Joint Multi-Channel Link Layer and Multi-Path Routing Design for Wireless Mesh Networks

Wai-Hong Tam[1] and Yu-Chee Tseng[1,2]
[1]Department of Computer Science
National Chiao-Tung University, Taiwan
[2]Department of Information and Computer Engineering
Chung-Yuan Christian University, Taiwan

Abstract— **In recent years, the wireless mesh network (WMN) attracts the interest of many people as a new broadband Internet access technology. However, increasing throughput is still an open and challenging research issue. One potential solution is to enable transceivers to utilize multiple channels dynamically. However, most of existing works do not consider the routing issue, and trivially use some popular single-path routing protocols like AODV and DSR. In this paper, we exploit the benefit of multi-path routing in multi-channel WMNs from the aspect of end-to-end throughput. Between medium access control and network layers, we propose a novel protocol named *Joint Multi-channel and Multi-path control* (JMM) which combines multi-channel link layer with multi-path routing. Dividing time into slots, JMM coordinates channel usage among slots and schedules traffic flows on dual paths. Our scheme efficiently and intelligently decomposes contending traffics over different channels, different time, and different paths, and hence leads to significant throughput improvement. To the best of our knowledge, this is the first work discussing the joint design of multi-channel control and multi-path routing for WMNs.**

I. INTRODUCTION

Wireless Mesh Networks (WMNs) are believed to be a promising technology to offer broadband wireless access to the Internet and to build self-organized networks in places where wired infrastructure is not available or not worthy to deploy [3]. A WMN consists of a collection of wireless *mesh routers*, which are able to self-configure themselves as a backbone and also serve as an access network to offer connectivity to end-users by standard radio interfaces like 802.11 [1]. A WMN typically has a *two-tier architecture* as shown in Fig. 1. On one hand, mesh routers self-organize themselves to form a wireless backbone, providing large coverage, connectivity, and robustness in the wireless domain. On the other hand, each mesh router is responsible of forwarding traffic on behalf of end-users in its coverage area. A logical separation is maintained between links connecting end-users and links forming the wireless backbone. One or more mesh routers with wired connections will serve as *gateways* to provide Internet access.

While benefiting from large coverage of multihop wireless connections, WMNs also inherit some scalability problems in terms of throughput, delay, and packet delivery ratio faced by all multihop wireless networks [9]. Previous studies have shown that end-to-end throughput of a flow may decrease rapidly as the number of hops increases [14], [25]. The main
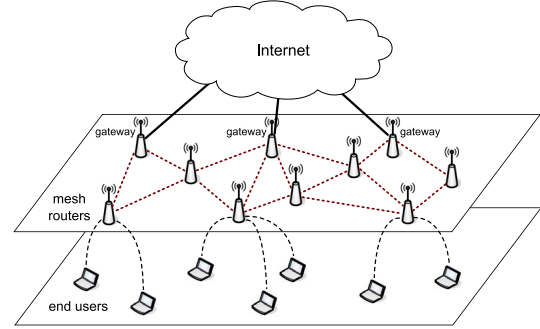

Fig. 1.   The two-tier architecture of wireless mesh networks.

reasons are as follows:

- Half-duplex property of the radios: Radios cannot transmit and receive at the same time. As a result, the capacity of relay nodes is halved.
- Broadcast nature of the wireless medium: When all nodes operate at a common communication channel, each node has to compete with neighboring nodes within extended hops, leading to a high collision probability as the traffic load increases.
- Difficulty of collision avoidance: In a multihop environment, the common phenomena of hidden and exposed terminals cause collision and unfairness, resulting in reduction of throughput.

There are several approaches to relieving the contention and collision problem, such as using directional antennas, implementing transmission power control [24], assisting by location information [20], and employing multiple channels. In this paper, we look for a more cost-effective solution by exploiting multiple non-overlapping channels using only one transceiver per host. While our goal is to improve network performance, we observe that using multiple channels alone is not very effective. Frequency diversity has to be exploited in concert with spatial and temporal reuse. We propose a protocol named *Joint Multi-channel and Multi-path control* (JMM), which can yield a significant performance improvement by decomposing the contending traffic over different channels, different time, and different paths.

