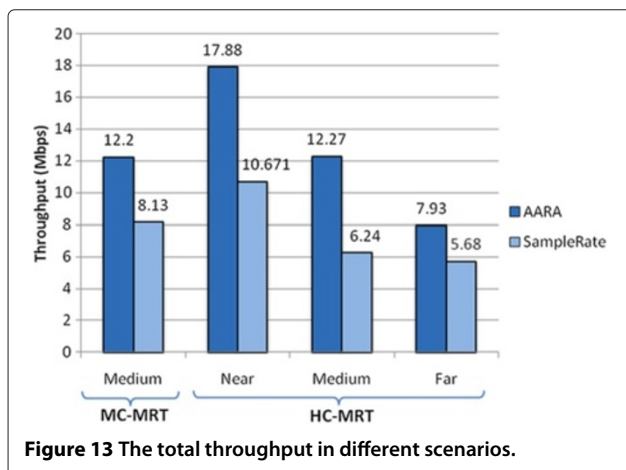


Figure 12 The effect of channel quality on the throughput and the rate-selection of AARA and SampleRate along the path in HC-MRT. We conduct the experiments in three distances between the MS and BS. The RSI values measured at the receiver when the train stops in the station are -46.30 , -63.19 , and -68.07 dBm as the scenarios of near, medium and far distance, respectively.

reasonable. It becomes easier for SampleRate to perceive the upgrade of the channel quality when the train enters STP phase since there is not so many accumulative transmission failure in the ARR phase. Thus, the higher bit-rate are not be blocked for so long. In other words, the channel qualities in the latter part of ARR phase and the STP phase are more similar, and thus the history record is partially credible than that in the medium distance scenario. Nevertheless, SampleRate still needs about 10s after the train stops in the station to raise the bit-rate. In this scenario, AARA also boosts the bit-rate earlier than it does in the medium scenario. After testing the bit-rate from the lowest one in ARR phase, AARA senses it is able to use higher rate promptly, and thus boosts the overall throughput.

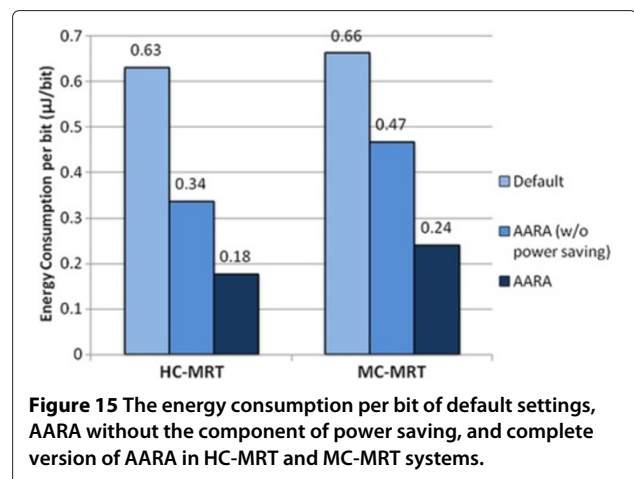
Figure 13 depicts the overall throughput of AARA and SampleRate in each scenario. In HC-MRT system, the result show that the throughput decrease as the distance between the MS and the BS getting further. The throughput of AARA outperforms that of SampleRate in each scenario, even when the channel quality is worse (fewer chance to employ higher bit-rate) and better (SampleRate is able to perceive the channel upgrade more quickly). Reasonably, the improvement of the performance is most remarkable in the medium distance scenario. The throughput of AARA almost doubles that of SampleRate in this scenario.

The overall throughput gain in Figure 13 is $1.50\times$ in MC-MRT, $1.68\times$ in the near distance scenario in HC-MRT, $1.97\times$ in the medium distance scenario in HC-MRT, and $1.40\times$ in the far distance scenario in HC-MRT. These experiments show that AARA, which is an enhancement of SampleRate according to out-of-band information, naturally performs better than that of SampleRate. Even when the channel quality is extremely bad (worse than our far scenario), AARA's throughput might only degrade to be almost as low as SampleRate's throughput.



Power saving

In addition to transmission performance, power saving is also deemed as a key issue. When the train departs from the platform, the MS and the BS are disconnected to each other. Leaving the wireless devices activated during this time consumes a substantial amount of energy. By contrast, if the wireless device is turned to sleep mode when data is not able to be transmitted, we can efficiently reduce power consumption. To achieve that, the MS has to automatically wake up the wireless device from sleep mode when the train is approaching to the next station. With the proposed mechanism of the train's movement estimation, it is easy to properly switch to sleep mode to save power, and to reactivate the device to transmit data. Specifically, we simply switch the wireless device into sleep mode at the beginning of CRU phase and reactivate it in the end of CRU phase. With the accurate phase estimation, AARA is able to reduce substantial power consumption without degrading the throughput.



