Utilizing Multiple Channels With Fewer Radios in Wireless Mesh Networks

Chia-Yu Ku, *Student Member, IEEE*, Ying-Dar Lin, Shiao-Li (Charles) Tsao, *Member, IEEE*, and Yuan-Cheng Lai

***Abstract*—Wireless mesh networking (WMN) is regarded as a low-cost technology for rapid wireless network deployment. However, in a single-channel WMN, the overlapped transmission ranges between relaying mesh points could introduce serious interference. Using multiple radios over multiple channels can decrease the interference and improve the capacity of a WMN, but it increases the cost of a mesh point. One possible solution to balance between interference, hence performance, and cost is utilizing fewer radios that switch among multiple channels instead of using per-channel radios. In this paper, we propose a channel-switching method, called the traffic-aware switching scheme (TRASS), for a mesh point with a limited number of radios. TRASS utilizes the existing IEEE 802.11 mechanisms, i.e., hybrid-coordination-function-controlled channel access and power saving, to avoid packet loss during channel switching. A TRASS mesh point monitors the occupied channel time to schedule radios that switch among channels. The implemented TRASS demonstrates 75% throughput improvement by (2, 1), i.e., two-channel single-radio, over (1, 1). (3, 2) and (3, 1) achieve 69.8% and 39.7%, respectively, of the throughput of (3, 3) in the simulated TRASS.**

***Index Terms*—Channel switching, IEEE 802.11s, multichannel, multiradio, wireless local area networks (WLANs), wireless mesh network (WMN).**

I. INTRODUCTION

INFRASTRUCTURE-BASED wireless local area networks (WLANs) extend their coverage by deploying a high density of access points (APs). With IEEE 802.11 [1], APs interconnect with each other through cables and switches. Stations (STAs) access the Internet through an AP. However, deploying a large number of APs that are interconnected through a *wired* infrastructure requires high cost and loses flexibility in deployment. Another low-cost solution is wireless mesh networking (WMN), where part of the wired infrastructure is replaced by *wireless* relay between APs.

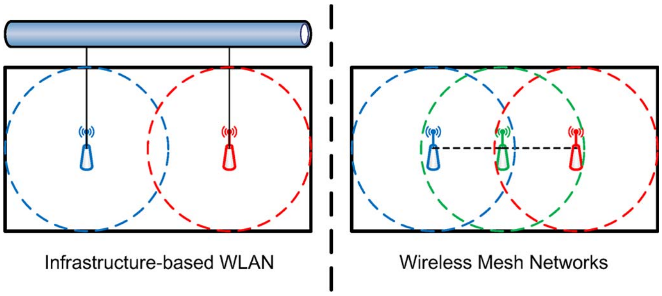


Fig. 1. Overlapped areas in infrastructure-based WLAN and WMN.

IEEE 802.11s [2] is an extension of IEEE 802.11 for mesh networking. It specifies a framework using the IEEE 802.11 Media Access Control/physical (MAC/PHY) layers to support data delivery over self-configuring multihop topologies. In IEEE 802.11s, mesh points (MPs) interconnect with each other through wireless links and forward packets hop by hop by covering each other within their transmission ranges. In Fig. 1, using a WMN to suffice a service range causes a larger overlapped transmission area than using infrastructure-based WLANs. In a single-channel WMN, the overlapped areas may cause *cochannel interference* and degrade the performance of the WMN.

To avoid the interference among wireless devices, one intuitive approach is using radios that operate in different channels among neighboring devices. Nevertheless, the more the number of radios a device has, the higher the hardware cost and power consumption that it suffers. One possible solution for a low-cost device is utilizing fewer radios that switch among channels. When a device has fewer radios, packet loss might occur in *unattended* channels. In a WMN, dropping packets that are forwarded through a long path wastes all consumed resources. Moreover, Internet-bound packets in a WMN aggregate toward mesh point portals (MPPs), which bridge wired networks. Thus, traffic distributions in a WMN are nonuniform. Monotonous channel switching, which does not adapt to different traffic loads of channels, might result in underutilization of *radios*. Therefore, the objectives of channel switching are to avoid packet loss and maximize the utilization of radios.

A number of studies investigated multichannel MAC protocols that utilize multiple radios, e.g., N radios [3]–[6], [7]–[10], or a single radio [11]–[15] for multiple channels, e.g., M channels. A summary of the previous works is shown in Table I. The protocols proposed in [3]–[6] enable wireless devices to use M radios to transmit data over M channels and provide policies to select feasible channels. To reduce the required number of

Manuscript received July 26, 2009; revised January 29, 2010 and May 27, 2010; accepted September 28, 2010. Date of publication November 11, 2010; date of current version January 20, 2011. The review of this paper was coordinated by Prof. Y. Cheng.

C.-Y. Ku is with the Department of Computer Science, National Chiao Tung University (NCTU), Hsinchu 30010, Taiwan, and also with the Realtek–NCTU Joint Research Center, Hsinchu 30010, Taiwan (e-mail: cyku@cs.nctu.edu.tw).

Y.-D. Lin and S.-L. Tsao are with the Department of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan (e-mail: ydlin@cs.nctu.edu.tw; sltsao@cs.nctu.edu.tw).

Y.-C. Lai is with the Department of Information Management, National Taiwan University of Science and Technology, Taipei 106, Taiwan (e-mail: laiyc@cs.ntust.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2010.2090915

TABLE I
RELATED WORK OF M-CHANNEL N-RADIO ISSUES

Type	Requirement	Main Idea	Extension
Multi-channel MAC protocols	M-channel M-radio	All as data channels Selecting an idle channel [3]	Based on signal strength [4]
			Based on round-trip time [5]
		1 control channel Based on the receiver`s SNR [6]	
	M-channel 2-radio	1 common channel for control 1 fixed radio in the common channel [7]	Regarding power control [8]
		1 fixed channel for receiving data 1 fixed radio in the reception channel [9]	The oldest packets first [10]
	M-channel 1-radio & global timer sync.	Common channel & time slot [11]	Receiver-based [12][14] Pre-determined pattern [15]
		Exchanging hopping sequences [13]	
Based on the current MAC	M-channel 1-radio & signaling protocol	Messages to leave/join a channel [16]	Tree-based sync. mech. [17] Round-robin slot duration [18] Using power-saving [19]

radios, [7]–[10] fix a radio in a predetermined channel for synchronization or reception and switch the other radios to transmit data. However, the approaches cannot be applied to single-radio devices, and the aforementioned fixed radio is usually underutilized. The single-radio studies [11]–[15] assume global timer synchronization among devices. Radios could switch to a predetermined channel at a specific time. Unfortunately, global timer synchronization in a WMN is difficult to achieve due to the unpredictable propagation delay of synchronization messages.

So and Vaidya [16] suggested a signaling protocol that coordinates the channel switching of devices without global timer synchronization. Wu *et al.* [17] further proposed a tree-based signaling protocol for synchronization. However, they focused on routing issues without considering channel-switching algorithms. Xu *et al.* [18] discussed asynchronous spectral multiplexing and the weighted round-robin (RR) scheme that considers slot duration. Nevertheless, they did not provide a concrete switching algorithm. The MultiNet adaptive scheme (MNAS) [19] suggested a switching algorithm based on the weighted RR with a fixed cycle. Although MNAS proposed a concrete method of channel switching, a fixed cycle that is ignorant of variant traffic loads in a WMN results in underutilization of radios, and its RR cannot select channels to maximize radio utilization.

This paper presents a practical approach for utilizing fewer radios to switch among channels in a WMN. The proposed method, called the traffic-aware switching scheme (TRASS), utilizes the existing IEEE 802.11 mechanisms to avoid packet loss and exercises a traffic-aware channel-switching algorithm to raise the utilization of radios. TRASS is a distributed channel-switching approach compatible with the legacy IEEE 802.11 devices and does *not* require global timer synchronization. We realize TRASS on a WLAN mesh test bed and evaluate its performance through both experiments and simulations.

The rest of this paper is organized as follows. Section II introduces IEEE 802.11 and 802.11s. Section III gives the problem statements of channel switching. The architecture, switching algorithm, and system design of TRASS are presented in Sections IV–VI, respectively. The performance evaluation of switching algorithms is discussed in Section VII. Finally, this paper is concluded in Section VIII.

II. BACKGROUND

This section introduces the IEEE 802.11 and the architecture of the IEEE 802.11s WMN. In a WMN, multiple radios or channel switching is needed to reduce cochannel interference as well as retain connectivity. When channel switching is applied, a mesh AP needs notification mechanisms to avoid packet loss. The adopted notification mechanisms in the IEEE 802.11 standard are also described.

A. IEEE 802.11 and IEEE 802.11s

IEEE 802.11 is a wireless communication technique that uses *three* nonoverlapping channels applied in the 2.4-GHz band, whereas *eight* nonoverlapping channels are applied in the 5-GHz band. Devices in the same channel might introduce cochannel interference. Cochannel interference results in collisions of frames and wastes available time of the medium. In single-channel WLANs, neighboring devices in the same channel might block one another and decrease throughput. Therefore, separating APs into different channels may decrease cochannel interference. However, in nonoverlapping channels, an 802.11-based multiradio device suffers board crosstalk, radiation leakage, and adjacent channel interference [5], [32], [33]. These problems are hardware dependent [5] and can be alleviated by utilizing hardware and radio frequency (RF) techniques such as shielding, bandpass filters, and antenna separation by distances or orientations [32], [33]. Nachtigall *et al.* [32] demonstrated that two radios that are separated 80 cm from each other and operate in channels 1 and 6, respectively, become noninterfering under the 2.4-GHz/orthogonal frequency-division multiplexing (OFDM) configuration. According to previous studies, an MP still can benefit from operating more than one radio in the same spectrum if hardware and RF could carefully be designed to eliminate the interference.