The primary contributions of this paper can be summarized as follows:

- We first point out that multi-path routing has to be used in concert with multi-channel design to improve end-to-end throughput. However, using single-path routing cannot achieve this goal.
- We introduce a novel protocol which combines multi-channel link layer with multi-path routing. This protocol is able to increase end-to-end throughput by decomposing the traffic over different channels, time, and space.
- In the route discovery phase of our multi-path routing protocol, we propose a GREQ forwarding strategy to reduce the number of broadcast messages. A new routing metric which explicitly accounts for the disjointness between paths and interference among links is proposed. According to this metric, it is easy to select two maximally disjoint paths with less interference.

In Sec. II, we compare single-path routing with multi-path routing in both single-channel and multi-channel environments to motivate our work. Sec. III reviews related work. The proposed JMM protocol is introduced in Sec. IV. Sec. V presents our simulation results. Finally, Sec. VI concludes the paper.

II. MOTIVATION

To motivate the problem, we first observe the upper bounds of end-to-end throughputs under (i) single-channel, single-path, (ii) multi-channel, single-path, (iii) single-channel, multi-path, and (iv) multi-channel, multi-path scenarios. We then show that case (iv) can achieve better performance.

A. Single-Channel, Single-Path (SCSP) Scenario

The most common combination is to use a single-channel MAC protocol like IEEE 802.11 with a single-path routing protocol like AODV (Ad-hoc On demand Distance Vector) [17]. In this case, packets travel along a chain of nodes toward their destinations. Successive packets on a single chain may interfere with each other as they move along, thus causing contention in the MAC layer.

In the SCSP scenario, we show that an ideal protocol could only achieve an end-to-end throughput at most $\frac{1}{3}$ of the effective MAC layer data rate. Consider the network in Fig. 2(a), where node A is the source and F is the sink. Assume for the moment that radios of nodes that are not neighbors do not interfere with each other. At time 1, A transmits the first packet to B. At time 2, A and B cannot transmit at the same time because B cannot receive and transmit simultaneously. At time 3, A and C cannot transmit at the same time because B cannot correctly hear A while C is sending. At time 4, A and D can send at the same time with the above assumption. Thus, a node can only send $\frac{1}{3}$ of the time.

However, if one assumes that radios can interfere with each other beyond the range at which they can communicate successfully, the situation is even worse. For example, in 802.11b, the interference range is about twice that of transmission range. Hence, in Fig. 2(a), node D's transmission will interfere with that from A to B. This may reduce a node's transmission
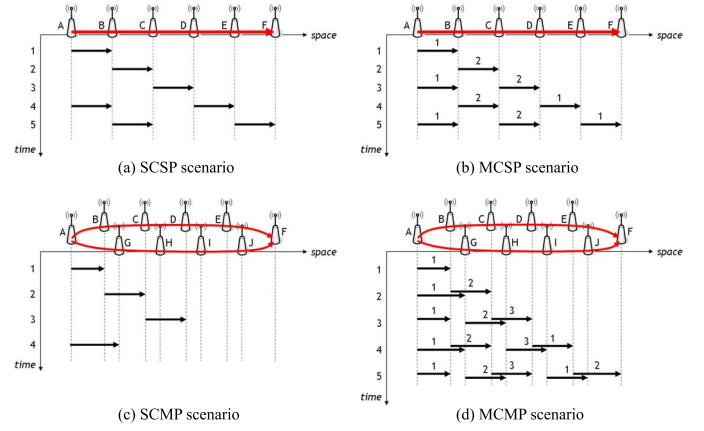

Fig. 2. Ideal packet scheduling in (a) SCSP, (b) MCSP, (c) SCMP, and (d) MCMP scenarios.

opportunity to $\frac{1}{4}$ of the time. As can be seen, the throughput is even more pessimistic.

B. Multi-Channel, Single-Path (MCSP) Scenario

The above analysis shows the impact due to the broadcast nature of wireless medium. To improve the end-to-end throughput, a lot of researchers have proposed multi-channel solutions. Allowing each transceiver to switch among different channels, instead of waiting in the same channel, the MAC protocol has to deal with channel selection and the multi-channel hidden terminal problems [19].

