INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2014; **27**:4407–4429
Published online 28 August 2013 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/dac.2622

An end-to-end channel allocation scheme for a wireless mesh network

Shiao-Li Tsao*,†, Jiun-Jang Su, Kuei-Li Huang, Yung-Chien Shih and Chien-Chao Tseng

Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan

SUMMARY

Co-channel interference seriously influences the throughput of a wireless mesh network. This study proposes an end-to-end channel allocation scheme (EECAS) that extends the radio-frequency-slot method to minimize co-channel interference. The EECAS first separates the transmission and reception of packets into two channels. This scheme can then classify the state of each radio-frequency-slot as transmitting, receiving, interfered, free, or parity. A node that initiates a communication session with a quality of service requirement can propagate a channel allocation request along the communication path to the destination. By checking the channel state, the EECAS can determine feasible radio-frequency-slot allocations for the end-to-end path. The simulation results in this study demonstrate that the proposed approach performs well in intra-mesh and inter-mesh communications, and it outperforms previous channel allocation schemes in end-to-end throughput. Copyright © 2013 John Wiley & Sons, Ltd.

Received 2 December 2012; Revised 3 May 2013; Accepted 2 July 2013

KEY WORDS: channel allocation; co-channel interference; multi-radio multi-channel; wireless mesh network

1. INTRODUCTION

The wireless mesh network (WMN) [1, 2] is one of the most promising technologies for fast and low-cost network deployment, and has attracted significant research from academia and industry [3]. Because of the increasing demands placed on WMNs, wireless network standards such as WiFi have incorporated the mesh access mode [4]. Unfortunately, a WMN suffers from a co-channel interference problem [5] when mesh nodes share the same wireless access channels. Therefore, the overall throughput of a WMN may not be able to increase substantially even with broadband physical layer technologies [6]. To reduce co-channel interference, researchers have proposed a number of solutions to allocate multiple non-overlapping channels to mesh nodes. Most of these studies have considered the co-channel interference between neighboring nodes, incorporated the transmit power control into dynamic channel allocation [7], maximized the total WMN throughput through the transmission control protocol (TCP) traffic control schemes [8–10], but they failed to address the end-to-end throughput and throughput guarantee issue in a WMN. The end-to-end throughput and throughput guarantee play key roles in real-time communication services such as voice over Internet protocol over WMNs [11].

This study proposes an end-to-end channel allocation scheme (EECAS) capable of providing throughput guarantees for end-to-end packet transmissions in WMNs. This approach models the

*Correspondence to: Shiao-Li Tsao, Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan.
†E-mail: sltsao@cs.nctu.edu.tw

Copyright © 2013 John Wiley & Sons, Ltd.

channel allocation as a directed graph and derives the solutions through set operations. The proposed approach can easily determine the channel allocation for an end-to-end communication path and simultaneously maintain the throughput. The results of this study demonstrate that the proposed mechanism improves channel utilization and achieves a better end-to-end throughput than other schemes.

The rest of the paper is organized as follows. Section 2 presents a summary of the related research on channel allocation in a WMN. Section 3 presents the main observation and contribution of this paper. Section 4 presents the proposed model and the channel allocation algorithm. Section 5 presents the NS-2 simulation results and performance comparisons with previous studies. Finally, Section 6 presents the conclusion.

2. RELATED WORKS

2.1. Centralized channel allocation

Most recent studies on channel allocation for multi-radio mesh networks tend to jointly solve channel allocation and routing problem. Centralized solutions [12–18] find the best combination of routes, channel assignments, and transmission schedules on all channels under a given network topology and traffic pattern. However, these optimization algorithms often fail to provide practical solutions for coordinating topology measurement and disseminating a channel assignment.

Previous studies [12, 13] assumed that the traffic demands are known beforehand and nodes can operate synchronously in a time-slotted mode. One study [12] presented a linear programming (LP) model to route the given traffic demands and maximize the system throughput subject to fairness constraints. This approach also considers the constraints on the number of radios and the sum of the flow rates on the same link in an interference range. The authors concluded that the problem is Non-deterministic Polynomial-time hard (NP-hard), and they solved the LP relaxation of the problem. Because the LP model may not always provide an optimal solution with a feasible channel assignment, the authors proposed an assignment algorithm to adjust the flows on the graph while ensuring feasible channel assignment. A post processing and flow scaling step further re-adjust the flow on this graph. Finally, a scheduling algorithm produces an interference-free link schedule.

Another study [13] formulated the traffic demands as a multi-commodity flow problem in which one among several different objectives can be defined. In addition to consider the same constraints in [12], this LP formulation uses time-indexed variables to solve the LP model and obtains a solution for the entire channel assignment, routing, and scheduling simultaneously.That study also proposed a heuristic channel scheduling and assignment algorithm on the basis of greedy coloring.

2.2. Distributed channel allocation

Distributed solutions that jointly solve channel allocation and routing problems appear in the literature [19]. Researchers have also proposed a number of solutions that focus on distributed channel allocation. Most of these methods assume an identical number of channels on all nodes [20–24] for better coordination, and relatively few dynamically allocate channels in a fully distributed fashion for asymmetric multi-radio WMNs.

The slotted seeded channel hopping (SSCH) method [25] was the very first radio-frequency-slot solution. The proposed EECAS also adopts this concept. The SSCH method implements a channel-hopping schedule and schedules packets for each channel on the basis of the seed knowledge. Nodes transmit the channel-hopping schedule to neighboring nodes and update the schedule on the basis of dynamic traffic conditions. The SSCH method is a decentralized protocol in which nodes can coordinate channel switching decisions in a distributed manner. However, SSCH schedules the channel hopping of a node to communicate with all of its neighboring nodes. This design does not consider an end-to-end issue in WMNs, and may cause redundant allocations. Thus, applying the radio-frequency-slot method to multi-radio WMNs may waste radio resources because of limited non-overlapping channels. One difference between the EECAS and SSCH is that the EECAS separates data and acknowledgement packets of transmissions to increase the efficiency of the channel utilization. Thus, the radio-frequency-slot concept can be applied to asymmetric multi-radio

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/dac

WMNs. The EECAS also allocates channels for end-to-end paths on demand, which reduces the channel waste on idle links.

Zhao *et al.* proposed the mesh distributed coordination function (MDCF), which also adopts the concept of radio-frequency-slot design and applied the MDCF to an IEEE 802.11 network [26]. Unlike previous radio-frequency-slot solutions, such as the SSCH, the MDCF considers an on-demand channel allocation; handles hidden stations, exposed stations, and capture problems; and offers a practical solution for multiple-channel WMNs. However, like other conventional radio-frequency-slot solutions, the MDCF does not separate the transmission and reception of packets into different channels, and therefore may not fully utilize radio resources because of intra-path and inter-path interferences.

The authors of [27] proposed a distributed channel assignment protocol in which one interface of each node is dedicated to a common channel. This protocol relies on the common channel across a WMN to ensure connectivity. Each node runs the distributed assignment protocol to find channels for its other radios. This study uses the multi-radio link quality source routing protocol [20] with the Weighted Cumulative Expected Transmission Time (WCETT) [20] metric as a routing protocol. The EECAS also reserves a common channel in exchanging control messages, but does not require a dedicated radio.The EECAS is also aware of channel usage within an interfered range. Thus, the EECAS can allocate channels without interference, use channels more efficiently, and achieve a higher throughput.

Hyacinth [28] is a distributed load-aware algorithm for channel assignment. This algorithm explicitly builds a spanning tree rooted at each gateway node. Each node separates its own radios into two groups and then independently chooses channels for radios in one group. Each node uses the channels in the group to communicate with its children. Channel assignment changes on a per-flow basis, as load conditions change in the network. However, this approach is difficult to apply to a large-scale mesh because of the complexity involved in assigning channels to radios. Moreover, this solution is designed for wireless Internet access in a WMN, and the algorithm works only for WMN gateways, whose connectivity graph must be a tree.

Routing over multi-radio access network (ROMA) [29], like Hyacinth, does not rely on a common channel to maintain network connectivity and optimizes channel assignments along routes between mesh nodes and a few gateways. A fundamental difference between ROMA and Hyacinth is that ROMA considers link losses and loss fluctuations. ROMA also uses the expected transmission time ETT [20, 30] metric for link measurement and extends the self-interference aware (SIM) [21] metric for route selection. Consequently, ROMA is aware of channel diversity for an end-to-end path. However, both of these methods fail to consider the influence of intra-mesh communications. These schemes build short-cut routes between nodes within an intra-mesh domain, and these short-cuts cause unexpected inter-path interference that affects the throughput.

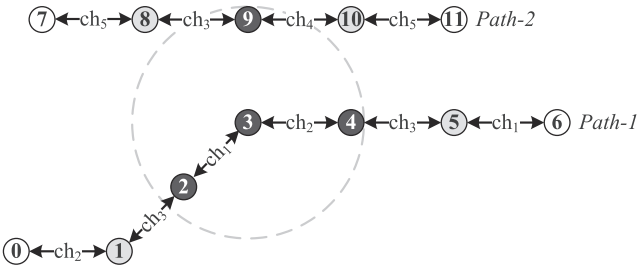
3. OBSERVATION AND MAIN CONTRIBUTION OF THE PROPOSED END-TO-END CHANNEL ALLOCATION SCHEME

The EECAS inherits the concept of the radio-frequency-slot design from the SSCH method. To incorporate the time division multiple access time slots into multiple frequency channels, the EECAS has a fixed and finer granularity of radio resources that can be easily and efficiently scheduled on the basis of a directed graph model. Neighboring nodes flexibly share the amount of radio resources according to their bandwidth needs, improving the utilization of radio resources on each frequency channel. The proposed EECAS extends the radio-frequency-slot method and considers a multi-radio design, end-to-end issue, and the separation of transmission and reception of packets on different channels. Although separating transmission and reception of packets on two different channels is not a new technique, this study shows that combining the two previously discussed techniques (i.e., radio-frequency-slot method and separation of transmission and reception of packets on different frequency-slot channels) can further reduce intra-path and inter-path interference between neighboring nodes.