We compute the amount of power AARA saves and show the experimental result in Figure 14. The duration for calculating power consumption is from the beginning of DEP phase when a passenger gets on the train, to the end of ARR phase when the train stops at the station after the next station. For example, in Figure 4b, a passenger gets on the train at the MRT Wanfang Hospital station, moves toward MRT Wanfang Community station, and finally gets off at the MRT Muzha station. The BS is settled in Wanfang Community station. We compute the energy consumption from the beginning of DEP phase in MRT Wanfang Hospital station to the end of ARR phase in MRT Muzha station. There are three operating modes in commodity wireless cards: transmit mode, standby mode, and sleep mode. Transmit mode denotes the wireless card is activated and is transmitting the packets; standby mode denotes the wireless card is also activated but not transmitting packets; sleep mode denotes the wireless card is not activated. If power saving mechanism is not included, the wireless adaptor only operates in transmit mode and standby mode during a passenger's entire trip. According to the data sheet of NIC we used [23], transmit mode and standby mode will consume about 1,650 and 1,250 mW, respectively. Given the power saving mechanism, in CRU phase, we switch the wireless adaptor to sleep mode, which only consumes 70 mW. In fact, the power consumption of accelerometer is fairly low to be negligible.

Figure 14 compares the energy consumption between AARA (with power saving) and default setting (without power saving) under the same condition stated above. In the experiments performed in HC-MRT and MC-MRT, AARA consumes, respectively only 54.2 and 46.6% of the total energy consumed by default settings. We see that the energy consumption in transmit mode is almost the same in AARA and default setting. However, AARA reduces the inessential power consumption in standby mode and increase the time operating the wireless device in sleep mode because the power consumed in sleep mode is much lower than in standby mode.

Figure 15 takes both energy consumption and transmission performance into consideration in the meanwhile. We compare the issue of energy consumption per bit in three experimental scenarios: default setting (SampleRate without power saving), AARA without and with the power saving mechanism. Here AARA without power saving mechanism only activates part of AARA's function—rate adaptation—but its function of shutting down the wireless card during CRU phase is disabled. This figure reveals that AARA's rate adaption component reduces the energy consumption per bit in terms of transmission performance. On the other hand, AARA's power saving component reduces the energy consumption per bit in terms of consuming less energy. The complete version of

AARA consumes only 28.6 and 36.4% of the energy to transmit a data bit compared with the default setting in either MRT system.

Discussion

In this work, we have designed and implemented the AARA in mobile WiFi devices. The experimental results demonstrated the feasibility of the proposed scheme. In our current prototyping, the accelerometer is attached to the mobile device. As acceleration information has been shown useful to assist wireless communications, we could seek to obtain acceleration information with various methods. For example, for radio system built-in for train vehicles, the acceleration information might possibly obtained from train control system. For radio system on motorcycles, the acceleration information might be obtained from its control panel. Accelerometer could be built-in internally (e.g. smartphones have built-in accelerometer) or connected externally (e.g. connected via USB port). We foresee this acceleration-assisted wireless communication technology to be applied in different forms.

Conclusion

In this article, we develop a new rate adaptation mechanism, AARA, to enhance SampleRate to adapt the transmission rate with the aid of accelerometer. AARA achieves the goal of boosting transmission performance on the MRT system. We use accelerometer to identify the patterns of the vibration and movement of the train, and to estimate the four movement phases of the train, regardless of the accelerometer's orientations. Afterwards, we utilize the estimation result to employ suitable bit-rate adaptation strategies in each phases. Crucially, as STP phase provides the best transmission opportunity, AARA attempts to properly employ higher bit-rates so as to seize the limited time to maximize the throughput. In addition, we take power efficiency for transmission into concern. The experiments conducted on the two types of Taipei MRT system, High-Capacity MRT and Medium-Capacity MRT, show that AARA improves both the transmission performance and the power efficiency compared with the default settings.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This work was also supported by the National Science Council, the National Taiwan University and the Intel Corporation under Grants NSC 100-2911-I-002-001, and 101R7501.