IEEE 802.11s defines mesh networking using the IEEE 802.11 MAC/PHY layers that support layer-two path selection protocols and data forwarding over the multihop transmissions. Fig. 2 illustrates the architecture of the IEEE 802.11s WMN. Each node that joins the WMN is called an MP. An MP that bridges heterogeneous networks is called an MPP. An MP that

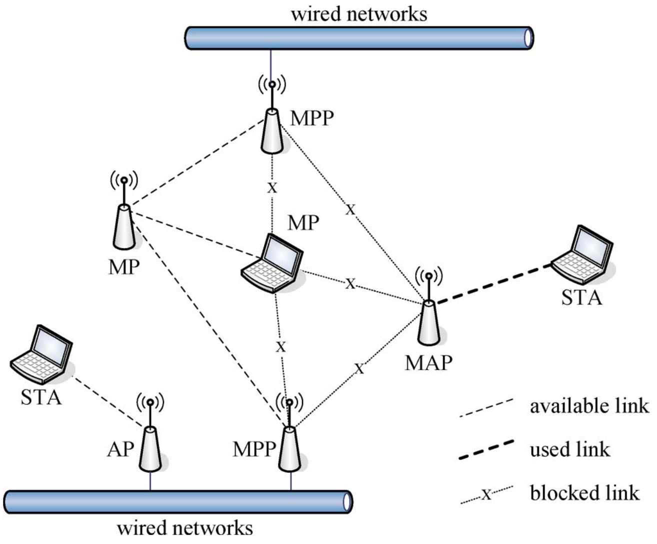


Fig. 2. Architecture of an IEEE 802.11s WMN.

also plays the role of an AP is called a mesh access point (MAP). Mostly, a user accesses the Internet through the WMN with a mobile STA associated with an MAP. The MAP forwards data from MPs to the MPP hop by hop. In a single-channel WMN, transmissions in a neighborhood may interfere with one another. Taking Fig. 2 as an example, the transmission between the STA and the MAP *occupies* the channel and potentially *blocks* the transmissions between the MP and the MPPs. Different from infrastructure-based WLANs, separating MAPs into different channels to reduce cochannel interference may result in the loss of *connectivity*. Therefore, utilizing multiple channels in a WMN needs to reduce cochannel interference and also retain connectivity.

B. Power-Saving Mechanism

The power-saving mechanism aims at reducing power consumption under low traffic loads. An MP enters its power-saving mode (PSM) after sending a beacon [1], [2], with the power management bit *set* to its neighboring MPs. Then, the neighboring MPs buffer the data bound for the MP and announce that step by the traffic indication map (TIM) element [1], [2], which is a bitmap that indicates if the buffered data exist. An MP should wake up at the beginning of each beacon interval and check the TIM element. If the buffered data exist, the MP might retrieve the buffered data by the PS-Poll [1], [2] and then return to its PSM. When an MP decides to wake up, it leaves its PSM and sends a beacon, with the power management bit *reset* to its neighboring MPs. Next, the MP waits for the neighboring MPs to transmit the buffered data to it. In infrastructure-based WLANs, an STA follows similar operations to handle the power-saving mechanism. Note that the standard prohibits an AP to enter its PSM.

C. HCCA

IEEE 802.11 provides the following two medium access coordination functions: 1) the distributed coordination function (DCF) [1] and 2) the point coordination function (PCF) [1]. The DCF is a contention-based medium access mechanism. On the other hand, the PCF is a contention-free (CF) medium access mechanism. A point coordinator that resides in an AP manages a centralized access control method. The PCF has been extended to the hybrid-coordination-function-controlled channel access (HCCA) mechanism [23] for providing the quality-of-service (QoS) accesses in WLANs. HCCA *interleaves* the contention-free period (CFP) [1], [23] and the contention period (CP) [1], [23] to control medium access. An AP enters the CFP by sending its beacon, including the CF parameter set [1], [23], which carries the correlative information of the CFP. All the STAs under the AP coverage transmit data after the AP polls them. The AP issues a CF-End [1], [23] to announce that the CFP is over. Then, the STAs return to the CP and contend with the medium for transmissions.

III. PROBLEM STATEMENT

A. Problems With Channel Switching

Coordination between switch points can improve the performance of a WMN. However, coordination between all switch points through message passing may introduce overheads, particularly considering long latency of message passing in a WMN. On the other hand, accurate coordination between switch points through global timer synchronization is difficult to achieve in a WMN. Our initial goal is to utilize multiple channels without introducing any new coordination protocol so that we can perform the channel switching based on the existing mechanisms. Therefore, we focus on allocating the *radios* of an MP to handle the traffic loads of channels by *channel switching*. Nevertheless, channel switching might cause collisions and packet loss during packet exchanges between MPs. These problems, which have been found by previous works, are illustrated shown in Fig. 3.

The first problem is called the *multichannel hidden terminal* [11]. In Fig. 3(a), MP_3 uses the request to send/clear to send (RTS/CTS) mechanism [1] to protect its transmission against collisions. However, MP_1 loses the CTS frame in Ch_i and does not update its NAV [1], which indicates how long the transmission occupies the medium. Then, a collision occurs when MP_1 *returns* and transmits its RTS in Ch_i. The second problem is called *deafness* [24], [25]. In Fig. 3(b), MP_1 and MP_2 stay in different channels but try to transmit packets to each other. Packet loss happens, because the receiver cannot receive the transmitted packets. The third problem is that traffic loads between MPs are *nonuniform*. In Fig. 3(c), MP_1 intends to transmit a large amount of data to MP_2 in Ch_i, whereas MP_3 has little traffic for MP_2 in Ch_j. MP_2 switches between the channels by RR with a fixed staying period. RR, which monotonously switches the radio to Ch_j, causes underutilization of the radio due to the little traffic. On the other hand, a short staying period incurs high *switching overhead*, whereas a long *staying period* might result in a long idle time. To improve radio utilization, a channel-switching algorithm should select the next channel and determine an appropriate staying period according to traffic loads.

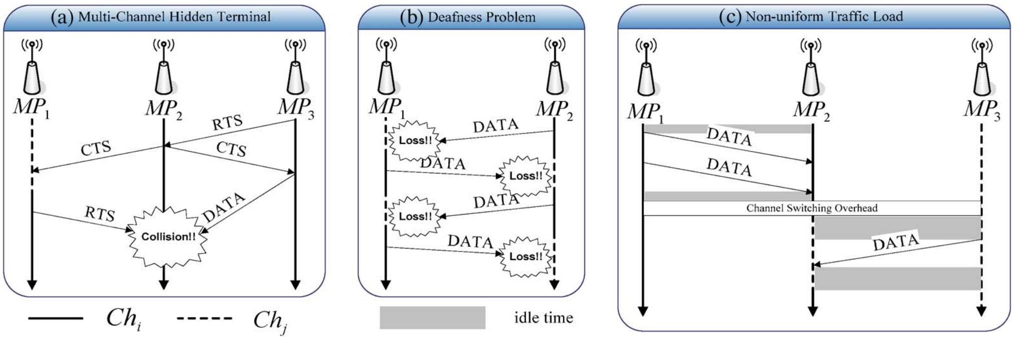


Fig. 3. Examples of the problems caused by channel switching.

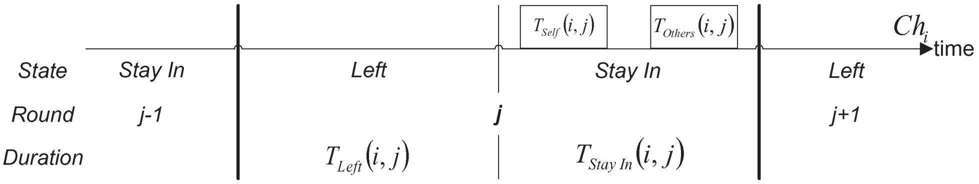


Fig. 4. TRASS round of Ch_i.

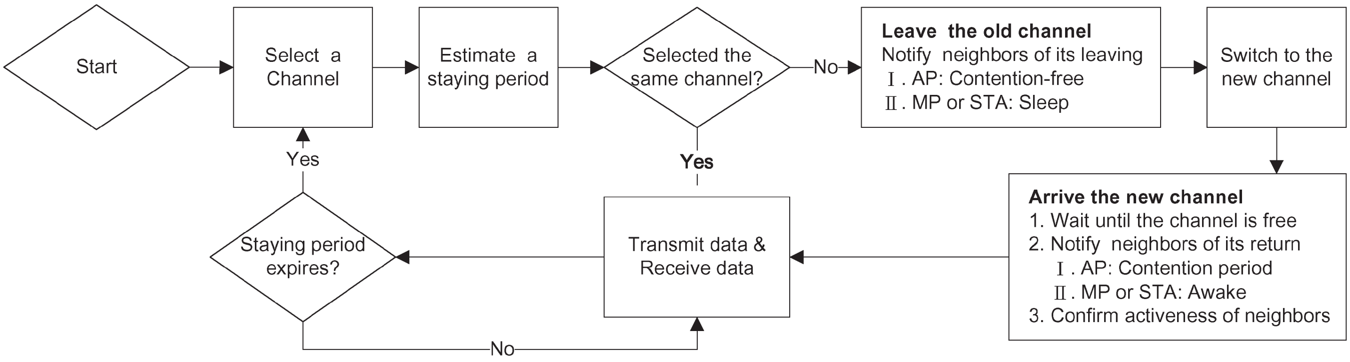


Fig. 5. Flowchart of TRASS.