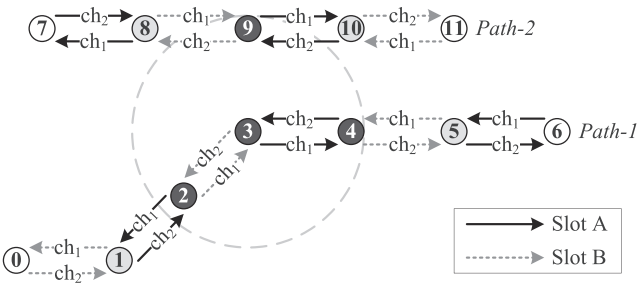
Figure 1 shows an example network consisting of 12 nodes, denoted from 0 to 11, and two routing paths, called *path-1* and *path-2*. Each node is equipped with two radio interfaces. In other words,

Copyright © 2013 John Wiley & Sons, Ltd.

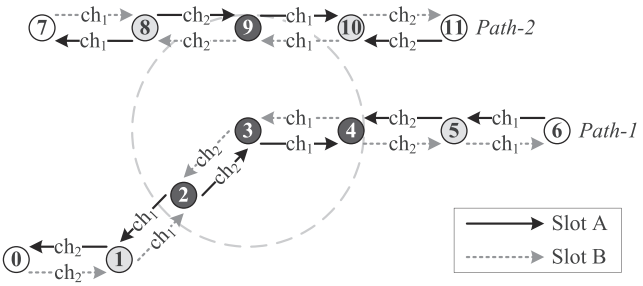
DOI: 10.1002/dac



(a) Conventional radio-frequency-slot scheme.



(b) Separating transmission and reception of packets.



(c) Separating transmissions on two different channels and slots.

Figure 1. Combining the radio-frequency-slot method and separation of transmission and reception of packets on different channels.

each node can be allocated up to two channels per slot. Figure 1a shows the results of applying the conventional radio-frequency-slot scheme. In this example, the system allocates channels for the hops of the two paths on the same slot. To avoid intra-path interference, the same channel cannot be allocated to three continuous hops of a path, so both *path-1* and *path-2* require three channels. To avoid inter-path interference, two neighboring nodes that belong to the two different paths cannot occupy the same channel. In Figure 1a, the in-used channels of node 3, which are ch_1 and ch_2, are different from the channels of node 9, which are ch_3 and ch_4. The conventional radio-frequency-slot scheme needs at least five non-overlapping channels. The number of required channels quickly grows as the network density or the number of radios increases.

Figure 1b shows that both *path-1* and *path-2* need two channels by employing the proposed concepts (i.e., separating transmission and reception of packets on different channels and scheduling the communications with different neighbors on different slots.) For example, at slot A, node 3 uses ch_1 to send packets and ch_2 to receive packets from node 4. Node 3 also uses ch_2 to send packets and ch_1 to receive packets from node 2 at slot B. Figure 1c also shows another alternative allocation. Separating transmissions on two channels and different slots can further improve the performance of end-to-end delay by minimizing slot switching. For example, in Figure 1c, the transmissions between node 2 and node 3 are separated on different slots. This approach can allocate various

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/dac

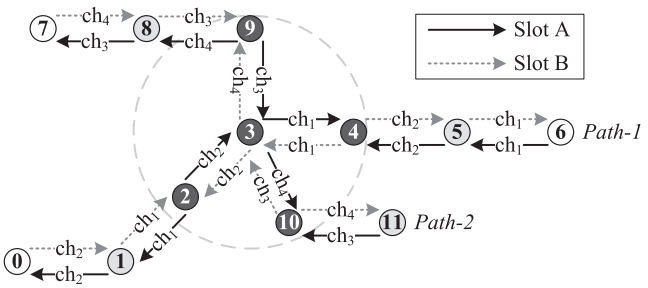


Figure 2. Observation of an asymmetric multi-radio architecture.

Table I. Symbols used in this paper.

Symbol	Description
V	A set of vertices, which represent nodes in a WMN
E	A set of edges, which represent logical channel between transmitter and receiver in a WMN
$G(V, E)$	A directed graph, which represents a WMN including nodes (V) and logical channels (E)
R_{com}	Radius of communication range of the node in a WMN
C^*	A set of all non-overlapping channels or *Cartesian product* of non-overlapping channels and time-slots
$D_{(u,v)}$	Distance between node u and node v
$S_u(k)$	State of channel slot k of node u, $\forall k \in C^*$
N_u	A set of neighbor nodes of node u
$ACS_{(u,v)}$	A set of channel slots that had been allocated on the edge (u, v)
$FCS_{(u,v)}$	A set of free channel slots on the edge (u, v)
$FCSC_{(u,v)}$	A set of free channel slot combinations on the edge (u, v)
$CCSC_{(u,v)}$	A set of candidate channel slot combinations on the edge (u, v), which is a subset of the $FCSC_{(u,v)}$

channels for the transmission from node 2 to node 3 and the transmission from node 3 to node 4 but schedule them on the same slot. Thus, packets can be quickly delivered from node 2 to node 4 without switching slots.

A node can communicate with all of its neighbors by applying the radio-frequency-slot scheme even if that node has fewer radios than its neighbors, and the number of radios on each node does not have to be equal. Thus, the proposed EECAS is applicable to an asymmetric multi-radio architecture, which is an efficient architecture for easing the hot-spot problem in a WMN. For example, in Figure 2, node 3 has four radios, and other nodes have two radios. There are two routing paths, and node 3 is the crossover node of the two paths. Node 3 functions as the mesh portal of a WMN. In this case, the system can allocate four different channels to node 3, and node 3 can simultaneously communicate with its neighbors. Therefore, the two paths do not have to share the bandwidth at the crossover node. The total number of channels in use is four. In summary, the hot-spot problem can be easily resolved by adding more radios when separating transmission and reception of packets in an asymmetric multi-radio WMN.

4. THE END-TO-END CHANNEL ALLOCATION SCHEME CHANNEL ALLOCATION MECHANISM

4.1. Definitions

The study presents a model of a WMN as a directed graph $G(V, E)$ and derives the solution of the channel allocation on the basis of set operations. Table I lists the symbols used in this paper. Assume that V is a set of vertices that represent all mesh nodes in a WMN, and u and v are mesh nodes in

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/dac

the set V. Then, define the relationships between u and v as

$$\begin{aligned} &\forall u, v \in V, \\ &D_{(u,v)} < R_{com} \Leftrightarrow (u,v) \in E \ \wedge \ (v,u) \in E. \end{aligned} \tag{1}$$

Equation (1) shows that if the distance between u and v, denoted as $D_{(u,v)}$, is less than the communication range of the mesh nodes, denoted as R_{com}, two directed edges, denoted as (u, v) and (v, u), exist in the graph $G(V, E)$. Each edge indicates a logical channel capable of directional communication between nodes. In addition, a sequence of edges indicates a directional path in a WMN.

The set of neighbors of a mesh node u, denoted as N_u, is defined as

$$\forall u \in V, \ N_u \equiv \{x \mid \forall x \in V, \ D_{(u,x)} < R_{com}\}. \tag{2}$$

Equation (2) shows that for any mesh node in V, denoted as x, if the distance between u and x is less than R_{com}, node x is included in N_u, and x is a neighbor of u. In other words, all nodes in the set N_u are within the signal coverage of u, and these nodes can directly communicate with u.

This study also defines the available logical channels in a multi-radio multi-channel WMN. A mesh node may have multiple radio interfaces. In addition, the transmission time can be divided into several time slots, like the time division multiple access approach. In each time slot, the radio interface can switch to a logical channel and communicate with peer nodes in the same channel. To consider the timing inaccuracies and various propagation delays of individual nodes, a guard period is usually introduced between two adjacent time slots. Therefore, perfect time synchronization is not necessary. Researchers have recently proposed several decentralized synchronization algorithms, such as [31, 32], that have a timing accuracy of less than 10 μs.

To synchronize time slots, each node in a network must send a data packet with the synchronous information to its neighbor nodes. The node also receives the data packet sent by its neighbor nodes, and then computes the propagation delay. According to [31], the slot differences converge rapidly, and they are all less than 10^{-6} after 15 iterations. Therefore, the time synchronization is scheduled at the parity channel slot of each cycle, and a new node just joining the network cooperates with other nodes after 15 cycles. Although this design may cause a long waiting time before channel slots become synchronized, it is feasible because WMN nodes do not frequently leave and join. The proposed approach uses uni-cast packets to exchange synchronous information. Therefore, the coordination overheads among nodes are $\sum_{i \in V} |N_i|$ messages per cycle, where $|N_i|$ is the number of neighbors of node i, and V is the set of all nodes in the network. In addition, the guard period is set to 10 μs to protect transmissions. Therefore, the overhead produced by the guard periods is 10 μs / 10 ms = 0.1% when the EECAS adopts a 10-ms channel slot.

The logical channel set can be expressed as the Cartesian product of non-overlapping channels and time slots. For example, assume that C denotes the set of non-overlapping logical channels and contains 1, 2, and 3 logical channels (i.e., $C = \{1, 2, 3\}$). Define S as the set of time slots containing slots a, b, c, and d. In other words, $S = \{a, b, c, d\}$. Thus, the new set of available logical channels can be denoted as $C^* = C \times S = \{\langle 1, a\rangle, \ \langle 2, a\rangle, \ ... \ \langle 3, d\rangle\}$,which includes 12 combinations. Element $\langle 1, a\rangle$ represents channel 1 at time slot a. For simplicity, refer to element $\langle 1, a\rangle$ as *channel slot* $\langle 1, a\rangle$.

Finally, consider two key terms: *directional logical channel* and *non-directional logical channel*. As mentioned previously, the proposed method separates packets in different directions and schedules them in different logical channels. For example, data and acknowledgement packets in a path between neighboring nodes are separated into two directional logical channels. On the other hand, a logical channel with data and acknowledgement packets together without separation is called a non-directional logical channel.

