Improving Access Point Association Protocols Through Channel Utilization and Adaptive Probing

Tingting Sun, Yanyong Zhang, and Wade Trappe, *Fellow, IEEE*

Abstract—We propose a distributed access point selection scheme by which nodes select an appropriate access point to associate with based upon each individual device's channel utilization. In this paper, we define channel utilization as the ratio of required bandwidth to estimated available bandwidth. By incorporating channel utilization into the access point selection protocol, we can effectively reduce unnecessary reassociations and improve upper layer performance such as throughput and packet delivery delay. We have further enhanced our association protocol by using reinforcement learning to dynamically schedule the probing of neighboring access points (APs), ultimately bringing down the probing overhead by learning from past experience. When channel utilization is combined with adaptive probing, we observe a significant performance improvement compared to traditional association approaches.

Index Terms—Wireless network, machine learning, access point association

1 Introduction

The high speed, low cost, and wide availability of Wi-Fi is making it an important player in providing Internet access to places like airports, hotels, sports venues and college campuses. With an ever-increasing number of "hotspots" being Wi-Fi-enabled, a wide range of new applications and wireless usage patterns are looming the surface, ranging from mobile video and VoIP [1], to location services, social networking and pervasive applications. This explosion in demand for connectivity will require that efficient and effective methods be used to optimally manage connectivity for nearby wireless networks.

The widespread Wi-Fi availability presents many choices to a wireless user in terms of which AP it can associate to [2]. Current commercial association schemes are mostly based on received signal strength indicator (RSSI) measurements or consecutive beacons lost, and perform poorly in many situations because they overlook the load and bandwidth of the AP. For example, suppose a mobile is running an on-line video application. Even though the signal quality is good (e.g., the mobile is physically close to the AP), the AP may be heavily loaded, and cannot satisfy the bandwidth requirement of the application. According to the signal-based approach, the mobile will stay with this AP, receiving a much degraded service.

There have been alternative proposals to address these shortcomings, such as the schemes in [3], [4]. However, most of these methods are either centralized or need special features from the APs. For example, they use a designated server or the APs to collect and analyze the bandwidth utilization, and distribute the association decisions throughout the network. As a result, these methods demand significant modifications to the existing infrastructure, therefore making it difficult and unrealistic to apply them.

In this paper, we propose a distributed association scheme based on the available bandwidth. Most Internet applications have a certain level of bandwidth requirement, but the bandwidth requirement is often overlooked when making association decisions. Instead of choosing the AP with the best signal strength (as in [5]) or the AP with the lightest load (as in [6]), we argue that a wireless node should choose an AP that can provide sufficient bandwidth. If at any time, the available bandwidth of the current AP drops below the required bandwidth level, the node should probe for a better AP and switch its association. The core of this scheme is an efficient and accurate available bandwidth estimation (ABE) method. The estimation method can give accurate estimates in different scenarios, e.g., when the wireless node is transmitting packets, when the wireless node is receiving packets, and when the wireless node is probing a channel. To our knowledge, this is one of the first papers to propose a bandwidth estimation scheme that covers all stages of the association process. Our estimation method relies on existing protocol framework, and does not introduce any additional overhead. The simulation results show that the bandwidth-based association scheme can improve the average per-node throughput by 38.4 percent compared to the signal-strength based scheme, and a factor of 29.1 percent by the load-based scheme.

We also identify the two potential problems for our bandwidth-based association scheme (the problem can also occur to any association scheme). One is over-probing: a node may decide to probe other APs when no better AP is available. Over-probing can adversely affect the performance since the node cannot transmit packets during the probing delay. We propose to address this problem by enhancing our scheme with reinforced learning so that

• *The authors are with the WINLAB, Rutgers University Technology Center of New Jersey 671 Route 1 South North Brunswick, NJ 08902-3390. E-mail: {yyzhangtrappe, trappe}@winlab.rutgers.edu.*

Manuscript received 22 Apr. 2014; revised 24 Mar. 2015; accepted 22 May 2015. Date of publication 19 June 2015; date of current version 31 Mar. 2016. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TMC.2015.2442254

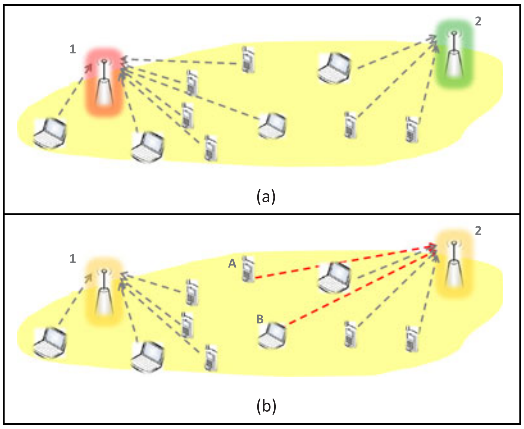


Fig. 1. An example scenario with two APs and 10 mobiles. In (a), we show an unbalanced situation caused by legacy 802.11 association protocols, and in (b), we show a more balanced association situation.

a node only probes when the expected gain is reasonable. The other problem is over-switching: a set of nodes may decide to switch to the same AP at the same time. Over-switching may lead to the *thrashing effect*, and we propose to address this problem by delayed switching or probabilistic switching. The simulations show that these enhancements can effectively alleviate the problems.

The rest of the paper is organized as follows. Section 2 discusses the background of the work, and sets up the stage for our association scheme. After presenting the bandwidth estimation method in Section 3, we explain our association scheme in Section 4. We compare the performance of three association schemes and present the simulation results in Section 5. In Section 6, we discuss our enhancement through reinforced learning to prevent the over-probing problem, and demonstrate its effectiveness through simulations. We summarize the related work in Section 7 and the concluding remarks in Section 8.

2 Overview of the Bandwidth-Based Association Scheme

2.1 Example Scenario

In Fig. 1 is the scenario picture with two APs and several mobile devices. Both APs can be reached by all the mobile nodes. According to the legacy association protocol, nodes make their association decisions only based on the received signal strength from the APs. Therefore, as indicated in Fig. 1a, 7 out of 10 nodes are associated with AP 1, causing load imbalance between the 2 APs. A better approach would be for the devices to take into consideration the load information of the APs. As shown in Fig. 1b, if nodes A and B switch to AP2, the load on the 2 APs will become more balanced. Several schemes have been proposed to achieve this purpose. For example, in [3], admission control and load-balancing algorithms are implemented in the AP to collect the state information of the network, such as available capacity in each cell, number of users per cell, etc. In [4], a network operation center (NOC) is employed to make and distribute association decisions, as well as balance load across all the APs. Both of the above methods, however, need significant modifications to the existing infrastructure by either adding extract components into the network or revising the

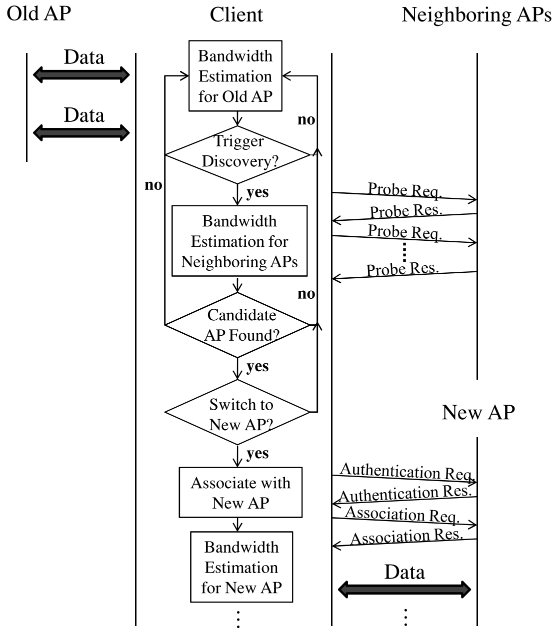


Fig. 2. The complete association protocol frame work.

communication protocol on the APs. In this paper, we design our distributed association management protocol to achieve better load balancing implicitly, by having the clients independently evaluate bandwidth estimation.