In the MCSP scenario, we show that an ideal multi-channel MAC protocol could achieve end-to-end throughput as high as $\frac{1}{2}$ of the effective MAC data rate. Consider the scenario in Fig. 2(b). Assume that the MAC protocol can always select an appropriate channel and schedule packets perfectly. At time 1, node A transmits the first packet to B on channel 1. At time 2, A and B cannot transmit at the same time because B cannot receive and transmit simultaneously. At time 3, A and C can send at the same time since they use different channels. We can see that if the MAC protocol can switch channels perfectly, A can continuously inject one packet every other slot. This leads to the factor of $\frac{1}{2}$. Because of the half-duplex property of radios, the bottleneck appears in the intermediate nodes.

C. Single-Channel, Multi-Path (SCMP) Scenario

In this SCMP scenario, packets are split along two disjoint paths leading toward destinations. We will show that the broadcast nature of wireless medium may degrade throughput significantly.

In fact, the SCMP scenario can only achieve an end-to-end throughput slightly higher than the SCSP scenario. Consider the network in Fig. 2(c), where there are two disjoint paths from source A to destination F. At time 1, node A transmits the first packet along the upper path to B. At time 2, only one of nodes A and B can transmit because they are competitors. We suppose that B wins in the contention. At time 3, A can not transmit on the lower path because C will interfere the reception of G. So A can only transmit on the lower path at time 4. So A can only inject a packet every three slots.

D. Multi-Channel, Multi-Path (MCMP) Scenario

Some may believe that the factor of $\frac{1}{2}$ is the best case. Below, we show that using a multi-channel MAC protocol combined with a multi-path routing protocol can overcome the bottleneck at intermediate nodes. In the MCMP scenario, we show that the ideal MAC end-to-end throughput can be as high as the effective MAC data rate. Consider the network in Fig. 2(d). Assume that the routing protocol can split packets properly and the MAC protocol can perform ideal channel switching and scheduling. At time 1, node A transmits a packet along the upper path to B on channel 1. At time 2, A transmits a packet along the lower path to G. At the same time B can transmit along the upper path because they use different channels. Afterward, A can alternate between these two paths in every slot. This concludes our derivation.

III. RELATED WORK

In the literature, a lot of efforts have been dedicated to multi-channel link protocols and multi-path routing protocols. However, these link layer protocols and routing protocols are investigated separately. This motivates us to design a joint protocol which combines these two approaches. Below, we review the related work in this field.

A. Multi-Channel MAC and Link Protocols

A lot of multi-channel link/MAC protocols focus on how to utilize multiple channels to reduce the contention and collision among stations. Depending on the number of radio interfaces per node, such protocols can be classified as *single-transceiver* schemes [19], [4], [26], [5] and *multi-transceiver* schemes [23], [28], [2].

For a single-transceiver system, the radio interface in each node needs to switch among channels. It may result in the *multi-channel hidden-terminal problem* [19]. The Multi-channel MAC (MMAC) protocol [19] proposes to embed a negotiation phase in the ATIM (Ad Hoc Traffic Indication Map) window that is periodically sent under the Power Save Mode (PSM). After the ATIM window, nodes may select different channels to transmit and receive packets. The Slotted Seeded Channel Hopping (SSCH) mechanism [4] divides the time axis into virtual channels. The hopping sequence of each virtual channel is determined by a (channel, seed) pair. SSCH requires a looser time synchronization than [19], but it has a higher channel switching overhead. The Multi-channel coordinated Temporal Topology control (MOTTO) [26] also divides the time axis into epochs. The active channel of an epoch is determined statically by the node's hop-count to a gateway and its direction (uplink or downlink).

B. Multi-Channel Routing Protocols

Several works consider utilizing multiple channels at the network layer [18], [10], [8], [12]. These works focus on how to assign channels to a flow and how to find the best path in a multi-channel environment.

The Hyacinth architecture [18] proposes a tree-based routing protocol for a multi-transceiver multi-channel WMN. From each gateway, a tree is constructed, along which packets are forwarded. Reference [10] proposes a CA-AODV protocol that combines channel assignment with AODV [17]. It assumes a system with one control channel and several data channels like DCA [23]. From the exchange of RREQ and RREP packets on the control channel, a source can achieve both route discovery and channel assignment of the flow.

A general multi-channel routing protocol can be designed by combining an existing single-channel routing protocol with a new routing metric by taking multi-channel effects into consideration. The WCETT (Weighted Cumulative Expected Transmission Time) metric in [8] is such a metric for routing in multi-radio multi-hop WMNs.