4.2. Channel allocation mechanism

The proposed EECAS includes two phases of operation. The first phase is called information maintenance, and consists of two operations: channel state table (CST) maintenance and exchange, and free channel slot (FCS) discovery. All mesh nodes periodically perform the procedures of the first

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/dac

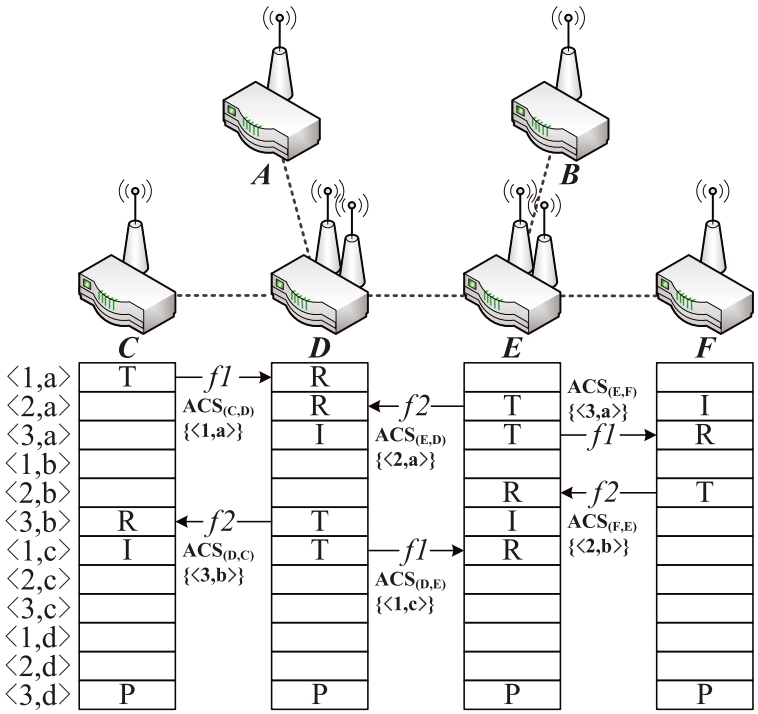


Figure 3. In-use channel awareness and soft-state associated with flows by applying the channel state tables (CSTs).

phase and can update information for the subsequent channel allocation. The concept behind the CST maintenance and exchange is similar to the distributed reservation protocol (DRP) [33]. The DRP, adopted by the WiMedia standard and applicable to WMNs [34], provides an infrastructure to exchange channel state information between neighboring nodes in a WMNS. Unlike the DRP, which exchanges information about whether the channel is in use, the EECAS classifies the channel into five conditions, and maintains the channel information of radio-frequency-slots. The second phase is called channel allocation and has two operations: free channel slot combination (FCSC) computing and channel decision. The second phase is performed on-demand, and is triggered when a communication is newly established or modified. The following examples provide an illustration of these procedures.

4.2.1. Channel state table maintenance and exchange. To find the channel slot in use, all mesh nodes must maintain a CST and the state of each channel slot. Each node must also broadcast its own CST to its neighbors, as Equation (2) shows. Figure 3 shows six mesh nodes, named A, B, ... F, where D and E are equipped with two radio interfaces. The channel slot set $C^* = \{\langle 1,a\rangle, \langle 2,a\rangle, \ldots \langle 3,d\rangle\}$ includes 12 available channel slots, and each node maintains its own CST. The channel slot state has five conditions: transmitting (T), receiving (R), interfered (I), free (F), and parity (P). Each node also keeps track of the lifetime of a channel slot in its own CST.

Figure 3 shows a data flow, called $f1$, from C to F, through D and E. Node C uses $\langle 1,a\rangle$ to send packets to D so that C marks the state of $\langle 1,a\rangle$ as 'T' in its CST and associates a lifetime $t_{\langle 1,a\rangle}$ to the channel slot. The lifetime is updated whenever packet is sent or received during the channel slot, and the channel slot is released if the lifetime expires. In other words, the lifetime associated with the channel slot is soft-state. Channel slot $\langle 1,a\rangle$ is also put into the set of allocated channel slots (ACSs) for the corresponding edge, denoted as $ACS_{(C,D)}$, indicating an in-use channel slot on the edge. Moreover, D marks the channel slot as 'R' in its CST because it receives packets from C through the channel slot.

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/dac

Equation (3) defines an interfered channel slot. Node u deems that channel slot k has interference, denoted as $S_u(k) = I$ and maintains the state in its CST, if at least one neighbor node x of node u is using the channel slot to send packets, denoted as $S_x(k) = T$, and node u is not the receiver. This condition is updated by sensing wireless carrier and/or after every CST exchanges. For example, A marks $\langle 1, a\rangle$ as 'I' in its CST because D uses the channel slot to send packets, and A is not the receiver. Finally, if a channel slot is not in the aforementioned states, the channel slot is marked as blank to represent its availability.

$$\forall u \in V,\ k \in C^*,\ S_u(k) = I \Leftrightarrow \exists x \in N_u,\ S_x(k) = T \tag{3}$$

To avoid channel conflict, a mesh node must know every channel slot's usage in the interference range. Thus, every node periodically broadcasts its own CST to its neighbors and stores the CSTs of all neighbors. Therefore, every node is aware of the interfered channels of its neighbors. This information can indicate channel slot usages of three hops away, and thus, every node can know the channel slot usage in the interference range. For example, if C knows that D deems $\langle 3, a\rangle$ to be interfered (Figure 3), then C surmises that the channel slot had been allocated within three hops' distance.

Finally, all nodes must reserve a parity channel slot for control message exchange. For example, in Figure 3, all nodes reserve channel slot $\langle 3, d\rangle$ and mark the channel slot as 'P' in their own CSTs to indicate the parity channel slot. Section 4.3 presents the details of this control mechanism.

4.2.2. Free channel slot discovery. To allocate channel slots without co-channel interference, a node must identify FCSs that have not been allocated. A mesh node performs the discovery procedure whenever it receives the CST of its neighbor. Solving channel allocation problem with directional logical channels can be viewed as allocating channel slots to directed edges. Therefore, a node must discover FCSs from its outgoing edges. Equation (4) presents the conditions of FCSs with directional logical channels. Assume that there are two neighboring nodes, called u and v, so that two directed edges, denoted as (u, v) and (v, u), exist. For the outgoing edge (u, v) of u, node u deems channel slot k as free on the edge and puts the channel slot into the $FCS_{(u,v)}$. This denotes the set of FCSs on the edge (u, v), if the following three conditions are satisfied: the state of k in the CST of u is free or interfered, denoted as $S_u(k) = F/I$; the state in the CST of v is free, denoted as $S_v(k) = F$; and the state in CSTs of all neighbors of u is not receiving packets. As mentioned earlier, node u has its own CST and the CSTs of its neighbors, and can therefore obtain $FCS_{(u,v)}$ and maintain the information.

$$\begin{aligned} &\forall u, v \in V,\ k \in C^*,\ k \in FCS_{(u,v)} \Leftrightarrow \\ &S_u(k) = F/I\ \wedge\ S_v(k) = F\ \wedge \\ &\quad \forall x \in N_u,\ S_x(k) \neq R \end{aligned} \tag{4}$$

Figure 4 shows a simple example of the FCS discovery with directional logical channels, where A and D are neighboring nodes such that A periodically receives D's CST. When A receives D's CST, A performs the FCS discovery algorithm (Figure 5) to update FCS of the outgoing edge (A, D). The result of $FCS_{(A,D)}$ is $\{\langle 1, b\rangle,\ \langle 2, b\rangle,\ \langle 2, c\rangle,\ \langle 3, c\rangle\}$, indicating that allocating any one of these channel slots to the edge (A, D) does not affect existing flows. Similarly, D can update its outgoing edges $FCS_{(D,A)}$, $FCS_{(D,B)}$, $FCS_{(D,E)}$, and so on.

In addition, channel slot $\langle 1, a\rangle$ is not included in $FCS_{(E,B)}$ because E has two radio interfaces, and the two interfaces have been assigned to other channel slots at time slot a. The node has no interface to be assigned at the same time slot. Thus, channel $\langle 1, a\rangle$ is removed from $FCS_{(E,B)}$. The proposed FCS discovery algorithm shown in Figure 5 can also handle this case.

This study also demonstrates how to obtain the FCSs when the conventional non-directional logical channel is applied. In this case, the channel allocation problem is to map channel slots to the edges of both directions between neighboring nodes. This is because the conventional non-directional logical channel assumes no direction on the link, and the communication packets may be exchanged in both directions. Therefore, the edges in both directions between two nodes are occupied if the link is assigned to the nodes. For example, node u deems channel slot k as free on

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/dac

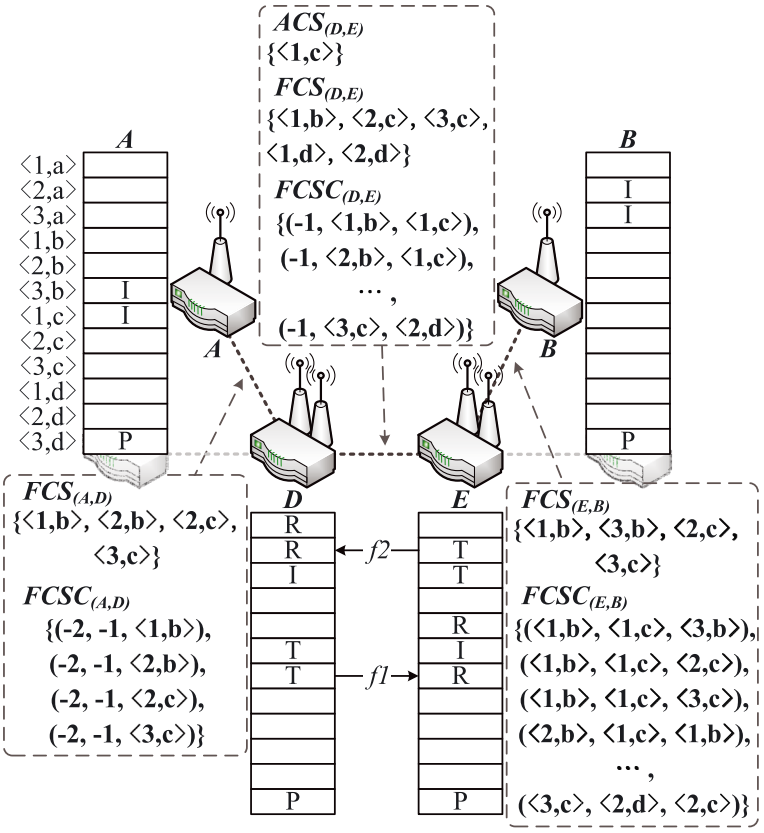


Figure 4. An example of free channel slot (FCS) discovery and free channel slot combination (FCSC) computing.