B. Problem Formulation

One formal statement of channel switching is given as follows. An MP that is applied in channel switching is called a *switch point*. Suppose that a switch point has N radios and stays in M channels, where M is larger than N. Fig. 4 shows the variables used by the channel-switching algorithm described in Section V.

For all Ch_i, $1 \leq i \leq M$, state $Stay\ In$ indicates that the switch point stays in Ch_i by assigning a radio to Ch_i. State $Left$ indicates that the switch point dose not stay in Ch_i. A TRASS round of a channel consists of a period in state $Left$ and a period in state $Stay\ In$. Each channel has its sequence of rounds. During the jth round of Ch_i, $j \geq 1$, $T_{Left}(i,j)$, and $T_{Stay\ In}(i,j)$ denote the duration in states $Left$ and $Stay\ In$, respectively. $T_{Self}(i,j)$ is the channel time occupied by the data transmitted and received by the switch point during $T_{Stay\ In}(i,j)$. $T_{Others}(i,j)$ is the channel time occupied by the data of the others during $T_{Stay\ In}(i,j)$. $D_{Done}(i,j)$ is the amount of transmitted and received data of the switch point during $T_{Stay\ In}(i,j)$. In fact, $T_{Self}(i,j)$ is equal to $D_{Done}(i,j)$ over the mean data rate. $T_{Buffered}(i,j)$ is the amount of data buffered by the switch point at the end of $T_{Left}(i,j)$.

Channel switching allocates the radio resources for traffic loads on channels. Inappropriate switching may result in idle time in the channels with little traffic and buffer overflow in the channels with heavy traffic. Idle channel time wastes available channel resources and decreases the radio utilization of a switch point. Low radio utilization decreases the throughput and potentially increases the buffer overflow. Therefore, the proposed switching algorithm intends to increase the radio utilization and decrease the idle channel time during each round. In other words, TRASS aims at raising $T_{Self}(i,j)/T_{Stay\ In}(i,j)$ and $(T_{Self}(i,j) + T_{Others}(i,j))/T_{Stay\ In}(i,j)$.

IV. TRAFFIC-AWARE SWITCHING SCHEME ARCHITECTURE AND MECHANISMS

A. Overview

The proposed TRASS is a distributed approach applied to each MP. MPs need no global timer synchronization, and their switching behaviors are also unsynchronized. TRASS includes two notification mechanisms to avoid packet loss and a channel-switching algorithm to decide *which* channel to switch to and *how long* it will stay according to the traffic loads. Fig. 5

illustrates the flow chart of TRASS for each free *radio*. A neighbor of a switch point denotes a wireless device which has built a wireless link with the switch point. Initially, TRASS selects a channel for each free radio in which to stay, respectively. Then, TRASS estimates a staying period according to the historical traffic information of the selected channel.

If the selected channel is the same as the *current* one, the switch point keeps transmitting and receiving data in the channel. Otherwise, the switch point utilizes the notification mechanisms specified in the IEEE 802.11 standards to avoid the *deafness* problem. According to whether the switch point plays as an AP in each channel, different mechanisms, as will later be described, are applied to notify its neighbors of its leaving. After switching to the new channel, the switch point waits until the channel is *free* to avoid the *multichannel hidden-terminal* problem. Then, the switch point utilizes the notification mechanisms, which will also described later, to announce its return. However, a neighbor applied channel switching might not stay in the current channel. Thus, the switch point verifies the *activeness* of the neighbors by sending probing messages before transmitting data. As the staying period ends, TRASS repeats the process for the freed radio.

B. Notification Mechanisms: Power Saving and HCCA

TRASS utilizes the existing IEEE 802.11 standards as notification mechanisms to avoid packet loss. The suitable mechanisms can notify the neighbors not to transmit data to the switch point. One way is that the switch point occupies the medium during channel switching. The mesh deterministic access (MDA) in IEEE 802.11s and the CTS-to-self protection mechanisms in IEEE 802.11b/g/n [20]–[22] belong to this category. However, these mechanisms waste the available channel time due to the occupancy of the channel without transmitting data. Thus, a more efficient way is that a switch point passes a message to a group of neighbors that may transmit data to it.

TRASS adopts HCCA and the power-saving mechanism from the WLAN standards [1], [2], [23] for compatibility with other non-TRASS devices. When the switch point acts as an MP or a STA, it applies the power-saving mechanism. However, an AP cannot enter its PSM. The switch point applies HCCA when it acts as an AP. Before leaving the old channel, the switch point enters its PSM or the CFP to demand its neighbors to buffer data for it. After arriving at the new channel, the switch point *leaves* its PSM or the CFP to *notify* the neighbors of its return. Then, the switch point retrieves the buffered data following the current MAC protocol. Moreover, TRASS may benefit by the further extended MAC protocols. For instance, IEEE 802.11e [23] can provide the QoS and assign the priorities to frames.

Avoiding the Deafness and Multichannel Hidden-Terminal Problems: Even applying the notification mechanisms, the deafness problem might still happen, because switch points are not synchronized. For example, a neighbor to which channel switching is also applied *leaves* the channel when the switch point stays in the *other* channel. The switch point is *unaware* of the notification messages announced by that neighbor. After the switch point returns, its transmission causes the deafness problem, because the neighbor has gone. To avoid the deafness problem, switch points need to verify whether the receivers stay in the current channel before transmitting data. A switch point may *passively* receive frames such as beacons from the neighbors or *actively* send probing messages such as probe requests [1] or null data [1]. However, switch points still might not *meet* each other due to their *mismatching* switching sequences. Therefore, the proposed algorithm takes that condition into consideration to decrease the *probability* of the mismatch. According to the simulations and the experiments on our test bed, the deafness problem rarely happens in TRASS.

To avoid the multichannel hidden terminal, switch points need to verify whether the new channel is occupied before transmitting data. A switching point can contend the medium after it hears a complete frame that contains the NAV or when the *longest* duration of a transmission has passed. The longest duration of a transmission is 32767 μs, which is defined in IEEE 802.11. Although the wait time wastes radio resources, it normally occurs under *low* traffic loads and could be acceptable. For heavy traffic load, we next measure the overhead of the wait time.

The overhead for resolving these two problems needs to be assessed. A switch point broadcasts a probe request, and each device in the transmission range may respond a probe response to the switch point. The time to process probe requests and probe responses at a switch point is related to several factors such as the number of devices in the transmission range. Moreover, the wait time to avoid the multichannel hidden terminal under heavy traffic loads varies according to traffic patterns. We measured the statistical overhead on a test bed. The measured overhead is about 2.5 ms under saturated traffic loads. The optimal switching interval measured in our experiments under saturated traffic loads is around 240 ms. In short, the overhead for solving these two problems costs about 2% of available channel time.

The operations of the notification mechanisms are illustrated by the following example. Fig. 6 shows the time series of the scenario that one radio switches between two channels. The *MAP* is a switch point and acts like an MAP. The *MAP* issues a beacon frame with the CF parameter set to announce the CFP before it leaves Ch_i. After switching to Ch_j, it waits to assure that Ch_j is available and then announces its return by sending its beacon with the power management bit reset. Now, the *MP* could transmit data to the *MAP* if the *MP* has queued up data for the *MAP*. If the *MAP* intends to transmit data to the *MP* whose activeness has not been confirmed, the *MAP* first issues a probe request to explore the *MP*. When the staying duration is over, the radio returns to Ch_i after the *MAP* sends its beacon with the power management bit set. The *MAP* announces its return by a CF-End to resume the CP after waiting to assure that Ch_i is not occupied by the others not in the BSS.

V. TRAFFIC-AWARE SWITCHING SCHEME ALGORITHM

Routing paths and traffic patterns determine the traffic load distributions seen by each switch point. In a WMN, the routing is relatively stable, and the timescale of routing path changes is usually much larger than that of channel switching. For example, the lifetime of a routing path in our test bed is between 60

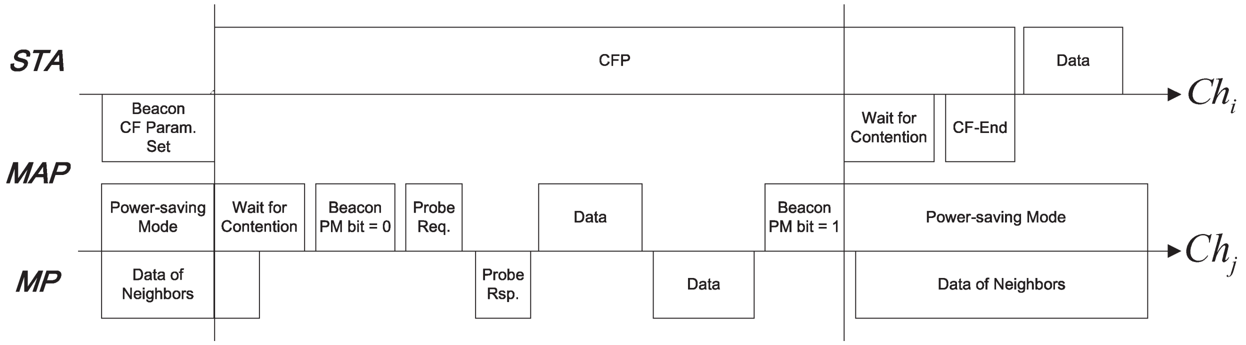


Fig. 6. One radio that switches between two channels.

and 500 s, whereas the channel-switching procedure frequently occurs every 100 ms. Thus, the routing protocol and the channel switching can be regarded as being loosely coupled, and they are separately handled in this paper. We assume that the routing has been determined by other protocols, and we concentrate on channel-switching and resource allocation issues.