2.2 Protocol Framework

Fig. 2 shows the protocol framework of our association scheme. The design philosophy is to incorporate available bandwidth estimation in different stages of the protocol, and use such information to make association decisions. The protocol starts from when the node is actively exchanging data frames with its current AP. While exchanging data frames, the client constantly monitors the available bandwidth of the AP. If the bandwidth drops below the required level, the client will start probing neighboring APs. The probing protocol already defined in the 802.11 MAC merely uses the received signal strength from APs. In this paper, we design a bandwidth estimation scheme for the probing phase, to take into consideration available bandwidth at each AP. By the end of probing, the client will choose a candidate AP based on its available bandwidth, load, and signal strength. Then the client will decide whether to switch to that candidate AP according to its policy. If the client decides to switch, it then initiates the association process by exchanging management frames with the new AP and resumes normal data transmissions.

3 Available Bandwidth Estimation

In this section, we discuss in depth how a node estimates the available bandwidth on a channel in several situations: when the node is transmitting packets, when the node is receiving packets, and when the node is probing a channel.

3.1 Frame Exchange Sequence in 802.11 Distributed Coordinated Function (DCF)

In this paper, we assume the underlying MAC protocol is 802.11 distributed coordinated function. Fig. 3 illustrates the

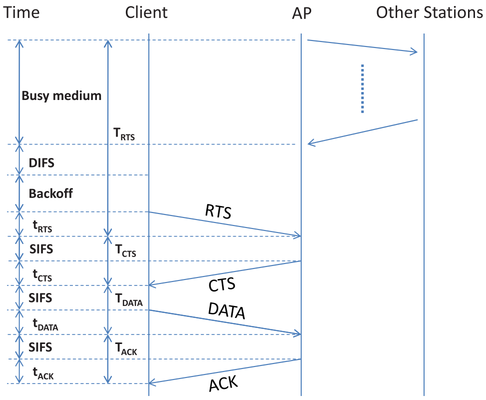


Fig. 3. Frame exchange sequence in 802.11 DCF.

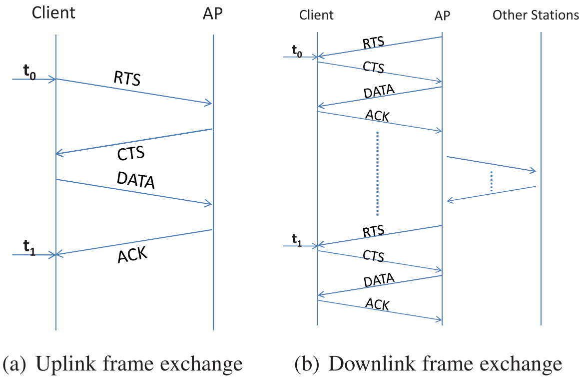


Fig. 4. Uplink and downlink frame exchange sequence.

frame exchange sequence in 802.11 DCF using an uplink example. Nodes that have data packets to transmit must first sense the channel until the channel is idle. Then after a short interval distributed inter-frame sequence (DIFS), each competing node chooses a random number from [0, CW] as its backoff timer, where CW is their individual contention window size. Once the backoff timer reaches zero, the node will transmit the request-to-send (RTS) frame first, and after an interval short inter-frame sequence (SIFS) the receiver responds with a clear-to-send (CTS) frame. Other nodes that hear either the RTS or the CTS frame will delay their transmissions until the end of the current frame exchange. Upon receiving the CTS frame, the sender waits for a duration of SIFS and sends its data frame. Finally, the receiver responds with an ACK frame after duration of SIFS. The absence of either a CTS or ACK frame causes timeout and retransmission.[1]

We use T_{RTS} to represent the RTS interval, defined as the interval between the time when the RTS frame is placed in the sending buffer and the time when the RTS frame successfully reaches the receiver. The interval consists of: the waiting time when the medium is busy (t_{busy}), a DIFS duration, the backoff duration ($t_{backoff}$), and the transmission delay of the RTS frame (t_{RTS}). Thus, T_{RTS} is calculated as:

$$T_{RTS} = t_{busy} + DIFS + t_{Backoff} + t_{RTS}.$$

Similarly, we denote the CTS interval T_{CTS}, DATA interval T_{DATA} and ACK interval T_{ACK} as:

$$\begin{aligned} T_{CTS} &= t_{CTS} + SIFS \\ T_{DATA} &= t_{DATA} + SIFS \\ T_{ACK} &= t_{ACK} + SIFS. \end{aligned}$$

The sum of these four intervals defines the duration of the message exchange sequence for a data packet.

3.2 Estimating Available Bandwidth via Uplink Traffic

We first look at bandwidth estimation for uplink traffic (the sequence shown in Fig. 4a). We define ABE_{uplink} (available bandwidth estimation for uplink traffic) as the following (S_{DATA} is the data size):

$$ABE_{uplink} = \frac{S_{DATA}}{T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK}} \tag{1}$$

$$= \frac{S_{DATA}}{t_1 - t_0}. \tag{2}$$

In Eq. (2), t_0 and t_1 are the start and end time of the frame transmission sequence (Fig. 4a). Here, we define the overhead T_{OH} as

$$T_{OH} = T_{RTS} + T_{CTS} + T_{ACK}.$$

Therefore, Eq. (1) can be further derived as:

$$\begin{aligned} ABE_{uplink} &= \frac{S_{DATA}}{T_{DATA} + T_{OH}} \\ &= \frac{1}{\frac{T_{DATA}}{S_{DATA}} + \frac{T_{OH}}{S_{DATA}}} \\ &= \frac{1}{\frac{1}{r_{DATA}} + \frac{T_{OH}}{S_{DATA}}}, \end{aligned} \tag{3}$$

where r_{DATA} is the PHY transmission rate for data frames. As shown in Eq. (3), ABE_{uplink} is related to r_{DATA}, T_{OH} and the frame size S_{DATA}. Here, we notice that the bandwidth estimation would be affected by the size of the data being transmitted, even if the network condition (contention level, etc) is the same. To eliminate the bias caused by applying different data sizes, we convert the measurement to a standard basis by using the same data size-the maximum data size (S_{DATA_MAX}). Hence, we redefine ABE_{uplink} as following:

$$\begin{aligned} ABE_{uplink} &= \frac{S_{DATA\_MAX}}{T_{OVERHEAD} + T_{DATA\_MAX}} \\ &= \frac{S_{DATA\_MAX}}{T_{OH} + \frac{S_{DATA\_MAX}}{r_{DATA}}}. \end{aligned} \tag{4}$$

By using Eq. (4), we can eliminate the inconsistency in bandwidth estimation when using different frame sizes.