Algorithm 1 FCS discovery

```
1: FCS_(src,dst) ⇐ {∅}
2: for each c ∈ channel slot set C* do
3:   if src and dst has at least one interface to be allocated the channel slot c then
4:     if (S_src(c) ="F" or S_src(c) ="I") and S_dst(c) ="F" then
5:       NonRecv_(src,dst) ⇐ 'true'
6:       for each n ∈ neighbor set N_src do
7:         if S_n(c) ="R" then
8:           NonRecv_(src,dst) ⇐ 'false'
9:         end if
10:      end for
11:      if NonRecv_(src,dst) = 'true' then
12:        FCS_(src,dst) ⇐ FCS_(src,dst) ∪ {c}
13:      end if
14:    end if
15:  end if
16: end for
```

Figure 5. The free channel slot discovery algorithm.

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/dac

Algorithm 2 FCSC computing

```
1: FCSC_(nbr,src) ⇐ the FCSC that is carried by FREQ and is sent from neighbor node nbr
2: FCSC_(src,dst) ⇐ {∅}
3: if ACS_(src,dst) ≠ {∅} and ∃a ∈ACS_(src,dst) satisfy the flow's bandwidth requirement then
4:   if src is the flow's source node then
5:     FCSC_(src,dst) ⇐ FCSC_(src,dst) ∪ {(−2, −1, a)}
6:   else
7:     for each (c1, c2, c3) ∈ FCSC_(nbr,src) do
8:       if c2 ≠ a and c3 ≠ a then
9:         FCSC_(src,dst) ⇐ FCSC_(src,dst) ∪ {(c2, c3, a)}
10:      end if
11:    end for
12:  end if
13: else
14:  for each c ∈ FCS_(src,dst) do
15:    if src is the flow's source node then
16:      FCSC_(src,dst) ⇐ FCSC_(src,dst) ∪ {(−2, −1, c)}
17:    else
18:      for each (c1, c2, c3) ∈ FCSC_(nbr,src) do
19:        if c2 ≠ c and c3 ≠ c then
20:          FCSC_(src,dst) ⇐ FCSC_(src,dst) ∪ {(c2, c3, c)}
21:        end if
22:      end for
23:    end if
24:  end for
25: end if
```

Figure 6. The free channel slot combination computing algorithm.

both edges (u, v) and (v, u), if the following conditions are satisfied: k is free in the CSTs of both u and v (denoted as $S_u(k) = F$ and $S_v(k) = F$), and none of k in the CSTs in all neighbors of u and v are receiving packets. Thus, channel slot k is included in both $FCS_{(u,v)}$ and $FCS_{(v,u)}$. In addition, nodes u and v must exchange information for FCS discovery because neither of them have sufficient information to know the channel states in the CSTs of another's neighbors.

$$\begin{aligned} &\forall u, v \in V,\ k \in C^*,\ k \in FCS_{(u,v)} \Leftrightarrow \\ &S_u(k) = F\ \wedge\ \forall x \in N_u,\ S_x(k) \neq R \wedge \\ &S_v(k) = F\ \wedge\ \forall y \in N_v,\ S_y(k) \neq R \end{aligned} \tag{5}$$

4.2.3. Free channel slot combination computing. The FCS discovery method can individually allocate channel slots for different edges and avoid inter-path interference. However, the FCS discovery method has not yet considered the end-to-end issue. Addressing the end-to-end issue can ensure that all edges along a routing path from source to destination can be successfully allocated with sufficient bandwidth for a specific flow. To guarantee the end-to-end throughput, all channel slot mapping of the nodes along a path and the channel slots for the flow must be determined to provide sufficient resources. The FCSC computing derives all channel slot mapping combinations for a flow's path, and is triggered when a node initiates a new data flow or modifies an existing flow. This procedure is performed node-by-node from the flow's source to the flow's destination. In other words, the nodes along the routing paths must perform the FCSC computing algorithm individually to derive channel allocations. Figure 6 shows this algorithm.

$$FCSC_{(A,D)} \equiv \left\{(-2\ -1\ c) \mid \forall c \in FCS_{(A,D)} \quad \vee c \in ACS_{(A,D)}\right\} \tag{6}$$

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/dac

$$FCSC_{(D,E)} \equiv \left\{Comb(p,c) \mid \forall p \in FCSC_{(A,D)} \quad \wedge (\forall c \in FCS_{(D,E)} \quad \vee c \in ACS_{(D,E)})\right\} \quad (7)$$

$$Comb((c_1\ c_2\ c_3)\ c_4) = \begin{cases} (c_2\ c_3\ c_4), & if\ c_2 \neq c_4\ and\ c_3 \neq c_4 \\ \emptyset, & otherwise \end{cases} \quad (8)$$

Figure 4 shows an example illustrating the FCSC computing method. Assume that A initiates a flow, called $f3$, from itself to B that triggers the FCSC computing procedure and then uni-casts a flow request (FREQ) message that carries the current outgoing edge's FCSC to the destination B. Node A must determine which combinations are available on the edge (A, D) and put these combinations into the corresponding set $FCSC_{(A,D)}$. A channel slot mapping combination for an edge of a flow's path is defined as a three-tuple that states available channel slot mappings for the neighboring three edges along the path. This definition ensures that any consecutive three edges of a flow's path are not allocated to the same channel slot, avoiding intra-path interference. Equation (6) shows the conditions of the combination on the first edge. Node A may pick in-use channel slots from $ACS_{(A,D)}$ if the remaining bandwidth of the channel slots satisfies the bandwidth requirement of the flow, or node A may collect all FCSs from $FCS_{(A,D)}$. These channel slots, denoted as c, are then placed on the third channel slot mapping of combinations to indicate available channel slots on edge (A, D). Because node A is the source of flow $f3$, the first and the second channel slot mappings of combinations are not necessary. In the proposed design, node A places two constants, '-2' and '-1', on the first and the second channel slot mapping, respectively, to indicate that no channel slot is needed. In this example, $ACS_{(A,D)}$ is empty, allowing A to collect channel slots only from $FCS_{(A,D)}$. The resulting $FCSC_{(A,D)}$ (Figure 4) is carried by the FREQ message, which is sent to the next node.

Upon receiving the FREQ message, node D also performs the FCSC computing algorithm to derive the second edge's combinations $FCSC_{(D,E)}$ and then puts the combinations into the FREQ message. In (7), node D may also pick channels from $ACS_{(D,E)}$ or $FCS_{(D,E)}$ and then perform the comb function to ensure that all combinations of the selected channel slots and the received FCSC satisfy the definition of channel slot mapping combination. In this example, assume that in-use channel slot $\langle 1, c\rangle$ in $ACS_{(D,E)}$ satisfies the bandwidth requirement of the flow and D picks the channel slot. Then, channel slot $\langle 1, c\rangle$ and the combinations in $FCSC_{(A,D)}$ are fed into the comb function to derive $FCSC_{(D,E)}$.

To avoid the intra-path interference, the same channel slot cannot be allocated to three consecutive edges of a flow's path. The comb function, shown in Equation (8), ensures that the second and third mappings of the input combination, denoted as c_2 and c_3, are not equal to the input channel slot, denoted as c_4.

Note that the first mapping, denoted as c_1, of the input combination is not significant because the corresponding edge of c_1 is beyond the interference range of the current edge. In this example, only channel slot $\langle 1, c\rangle$ and combinations in $FCSC_{(A,D)}$ are fed into the function, and Figure 4 shows the result $FCSC_{(D,E)}$. In addition, the third mapping of all combinations in $FCSC_{(D,E)}$ is $\langle 1, c\rangle$, and the channel slot is allocated to the edge (D, E). This indicates that multiple flows sharing the same edges may be aggregated into the same channel slot on the common edges. Furthermore, a node may also collect all free channel slots from the FCS to derive the FCSC. For example, because $ACS_{(E,B)}$ is empty, node E can only collect channel slots from $FCS_{(E,B)}$ and derive $FCSC_{(E,B)}$.

4.2.4. Channel decision. When the destination receives a FREQ message carrying the last edge's FCSC, the FCSC computing procedure is completed. The destination then triggers the channel decision procedure and uni-casts a flow reply (FREP) message back to the source to confirm the resource allocation. The channel allocation of the edges in a path is confirmed in a reversed order. Several channel allocation choices may satisfy a connection request. A simple decision policy for the channel allocation is to randomly pick a combination from the set of candidate channel slot combinations (CCSC), which is a subset of the FCSC. The last edge's CCSC is initially equal to the edge's FCSC. For example, the last edge's $CCSC_{(E,B)}$ is equal to the edge's $FCSC_{(E,B)}$, and node B randomly picks a combination such as $(\langle 1, b\rangle, \langle 1, c\rangle, \langle 3, b\rangle)$ from $CCSC_{(E,B)}$, denoted as $fcs_{(E,B)}$. Node B

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/dac

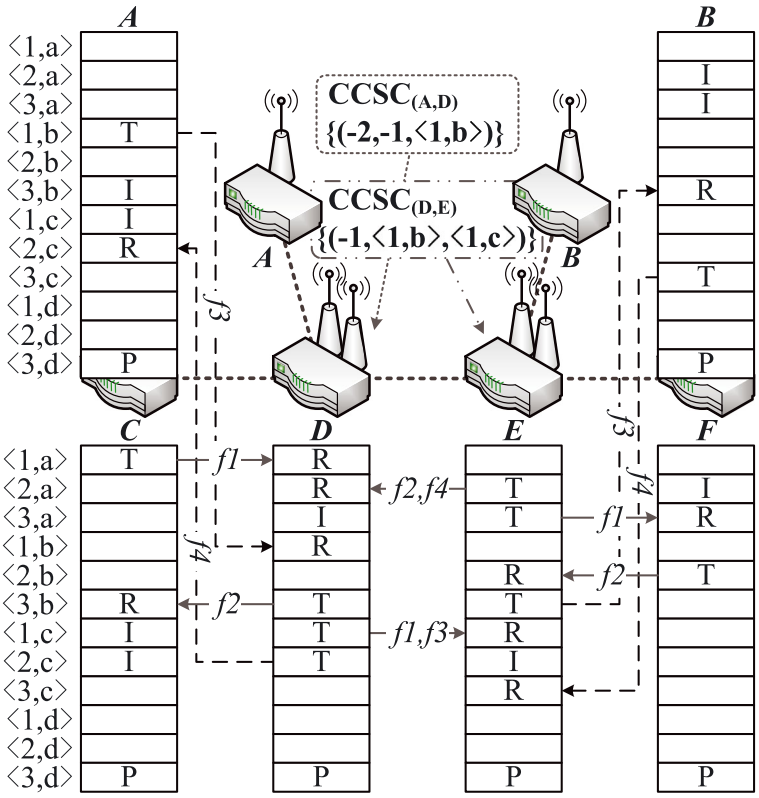


Figure 7. An example of end-to-end channel allocation scheme channel allocation.

sets channel slot $\langle 3, b\rangle$ as the receiving state in its CST and uni-casts the FREP message that carries the decision back to the source.