To deal with nonuniform traffic loads of channels, the channels with higher traffic loads should acquire more resources at the switch point to raise the utilization of radios. One general strategy is either prioritizing the selection of channels with a *fixed* staying period or in a RR basis with *variable* staying periods. However, an unsuitable staying period might incur a high switching overhead or underutilization. RR scheduling might waste time on the channels that carry little traffic and cannot serve the channels that carry heavy traffic in time.

TRASS considers both *prioritized* channel selection and *variable* staying periods and divides the channel-switching procedure into two phases. The first phase selects *in which* channel to stay based on the radio utilization in each channel. The radio utilization represents the percentage of time that the switch point spends in transmitting and receiving data during the staying period in a channel. TRASS selects the channel with the *highest* radio utilization. On the other hand, because of the medium access control overheads [26]–[28], the channel resources cannot be fully utilized. Thus, the second phase estimates the duration of stay in the selected channel according to the target channel utilization, which indicates the percentage of time available to transmit data. The traffic loads of neighboring devices are also considered, because the switch point shares the channel resources with them. TRASS considers short staying periods when a switch point does not have much traffic to handle. In this situation, short staying periods that introduce channel-switching overheads become acceptable. On the other hand, TRASS allocates long staying periods to the channels so that switching overheads can be reduced when the MP has a heavy traffic workload.

A. Channel Selection

For a radio, TRASS selects the next channel according to the radio utilization of each channel during its previous staying periods. Selecting a channel with low radio utilization causes the switch point to be idle in the channel and potentially results in buffer overflow in other channels with high radio utilization. Thus, TRASS selects the channel with the *highest* radio utilization. The traffic loads of the channels with lower radio utilization would be accumulated when the switch point relieves the traffic loads of the channel with the highest radio utilization. In summary, TRASS could increase the radio utilization of the channels with low traffic loads and avoid the buffer overflow in the channels with high traffic loads. A comparison with the weighted RR proposed in [19] is given in Section VII.

Extended Radio Utilization: In wireless communications, the radio utilization is evaluated by the occupied channel time rather than the amount of data, because the transmission rates are variable. For example, transmitting data of 594 B spends 54 μs at 11 Mb/s or 11 μs at 54 Mb/s. Thus, TRASS defines the radio utilization during the $(j-1)$th round of Ch_i as $T_{Self}(i,j-1)/T_{Stay\ In}(i,j-1)$. To adapt the algorithm to various traffic patterns, historical information is taken into account. TRASS uses the *weighted moving average* of radio utilization, i.e.,

$$(1-\alpha_{i,j-1}) \times \frac{\sum_{k=0}^{j-2} T_{Self}(i,k)}{\sum_{k=0}^{j-2} T_{Stay\ In}(i,k)} + \alpha_{i,j-1} \times \frac{T_{Self}(i,j-1)}{T_{Stay\ In}(i,j-1)}$$

where $\alpha_{i,j-1}$ is a parameter that indicates the weight of the $(j-1)$th round of Ch_i. The zeroth round of Ch_i is defined as the initial historical information. TRASS assumes that the zeroth round can achieve the target channel utilization, and it can fully be utilized by the switch point. This design gives new channels opportunities to be selected in the initial stage. Nevertheless, a busy channel might always be selected and starve the other channels following the aforementioned equation. To avoid the starvation, an *aging* mechanism based on the duration in state $Left$ is applied. The equation of the "extended" radio utilization of Ch_i is

$$\frac{(1-\alpha_{i,j-1}) \times \sum_{k=0}^{j-2} T_{Self}(i,k)}{\sum_{k=0}^{j-2} T_{Stay\ In}(i,k)} + \frac{\alpha_{i,j-1} \times T_{Self}(i,j-1)}{T_{Stay\ In}(i,j-1)} + \frac{T_{Left}(i,j)}{\beta_{i,j}} \qquad (1)$$

where $\beta_{i,j}$ denotes the maximum tolerable duration in state $Left$ during the jth round of Ch_i. When $T_{Left}(i,j)$ exceeds $\beta_{i,j}$, the value of (1) becomes greater than 1, and then, TRASS selects Ch_i as the next channel in which to stay. For switch points that do not meet each other for a long time, this mechanism also increases their opportunities. $\beta_{i,j}$ affects the total switching overhead, whereas $\alpha_{i,j}$ affects the prediction of traffic loads. In this paper, $\alpha_{i,j}$ and $\beta_{i,j}$ are determined according to the experiments on the test bed.

B. Channel Time Allocation

Target Channel Utilization: After Ch_i is selected, TRASS estimates a suitable staying period for Ch_i to deal with the traffic load of the switch point. Because the switch point shares the channel resource with the others in Ch_i, the occupied channel time of both the switch point and the others are taken into account. To fully utilize the available channel time, TRASS estimates the staying period that could achieve the *target channel utilization*, U, i.e.,

$$\frac{T_{Self}(i,j)+T_{Others}(i,j)}{T_{Stay\ In}(i,j)}=U. \tag{2}$$

In this paper, U is regarded as a constant and is measured as 46.91% on the test bed. To solve $T_{Stay\ In}(i,j)$, we have to estimate $T_{Self}(i,j)$ and $T_{Others}(i,j)$ above all.

Estimating $T_{Self}(i,j)$ and $T_{Others}(i,j)$: $T_{Others}(i,j)/T_{Stay\ In}(i,j)$ may change according to different traffic load distributions. To adapt to various traffic load distributions, TRASS uses the weighted moving average of $T_{Others}(i,j-1)/T_{Stay\ In}(i,j-1)$ to estimate $T_{Others}(i,j)$. The weighted moving average is

$$(1-\gamma_{i,j-1})\times\frac{\sum_{k=0}^{j-2}T_{Others}(i,k)}{\sum_{k=0}^{j-2}T_{Stay\ In}(i,k)} + \gamma_{i,j-1}\times\frac{T_{Others}(i,j-1)}{T_{Stay\ In}(i,j-1)} \tag{3}$$

where $\gamma_{i,j-1}$ is a parameter that indicates the weight of the $(j-1)$th round of Ch_i. To simplify the representation, the weighted moving average is represented as $T^*_{Others}(i,j-1)/T^*_{Stay\ In}(i,j-1)$. Then, the estimated $T_{Others}(i,j)$ by the moving average is

$$T^*_{Others}(i,j-1)\times\frac{T_{Stay\ In}(i,j)}{T^*_{Stay\ In}(i,j-1)}. \tag{4}$$

$T_{Self}(i,j)$ includes the occupied channel time for the buffered data by the switch point and its neighbors. We give the assumption that the buffered data of the neighbors proportionally grows with $T_{Left}(i,j)$, because the neighbors accumulate the data for the switch point during the period in state $Left$. Based on the assumption, $T_{Self}(i,j)$ is approximated as

$$T_{Self}(i,j-1)\times\frac{T_{Left}(i,j)}{T_{Left}(i,j-1)}$$

by proportionally adjusting $T_{Self}(i,j-1)$ based on the duration in state $Left$ of the $(j-1)$th and jth rounds. On the other hand, the exact amount of the buffered data by the switch point is known at the end of $T_{Left}(i,j)$. We assume that the contributed occupied channel time by the buffered data is in proportion to the transmitted or received data. Based on the assumption, we estimate $T_{Self}(i,j)$ as

$$T_{Self}(i,j-1)\times\frac{T_{Left}(i,j)}{T_{Left}(i,j-1)} \times\frac{D_{Done}(i,j-1)+D_{Buffered}(i,j)}{D_{Done}(i,j-1)} \tag{5}$$

i.e., enlarging the estimated occupied channel time by the total amount of data over the transmitted and received data during the $(j-1)$th round. When no datum is transmitted and received during the $(j-1)$th round, TRASS sets $D_{Done}(1,j-1)$ as the size of the frames used by the notification mechanisms.

Finding $T_{Stay\ In}(i,j)$: The only unknown variable now is $T_{Stay\ In}(i,j)$ in (2) and (4). We could get $T_{Stay\ In}(i,j)$ by substituting the estimated $T_{Self}(i,j)$ and $T_{Others}(i,j)$ in (5) and (4), respectively, into (2). Then, $T_{Stay\ In}(i,j)$ is

$$T_{Stay\ In}(i,j) = \begin{cases} MinTime, & \text{if } T_{Stay\ In}(i,j) < MinTime \\ T_{Stay\ In}(i,j-1), & \text{if } \frac{T^*_{Others}(i,j-1)}{T^*_{Stay\ In}(i,j-1)} \geq U \\ \frac{T_{Self}(i,j)}{U-\frac{T^*_{Others}(i,j-1)}{T^*_{Stay\ In}(i,j-1)}}, & \text{otherwise} \end{cases} \tag{6}$$

where $T_{Self}(i,j)$ is not expanded to simplify the representation. $MinTime$ denotes the duration required by the notification mechanisms for transmitting data. The special case in (6) is that it is very congested to achieve the target channel utilization, because $T^*_{Others}(i,j-1)$ causes the overloading of Ch_i. In that case, TRASS can never reach U by adjusting the length of the staying period. Then, the staying period is given as the last staying period until overloading is relieved.