3.3 Simulation Validation

Several simulations are carried out to validate the correctness of Eq. (4). The simulations are conducted in the Qualnet 4.5 simulator with one AP and two nodes, A and B. The PHY layer transmission rate is 2Mbps, and the traffic

1. In this work, we consider RTS/CTS-enabled DCF MAC in the bandwidth estimation. However, the estimation method can be easily extended to other cases.

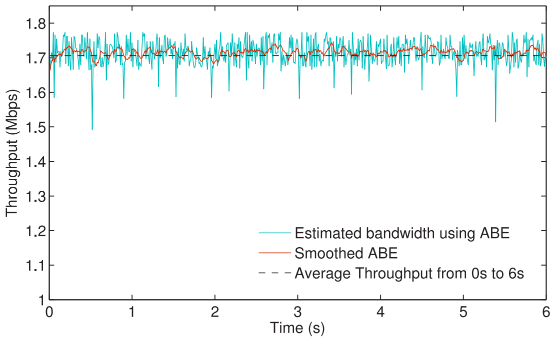


Fig. 5. ABE and smoothed ABE compared with the actual effective bandwidth when using maximum frame size.

is CBR. We first configure the simulations to achieve the maximum bandwidth. As discussed before, the bandwidth reaches its maximum with the largest allowable frame size. Since Qualnet 4.5 does not implement MAC layer fragmentation, the maximum frame size is determined by IP layer fragmentation. The IP fragment threshold is 2048 bytes in the simulator. The part of the original packet is divided into fragments, each, except the last one, being an integer multiple of 8 octets long. Thus, the maximum fragment size using IPv4 protocol is $\lfloor\frac{2048-20(IPv4header)}{8}\rfloor * 8 = 2024$. Therefore, in order to generate such a fragmentation, the data packet should be no less than 2024-8(UDP header)=2016 bytes, which gives the size of MAC layer frames 2016+8 (UDP header)+20(IP header)+28(MAC header)=2072 bytes. In our simulations, the average time spent sending one frame is 2.5 ms. Hence, we let the CBR client transmit 500 packets per second in order to saturate the channel. The simulation runs for 60 seconds, and the CBR throughput shown by Qualnet is 1.667 Mbps. Taking into account of the overhead from adding UDP and IP headers, we get the MAC layer throughput of $\frac{1.667}{2016} * 2072 = .1.706Mbps$ (saturation throughput).

Next, we estimate ABE_{uplink} on a per-frame basis using Eq. (4). When sending frames to the AP, node A records the time when the frame is placed in the sending buffer (t_0) and the time when the ACK is received (t_1). The bandwidth can then be estimated as

$$ABE_{uplink} = \frac{S_{DATA\_MAX}}{t_1 - t_0}. \tag{5}$$

The results calculated using Eq. (5) are presented in Fig. 5, showing the ABE_{uplink} values during the 6 second simulation period. The horizontal dash line represents the simulated bandwidth 1.706 Mbps, and the blue line plots the ABE_{uplink} values calculated on a per-frame basis using Eq. (5). Though the estimated values are close to the average throughput, there are a lot of noises and jitters. Since we will be making critical decisions based on the ABE values, these noises may severely undermine the reliability of the protocol. In order to reduce the noises, we average the ABE values over k samples. Fig. 5 shows the resulting estimation values (represented by the redline) using $k = 10$ can greatly reduce jitters compared to original values.

Next, under the same setting, we run another set of simulations with varying packet sizes S_{DATA}. Since the ABE calculation uses the maximum frame size, we use the following equation to normalize the estimation when using varying packet sizes:

$$\begin{aligned} ABE_{uplink} &= \frac{S_{DATA\_MAX}}{T_{OH} + T_{DATA\_MAX}} \\ &= \frac{S_{DATA\_MAX}}{(t_1 - t_0 - \frac{S_{DATA}}{r_{DATA}}) + \frac{S_{DATA\_MAX}}{r_{DATA}}}. \end{aligned} \tag{6}$$

3.4 Estimating Available Bandwidth via Downlink Traffic

The downlink traffic actually is the traffic pattern most often seen in WLAN, given that more than 90 percent of the WLAN traffic is from AP to client. The downlink bandwidth estimation is different from the uplink case. Since it is the AP who initiates transmissions, the client would not know when the AP places the RTS into the sending buffer(t_0). To address this issue, we mark the time when RTS is received as t_0, and the time when the following RTS is received by the client as t_1, as illustrated in Fig. 4b. Therefore, $ABE_{downlink}$ can be written in the same form as ABE_{uplink} in Eq. (6), and we use ABE_{DATA} to unify the above two estimations in the following equation:

$$ABE_{DATA} = \frac{S_{DATA\_MAX}}{(t_1 - t_0 - \frac{S_{DATA}}{r_{DATA}}) + \frac{S_{DATA\_MAX}}{r_{DATA}}}. \tag{7}$$

Note that in the above equation, we use consecutive downlink data packets, which are packets dedicated for the same receiver, and are placed back-to-back in the sender's queue. Given the dominate volume of downlink traffic in the overall WLAN traffic, and the relatively large size of the contents to be downloaded and the relatively small packet size limit in 802.11, the probability of consecutive downlink packet delivery is very high, which makes it a viable approach to use consecutive packets for downlink bandwidth estimation.

Technically, this can be achieved by only looking at frames that are fragmented from the same packet, usually with "1" as the MF (more fragments) bit in the IP header. An improved approach is to have the client examine the size of the current frame. If the frame size is S_{DATA_MAX}, very likely the packet has been fragmented, and the client can use successive fragments to estimate bandwidth until it receives a fragment smaller than S_{DATA_MAX}. This may avoid the delay of passing the frame to the IP layer to check its MF bit.

3.5 Estimating Available Bandwidth while Probing

A node not only needs to estimate the available bandwidth for its current association, as discussed in the previous two sections, but also needs to do so for other APs. This is crucial for a node looking to improve its association quality, so that it can not only assess the current association, but also search for potential association that offers better quality. Specifically, when the current association cannot satisfy its bandwidth requirement, a node needs to examine the available bandwidth of other APs and may change its association to an AP that offers more bandwidth.