C. Multi-Path Routing Protocols

Recently, multi-path routing in WMNs has received some attention [16], [21], [13], [27], [15].

Based on Directed Acyclic Graph (DAG), TORA [16] can support multiple-path routing. However, it does not guarantee disjointness of paths. DSR [11] can also find multiple paths, naturally by its flooding behavior. But sometimes only small portions of the found paths are disjoint. The Split Multipath Routing (SMR) [13] can solve this problem because duplicate RREQs are not dropped, but this is at the cost or more RREQs.

AODVM [27] is an extension to AODV for finding reliable routing paths. Duplicate RREQs are not discarded by intermediate nodes. Again, the routing overhead is high. AOMDV [15] is also an extension to AODV for computing multiple loop-free and link-disjoint paths. It uses the notion of "advertised hop count" to guarantee loop-freedom and uses a particular property of flooding to achieve link-disjointness.

IV. JMM PROTOCOL

A. Protocol Architecture

We assume that each node is equipped with an off-the-shelf 802.11 wireless adapter with a half-duplex radio which is allowed to switch among different channels and runs the 802.11 MAC protocol. The proposed JMM protocol is a cross-layer design on top of the 802.11 MAC layer and does not require any change to the 802.11 MAC and hardware. It is composed of a *multi-channel link layer part* and a *multi-path routing part*. These two parts cooperate with each other tightly. JMM has the following functionalities.

1) It decides the receiving channel of each node based on neighborhood information (see Sec. IV-C.1).
2) It constructs a dual path from each node to its gateway (see Sec. IV-C.2).
3) It conducts slot assignment for each node's superframes (see Sec. IV-B.2 and IV-C.4).
4) It schedules and forwards packets and adjusts the ratio of transmitting slots to receiving slots for each node (see Sec. IV-B.3).

Our presentation is bottom-up, from the link layer part (items 3 and 4) toward the routing part (items 1 and 2).
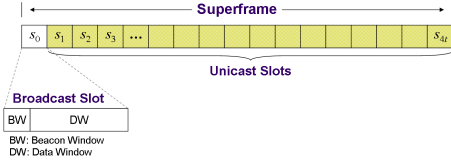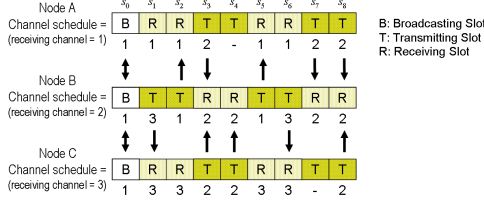
Fig. 3. The superframe structure.

Fig. 4. An example of channel schedule.

Fig. 5. The TF and RF patterns.

B. Multi-Channel Link Layer Part

The link layer has two functionalities: *channel scheduling* and *packet scheduling*. The former is to control which channel the transceiver should stay on, and the latter is to schedule when a packet can be sent. Our design can avoid the *multi-channel hidden-terminal problem* [19].

1) Superframe Structure: The time axis is divided into slots of a fixed length l. Slots are organized into *superframes*. A slot may be designated as a transmitting slot or a receiving slot. We will determine the channels to be used in slots of a superframe. Our channel assignment strategy is *receiver-based*. The structure of a superframe is shown in Fig. 3. Superframes are loosely synchronized in time. Each superframe comprises $4t + 1$ slots, marked as s_0, s_1, ..., s_{4t}, where t is an integer. Slot s_0 is a *broadcast slot* in which only beacons and broadcast messages can be sent. Each broadcast slot is led by a *beacon window*, followed by a *data window*. Beacons also serve to synchronize stations' clocks. To ensure network connectivity, all nodes should stay on a pre-defined common channel in slot s_0. The remaining $4t$ slots are *unicast slots*, whose channels will be decided dynamically.

The receiver-based channel assignment helps two nodes to switch to the same channel for communication. Unicast slots are designnated as transmitting/receiving slots (refer to Sec. IV-B.2). A node will select a *receiving channel* for its receiving slots (refer to Sec. IV-C.1). Nearby nodes will try to avoid using the same receiving channel. During a receiving slot, a node will stay on its receiving channel. During a transmitting slot, a node can switch to its receiver's receiving channel and stay on that channel until the end of the slot. Hence, two nodes can communicate only if one is in a transmitting slot and the other is in a receiving slot. After switching to a new channel, a node first remains silent for a duration equals to the maximum packet transmission time so as to avoid the multi-channel hidden terminal problem which is resulted by loose time synchronization. Therefore, JMM does not require very precise clock synchronization.