C. Example of the TRASS Algorithm

The scenario in Fig. 7 is an example of explaining the operation of the TRASS algorithm.

There are two channels that share one radio. Suppose that $U=0.9$, $\alpha_{1,j}=\alpha_{2,j}=1$, $\beta_{1,j}=\beta_{2,j}=100$, and $\gamma_{1,j}=\gamma_{2,j}=1$ for all $j\geq 0$ and that there are no buffered data for each channel. TRASS is now run at the time marked by the solid line. TRASS first selects a channel according to (1). Because the extended radio utilization of Ch_1 and Ch_2 is $4/20+12/100=0.32$ and $2/12+0/100=0.167$, respectively, Ch_1 is selected. Then, estimating $T_{Self}(1,j)$ by (5), $T^*_{Others}(1,j-1)/T^*_{Stay\ In}(1,j-1)$ by (3), and $T_{Stay\ In}(1,j)$ by (6), we have $T_{Self}(1,j)$ as $4\times(12/8)=6$, $T^*_{Others}(1,j-1)/T^*_{Stay\ In}(1,j-1)$ as $10/20=0.5$, and $T_{Stay\ In}(1,j)$ as $6/(0.9-0.5)=15$ ms, respectively.

VI. IMPLEMENTATION

The proposed notification mechanisms and the channel-switching algorithm are implemented on our WLAN mesh test bed based on a commercial-off-the-shelf WLAN chipset: Realtek RTL8186 [29]. The WLAN mesh solution that conforms

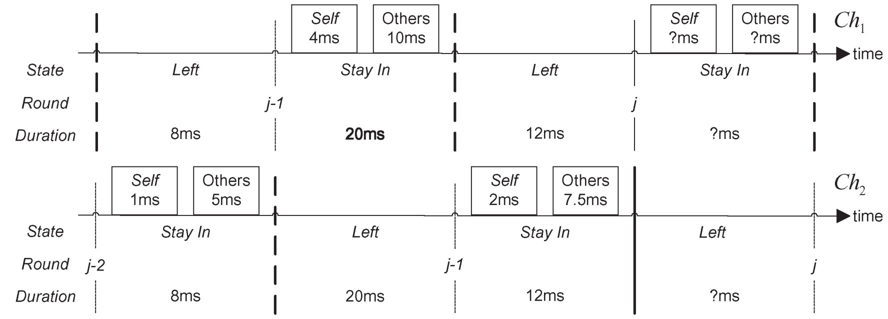


Fig. 7. Example of the channel-switching algorithm.

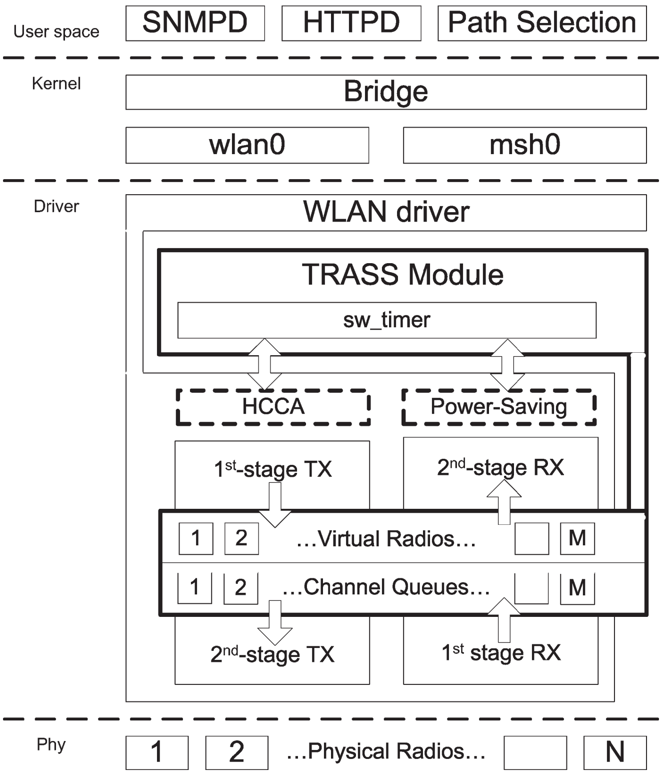


Fig. 8. System architecture of the implementation on RTL8186.

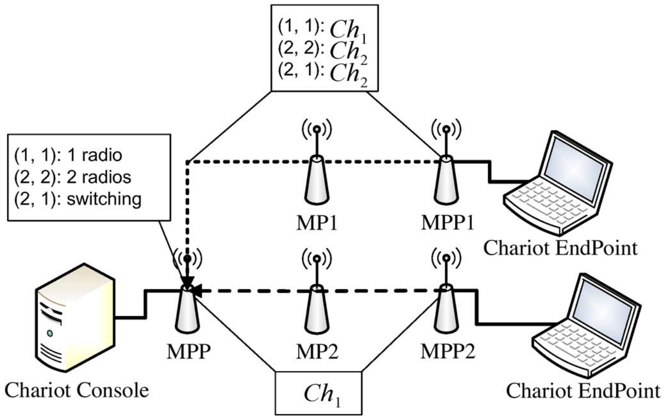


Fig. 9. Experiment topology of throughput evaluation by Chariot.

to the IEEE 802.11s draft revision 2.03 was developed and investigated in [30]. The architecture of RTL8186 and the implementation of TRASS are shown in Fig. 8. There are some daemon programs in the user space and a kernel-level bridge module with interfaces to infrastructure-based WLANs and WMN. The function that supports multichannel transmissions is developed on the data path of the WLAN driver. Because an MP in WMN is responsible for forwarding frames among different channels, such frames must be *intercepted* on the data path and *buffered* until the neighbors return to the corresponding channels. To achieve transparent interception and multichannel transmissions, the transmission procedure (TX) and the reception procedure (RX) are divided into two stages. The first-stage TX/RX is responsible for capturing the requests from either the upper layer or the hardware. On the other hand, the second-stage TX/RX is responsible for dealing with the operations such as encapsulating data to transmit and deencapsulating frames to extract data. To intercept frames without influencing the original data path, the module of the multichannel transmissions lies between the first- and second-stage TX/RX. This module includes a *virtual-radio* interface, which hides the hardware constraints, and a *channel queue* for each channel to realize the multichannel transmissions. The frames that pass the first-stage TX to the virtual radio of an unattended channel are queued up at the corresponding channel queue. The frames are later handled by physical radios when the corresponding channel is *selected* by TRASS.

TRASS is completely implemented in the Linux kernel for efficiency. In Fig. 8, the notification mechanisms are embedded in the WLAN driver, because it *reuses* original functions offered by the IEEE 802.11 MAC/PHY layers, including the power saving and HCCA. On the other hand, the channel-switching algorithm and the channel queues are implemented as separate modules. The channel-switching algorithm is implemented as a *timer handler*: sw_timer. When a staying period expires, a timer triggers the switching algorithm and performs channel switching. The frames of the selected channel are passed to the entry of the second-stage TX, whereas the frames of the other channels are *buffered* until these channels are again selected.

VII. EVALUATION

This section investigates the performance of TRASS. The evaluation of the two-channel one-radio configuration (2, 1) is measured on the test bed. The result is compared with (1, 1), (2, 2), and the optimal throughput. The throughput of the more

TABLE II
REALISTIC PARAMETER VALUES IN THE TEST BED AND SIMULATIONS. DEPLOYMENT AREA: 640 × 640 m^2 (SQUARE METERS)

Parameter Name	Value
SIFS	10 (us)
DIFS	30 (us)
Max. contention window size	1024
Min. contention window size	32
Contention window time slot	10 (us)
Supported transmission rate	1, 2, 5.5, 11, 6, 9, 12, 18, 24, 36, 48, 54 (Mbps)
Channel switching overhead	6 (ms)
Queue size	64 (Frames)
Deployment Area	640 * 640 (m2)
Transmission Range	160 (m)
Deployed mesh points	256
Portal Proportion	25 (%)
Packet loss probability	0.04
Target channel utilization	46.91%
$\alpha_{i,j}$	0.5
$\beta_{i,j}$	1000 (ms)
$\gamma_{i,j}$	1

generic (M, N) is evaluated through simulations over the topology extended from the test bed and the large-scale topologies. Finally, the channel-switching algorithms, including MNAS [19], are compared.

A. Evaluation Environment

This experiment examines the throughput enhancement by applying the proposed approach on our WLAN mesh test bed. NetIQ Chariot [31] is used to measure the throughput. The experimental topology with an MPP that interconnects two transmission paths is shown in Fig. 9.

To examine the maximal throughput, the two endpoints generate best effort UDP data streams to the Chariot console through two-hop paths. (1, 1), (2, 1), and (2, 2), as shown in Fig. 9, are measured and evaluated. In (1, 1), the *MPP* is equipped with one radio, and the two paths are assigned in Ch_1. In (2, 2), the *MPP* is equipped with two radios, and the two paths are assigned in Ch_1 and Ch_2, respectively. Because the WLAN chipset, RTL8186, supports only one radio, the switch point utilizes two RTL8186 chipsets to emulate (2, 2). In (2, 1), channel switching is applied to the MPP equipped with one radio to deal with the two paths, which are assigned in Ch_1 and Ch_2, respectively.