Upon receiving the FREP message, each node along the path configures the channel slot as the transmitting state and records the bandwidth requirement in its CST. For example, E configures channel slot $\langle 3, b\rangle$ as the transmitting state in its CST. As Figure 7 shows, flow $f3$ has been established on edge (E, B). One edge's channel decision determines the subsequent edge's CCSC because a combination designates the current edge's mapping and states mappings of the next two edges along the reversed path. For example, node E must derive $CCSC_{(D,E)}$ before it makes the decision for edge (D, E). $fcs_{(E,B)}$ carried by the FREP message determines which channel slot combinations are included in $CCSC_{(D,E)}$ (Figure 7). A combination in $FCSC_{(D,E)}$ is also included in $CCSC_{(D,E)}$ if the second and third mappings of the combination are equal to the first and second mappings of $fcs_{(E,B)}$. In this example, only one combination $(-1, \langle 1, b\rangle, \langle 1, c\rangle)$ satisfies this condition. This restriction indicates that node E must decide only the first mapping of combinations in $CCSC_{(D,E)}$ because other mappings have been decided.

Finally, when the source receives the FREP message, the channel decision procedure and the second phase are completed, and the source can send the flow's packets. As Figure 7 shows, flow $f3$ has been established from A to B, and flow $f4$ may be established to transmit $f3$'s acknowledgement packets. In addition, flows $f1$ and $f3$ share the channel slot $\langle 1, c\rangle$ in their common edge (D, E). This indicates that the EECAS allows multiple flows to share or contend the capacity of a channel.

4.3. Control mechanism

4.3.1. Dedicated control radio. Arranging a dedicated control radio for each node in the network is the simplest implementation of a control mechanism. In this setup, all nodes configure a dedicated

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/dac

radio and always switch to the common channel and run the carrier sense multiple access/collision avoidance on the radio. Therefore, control messages and synchronous information can be delivered over the radio. However, this dedicated radio approach is not efficient because the radio may be idle most of the time.

4.3.2. Parity channel slot. An alternative approach of exchanging control messages and synchronous information is that all nodes reserve a common parity channel slot on each cycle (like SSCH) and all nodes switch to a common channel during the slot. Therefore, nodes run the carrier sense multiple access/collision avoidance on the parity channel slot to avoid collision. In addition, we can reserve multiple parity channel slots within a cycle to improve performance in delivering of control messages. The overhead of parity channel slots of node x is $|PCS|/(|C^*| \times I_x)$, where $|PCS|$ is the number of parity channel slots per cycle, $|C^*|$ is the total number of channel slots per cycle, and I_x is the number of radio interfaces equipped on node x.

4.4. Recovery mechanism

In the channel decision state, some nodes may select unsuitable channel slots because of inconsistent CSTs. The proposed method uses a simple recovery mechanism. If some nodes along the allocating path detect an unsuitable decision, the decision node is notified and must select new channel slots. If the source node does not receive an FREP message and the FREQ is timeout, the source node resends the FREQ message and triggers the FCSC computing procedure again.

4.5. Choosing routes

4.5.1. Link metric. The most popular link metric is expected transmission count (ETX) [35, 36] and its extension ETT [20, 30]. The ETX metric measures the expected number of transmissions, including retransmissions, required to send a packet on a link. The derivation of ETX starts with measurements of the underlying packet loss probability in both forward and reversed directions, denoted as p_f and p_r, and then calculates the expected number of transmissions. Let p denote the probability that a packet transmission has failed (i.e., $p = 1-(1-p_f)\times(1-p_r)$). Let $s(k)$ denote the probability that the packet is successfully delivered after k attempts (i.e., $s(k) = p^{k-1}\times(1-p)$). Finally, we can derive the expected number of transmission as

$$ETX = \sum_{k=1}^{\infty} k \times s(k) = \frac{1}{1-p}. \tag{9}$$

Although the ETX metric performs better than the shortest path routing, it does not necessarily select efficient routes in an environment with various data rates or multiple radios. The ETT metric addresses this issue by taking the link bandwidth into account. Let P denote the size of the packet (for example, 1024 bytes), and let B denote the bandwidth (raw data rate) of the link. Then,

$$ETT = ETX \times \frac{P}{B}. \tag{10}$$

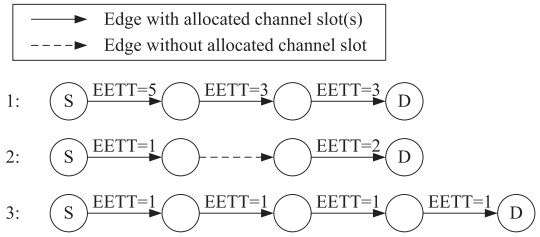
To calculate the transmitting time of an edge in the proposed EECAS, modify the conventional ETT metric and define the edge ETT(EETT) metric. For an edge (u, v), define $EETT_{(u,v)}$ as follows:

$$EETT_{(u,v)} = \frac{\sum_{i \in ACS_{(u,v)}} ETT_i}{|ACS_{(u,v)}|}. \tag{11}$$

The proposed scheme uses the ETT to calculate the transmitting time of an ACS. To calculate the value of channel slot i on edge (u, v), denoted as ETT_i, it is necessary to know both the loss rate of channel slot i and that of its reversed edge (ACS(s) on edge (v, u)). The rate can be approximated by sending a packet on the specified channel slot and gathering an acknowledgement from the reversed link. Therefore, according to Equation (11), the EETT is the average transmission time of the ACSs on a specified edge.

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/dac



Path	Sum	H	M (β=0.1)	M (β=0.5)	M (β=0.9)
1	11	0	9.9	5.5	1.1
2	3	1	2.8	2.0	1.2
3	4	0	3.6	2.0	0.4

Figure 8. Example of route stretch by applying the end-to-end channel allocation scheme's path metric.

4.5.2. Path metric. WECTT [20] and SIM [21] are two path metrics that resolve the tension of choosing between shorter paths with lower total transmission overhead (e.g., the sum of ETTs) and longer paths consisting of better links. This study does not consider the channel diversity on a path to minimize intra-path interference because the EECAS can allocate channels for paths without interference. Therefore, the EECAS modifies these two path metrics, accounts for the overhead of channel allocation, and calculates the path metric as follows:

$$M = (1-\beta) \times \sum_{j \in path} EETT_j + \beta \times H. \tag{12}$$

The path metric of the EECAS (M) is a linear combination of path overhead and allocation overhead with a parameter β, which implies a trade-off between them. The path overhead is approximated by the sum of EETTs along a routing path, and the allocation overhead (H) is the sum of edges without the ACS.

Figure 8 shows an example. Consider three possible paths from the source, S, to the destination, D. Figure 8 shows the EETTs on the edges of these paths and M values for $\beta = 0.1$, $\beta = 0.5$,and $\beta = 0.9$.

The first path is one of the shortest paths, and all edges along this path have ACS(s). Unfortunately, its sum of EETTs, which is 11, is higher than the sums of other paths. This implies that this path suffers from heavy transmission loads or that multiple flows share the same resources.

Next, consider the second and third paths. The second path is another shortest path, and its sum of EETTs is the smallest one, but it contains an edge without a channel slot. If $\beta = 0.1$, the overhead of allocating channel slot for those edges without ACSs is undervalued. Therefore, this path was chosen. On the other hand, even though the length of the third path is longer than that of the second path, the edges on the third path all have ACSs, and the sum of EETTs is close to the sum of the second path. In this case, if β is more than 0.5, which implies that the overhead of allocating channel slots is taken into account, the third path is better than the second path. Section 5 presents a further investigation of system performance at different β values.

5. SIMULATION RESULTS

The simulations were conducted to evaluate the performance improvements of the proposed EECAS. The NS-2 simulator was used, and various network configurations were considered. The simulations assume IEEE 802.11a as the physical layer technology, and the raw data rate is set to 54 Mbps. Previous research [37] indicates that the channel switching delay of an IEEE 802.11a

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/dac

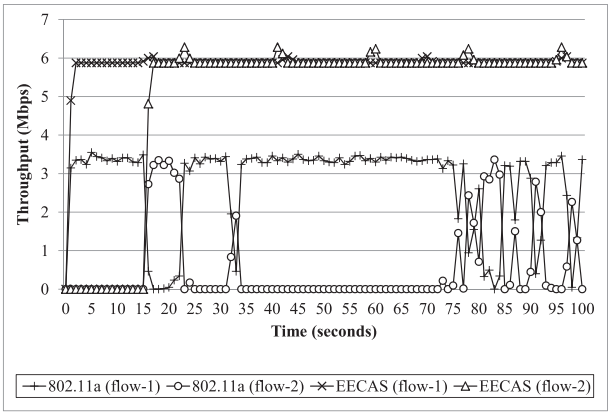


Figure 9. Throughput comparison between the end-to-end channel allocation scheme and the conventional IEEE 802.11a.

interface is between 40 to 80 μs. The authors of [38] further proposed a channel assignment reconfiguration model to quantitatively evaluate network throughput degradation and channel switching overhead. This study assumes a fixed overhead of 40 μs for channel switching, and future research will consider a more complete cost model, such as that in [38].

5.1. Comparison between the end-to-end channel allocation scheme and the conventional IEEE 802.11a

The first simulation in this study evaluates the throughput improvement by comparing the proposed EECAS against the conventional IEEE 802.11a. This simulation configures 10 mesh nodes, each with only one radio interface as a 2-row-by-5-column mesh network, and every node can hear neighboring nodes. Two 4-hop flows, which do not share any node or edge, were established. The EECAS uses a 10-ms channel slot, and therefore, the switching overhead is 40 μs / 10 ms = 0.4%. The two flows, called *flow-1* and *flow-2*, were then sequentially activated at 0 and 15s, respectively, and the two flows generated constant bit rate (CBR) user datagram protocol (UDP) packets.