Fig. 6 shows the frame exchange sequence during probing. First, after tuning to a specific channel, a client will send a broadcast $Probe\_Request$ frame. The AP that receives

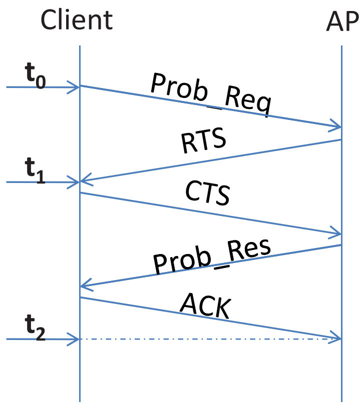


Fig. 6. Frame exchange sequence during probing.

the frame will go through a complete frame exchange sequence: RTS $\rightarrow$ CTS $\rightarrow$ *Probe_Response* $\rightarrow$ ACK. Similar to the previous bandwidth estimation methods, clients need to know the start time and the end time of the sequence to calculate the ABE. Here, the client can mark the time when ACK is transmitted (t_2) as the end time. The start time of the sequence is not available to the client, but we can record the time when the Probe_Requst frame is placed in the client's sending buffer as t_0 and the time when the client receives RTS as t_1. Note that the transmission of a Prob_Reqest frame is the same as RTS in that they both need to contend for channel, so we assume the interval $t_1 - t_0$ the same as the period for transmitting two RTS frames, thus T_{RTS} as $\frac{t_1-t_0}{2}$. Using $t_2 - t_1$ to replace $T_{CTS} + T_{DATA} + T_{ACK}$ in Eq. (1), we can calculate $ABE_{Probing}$ as:

$$ABE_{Probing} = \frac{S_{Probe\_Response}}{\frac{t_1-t_0}{2} + (t_2 - t_1)} = \frac{S_{Probe\_Response}}{t_2 - \frac{t_1}{2} - \frac{t_0}{2}}. \quad (8)$$

After normalization using the maximum frame size S_{DATA_MAX}, and using t' to substitute $t_2 - \frac{t_1}{2} - \frac{t_0}{2}$, Eq. (8) becomes:

$$ABE_{Probing} = \frac{S_{DATA\_MAX}}{t' - \frac{S_{Probe\_Response}}{r_{DATA}} + \frac{S_{DATA\_MAX}}{r_{DATA}}}. \quad (9)$$

To validation this equation, we use the same simulation setup as before, and apply Eq. (9) to the probing phase and got the average $ABE_{Probing}$ of 1.69 Mbps. We note that this value is in agreement with the ABE value in the transmission phase (1.706 Mbps). We further tested the scenarios where there are more than 1 incumbent APs in each channel, and the bandwidth estimation for the APs through probing are in agreement with the estimation done through data transmission by existing associated nodes. The ability to do bandwidth estimation in the probing phase makes the ABE approach superior than other bandwidth estimation approaches such as EVA [6], in that it not only can assess the current association, but also search for potential association that offers better quality.

4 Access Point Association Policy Using Channel Utilization (CU)

In this section, we present our access point association policy using channel utilization. Specifically, we discuss the policies of when to start probing other APs, which AP to choose as the next candidate, and when to switch association.

4.1 Selecting a New AP Candidate

We take the viewpoint that a wireless node determines which AP to associate based on its own bandwidth requirement and the available bandwidth on the AP. A node's bandwidth requirement can change with time. For example, a client may need 200 Kbps from 2:00 pm to 3:00 pm for VOIP calls, and 1 Mbps from 3:00 to 4:00 pm for on-line video.

Based on this, we define channel utilization as the following:

$$CU = \frac{BandwidthRequirement}{Availablebandwidth}. \quad (10)$$

Here is an example to illustrate the use of CU. Say during 2:00 and 4:00 pm, the available bandwidth for the client's current association is 800 Kbps. Then in the VOIP session, the CU is 200 Kbps/800 Kbps $= 25\%$, but during the video session, the CU becomes 1 Mbps/800 Kbps $= 125\%$, indicating the current association will not meet the client's bandwidth requirement, and a new association is preferred.

We set the following threshold values: $CU_{probing}$ and $RSSI_{threshold}$. Probing will be triggered if **either** of the following criteria is met: (a)

1) $CU \geq CU_{probing}$
2) $rssi \leq RSSI_{threshold}$.

The first criteria is mainly used to guarantee the client bandwidth requirement, while the second criteria is inherited from the legacy 802.11 MAC protocol to guarantee the signal quality.

After probing all the channels and collecting statistics from each AP (e.g., signal strength, available bandwidth), the candidate AP is chosen based on **either** of the following conditions. (a)

1) $rssi \leq R$, and $rssi' - rssi \geq \delta_R$;
2) $CU \geq CU_{probing}$, and $CU' \leq CU_{probing}$, and $rssi' > R$,

where $rssi$ and CU are statistics of the current channel, while $rssi'$ and CU' are statistics for the candidate channel. R, δ_R, and $CU_{probing}$ are threshold values.

Here, we note that the choice for the threshold values $CU_{probing}$ is at the discretion of the users. In our work, we consider it to be a minimum bandwidth requirement to guarantee the quality of service. Generally speaking, reasonable lower $CU_{probing}$ usually leads to better communication quality, but runs the risk of under-utilizing the network resource and may lead to more frequent switching and overhead.

4.2 AP Switching Policy

To better understand the proposed switching policy, let us consider an example where there are 2 APs A and B, with several clients associated with A. Suppose a new client with a heavy load joins A's network, which will seriously degrade the performance of the existing clients. If the clients all start to look for a better candidate, and switch to B at the same time (though independently), they will soon saturate B, and need to switch again. The worst case would be for

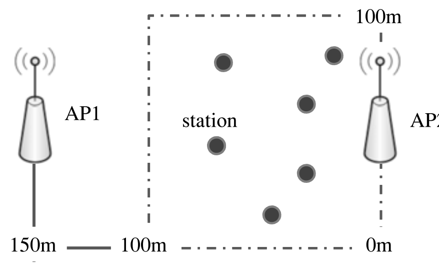


Fig. 7. The topology of simulation.

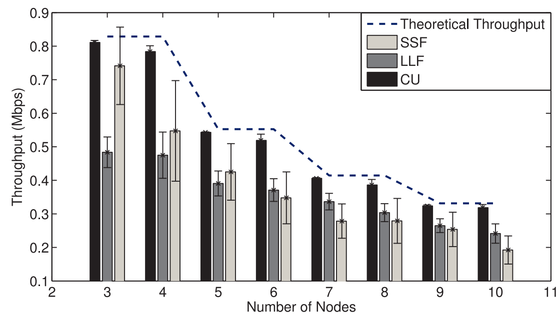


Fig. 8. Average per-station throughput.

the clients to bounce back and forth between A and B, leading to a phenomenon known as "thrashing". In fact, we observe several trashing situations in our simulations. In some extreme cases, multiple clients switched between the same source and destination APs within less than 1 second. Some other clients only stayed briefly with the current AP before switching to another one.

To address the above issues, we proposed the following three switching strategies:

- *Immediate switch.* This is the baseline scheme in which once a candidate AP is chosen, a node immediately switches its association.
- *Delayed switch*. In this scheme, after selecting the candidate AP, the client postpones the switching by T_{delay}, which is a random value from [0, $Delay_{max}$]. During T_{delay}, normal transmission continues and the node keeps monitoring the bandwidth of the current AP. After T_{delay}, the client switches if the CU level remains above the threshold.
- *Probabilistic switch*. In probabilistic switch, the node switches to the candidate AP with a probability p.

5 Performance Evaluation

In this section, we present the comparative performance of different AP association and switching policies.