An example is in Fig. 4. In s_0, all nodes stay on the common channel 1. In s_1, node B wants to send packets to C, so B
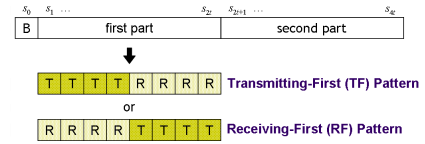
switches to C's receiving channel 3. Suppose that A also wants to send packets to B. Since s_1 and s_2 of A are receiving slots, it has to wait until s_3 to transmit. Note that since both A and C want to send packets to B in s_3, they will use 802.11's CSMA/CA mechanism to contend for the medium.

2) Transmitting and Receiving Patterns: Unicast slots of a superframe are designated as transmitting/receiving slots. However, since traffics on mesh networks are quite stable, slot assignment will not be changed too frequently. In each superframe, unicast slots s_1 to s_{4t} are evenly divided into two parts, with the fist part from s_1 to s_{2t} and the second part from s_{2t+1} to s_{4t}. One part is designated as the *upstream part* for communication with the node's upstream nodes (with respect to the node's gateway), and the other part is the *downstream part* for communication with its downstream nodes. These two parts are of the same length because for a relay node, the amount of traffics to and from upstream nodes is likely to be equal to that to and from downstream nodes.

Each part can follow a *Transmitting-First (TF) pattern* or a *Receiving-First (RF) pattern* as shown in Fig. 5. In a TF pattern, the first half is all transmitting slots, and the second half is all receiving slots. Contrarily, in a RF pattern, the first half is all receiving slots, and the second half is all transmitting slots. Considering the patterns of the first and the second parts, there are four types of superframe patterns, namely TF-TF, RF-RF, TF-RF, and RF-TF types. The ratio of the number of transmitting slots to the number of receiving slots can be adjusted dynamically (refer to Sec. IV-B.3).

3) Dynamic Adjustment of the T/R Ratio: Recall that each superframe has an upstream part and a downstream part. The ratio of the number T of transmitting slots to the number R of receiving slots in each upstream part, call T/R ratio, can be dynamically adjusted in a per node basis. Since in a relay node the amount of traffics from upstream nodes is likely to be equal to that to downstream nodes, the number of receiving slots in the upstream part should equal the number of transmitting slots in the downstream part. Similarly, the transmitting slots in the upstream part should equal the number of receiving slots in the downstream part. Therefore, in the downstream part, we can let T be the number of receiving slots and R be the number of transmitting slots.

The T/R ratio of each node is adjusted dynamically during runtime. Initially, we set $T = R = t$. A node should monitor the actual traffic through itself. Assume that the actual transmitting and receiving traffics on the upstream part are T_{actual} and R_{actual}, respectively. We then compute new weighted averages T_{smooth} and R_{smooth} as follows:

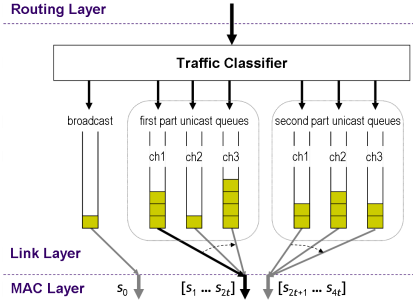$$T_{smooth} \leftarrow \alpha * T_{actual} + (1 - \alpha) * T_{smooth}; \qquad (1)$$

Fig. 6. The broadcast and unicast queues at the link layer.

Fig. 7. A channel selection example.

$$R_{smooth} \leftarrow \alpha * R_{actual} + (1 - \alpha) * R_{smooth}. \qquad (2)$$

The values of T and R will be changed slowly by the following rules:

```
if (T_smooth/T)/(R_smooth/R) > Threshold_h and R > 1 then
    T ← T+1;
    R ← R−1;
endif

if (T_smooth/T)/(R_smooth/R) < Threshold_l and T > 1 then
    T ← T−1;
    R ← R+1;
endif
```

T_{smooth}/T and R_{smooth}/R are the utilizations of transmitting and receiving slots, respectively. If the utilization ratio of transmitting to receiving slots is higher than a threshold $Threshold_h$, we increase T and decrease R by one. If the utilization ratio is lower than a threshold $Threshold_l$, a reverse process is preformed.