Author details

¹Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan. ²Department of Electrical Engineering, Graduate Institute of Communication Engineering, Intel-NTU Connected Context Computing Center, National Taiwan University, Taipei 106, Taiwan.

Received: 1 June 2011 Accepted: 30 May 2012

Published: 7 August 2012

References

1. Y Zheng, Q Li, Y Chen, X Xie, WY Ma, in *Proceedings of the 10th international conference on Ubiquitous computing (UbiComp)*, 2008. Understanding mobility based on GPS data, pp. 312–321
2. D Patterson, L Liao, D Fox, H Kautz, in *Proceedings of the fifth international conference on Ubiquitous computing (UbiComp)*, 2003. Inferring high-level behavior from low-level sensors, pp. 73–89
3. S Reddy, M Mun, J Burke, D Estrin, M Hansen, M Srivastava, Using mobile phones to determine transportation modes, *ACM Transactions on Sensor Networks (TOSN)*. **vol. 6**(no. 2), 1–27 (2010)
4. M Lacage, M Manshaei, T Turetli, in *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems (MSWiM)*, 2004. IEEE 802.11 rate adaptation: a practical approach, pp. 126–134
5. Onoe Rate Control. [<http://madwifi-project.org/wiki/UserDocs/RateControl>]
6. SHY Wong, H Yang, S Lu, V Bharghavan, in *Proceedings of the 12th ACM international conference on Mobile computing and networking (MobiCom)*, 2006. Robust rate adaptation for 802.11 wireless networks, pp. 146–157
7. JC Bicket, Bit-rate Selection in Wireless Networks. Master's thesis, MIT, 2005
8. P Shankar, T Nadeem, J Rosca, L Iftode, in *IEEE International Conference on Network Protocols (ICNP)*, 2008. CARS: Context-aware rate selection for vehicular networks, pp. 1–12
9. A Rahmati, L Zhong, in *Proceedings of the fifth international conference on Mobile systems, applications, and services (MobiSys)*, 2007. Context-for-wireless: context-sensitive energy-efficient wireless data transfer, pp. 165–178
10. A Joshi, P Kulkarni, in *Proceedings of the ACM SIGCOMM 2010 conference, Poster Session*, 2010. Vehicular wifi access and rate adaptation, pp. 423–424
11. N Santhapuri, J Manweiler, S Sen, X Bao, R Choudhury, S Nelakuditi, in *17th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN)*, 2010. Sensor assisted wireless communication, pp. 1–5
12. J Ott, D Kutscher, in *Proceedings of the 23rd Conference of the IEEE Communications Society (INFOCOM)*, 2004, vol. 1. Drive-thru Internet: IEEE 802.11b for "automobile" users,
13. D Hadaller, S Keshav, T Brecht, S Agarwal, in *Proceedings of the fifth international conference on Mobile systems, applications and services (MobiSys)*, 2007. Vehicular opportunistic communication under the microscope, pp. 206–219
14. J Eriksson, S Madden, in *Proceedings of the 14th ACM international conference on Mobile computing and networking (MobiCom)*, 2008. Cabernet: vehicular content delivery using WiFi, pp. 199–210
15. J Zhao, T Arnold, Y Zhang, G Cao, in *Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking (VANET)*, 2008. Extending drive-thru data access by vehicle-to-vehicle relay, pp. 66–75
16. A Giannoulis, M Fiore, E Knightly, in *Proceedings of the 6th international conference on Mobile systems, applications, and services (MobiSys)*, 2008. Supporting vehicular mobility in urban multi-hop wireless networks, pp. 54–66
17. P Deshpande, A Kashyap, C Sung, S Das, in *Proceeding of the 7th international conference on Mobile systems, applications, and services (MobiSys)*, 2009. Predictive methods for improved vehicular WiFi access, pp. 263–276
18. Multiband Atheros Driver For WiFi. [<http://madwifi-project.org>]
19. Iperf. [<http://sourceforge.net/projects/iperf>]
20. Wireshark. [<http://www.wireshark.org/>]
21. Radiotap. [<http://www.radiotap.org/>]
22. Taipei Metro. [<http://english.trtc.com.tw>]
23. Data Sheet of Dlink DWL-AG660. [<http://www.dlink.com/products/?pid=DWL-AG660>]

doi:10.1186/1687-1499-2012-246

Cite this article as: Lai et al.: Accelerometer-assisted 802.11 rate adaptation on mobile WiFi access. *EURASIP Journal on Wireless Communications and Networking* 2012 **2012**:246.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com