We conduct simulation-based studies using the Qualnet 4.5 simulator, with 802.11b as the PHY module at a 2 Mbps transmission rate. The transmission power is 20 dBm power. The considered topology is shown in Fig. 7, which consists of two APs and multiple wireless stations. The two APs are separated by 150 m from each other, and the stations are randomly placed in an area of $100 \times 100\ m^2$, closer to AP2. In this topology, we expect more stations to associate with AP2 in the legacy 802.11 protocol or any signal strength based association protocol. By using our CU metric, despite how the nodes are located, the load on the two APs will be more balanced. Different channel frequencies are assigned to each AP to avoid inter-channel interference. The carrier sensing range is set to 200 m so that both APs can be detected by all the clients. The combined load of the stations is set to the total capacity of the 2 APs, which is approximately 3.4 Mbps as discussed in Section 3.

We first compare our CU-based protocol with two baseline protocols, signal strength first (SSF) and least load first (LLF). In SSF, a node chooses the AP with the best signal-noise ratio (SNR). Whenever the SNR of the current association is below a threshold (e.g., -83 dBm), the node will switch to a AP with an SNR greater than the current SNR by at least a small margin (e.g., 2 dBm). LLF is a load-based protocol in which a node probes the neighboring APs periodically and associates with the AP with the least load ([6]). We note that our CU protocol differs from LLF in that CU only switches to a new AP when the current association cannot satisfy the node's bandwidth requirement while LLF always looks for a least-loaded AP no matter whether the current association is sufficient. In LLF, the node probes every 0.1 second as used in [6].

In our results, each data point is the average over 10 simulation runs with random node placements.

5.1 Throughput

Fig. 8 shows per-station throughput for the three association schemes when the number of wireless stations increases from 3 to 10. We observe that the proposed CU protocol shows a significant improvement over the other two protocols. Compared with SSF, CU achieves a 38.4 percent gain on average, and a 65.7 percent improvement in certain scenarios (when we have 10 nodes). More importantly, the CU throughput is very close to the best throughput in theory shown in the dotted line.

We also show the standard deviation of the throughput measurements in Fig. 8. Since each station has the same load, the standard deviation in throughput can reflect the extent to which these stations are treated equally, further indicating whether the load on each AP is balanced. The legacy SSF protocol shows the largest variance for all cases. LLF shows an improved variance on throughput because it takes the load on each AP into consideration, but our CU-based protocol displays the smallest standard deviation for all cases. This suggests that the CU protocol can achieve the best load balance among the three protocols.

In addition to per-station throughput, we also present the overall system-wide throughput in Fig. 9. The CU protocol achieves the highest system throughput. As the system load goes up, the throughput first increases and then remains at 3.2 Mbps (the system capacity is 3.4 Mbps). The maximum achievable throughput for SSF and LLF is much lower, 2.3 Mbps and 2.5 Mbps respectively.

5.2 Delay and Jitter

Another important metric for protocols supporting applications with QoS requirements is packet arrival delay and jitter. We compare the end-to-end delay and jitter (defined as the variance of the delay) for the three protocols in Fig. 10. Among the three protocols, the CU protocol is the best. It generates very low end-to-end delay (0.044 s in average) and jitter (0.015s in average). SSF fares the worst, with its delay varying dramatically from 0.22 to 1.92 sec, with an

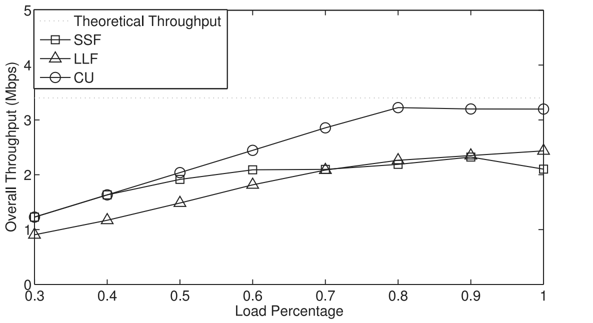


Fig. 9. System throughput when the number of stations is 10.

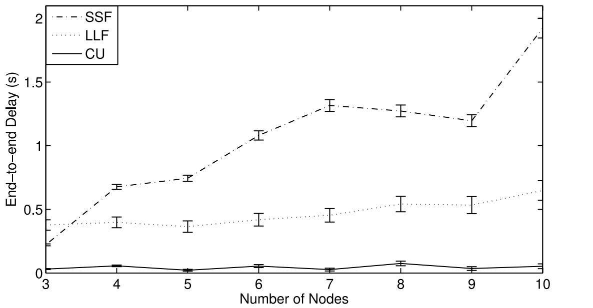


Fig. 10. Delay and Jitter among contending stations.

average jitter of 0.038 sec. LLF has an average delay of 0.47 sec, and average jitter of 0.05 sec.

5.3 AP Switching Policy

Any association protocol can potentially lead to an excessive number of reassociations, dropped packets, or even thrashing. The CU protocol has this potential problem as well, and we can alleviate this problem by controlling when or whether to switch, such as in Delay Switch and Probabilistic Switch discussed in Section 4.2. Here we look at their effectiveness in reducing the number of reassociations and dropped packets in Figs. 11 and 12. we can see these two switching policies are quite effective. For example, Probabilistic Switch with $p = 0.2$ can reduce the average number of reassociations by 91 percent, and can reduce the number of dropped packets by 90 percent.

5.4 Incorporating Adaptive Rates

We note that in reality auto rate fallback (ARF) [7] is widely used in 802.11 WLAN devices, in which the nodes will adaptively adjust the physical transmission rate. With ARF, a station increases its physical transmission rate when a certain number of transmission success occur consecutively. In other case, the station decreases its rate upon certain number of transmission failures. In evaluating the effect of ARF, We use the same simulation setting as the previous experiments, except the transmission rate is set to 11 Mbps. Here we compare the overall throughput of the CU scheme with and without ARF, and the results are presented in Fig. 13. We notice that the ARF algorithm severely degrades the performance of the network when the contention increases. This is because when having transmission failures, ARF does not distinguish failures caused by channel errors from those caused by frame collisions, therefore suffers from the increasing amount of traffic.

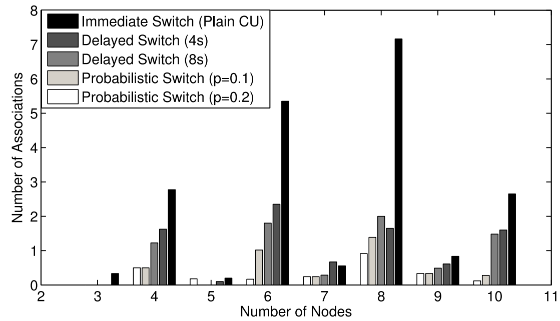


Fig. 11. Average number of per-node reassociations with CU and its APS variants.

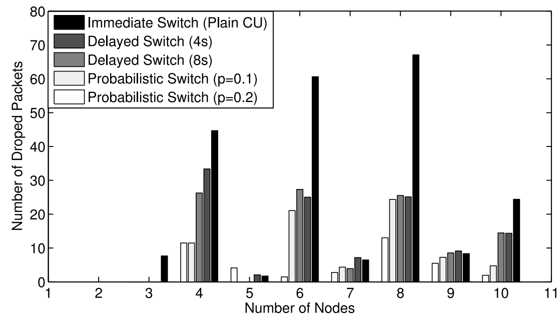


Fig. 12. Average per-node packets drop with CU and its APS variants.