To have more large-scale comparisons, the throughput of the WMN under larger (M, N) is evaluated by a simulator written in the C language. Two different topologies of the simulations are investigated. One of these topologies is extended from the test bed. The *MPP* utilizes N radios to handle M two-hop paths in M channels to examine the improvement of multiple radios. The other topology is randomly deployed to examine the throughput enhancement over a large-scale WMN. To obtain realistic simulation results, the system parameters of RTL8186 are used in the test bed and simulations. The parameter values in the simulations are summarized in Table II.

All the MAC parameters are measured in a shielding box. The channel-switching overhead is 6 ms, which is the spent time of the channel-switching function in the driver. The packet loss probability is around 4% on the average. The transmission range, which is assumed to be the same as the interference range, is measured as around 160 m in practice. The target channel utilization measured in noninterfered environments is 46.91%. $\alpha_{i,j}$ is set to 0.5 to refer to both the last round and historical information. $\beta_{i,j}$ is determined as 1000 ms based on the experimental results measured at the maximum throughput. $\gamma_{i,j}$ is set to 1, because the background traffic load is controlled as a fixed value in our experiments. Furthermore, the deployment area refers to the features of the campus. The simulations assume that the deployed number of MPs is 256, and 25% of these MPs play the role of MPP to bridge the Ethernet.

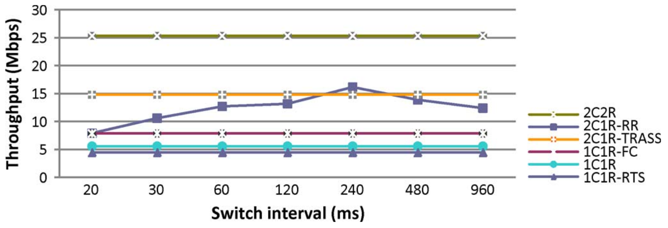


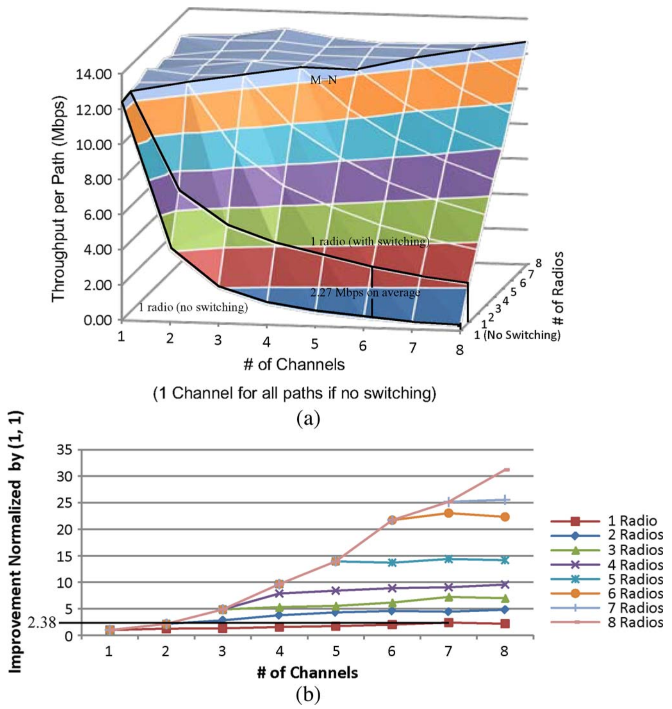
Fig. 10. Evaluation result of throughput.

B. Throughput Evaluation of (2, 1) by Test Bed

To examine the performance of (2, 1) to which TRASS is applied, (1, 1), (2, 2), and (2, 1) to which RR is applied are compared. The results are depicted in Fig. 10.

The throughput of 2C1R-TRASS, which implies TRASS applied to (2, 1), can achieve 14.8 Mb/s. 2C1R-RR, which implies RR applied to (2, 1), reaches its maximum, i.e., 16.2 Mb/s, at 240 ms. When the staying period is short, the switching overhead degrades the throughput. When its staying period exceeds 240 ms, buffer overflow occurs. TRASS reaches 91% of the above maximum *all* the time.

Three possible extended configurations of (1, 1) are all examined to obtain the maximum throughput. In 1C1R, *MPP1* and *MPP2* can only sense *MP1* and *MP2*, respectively. The



(a)

(b)

Fig. 11. (a) Simulation result of the extended topologies. (b) Simulation result of the large-scale topologies.

hidden-terminal problem might occur. 1C1R-RTS means that the RTS/CTS [1] is applied to 1C1R to avoid hidden terminals. In 1C1R-FC, all devices reside in the same collision domain, and no hidden terminal exists. In our experimental result, 1C1R-FC has better performance due to at most two concurrent transmissions in either 1C1R or 1C1R-RTS. The gain of concurrent transmissions is limited and cannot overcome the overheads of RTS/CTS and the hidden-terminal problem. The performance of the 2C1R-TRASS is nearly *twice* as high as 1C1R-FC, because the available channel resource of (2, 1) is twice as (1, 1). Furthermore, the hidden-terminal problem at the switch point is alleviated by separating *MP1* and *MP2* into different channels. The experimental results demonstrate the improvement of (2, 1), which is contributed by multiple channels that eliminate the cochannel interference.

A comparison with 2C2R shows that the throughput of 2C1R-TRASS *exceeds half* of 2C2R. One observation is that a burst of traffic appears when the switch point arrives at a new channel. The burst of traffic results from the buffered data at the intermediate MP. The intermediate MP can pass the buffered data through the forwarding procedure, which is potentially a bottleneck when the switch point stays in the other channel. After the switch point returns to the channel, the intermediate MP may directly transmit the buffered data and enhances the utilization of the radio.

C. Throughput of (M, N) by Simulation

Fig. 11(a) exhibits the maximum throughput per path over the extended topologies. Fig. 11(b) shows the throughput improvement normalized by (1, 1) over large-scale topologies. In Fig. 11(a), the throughput significantly decreases when the radios are fewer than the channels due to the reduced capacity of the switch point. Similarly, Fig. 11(b) shows that the throughput improvement is limited by the number of radios. As a result, channel switching is unsuitable to deal with heavy traffic loads. However, the difference between the bold lines in Fig. 11(a) shows that a single radio with channel switching enhances 2.27 Mb/s, because traffic can be dispersed into different channels to reduce the interference.

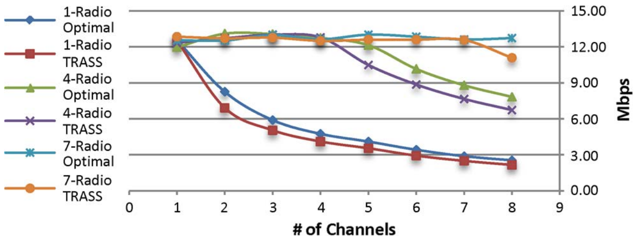


Fig. 12. Maximum throughput of TRASS and the optimal solution.

Moreover, (3, 2) and (3, 1), which achieve 69.8% and 39.7%, respectively, of the throughput of (3, 3) shows that TRASS fully utilizes the radios and notably enhances the throughput. Fig. 11(b) also exhibits significant throughput improvement over the multichannel scenarios, because the cochannel interference is more serious in such a large-scale topology than the chain topologies. The large-scale topology includes 256 MPs equipped with up to eight radios that switch among at most eight channels. Even equipped with a single radio at each MP, the bold line shows that the throughput reaches *twice* the single-channel scenario.

Fig. 12 gives a comparison of the maximum throughput between TRASS and the optimal solution.

The optimal solution is regarded as the RR that operates at the optimal interval obtained by extensive experiments. TRASS approximates the optimal solution well. The result shows that the gap slowly increases with the number of radios, because the error of estimation is insignificant.

D. Performance Comparisons of Channel-Switching Algorithms

To investigate channel-switching algorithms, the experiments for observing the adaptation to different traffic loads are examined. The packet loss ratios and the occupied channel time are evaluated through the experiments on the test bed. On the other hand, we investigate the time percentage of radio usage by the simulations, which are based on the same configuration of the test bed. For ease and accuracy of controlling traffic generation, the traffic sources stay in different channels and generate traffic to a switch point through a direct link. In the experiment topology, the switch point acts as an MAP. The other two devices act as a STA and an MP, respectively. The STA and the MP stay in different channels and generate different traffic loads to the MAP by the Linux socket programs. The developed server program that runs on the MAP includes a Transmission Control Protocol (TCP) echo server and a receiver of User Datagram Protocol (UDP) data streams. On the other hand, the developed client program that runs on the STA and the MP could generate both TCP and UDP data streams of random packet

TABLE III
PARAMETER VALUES OF THE CHANNEL-SWITCHING ALGORITHMS

Module Name	Parameter Name	Value
Round-Robin	Switching period	150 (ms)
	SW queue size	64 (frames)
MNAS	Switching cycle	300 (ms)
	SW queue size	64 (frames)
TRASS	$\alpha_{i,j}$	0.5
	$\beta_{i,j}$	300 (ms)
	$\gamma_{i,j}$	1
	U	46.91%
	SW queue size	64 (frames)
Packet generation (Socket programs)	Back-off window size	1/8, 1/16, 1/32, 1/64, 1/128 (s)
	Packet size	150~1500 (Bytes)
Platform (RTL8186)	Channel switching overhead	6 (ms)
	TX queue size of HW	256 (packets)
	Jiffies	10 (ms)

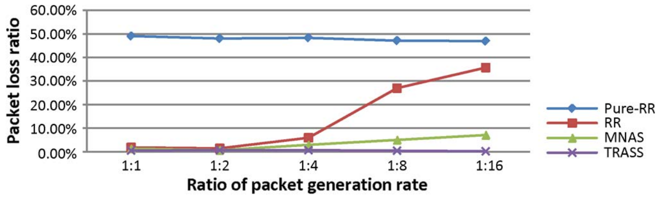


Fig. 13. Packet loss ratios of the channel-switching algorithms.

sizes and control packet generation rates by a configurable random back-off window.