Figure 9 shows the variation of the maximal throughput by applying the EECAS to IEEE 802.11a and the conventional IEEE 802.11a. The horizontal axis of this figure indicates the simulation time in seconds, and the vertical axis indicates the throughput in megabit per second. The maximal throughput with the conventional IEEE 802.11a infrastructure mode (i.e., one-hop) is approximately 13 Mbps, but an IEEE 802.11a WMN encounters serious interferences. For an IEEE 802.11a WMN, the throughput of *flow-1* drops to 3.4 Mbps because of intra-path interference. Moreover, *flow-1* and *flow-2* suffer from inter-path interference such that the two flows influence each other and cannot maintain a stable throughput.

The EECAS allows mesh nodes to use various channels. Figure 9 shows the maximal throughput of both *flow-1*, and *flow-2* is approximately 5.9 Mbps when applying the EECAS. A single radio mesh node requires approximately half of the time to receive and half of the time to forward packets (i.e., the ideal throughput is approximately 6.5 Mbps). These results show that the EECAS introduces an overhead of approximately 9.23% in the parity channel slot, channel switching delay, and guard period. The throughput of the EECAS is more stable than that of the conventional IEEE 802.11a.

Figure 10 considers the results after the 15th second of the first simulation and illustrates the complementary cumulative distribution function of the throughput of *flow-1* and *flow-2*. Figure 10 shows that the EECAS ensures a constant throughput of approximately 6 Mbps for both *flow-1* and *flow-2*, but the throughputs of *flow-1* and *flow-2* vary when the conventional IEEE 802.11a is employed. The EECAS achieves a higher throughput and provides a better throughput guarantee than the conventional IEEE 802.11a.

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/dac

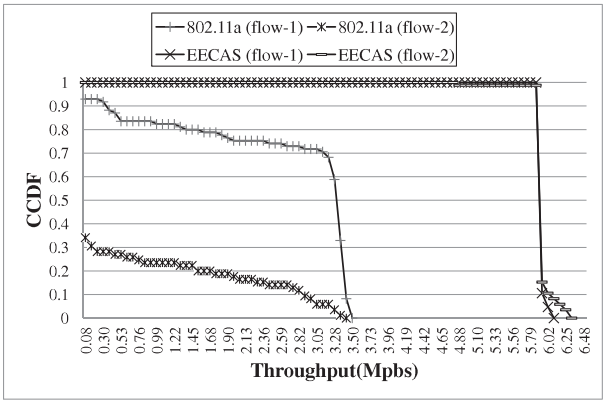


Figure 10. The complementary cumulative distribution function of the simulation results after the 15th second shown in Figure 9.

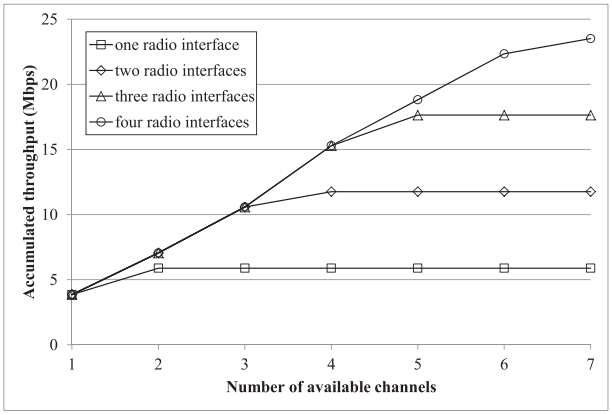


Figure 11. Effects of varying available radio channels on the upper bound of capacity.

5.2. The effect of available resources

Figure 11 shows the effects of varying the number of available radio channels on the upper bound of the capacity. The *x*-axis of this figure indicates the number of available non-overlapping channels, and the *y*-axis indicates the aggregated throughput in megabit per second, which is the maximum throughput a node can send or receive. These simulations were conducted on a 24-node grid network, and the maximum hop-count between any two nodes was limited to four hops. The number of radio interfaces per node was also varied to evaluate the effect of the number of radio interfaces on channel utilization efficiency. The aggregated throughput increases monotonically with the number of available non-overlapping channels when the number of radio interfaces per node remains constant. When each node has one radio interface, the aggregated throughput becomes saturated at approximately two channels. When the number of radio interfaces on each node increases to four, the node can use up to seven channels before its performance saturates.

5.3. Single flow throughput

To ensure a fair comparison, this study evaluates these approaches with similar assumptions and objective functions. According to recent surveys [39–41], Hyacinth [28] and ROMA [29] are the most representative-distributed channel assignment protocol for WMNs. Therefore, this study investigates the performance differences between the proposed scheme and these two protocols.

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/dac

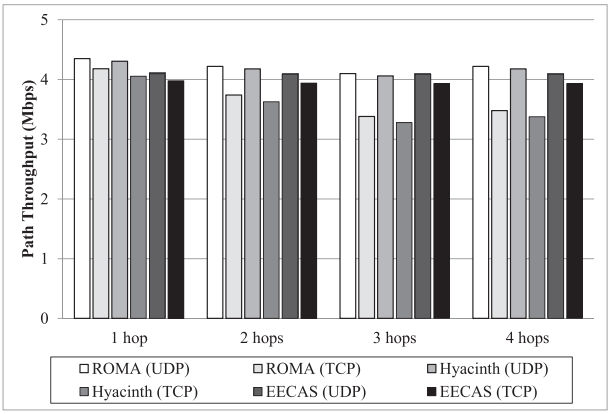


Figure 12. Throughput of paths under various hop-counts.

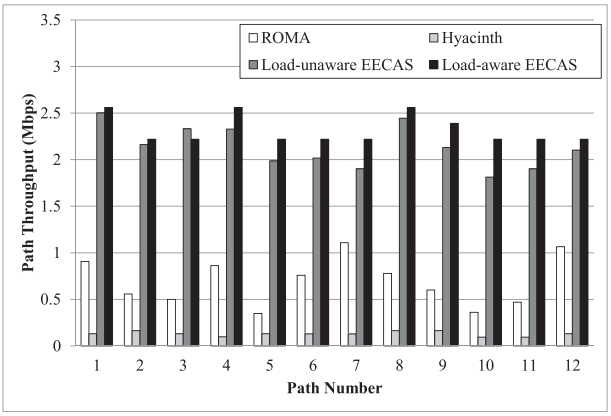


Figure 13. Throughput of multiple randomly chosen simultaneous flows.

Figure 12 shows the throughput of paths under various hop-counts for the EECAS, ROMA, and Hyacinth. Simulations were conducted on a 24-node dual-radio grid network. Each run of the simulation randomly assigned one of the 24 nodes as the gateway, and initiated a UDP flow from the gateway to each of 23 non-gateway nodes, one at a time. The request to send/clear to send was enabled, and 10 runs were conducted. In this configuration, ROMA and Hyacinth generated similar solutions. They both built a spanning tree rooted at the gateway and then assigned channels for links with various hop counts. In the single-flow examination, only one flow is active at a time. Thus, ROMA and Hyacinth did not suffer intra-path and inter-path interference, and performed well. This simulation showed a 6% degradation in the EECAS compared with ROMA because of protocol overheads, such as channel switching delay and guard period.

The same single-flow simulation was repeated to evaluate the throughput of TCP flows. Figure 12 shows that TCP flows in the EECAS achieved only marginally lower throughput compared with the UDP flows, even for longer paths. For example, in 4-hop paths, the TCP throughput is 3.93 Mbps, which is 4% lower than the UDP throughput of 4.09 Mbps.

5.4. Multiple flows throughput

The simulation of this study then measured the throughput of multiple randomly chosen simultaneous flows. These simulations were also conducted on a 24-node dual-radio grid network. Each run divided the 24 nodes into 12 source-destination pairs, and started a UDP flow on each pair. For path decision, two configurations were applied: (i) the EECAS's metric (load-aware), $\beta = 0.5$; and (ii) shortest-path only (load-unaware). There were 12 runs in total. Figure 13 shows the path throughput

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/dac

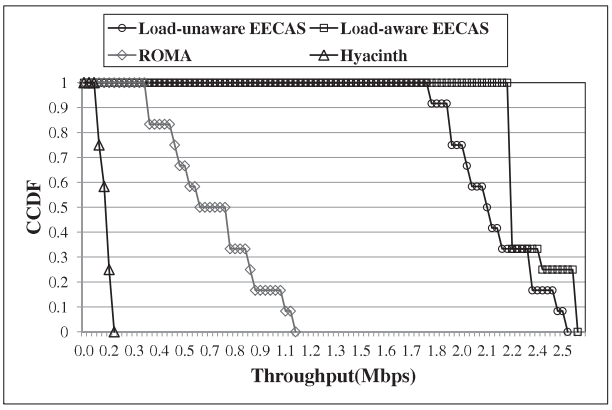


Figure 14. The complementary cumulative distribution function of throughputs of flows in a wireless mesh network.

of each path with the two configurations, and compares the performances of the EECAS, ROMA, and Hyacinth in this scenario.

Most of the paths with the load-aware EECAS achieved a higher performance than the load-unaware EECAS. Applying the EECAS metric caused a longer path or a path sharing bandwidth from other paths, causing lower performance in particular paths, such as path 3 in Figure 13. However, the network throughput by employing the load-aware EECAS reached 2.32 Mbps, representing an 8% improvement compared with the load-unaware EECAS. In addition, none of the paths starved when either the load-aware or load-unaware EECAS is applied.

To compare with ROMA and Hyacinth, we randomly choose a node to act as the gateway, and let both ROMA and Hyacinth build a spanning tree rooted from the gateway. As Figure 13 shows, the variations of path throughput with the EECAS, ROMA, and Hyacinth are dissimilar even though they have same source-destination pairs. This is primarily because a channel allocation mechanism dominates routes for these pairs. With Hyacinth, most of the routes travel through the gateway, causing serious interference among these routes. Thus, path throughput decreases significantly. ROMA allocates channels for all nodes residing on the same routing level on the basis of the spanning tree and preserves cross links between paths. Thus, most source-destination pairs can find shortcut routes, which helps improve path throughput. However, neither ROMA nor Hyacinth are feasible in this intra-mesh communication scenario because of their serious inter-path interference.