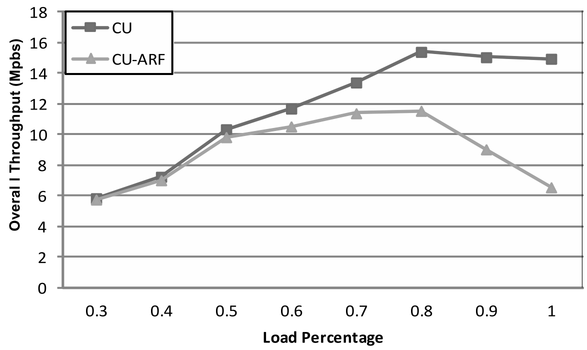


Fig. 13. System throughput for CU and CU-ARF.

6 Enhancing the CU Protocol through Reinforcement Learning

The main advantage of CU over SSF is that the former protocol can make a node start probing for a better AP much earlier. The implicit assumption is that a node can find a better AP by probing. This assumption, however, does not always hold. If no better AP is available, then probing would not solve the problem but further hurt the performance. This adverse effect will be more pronounced when all the APs in the system are heavily loaded.

To address this issue, we enhance the CU protocol by adopting reinforcement learning in determining whether a node needs to probe for a better AP when the available bandwidth of the current AP is not sufficient for the node.

6.1 Reinforcement Learning Model

Formally, the basic reinforcement learning model, as applied to Markov decision process (MDPs), consists of: a set of environment states S, a set of actions A, and a set of scalar "rewards" R.

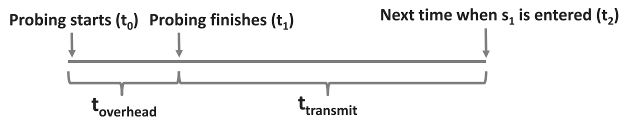


Fig. 14. Illustration of transmission gain and association overhead when probing occurs.

We build the following reinforcement model:

1) $S = (s_1, s_2)$. s_1 represents the state in which the available bandwidth of the current AP falls below the required level, while s_2 represents the opposite state.
2) $A = (a_1, a_2)$, where action a_1 is to probe neighboring nodes and action a_2 is not to probe.

Next we define the reward functions $r(s_i, a_i)$. The reward function for state s_2 is straightforward:

$$r(s_2, a_1) = 0$$
$$r(s_2, a_2) = 1.$$

We use this reward definition to refrain probing when the available bandwidth of the current association can meet the bandwidth requirement.

Next let us look at the reward functions at state s_1. The reward for each action is determined by the data transmission gain since last time when the same action was taken. In this way, we are predicting the reward of the near future using the reward of the near past. This is illustrated in Fig. 14. Suppose that at time t_2, we are in state s_1, and need to decide whether to probe other APs. We have recorded the time when last probing started (t_0) as well as the time when it finished (t_1). Thus we can derive the probing overhead $t_{overhead} = t_1 - t_0$, and the transmission time since last probing $t_{transmit} = t_2 - t_1$. We have the following reward function definitions:

$$r(s_1, a_1) = t_{transmit} \times ABE$$
$$r(s_1, a_2) = (t_{transmit} + t_{overhead}) \times ABE'.$$

Here, ABE is the estimated bandwidth since last probing, and ABE' is the estimated bandwidth before last probing. Therefore, $r(s_1, a_1)$ is the transmission gain due to last probing, and $r(s_1, a_2)$ is the transmission gain that could have been achieved if we chose not to probe at time t_0.

We use function Q to calculates the quality of a state-action combination:

$$Q : S \times A \rightarrow \mathbb{R}. \qquad (11)$$

Before learning starts, Q returns a fixed initial value. Then, every time when the agent is given a reward (due to a state change), new values are calculated for every state/action pair. The core of the algorithm is a value iteration update. It assumes the old value and makes corrections based on the updated information. Q is updated based on the following equation:

$$Q(s_i, a_j) = (1 - \alpha)Q(s_i, a_j) + \alpha r(s_i, a_j), \qquad (12)$$

where $\alpha (0 < \alpha \leq 1)$ is the learning rate. It determines to what extent the newly acquired information will override the old information.

Next, we give the ϵ-greedy learning policy for our Reinforcement CU protocol:

Policy. At each state s_i, the agent picks the action $a_j = \text{argmax}_a Q(s_i, a)$ with probability 1-ϵ, and picks a random action with probability ϵ.

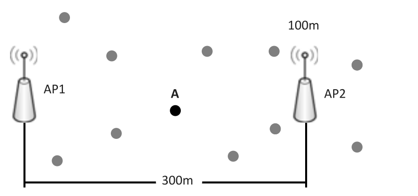
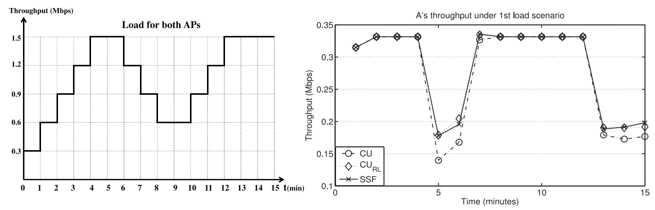


Fig. 15. Simulation topology of 2 APs with variable load.



(a) AP load pattern 1 (b) A's throughput

Fig. 16. Throughput under load scenario 1.

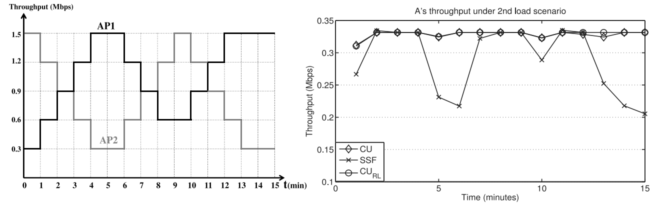
6.2 Performance Evaluation

6.2.1 *Static Scenarios*

Next we compare the performance of the enhanced CU algorithm (referred to as RL-CU) with CU and SSF. Fig. 15 shows the simulation topology with two APs and a few wireless nodes. All nodes except A are used to create background traffic on the two APs and their associations are fixed. A can change its association according to different association protocols, and we report the performance observed on node A.

In the first experiment, the two APs have the same background load pattern, as shown in Fig. 16a. It is intuitive that there won't be much gain to switch AP associations in this case. From Fig. 16b, we observe that around the time minute 5 and 15, the throughput drops due to the increase of background traffic. When the network is congested, CU suffers from large probing cost, and it performs worse than SSF. However, RL-CU can avoid this problem and delivers the best performance among the three.

Fig. 17a shows another scenario in which the load on one AP is always noticeably heavier than the other. In such scenarios, CU has advantage over SSF since a node can choose a suitable AP to associate. In Fig. 17b, we can see RL-CU can perform just as well as CU, while the performance for SSF is much worse than the other two. Therefore, to summarize, by using RL-CU to adjust probing schedules, we can achieve the best performance of both worlds.



(a) AP load pattern 2 (b) A's throughput

Fig. 17. Throughput under load scenario 2.

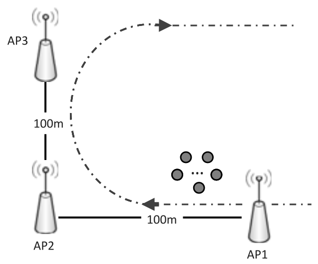


Fig. 18. Simulation topology for mobile nodes with bursty traffic.