The three algorithms—RR, MNAS, and TRASS—are investigated as follows. RR manipulates radios switching among channels in sequence with a fixed staying period. To compare with TRASS, MNAS as proposed in [19] is examined, but the parameter values of MNAS are based on our system and the test bed. MNAS manipulates radios switching among channels in a sequence with a fixed switching cycle. The switching cycle is divided into M slots whose lengths are in proportion to the observed number of frames in each channel during the *last* switching cycle. All the examined algorithms use our notification mechanisms of TRASS. To verify the effectiveness of the notification mechanisms, RR *without* the notification mechanisms (Pure-RR) is also examined. The experimental parameters are summarized in Table III.

In the experiment of packet loss, each client on the STA or the MP generates best effort UDP data streams to the server on the MAP. The packet loss ratio of each algorithm is shown in Fig. 13.

The packet loss ratio of Pure-RR is close to 50%, because the MAP switches at a fixed interval. Half of the packets might be lost due to half of the time in unattended channels. However, a few packet losses still occur, even when the notification mechanisms are applied. Because the hardware queue on RTL8186 cannot be controlled by the notification mechanisms, certain frames that are queued up by the hardware might be transmitted in incorrect channels. Another notable phenomenon is that the packet loss ratio of RR and MNAS becomes *higher* as the packet generation rate increases. The packet loss occurs due to buffer overflow. The buffer overflow of TRASS is much less than the other algorithms, because TRASS selects the channel with the highest radio utilization to reduce buffer overflow and estimates the staying period without the constraints such as a fixed switching cycle. Both RR and MNAS have a fixed switching cycle. RR wastes resources on the channels with little traffic. Moreover, TRASS takes the buffered data into account. This condition is important to correctly estimate the staying duration, particularly during heavy loads, which introduce a significant amount of buffered data.

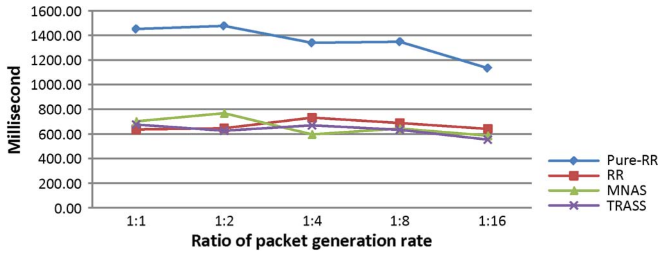


Fig. 14. Occupied channel time of the channel-switching algorithms.

In the experiment of occupied channel time, each client generates the same volume of TCP traffic to the server for each algorithm. The experimental results are shown in Fig. 14.

The occupied channel time of Pure-RR is much higher than the other algorithms. Because frame losses result in retransmissions, the available channel time is wasted by retransmissions. Furthermore, the occupied channel time of the algorithms with the notification mechanisms is very close. One observation is that no packet loss is caused by buffer overflow when the ratio of packet generation rates is high. Because the TCP congestion control reduces the transmission rate, buffer overflow is lessened, and the occupied channel time does not increase.

Fig. 15 illustrates the distribution of radio resources in terms of time percentages of switching overhead, idle time, and radio utilization by applying different algorithms.

Because both RR and MNAS schedule channels every fixed cycle, the switching overhead is independent of the traffic loads and retains a fixed percentage. On the other hand, TRASS frequently switches when the traffic load is low, whereas the switching overheads are acceptable. On the other hand, TRASS reduces the switching overheads by allocating long staying periods for serving heavy traffic workloads. This figure also shows that TRASS can achieve higher radio utilization than RR and MNAS.

VIII. CONCLUSION

This paper has provided a low-cost solution through channel switching to utilize multiple channels in a WMN, particularly for IEEE 802.11s. The proposed method called TRASS reuses the existing IEEE 802.11 mechanisms to avoid packet loss during channel switching. On the other hand, the TRASS algorithm manages channel switching to raise radio utilization according to traffic loads. TRASS selects the next channel with the highest radio utilization. Then, it estimates the staying period based on the target channel utilization to improve $(T_{Self}(i,j) + T_{Others}(i,j))/T_{Stay\ In}(i,j)$.

The experimental results show that TRASS highly adapts to traffic loads of channels. The buffer overflow is decreased, and

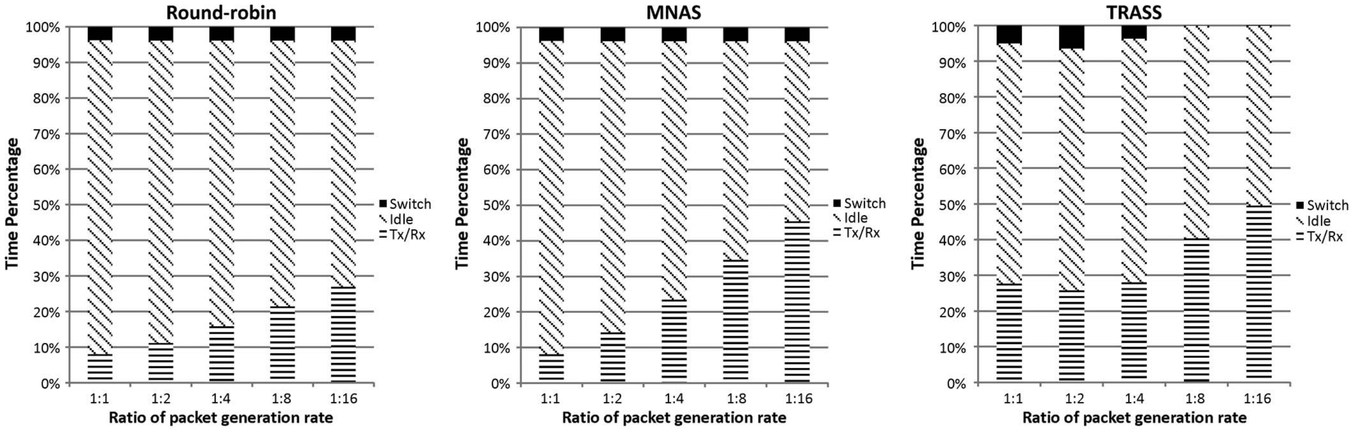


Fig. 15. Time percentages of switching overhead, idle time, and radio utilization.

$T_{Self}(i,j)/T_{Stay\ In}(i,j)$ is increased. To investigate channel switching, the two-channel single-radio configuration is examined on the test bed. TRASS improves the throughput by 75% over the single-channel WMN. Furthermore, the capacity of the generic multichannel multiradio scenarios is evaluated through simulations. The simulation results exhibit that channel switching significantly enhances throughput by minimizing the cochannel interference. The simulations also provide a reference to vendors in designing products and deploying the WMN.

Further issues of channel switching in a multichannel WMN deserve future research. A lightweight coordination protocol between switch points, e.g., message-passing coordination mechanisms, will be considered in our next mesh system design. Coordinating switch points could eliminate cochannel interference and realize the dynamic channel assignment. Moreover, the routing and the channel switching could jointly be considered and optimized, and we will consider this improvement in our future work. Because traffic distribution may highly affect the performance of channel-switching algorithms, further analytical modeling and optimization are needed. Among these approaches, dynamic parameter tuning according to traffic distributions could further be investigated. Ways of minimizing the probability of the deafness problem in the unsynchronized channel switching also deserve further research.

REFERENCES

[1] IEEE 802.11 Working Group, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specs.*, Sep. 1999.

[2] IEEE 802.11 Working Group, *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specs.—Amend. 10: Mesh Networking*, Nov. 2008.

[3] A. Nasipuri, J. Zhuang, and S. R. Das, "A multichannel CSMA MAC protocol for multihop wireless networks," in *Proc. IEEE WCNC*, Sep. 1999, vol. 3, pp. 1402–1406.

[4] A. Nasipuri and S. R. Das, "Multichannel CSMA with signal-power-based channel selection for multihop wireless networks," in *Proc. IEEE VTC*, Sep. 2000, vol. 1, pp. 211–218.

[5] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, "A multiradio unification protocol for IEEE 802.11 wireless networks," in *Proc. IEEE Int. Conf. BroadNets*, 2004, pp. 344–354.

[6] N. Jain, S. R. Das, and A. Nasipuri, "A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks," in *Proc. IEEE Int. IC3N*, Oct. 2001, pp. 432–439.