Figure 14 further illustrates the complementary cumulative distribution function of the throughputs of flows in a WMN. Figure 14 depicts that Hyacinth can provide a near constant throughput of flows in a WMN, but it offers the lowest performance compared with other approaches. The load-aware EECAS method achieves the highest throughput among all methods and provides a quite good throughput guarantee for flows in a WMN.

This study also investigates the effects of various percentages of intra-mesh communications on average path throughput. This simulation set up 12 paths and varied percentages of intra-mesh and outgoing paths. The outgoing paths mean the communications from nodes to Internet through the gateway. As Figure 15 shows all three schemes can achieve approximately 0.7 Mbps in average path throughput when all paths are outgoing paths. In other words, all paths contend for gateway resources. As the percentage of intra-mesh communications increases, the EECAS, ROMA, and Hyacinth perform differently. All three schemes should improve performance in path throughput by reducing contention for the same resource, but ROMA and Hyacinth fail to achieve the expected results because of inter-path interference. Although ROMA can route packets with shortcut paths, these paths still cause interference such that path throughput is inefficient. The EECAS guarantees no inter-path interference, producing fewer contentions for the same resource and improving path throughput. Figure 15 also shows the standard deviation of the throughputs of flows. Although Hyacinth provides the lowest standard deviation of throughput compared with ROMA and the EECAS, it achieves the lowest performance among the three approaches.The EECAS method can achieve the highest performance, and provide a better throughput guarantee than ROMA.

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/dac

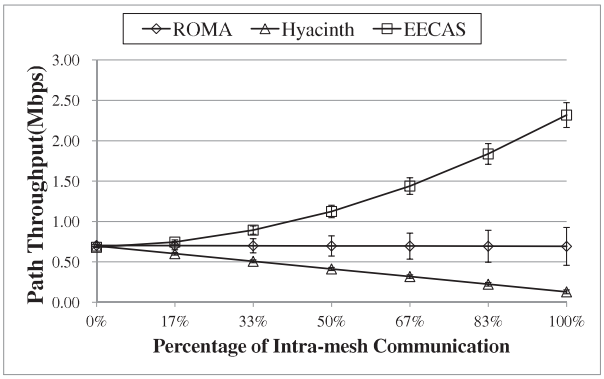


Figure 15. Effect on average path throughput/standard deviation because of various percentages of intra-mesh communications.

5.5. Control message latency

Hyacinth, ROMA, and the proposed EECAS are all distributed channel allocation schemes. Each node must involve local computation and memory space, collaborate with other nodes, and determine the channel allocation for a WMN. In Hyacinth and ROMA, each node only checks its own available logical channels, which can be specified by the frequencies and time slots, and the status of its neighboring nodes. Therefore, the computation and space complexity of Hyacinth and ROMA are related to the number of logical channels and the number of neighbors for each node. To maximize the channel reuse, the proposed EECAS checks the channel status of the neighbors for a node. Therefore, each node must maintain its own channel state table and the channel state tables from its neighboring nodes, and check these tables for channel allocations. The worst-case computation and space complexity of the proposed EECAS become a square of the number of logical channels multiplying the number of neighboring nodes. The computation and space complexity of the EECAS are higher than those of Hyacinth and ROMA. For a practical WMN deployment, the numbers of logical channels and neighboring nodes are not very large. The simulations show that the overheads of local computation and memory space for the proposed EECAS are affordable. In addition to the local computation, Hyacinth, ROMA, and the proposed EECAS all involve a set of procedures and protocols exchanged between nodes before a channel allocation can be determined. Therefore, the proposed protocol was implemented in NS-2 to evaluate its protocol latency.

The latency of route selection and channel allocation in the EECAS is affected by the number of hop-counts, the length of the channel slot (Ls), and length of the parity slot cycle (Pc). This is because control messages are exchanged only in parity slot periods. In other words, multiple parity slots can be arranged in a channel slot cycle to reduce the latency. The EECAS typically spends two round-trip times performing route selection and channel allocation for a new request. Figure 16 shows the total latency of route selection and channel allocation under various numbers of hop-counts. The latency increases slightly as the hop-count increases. When $Ls = 5$ ms and $Pc = 5$, the latency for a 1-hop path is approximately 17.34 ms, which is roughly five times than that of ROMA's latency. The latency increases to 23 ms as the hop count increases to 4. If the length of the channel slot ($Ls = 10$ ms) is double, the latency increases to 26.55 ms (and 36.67 ms) in 1-hop (and 4-hop) paths. If the cycle length of parity slot doubles ($Pc = 10$), the latency increases to 35.97 ms (and 43.89 ms) in 1-hop (and 4-hop) paths. Unlike ad-hoc networks, the nodes in mesh networks are stationary and act as the backbone of the network. Therefore, it is acceptable for a channel allocation scheme such as the EECAS to spend more efforts on route selection and channel allocation.

5.6. Route stretch

The EECAS uses the parameter β to form a trade-off between path performance and channel allocation overhead. To study the effects of β, this study uses a 5×5 mesh network consisting of one

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/dac

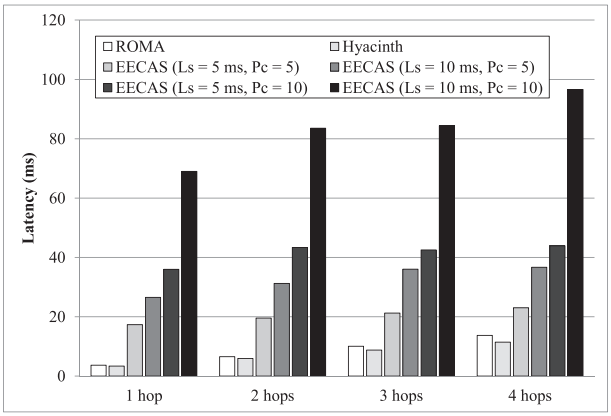


Figure 16. Total latency of route selection and channel allocation under various numbers of hop-counts.

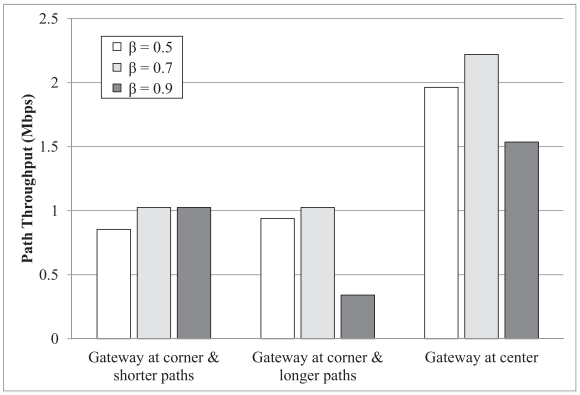


Figure 17. Effect of various network topologies and β values on throughput.

node as the gateway and 24 dual-radio non-gateway nodes. The gateway is a bottleneck if multiple non-gateway nodes simultaneously generate traffics and throw packets to the gateway. Fortunately, the EECAS allows nodes to be equipped with various numbers of radio interfaces because of different roles in a network. Thus, the gateway was equipped with four radios to solve the potential hot spot problem. In simulation, 12 source nodes selected from the 24 non-gateway nodes generated data flows to the gateway. These simulations tested three topologies: (i) placing the gateway at a corner of the network and the source nodes near the gateway; (ii) placing the gateway at a corner, but placing the source nodes opposite to the gateway; and (iii) placing the gateway at the center of the network and randomly choosing the source nodes from other non-gateway nodes. The parameter β was also varied to investigate its effect on the average path length and the average path throughput from the source nodes to the gateway. Three values of β (0.5, 0.7, and 0.9) were used, and three simulations were conduced for each value.

In the topology (i), the average path lengths of the three runs were 2.75, 2.5, and 2.5, and the average path throughputs were 0.85, 1.02, and 1.02 Mbps for $\beta = 0.5$, 0.7, and 0.9, respectively (Figure 17). In the topology (ii), the average path lengths were 3.67, 3.34, and 3.34, and the average path throughputs were 0.94, 1.02, and 0.34 Mbps. As β decreased, many of the sub-optimal paths were replaced by high-performance longer paths. For example, some 3-hop paths were replaced by 4-hop or 5-hop paths as β decreased. However, too many long paths can waste network resources and cause a decrease in the path throughput of subsequent flows. On the other hand, as β increases, many flows, especially in topology (ii), form common paths and share the bandwidth of channel slots in these common paths. However, when many flows share the limited bandwidth of common paths, they decrease the path throughput of subsequent flows. In the last topology (ii), the average

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/dac

path lengths were 2, 1.75, and 1.67, and the average path throughputs are 1.96, 2.22, and 1.54 Mbps for $\beta = 0.5$, 0.7, and 0.9, respectively. Most of the paths in topology (iii) are shorter than the paths in other two topologies, and can achieve higher throughput because they have fewer common paths than other topologies. In summary, higher β values improve performance at the cost of increased channel allocating overhead, whereas smaller β values tend to produce longer paths with higher throughput. Assigning $\beta = 0.7$ leads to better performance in all three simulation topologies.

6. CONCLUSION

This study presents the proposed EECAS, a channel allocation mechanism for end-to-end communication in a WMN. The proposed method extends the radio-frequency-slot method and considers the channel allocation from an end-to-end point of view to improve channel utilization while maintaining the end-to-end throughput of data flows. The proposed EECAS performs well in intra-mesh and inter-mesh communications. The simulation results in this study show that the load-aware EECAS achieves an 8% improvement over the load-unaware EECAS, and also achieves a 231% improvement over ROMA for intra-mesh communications.

ACKNOWLEDGEMENTS

The authors would like to thank MediaTek Inc. and National Science Council of the Republic of China for financially supporting this research under Contract No.100-2221-E-009-072-MY3, 101-2221-E-009-031-MY3, 101-2219-E-009-028, 102-2219-E-009-006-, 102-2220-E-009-020-, 101-2918-I-009-004-, 102-3113-P-006-015-, and NSC99-2915-I-009-064.