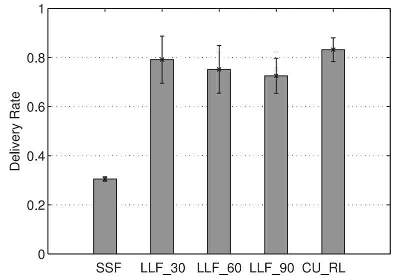


Fig. 20. Delivery rate for mobile continuous traffic scenario.

6.2.2 Mobile Scenarios

Next, we use another set of experiments to demonstrate the performance in mobile scenarios. In these experiments, a group of 15 nodes will in turn visit the three access points. The simulation topology pattern is shown in Fig. 18. Again, we tested two traffic patterns, which are both shown in Fig. 19. The first traffic pattern represents continuous traffic, with each node continually contributes an individual traffic of 0.33 Mbps. The second pattern represents bursty traffic. The traffic generated by each node has a mean inter-arrival rate of 30 seconds and each traffic burst lasts for 10 seconds. For both scenarios, there are existing background traffic for the three APs (0.1, 0.1 and 0.2 Mbps for A, B, C respectively). For each scenario, we tested the performance of three protocols, SSF, LLF, and CU_ML. For LLF, we vary the probing interval to three different values: 30, 60, 90 s.

The result for continuous traffic is shown in Fig. 20. CU_ML's packet delivery rate reaches 83.18 percent. The performance for LLF varies with the probing intervals. The delivery rate decreases from 79.13 to 72.55 percent when the probing interval increases from 30 to 60s, indicating that the performance suffers when the probing is not frequent enough to discover better gateways to associate. The performance for SSF is the worst of all three, with only 30 percent delivery rate. This is because all nodes associate with the gateway that they approach first, which is AP1, and ignore other gateways that appear later and may become better choices for association.

The result for bursty traffic is shown in Fig. 21. CU_ML's packet delivery rate reaches 87.45 percent, which is significantly higher than the SSF(37.84 percent) and LLF(from 37.75 to 46.06 percent). The reason for SSF's suboptimal performance is that the group of nodes tend to choose the same AP as their common strongest signal strength AP, thus it's always the one and only AP that get flooded. LLF's performance is caused by the bursty nature of the traffic. Since the duration of the traffic is short, the periodic probing in LLF is less able to capture the burst of traffic. In contrast, CU_RL is able to constantly adjust its probing on-demand, therefore delivers the best results.

7 Related Work

7.1 Bandwidth Estimation

Various bandwidth estimation schemes have been proposed for 802.11 networks. The bandwidth estimation method EVA in [6] is similar to our ABE concept. However, EVA can only be used for evaluating data traffic. By contrast, the ABE scheme can easily incorporate probing frames into uplink/downlink data traffic. Unlike the EVA protocol which always favors APs with the best bandwidth, we compare the estimated bandwidth with the individual client's bandwidth requirement, which further reduces the reassociation overhead. In [8], the authors propose to evaluate the potential AP bandwidth by passively measuring beacon timings from a particular AP. We note that using beacons can only provide downlink bandwidth estimates, which usually gives inaccurate evaluation of the overall channel condition.

7.2 Bandwidth Utilization

Another class of related work focus on improving the bandwidth utilization by reducing the channel scanning delay, either by reducing scanned channels, or reducing the time consumed in scanning each channel (active scanning). In [9], the paper adopts proactive association to avoid losing connectivity. The scanning phase is shortened in the association process by only scanning the APs on the same or overlapping channels, and in-band scanning and switching is given higher priority in order to reduce channel switching time. In [5], the authors propose to decouple scanning from actual handoff by interleaving scanning into data transmission, during which a sleep request is sent to the AP to buffer packets before the scan finishes. In [10], vehicular mobility is considered in MESH networks with a scalable multi-tier

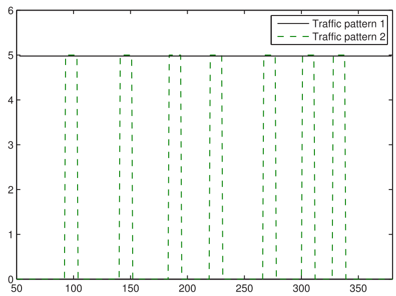


Fig. 19. Traffic pattern for mobile scenarios.

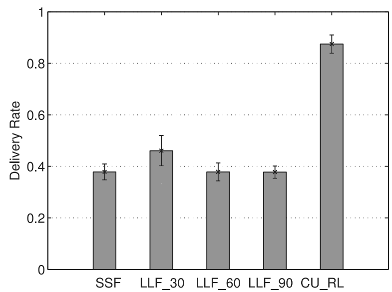


Fig. 21. Delivery rate for mobile bursty traffic scenario.

architecture. A set of policies that employ smoothed AP-client signal quality coupled with per-AP quality scores are designed, which characterize the inherent inability of the mesh architecture to provide uniform bandwidth to all spatial locations. In [11], the authors proposed HJam, which is based on OFDM-based, and aims to fully explore the physical layer features of OFDM. It combined data packets and a number of control messages to be transmitted together. In [12], the author implemented an AP association strategy that maximizes the minimum throughput among all users at the cost of an acceptable overhead.

The above mentioned works mainly use RSSI-based metrics and did not address the load-balancing problems associated with it.

7.3 Load Balancing

The problem of load balancing has been extensively studied. In [3], explicit channel switching and network-directed roaming are used to provide hot-spot congestion relief while maintaining pre-negotiated user bandwidth agreements with the network. In [13], two access point selection algorithms are proposed to maximize the average throughput and minimum throughput of stations. This work as well as many others require significant modification to the current communication protocol. Another group of works applies a centralized approach to collect and distribute statistics from the network and distribute to AP and clients, such as [4], [14], [15], [16]. In [4], a NOC is needed to make association decisions, as well as balancing load of all the AP. In [14], proportional fair (or time-based fair scheduling) provides a balanced tradeoff between fairness and network throughput. The function is implemented in a central management server, and the approximation algorithms can be used for periodic offline optimization. The similar approaches are also applied in cellular network studies, where to achieve fully reuse of frequency and alleviate severe intercell interference. [17], [18], [19], [20] The basic idea is to schedule users across cells so that they do not severely interfere with each other. Earlier work on load balancing also mostly assumed a centralized controller so that users are scheduled across cells.

Unlike the above mentioned works, the ABE protocol we proposed allows the clients to make distributed decisions and requires no modifications to the current infrastructure. In [21], the authors proposed a sociality-aware access point selection scheme targeting the Enterprise environment, where there is strong social relationships of users.

8 Conclusion and Future Work

As wireless LAN hotspots become more prevalent and experience many more users, an efficient access point association protocol is in a great demand. Most of existing association protocols rely on either the received signal strength of the access point, or the load of the access point, without considering the bandwidth requirement of the user nor the available bandwidth at the access point. Those schemes that do consider the bandwidth factors, however, are mostly centralized schemes. In this paper, we set out to fill this void by designing a distributed access point association protocol based upon the bandwidth situation.