4) Packet Queues: When packets arrive, we need to allocate them to transmitting slots for transmission. JMM dispatches packets into a broadcast queue and two groups of unicast queues as shown in Fig. 6, where we assume that there are three non-overlap channels. Broadcast packets are enqueued in the *broadcast queue*, while unicast packets are classified as the *first part* or the *second part* and then are enqueued in the corresponding queues based on the receiving channels of receivers (refer to Sec. IV-C.5). The number of queues in each part is equal to the number of channels in the system.

The broadcast queue is served in broadcast slots. The first part unicast queues are served by transmitting slots of slots s_1 to s_{2t} in a round-robin manner. Each transmitting slot will serve one queue by switching to the channel of that queue, until the queue is empty or the slot expires. The second part unicast queues are served by transmitting slots of slots s_{2t+1} to s_{4t} in a similar way.

5) Permutation of Slots: In the above discussion, transmitting and receiving slots are clustered together. In practice, we can permute the slot sequence of a superframe to obtain some degree of randomness among these slots. The same permutation of $4t$ elements should be applied to all nodes.

C. Multi-Path Routing Part

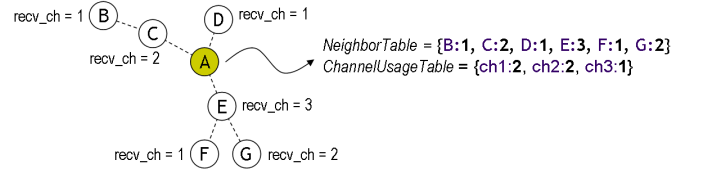The goal of the routing part is to construct two paths to the gateway. Since finding the best two paths requires the channel information provided by the link layer part, we first describe how a node selects its receiving channel. We then present the multi-path route discovery phase and our path selection metric. Finally, we describe our packet scheduling scheme to exploit the benefit of multi-path routing. In JMM, route discovery is done in an on-demand manner. However, the selection of receiving channel will be changed less frequently.

1) Selection of Receiving Channels: When a node is first turned on, it can choose any channel as its receiving channel. Periodically, each node broadcasts its receiving channel to its 2-hop neighbors. This can be achieved by broadcasting a HELLO message carrying a node's direct neighbors' receiving channels. Each node maintains a *NeighborTable* containing the receiving channels of its 2-hop neighbors and a *ChannelUsageTable* to count the number of nodes using each channel. For example, Fig. 7 shows these tables of node A.

A node will choose the least used channel as its receiving channel. To prevent unnecessary fluctuation, when a node finds a better channel than its current receiving channel, it will only switch to that channel with a probability p.

2) Dual-Path Route Discovery: Our goal is to find from each node two paths to its gateway that are as disjoint as possible. However, a dilemma is: on one hand, we would like to avoid network-wide flooding of route search packets, while on the other hand, we do not expect too many duplicate route search packets being discarded by intermediate nodes.

Below, we propose an efficient discovery strategy to find a dual-path to each gateway in the network. A Gateway REQuest (GREQ) packet is used for this purpose. Instead of blindly flooding, limited rebroadcasts of GREQs are invoked. The format of GREQ is shown in Table I. The route discovery is performed in an incremental way. So when a node issues a GREQ, we can assume that each existing node has already established two paths to its gateway. For each node, let $gwAddr$ be its selected gateway and $hopCount$ be the length of the shorter path of its dual-path. When an intermediate node R receives a GREQ, the procedure in Fig. 8 is executed. It first checks whether the sequence number is up-to-date (lines 2-6). Then it verifies if its slot schedule mismatches with that of the transmitter (lines 7-9). Note that a "mismatch" happens when the superframe patterns of two neighboring nodes are the same (i.e., they choose the same type from TF-TF, RF-RF, TF-RF, and RF-TF), in which case these two nodes cannot communicate with each other. The $gwAddr$ and $hopCount$ fields guarantee that the GREQ packet is forwarded toward the gateway indicated in the $gwAddr$ flied and the $hopCount$ value progressively decreases on its way to the gateway (lines