REFERENCES

1. Bruno R, Conti M, Gregori E. Mesh networks: commodity multihop ad hoc networks. *IEEE Communications Magazine* 2005; **43**:123–131.
2. Akyildiz IF, Wang XD, Wang WL. Wireless mesh networks: a survey. *Computer Networks-the International Journal of Computer and Telecommunications Networking* 2005; **47**:445–487.
3. Uludag S, Imboden T, Akkaya K. A taxonomy and evaluation for developing 802.11-based wireless mesh network testbeds. *International Journal of Communication Systems* 2012; **25**(8):963–990.
4. Hiertz GR, Denteneer D, Max S, Taori R, Cardona J, Berlemann L, Walke B. IEEE 802.11s: the WLAN mesh standard. *IEEE Wireless Communications* 2010; **17**(1):104–111.
5. Jain K, Padhye J, Padmanabhan VN, Qiu LL. Impact of interference on multi-hop wireless network performance. *Wireless Networks* 2005; **11**:471–487.
6. Hunchangsith K, Bialkowski ME, Portmann M, Tan WL. Analytical model for approximating node throughputs in wireless mesh networks. *International Journal of Communication Systems* 12 Jul 2012. DOI: 10.1002/dac.2383.
7. Yu M, Ma X, Su W, Tung L. A new joint strategy of radio channel allocation and power control for wireless mesh networks. *Computer Communications* 2012; **35**:196–206.
8. Ye J, Wang J-X, Huang J-W. A cross-layer TCP for providing fairness in wireless mesh networks. *International Journal of Communication Systems* 2011; **24**(12):1611–1626.
9. Hou T-C, Hsu C-W, Wu C-S. A delay-based transport layer mechanism for fair TCP throughput over 802.11 multihop wireless mesh networks. *International Journal of Communication Systems* 2011; **24**(8):1015–1032.
10. Lin PJ, Dow CR, Hsuan P, Hwang SF. An efficient traffic control system using dynamic thresholding techniques in wireless mesh networks. *International Journal of Communication Systems* 2011; **24**(3):325–346.
11. Li Y, Yang Y, Zhou L, Wei A, Cao C. QoS-aware fair packet scheduling in IEEE 802.16 wireless mesh networks. *International Journal of Communication Systems* 2010; **23**(6-7):901–917.
12. Alicherry M, Bhatia R, Li L. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. *Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom)*, Cologne, Germany, August 28–September 2, 2005; 58–72.
13. Kodialam M, Nandagopal T. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. *MobiCom '05*, Cologne, Germany, August 28-September 2, 2005.
14. Mohsenian-Rad A, Wong V. Joint logical topology design, interface assignment, channel allocation and routing for multi-channel wireless mesh networks. *IEEE Transactions on Wireless Communications* December 2007; **6**(12):4432–4440.
15. Merlin S, Vaidya NH, Zorzi M. Resource allocation in multi-radio multi-channel multi-hop wireless networks. *UIUC Technical Report*, 2007.
16. Marina M, Das S. A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks. *Proceedings of Broadnets, 2nd International Conference on Broadband Networks (BroadNets 2005)*, Boston, USA, Vol. 1, Oct. 2005; 381–390.

Copyright © 2013 John Wiley & Sons, Ltd.

17. Raniwala A, Gopalan K, cker Chiueh T. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *ACM SIGMOBILE Mobile Computing and Communications Review* 2004; **8**(2):50–65.
18. Dutta P, Jaiswal S, Rastogi R. Routing and channel allocation in rural wireless mesh networks. *26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, Anchorage, USA, 6-12 May 2007; 598–606.
19. Peng Y, Yu Y, Guo L, Jiang D, Gai Q. An efficient joint channel assignment and QoS routing protocol for IEEE 802.11 multi-radio multi-channel wireless mesh networks. *Journal of Network and Computer Applications* March 2013; **36**(2):843–857.
20. Draves R, Padhye J, Zill B. Routing in multi-radio, multi-hop wireless mesh networks. *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom '04)*, Philadelphia, USA, September 26-October 1, 2004; 114–128.
21. Das SM, Wu Y, Chandra R, Hu YC. Context based routing: Technique, applications and experience. *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, San Francisco, USA, 2008; 379–392.
22. Valarmathi K, Malmurugan N. Distributed multichannel assignment with congestion control in wireless mesh networks. *International Journal of Communication Systems* 2011; **24**(12):1584–1594.
23. Sun W, Fu T, Xia F, Qin Z, Cong R. A dynamic channel assignment strategy based on cross-layer design for wireless mesh networks. *International Journal of Communication Systems* 2012; **25**(9):1122–1138.
24. Biswas S. Meraki networks' next generation multi-radio mesh platform, private communication, 2008.
25. Bahl P, Chandra R, Dunagan J. SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. *Proceedings of the 10th annual international conference on Mobile computing and networking (MobiCom 2004)*, Philadelphia, PA, USA, 2004.
26. Zhao R, Walke B, Hiertz GR. An efficient IEEE 802.11 ESS mesh network supporting quality-of-service. *IEEE Journal on Selected Areas in Communications* 2006; **24**(11):2005–2017.
27. Ko B, Misra V, Padhye J, Rubenstein D. Distributed channel assignment in multi-radio 802.11 networks. *IEEE Wireless Communications and Networking Conference*, Kowloon, Hong Kong, 11-15 March 2007; 3978–3983.
28. Raniwala A, cker Chiueh T. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh networks. *24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, Miami, USA, 13-17 March 2005; 2223–2234.
29. Dhananjay A, Zhang H, Li J, Subramanian L. Practical, distributed channel assignment and routing in dual-radio mesh networks. *SIGCOMM'09*, Barcelona, Spain, August 17-21, 2009.
30. Bicket J, Aguayo D, Biswas S, Morris R. Architecture and evaluation of an unplanned 802.11b mesh network. *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (MobiCom '05)*, Cologne, Germany, August 28-September 2, 2005; 31–42.
31. Yang Q, Shi J, Xiao M, Chen H. A decentralized slot synchronization algorithm for TDMA-based ad hoc networks. *International Conference on Wireless Communications, Networking and Mobile Computing, (WiCom 2007)*, Shanghai, China, 21-25 Sept. 2007; 1717–1721.
32. Wang F, Zeng P, Yu H. Slot time synchronization for TDMA-based ad hoc networks. *International Symposium on Computer Science and Computational Technology (ISCSCT '08)*, Shanghai, China, 20-22 Dec. 2008; 544–548.
33. Hiertz GR, Habetha J, May P, Weib E, Bagul R, Mangold S. A decentralized reservation scheme for IEEE 802.11 ad-hoc networks. *14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications (PIMRC 2003)*, Beijing, China, 7-10 Sept. 2003; 2576–2580.
34. Rosier H, Sambale K. Performance evaluation of channel access protocols in overlapping ECMA-368 wireless personal area networks. *IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC 2012)*, Sydney, Australia, 9-12 September 2012; 362–368.
35. De Couto DSJ, Aguayo D, Bicket J, Morris R. A high-throughput path metric for multi-hop wireless routing. *Proceedings of the 9th ACM International Conference on Mobile Computing and Networking (MobiCom '03)*, San Diego, California, September 2003.
36. Draves R, Padhye J, Zill B. Comparison of routing metrics for static multi-hop wireless networks. *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '04)*, Portland, USA, Aug. 30-Sept 3, 2004; 133-144.
37. Herzel F, Fischer G, Gustat H. An integrated CMOS RF synthesizer for 802.11a wireless LAN. *IEEE Journal of Solid-state Circuits* 2003; **18**(10):1767–1770.
38. Antony Franklin A, Balachandran A, Siva Ram Murthy C. Online reconfiguration of channel assignment in multi-channel multi-radio wireless mesh networks. *Computer Communications* 2012; **35**:2004–2013.
39. Gabale V, Raman B, Dutta P, Kalyanraman S. A classification framework for scheduling algorithms in wireless mesh networks. *IEEE Communications Surveys and Tutorials* First Quarter 2013; **15**(1):199–222.
40. Benyamina D, Hafid A, Gendreau M. Wireless mesh networks design - A survey. *IEEE Communications Surveys and Tutorials* 2012; **14**(2):299–310.
41. Ding Y, Xiao L. Channel allocation in multi-channel wireless mesh networks. *Computer Communications* 2011; **34**:803–815.

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/dac

AUTHORS' BIOGRAPHIES

Shiao-Li Tsao earned his Doctor of Philosophy (PhD) degree in Engineering Science from the National Cheng Kung University, Taiwan in 1999. His research interests include energy-aware computing, embedded software and system, and mobile communication and wireless network. He was a visiting scholar at Bell Labs, Lucent Technologies, USA, in the summer of 1998 and a visiting professor at the Department of Electrical and Computer Engineering, University of Waterloo, Canada, in the summer of 2007 and at the Department of Computer Science, ETH Zurich, Switzerland, in the summer of 2010 and 2011, and in 2012-2013. Prof. Tsao is currently an associate professor of the Department of Computer Science of National Chiao Tung University.

Jiun-Jang Su received his Bachelor of Science (BS) degree in Computer Science and Information Engineering in the National Chiao Tung University in 1997 and Master of Science degree in Electrical Engineering in the National Taiwan University of Science and Technology in 2001. He currently is a PhD candidate in Computer Science in the National Chiao Tung University. His research interests include wireless networks and wireless communication.

Kuei-Li Huang received the BS degree in Mathematics from the National Chung-Cheng University, Chia-I, Taiwan, in 2001 and the MS and the PhD degrees in Computer Science from the National Chiao-Tung University, Hsinchu, Taiwan, in 2003 and 2012, respectively. Currently, he is an engineer in the Industrial Technology Research Institute, Hsinchu, Taiwan. His research interests include wireless communications, computer networks, and mathematical modeling.

Yung-Chien Shih earned his PhD degree in Computer Science and Information Engineering from National Chiao-Tung University in 2011 and BS degree in Computer Science and Engineering from the Yuan Ze University in 2001. Dr. Shih is currently a senior engineer of MediaTek Inc., Hsinchu Science Park, Taiwan. His research interests include wireless communications and high speed communication networks.

Chien-Chao Tseng received his BS degree in Industrial Engineering from the National Tsing-Hua University, Hsin-Chu, Taiwan, in 1981 and MS and PhD degrees in Computer Science from the Southern Methodist University, Dallas, Texas, USA, in 1986 and 1989, respectively. He is currently a professor in the Department of Computer Science at National Chiao-Tung University, Hsin-Chu, Taiwan. His research interests include wireless internet, heterogeneous networks, and mobile computing.

Copyright © 2013 John Wiley & Sons, Ltd.

DOI: 10.1002/dac