Our association protocol is centered around light-weight and accurate estimation of bandwidth and channel utilization. We also adopt techniques that can avoid unnecessary reassociations among multiple access points. Further, we employ reinforcement learning techniques to determine whether we need to probe other access points when the current one seems to have insufficient bandwidth. Considering these factors, we compare our scheme with existing ones through extensive simulation studies. Our simulation results show that the proposed CU scheme can outperform existing schemes in many situations. In stationary scenarios, the basic CU association scheme can improve the average per-node throughput by 38.4 percent compared to the signal-strength based scheme, and a factor of 29.1 percent by the load-based scheme. In bursty mobile traffic scenarios, the reinforced CU scheme achieves a packet delivery rate of 87.45 percent, which is significantly higher than the signal strength based scheme (37.84 percent) and the load based scheme (from 37.75 to 46.06 percent).

References

[1] A. J. Nicholson, Y. Chawathe, M. Y. Chen, B. D. Noble, and D. Wetherall, "Improved access point selection," in *Proc. 4th Int. Conf. Mobile Syst., Appl. Services*, 2006, pp. 233–245.

[2] A. Akella, G. Judd, S. Seshan, and P. Steenkiste, "Self-management in chaotic wireless deployments," in *Proc. 11th Annu. Int. Conf. Mobile Comput. Netw.*, 2005, pp. 185–199.

[3] A. Balachandran, P. Bahl, and G. Voelker, "Hot-spot congestion relief in public-area wireless networks," in *Proc. 4th Workshop Mobile Comput. Syst. Appl.*, 2002, pp. 70–80.

[4] Y. Bejerano, S.-J. Han, and L. E. Li, "Fairness and load balancing in wireless LANs using association control," in *Proc. 10th Annu. Int. Conf. Mobile Comput. Netw.*, 2004, pp. 315–329.

[5] H. Wu, K. Tan, Y. Zhang, and Q. Zhang, "Proactive scan: Fast handoff with smart triggers for 802.11 wireless LAN," in *Proc. 26th IEEE Int. Conf. Comput. Commun.*, May 2007, pp. 749–757.

[6] H. Lee, S. Kim, O. Lee, S. Choi, and S.-J. Lee, "Available bandwidth-based in IEEE 802.11 wireless LANs," in *Proc. Int. Symp. Model., Anal. Simul. Wireless Mobile Syst.*, 2008, pp. 132–139.

[7] Kamerman, Ad and Monteban, Leo, "WaveLAN-II: A high-performance wireless LAN for the unlicensed band," *Bell Labs Tech. J.*, vol. 2, pp. 118–133, 1997.

[8] S. Vasudevan, K. Papagiannaki, C. Diot, J. Kurose, and D. Towsley, "Facilitating access point selection in IEEE 802.11 wireless networks," in *Proc. 5th ACM SIGCOMM Conf. Internet Meas.*, 2005, pp. 26–26.

[9] V. Mhatre and K. Papagiannaki, "Using smart triggers for improved user performance in 802.11 wireless networks," in *Proc. 4th Int. Conf. Mobile Syst., Appl., Services*, 2006, pp. 246–259.

[10] A. Giannoulis, M. Fiore, and E. W. Knightly, "Supporting vehicular mobility in urban multi-hop wireless networks," in *Proc. 6th Int. Conf. Mobile Syst., Appl., Services*, 2008, pp. 54–66.

[11] K. Wu, H. Li, L. Wang, Y. Yi, Y. Liu, Q. Zhang, and L. Ni, "Hjam: Attachment transmission in WLANs," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 1449–1457.

[12] F. Xu, C. Tan, Q. Li, G. Yan, and J. Wu, "Designing a practical access point association protocol," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[13] A. Fujiwara, Y. Sagara, and M. Nakamura, "Access point selection algorithms for maximizing throughputs in wireless LAN environment," in *Proc. 13th Int. Conf. Parallel Distrib. Syst.*, 2007, pp. 1–8.

[14] L. E. Li, M. Pal, and Y. R. Yang, "Proportional fairness in multi-rate wireless LANs," in *Proc. 23rd Annu. Joint Conf. IEEE Comput. Commun. Soc.*, Apr. 2004, pp. 1004–1012.

[15] M. Abusubaih, J. Gross, S. Wiethoelter, and A. Wolisz, "On access point selection in IEEE 802.11 wireless local area networks," in *Proc. 31st IEEE Conf. Local Comput. Netw.*, 2006, pp. 879–886.

[16] K. Sundaresan and K. Papagiannaki, "The need for cross-layer information in access point selection algorithms," in *Proc. 6th ACM SIGCOMM Conf. Internet Meas.*, 2006, pp. 257–262.

[17] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2706–2716, Jun. 2013.
[18] K. Son, S. Chong, and G. Veciana, "Dynamic association for load balancing and interference avoidance in multi-cell networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 7, pp. 3566–3576, Jul. 2009.
[19] B. Rengarajan and G. de Veciana, "Architecture and abstractions for environment and traffic-aware system-level coordination of wireless networks," *IEEE/ACM Trans. Netw.*, vol. 19, no. 3, pp. 721–734, Jun. 2011.
[20] T. Bu, L. Li, and R. Ramjee, "Generalized proportional fair scheduling in third generation wireless data networks," in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 1–12.
[21] G. Xue, Q. He, H. Zhu, T. He, and Y. Liu, "Sociality-aware access point selection in enterprise wireless lans," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 10, pp. 2069–2078, Oct. 2013.

Tingting Sun received the PhD degree in computer engineering from Rutgers University in 2014. Her advisor was professor Yanyong Zhang, and she also worked with professors Wade Trappe and Marco Gruteser on various projects. Her research interests include location-aware systems, mobile networking, and resource management in WLAN and mobile ad hoc networks.

Yanyong Zhang is currently a professor in the Electrical and Computer Engineering Department, Rutgers University. She is also a member of the Wireless Information Networks Laboratory (Winlab). She has 18 years of research experience in the areas of sensor networks, mobile computing, and high-performance computing, and has published more than 90 technical papers in these fields. She is the associate editor for *IEEE Transactions on Mobile Computing* (TMC) and *IEEE Transactions on Service Computing* (TSC). Her research is mainly funded by the National Science Foundation (NSF), including an NSF Career Award.

Wade Trappe is a professor in the Electrical and Computer Engineering Department, Rutgers University, and an associate director of the Wireless Information Network Laboratory (WINLAB), where he directs WINLABs research in wireless security. He has led several federally funded projects in the area of cybersecurity and communication systems, projects involving security and privacy for sensor networks, physical layer security for wireless systems, a security framework for cognitive radios, the development of wireless testbed resources (the ORBIT testbed, www.orbit-lab.org), and new RFID technologies. He served as an editor for the *IEEE Transactions on Information Forensics and Security* (*TIFS*), *IEEE Signal Processing Magazine* (*SPM*), and *IEEE Transactions on Mobile Computing* (*TMC*). He served as the lead guest editor for September 2011 special issue of the *Transactions on Information Forensics and Security* on Using the Physical Layer for Securing the Next Generation of Communication Systems and served as the IEEE Signal Processing Society representative to the governing board of *TMC*. He is a fellow of the IEEE and is currently IEEE SPS Regional Director for Region 1-6.