[7] S. L. Wu, C. Y. Lin, Y. C. Tseng, and J. P. Sheu, "A new multichannel MAC protocol with on-demand channel assignment for multihop mobile ad hoc networks," in *Proc. IEEE I-SPAN*, Dec. 2000, pp. 232–237.

[8] Y. C. Tseng, S. L. Wu, C. Y. Lin, and J. P. Sheu, "A multichannel MAC protocol with power control for multihop mobile ad hoc networks," in *Proc. IEEE ICDCSW*, Apr. 2001, pp. 419–424.

[9] P. Kyasanur and N. H. Vaidya, "Routing and interface assignment in multichannel multi-interface wireless networks," in *Proc. IEEE WCNC*, Mar. 2005, vol. 4, pp. 2051–2056.

[10] P. Kyasanur and N. H. Vaidya, "Routing and link-layer protocols for multichannel multi-interface ad hoc wireless networks," *ACM Mobile Comput. Commun. Rev.*, vol. 10, no. 1, pp. 31–43, Jan. 2006.

[11] J. So and N. H. Vaidya, "Multichannel MAC for ad hoc networks: Handling multichannel hidden terminals using a single transceiver," in *Proc. ACM Int. Symp. MOBIHOC*, May 2004, pp. 222–233.

[12] W. H. Tam and Y. C. Tseng, "Joint multichannel link layer and multipath routing design for wireless mesh networks," in *Proc. IEEE INFOCOM*, May 2007, pp. 2081–2089.

[13] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad hoc wireless networks," in *Proc. ACM Int. Symp. MOBIHOC*, Sep. 2004, pp. 216–230.

[14] A. Sharma and E. M. Belding, "FreeMAC: Framework for multichannel MAC development on 802.11 hardware," in *Proc. ACM Int. Workshop Programmable Routers Extensible Serv. Tomorrow*, Aug. 2008, pp. 69–74.

[15] A. Asterjadhi, N. Baldo, and M. Zorzi, "A distributed network coded control channel for multihop cognitive radio networks," *IEEE Netw.—Special Issue on Networking Over Multihop Cognitive Networks*, vol. 23, no. 4, pp. 26–32, Jul./Aug. 2009.

[16] J. So and N. H. Vaidya, "A routing protocol for utilizing multiple channels in multihop wireless networks with a single transceiver," Univ. Illinois Urbana-Champaign, Urbana, IL, Tech. Rep., Oct. 2004.

[17] G. Wu, S. Singh, and T. C. Chiueh, "Implementation of dynamic channel switching on IEEE-802.11-based wireless mesh networks," in *Proc. ACM 4th Annu. Int. Conf. Wireless Internet*, 2008, pp. 39:1–39:9.

[18] W. Xu, T. Wang, and Y. Zhang, "Defending wireless sensor networks from radio interference through channel adaptation," *ACM Trans. Sens. Netw.*, vol. 4, no. 4, pp. 1–34, Aug. 2008.

[19] R. Chandra and P. Bahl, "MultiNet: Connecting to multiple IEEE 802.11 networks using a single wireless card," in *Proc. IEEE INFOCOM*, Mar. 2004, vol. 2, pp. 882–893.

[20] IEEE Std. 802.11 Working Group, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specs.: Higher Speed Physical Layer Extension in the 2.4-GHz Band*, Sep. 1999.

[21] IEEE Std. 802.11 Working Group, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specs.—Amend. 4: Further Higher Data Rate Extension in the 2.4-GHz Band*, Jun. 2003.

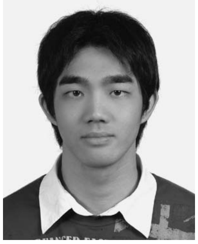
[22] IEEE Std. 802.11 Working Group, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specs.—Amend. 4: Enhancements for Higher Throughput*, Sep. 2007.

[23] IEEE Std. 802.11 Working Group, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specs.—Amend. 8: Medium Access Control (MAC) Quality-of-Service Enhancements*, Nov. 2005.

[24] Y. Li and A. M. Safwat, "Efficient deafness avoidance in wireless ad hoc and sensor networks with directional antennas," in *Proc. ACM Int. Workshop PE-WASUN*, Oct. 2005, pp. 175–180.

[25] R. R. Choudhury and N. H. Vaidya, "Deafness: A MAC problem in ad hoc networks when using directional antennas," in *Proc. IEEE ICNP*, Oct. 2004, pp. 283–292.

[26] Y. C. Tay and K. C. Chua, "A capacity analysis for IEEE 802.11 MAC protocol," *Wireless Netw.*, vol. 7, no. 2, pp. 159–171, Mar. 2001.
[27] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.
[28] H. S. Chhaya and S. Gupta, "Performance modeling of asynchronous data transfer methods of IEEE 802.11 MAC protocol," *Wireless Netw.*, vol. 3, no. 3, pp. 217–234, Aug. 1997.
[29] Realtek Semiconductor Corp. [Online]. Available: http://www.realtek.com.tw/products/
[30] Y. D. Lin, S. L. Tsao, S. L. Chang, S. Y. Cheng, and C. Y. Ku, "Design issues and experimental studies of wireless LAN mesh," *IEEE Wireless Commun.*, vol. 17, no. 2, pp. 32–40, Apr. 2010.
[31] Ixia, Inc. [Online]. Available: http://www.ixiacom.com/
[32] J. Nachtigall, A. Zubow, and J. P. Redlich, "The impact of adjacent channel interference in multiradio systems using IEEE 802.11," in *Proc. IWCMC*, 2008, pp. 874–881.
[33] C. M. Cheng, P. H. Hsiao, H. T. Kung, and D. Vlah, "Adjacent channel interference in dual-radio 802.11 nodes and its impact on multihop networking," in *Proc. IEEE GLOBECOM*, Nov. 2006, pp. 1–6.



Chia-Yu Ku (S'10) received the B.Sc. and M.S. degrees in mathematics and computer science from the National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 2005 and 2007, respectively. He is currently working toward the Ph.D. degree with the Department of Computer Science, NCTU.

From 2006 to 2009, he was with the Realtek–NCTU Joint Research Center, Hsinchu. Since 2009, he has been with the Network Benchmarking Laboratories (www.nbl.org.tw). His research interests include wireless mesh networking, multichannel multiradio, mobility management, integration of wireless networks, and performance evaluation.

Ying-Dar Lin received the B.Eng. degree in computer science and information engineering from the National Taiwan University, Taipei, Taiwan, in 1988 and the M.S. and Ph.D. degrees in computer science from the University of California at Los Angeles, in 1990 and 1993, respectively.

In 1993, he joined the faculty of the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, where he has been a Professor since 1999. During 2007–2008, he spent his sabbatical year as a Visiting Scholar with Cisco Systems, San Jose, CA. He is also the Founder and has been the Director of Network Benchmarking Laboratories (www.nbl.org.tw) since 2002. In 2002, he cofounded L7 Networks Inc., which addressed the content networking markets with the technologies of deep-packet inspection and was later acquired by D-Link Corporation. He is a coauthor (with R.-H. Hwang and F. Baker) of the textbook *Computer Networks: An Open Source Approach* (New York: McGraw-Hill, Feb. 2011). He is currently on the Editorial Boards of *Computer Communications* and *Computer Networks*. His research interests include the design, analysis, implementation, and benchmarking of network protocols and algorithms, quality of services, network security, deep-packet inspection, embedded hardware software codesign, wireless mesh, and peer-to-peer networking.

Dr. Lin is currently on the Editorial Board of the *IEEE Communications Magazine*, IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and IEEE COMMUNICATIONS LETTERS.

Shiao-Li (Charles) Tsao (M'04) received the Ph.D. degree in engineering science from the National Cheng Kung University, Tainan, Taiwan, in 1999.

From 1999 to 2003, he was with the Computers and Communications Research Laboratories, Industrial Technology Research Institute (ITRI), as a Researcher and a Section Manager. He was a visiting Professor with the University of Waterloo, Waterloo, ON, Canada, in the summer of 2007 and the Swiss Federal Institute of Technology, Zurich, Switzerland, in the summer of 2010. He is currently an Associate Professor with the Department of Computer Science, National Chiao Tung University (NCTU), Hsinchu, Taiwan. He has published more than 70 international journal papers and conference proceedings and is the holder of or has applied for 18 U.S. patents. His research interests include mobile communication and wireless networks, as well as embedded software and systems.

Prof. Tsao is the recipient of the Research Achievement Award from ITRI in 2000 and 2004, the Highly Cited Patent Award from ITRI in 2007, the Outstanding Project Award from the Ministry of Economic Affairs (MOEA) in 2003, and the Advanced Technologies Award from MOEA in 2003. He is also the recipient of the Young Researcher Award from the Pan Wen-Yuan Foundation, the Young Engineer Award from the Chinese Institute of Electrical Engineering in 2007, the Outstanding Teaching Award from NCTU, and the K. T. Li Outstanding Young Scholar Award from the Association for Computing Machinery Taipei/Taiwan Chapter in 2008.

Yuan-Cheng Lai received the Ph.D. degree from the National Chiao Tung University, Hsinchu, Taiwan, in 1997.

In August 1998, he joined the faculty of the Department of Computer Science and Information Science, National Cheng Kung University, Tainan, Taiwan. In August 2001, he joined the faculty of the Department of Information Management, National Taiwan University of Science and Technology, Taipei, Taiwan, where he has been a Professor since February 2008. His research interests include performance analysis, protocol design, wireless networks, and web-based applications.