Field	Initial value	Meanings
$seqNum$	seqNum at S	the sequence number
$srcAddr$	S	the source address
$gwAddr$	unknown	the gateway address of the mesh network
$hopCount$	∞	the smallest number of hops to the gateway
$pathRecord$	{S}	the list of node records on the path

```
/*Executed when a non-gateway node R receives a GREQ from a node T */
01. begin
02.     if GREQ.seqNum < R.seqNum[srcAddr] then
03.         discard and exit;
04.     else
05.         R.seqNum[srcAddr] ← GREQ.seqNum;
06.     endif
07.     if the slot schedules of R and T mismatch then
08.         discard and exit;
09.     endif
10.     if GREQ.gwAddr ≠ unknown and
            GREQ.gwAddr ≠ R.gwAddr then
11.         discard and exit;
12.     endif
        /* Ensure that hopCount progressively decreases */
13.     if GREQ.hopCount < R.hopCount then
14.         discard and exit;
15.     elseif GREQ.hopCount = R.hopCount then
16.         if R ∈ GREQ.pathRecord then
17.             discard and exit;
18.         endif
19.     endif
20.     send GREQ(GREQ.seqNum, GREQ.srcAddr, R.gwAddr,
                    R.hopCount, GREQ.pathRecord ∪ {R});
21. end
```

Fig. 8.   The GREQ propagation procedure of a non-gateway node.

10-19). This forwarding strategy can significantly reduce the rebroadcast overhead while traversing most wireless links. Finally, the node rebroadcasts the GREQ packets (line 20).

An example of the GREQ propagation procedure is shown in Fig. 9. The links indicated by dashed lines mean that the corresponding GREQs are discarded.

3) Path Selection Metric: After the above procedure, each gateway will collect a number of GREQs each carrying a path. Since our goal is to find a dual-path, the gateway will use a metric function to evaluate each pair of paths. For example, the gateway X in Fig. 9 will collect $n = 4$ paths, S-C-A-X, S-D-B-X, S-C-B-X, and S-D-A-X, from the route discovery initiated by S. So there are totally $\binom{n+2-1}{2} = 10$ path pairs
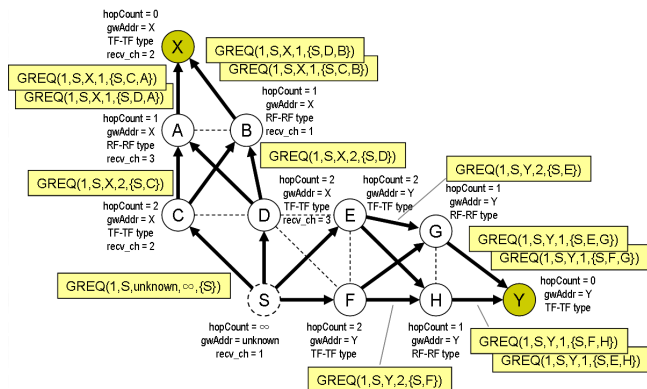

Fig. 9.   A GREQ propagation example.

to be evaluated. Note that the combination with repetition is used because a path may serve as both paths of a dual-path in case that there is no good choice. The path pair with the lowest metric will be selected and two Gateway REPly (GREP) packets are unicast along the reverse directions to the source node. Then the source node will collect all GREP packets from different gateways and select the dual-path with the best path metric by sending two GREP ACKnowledgement (GREP_ACK) packets to the selected gateway along the dual-path.

The input of the path metric function is a path pair (P_1, P_2) and the output is affected by 3 factors V_{node}, V_{chl}, and V_{qlty}. V_{node} is the number of common nodes between P_1 and P_2 excluding the source node and the gateway. V_{chl} is defined as

$$V_{chl} = CN(P_1) + CN(P_2) + \delta(P_1, P_2), \qquad (3)$$

where $CN(P_i)$ is the number of *channel contending pairs* along P_i, where two nodes on P_i are called a channel contending pair if they are within 2 hops and use the same receiving channel. For example, $CN(S\text{-}C\text{-}B\text{-}X) = 2$ because (S, A) and (C, X) are channel-contending pairs. Function $\delta(P_1, P_2) = 1$ if the difference of the lengths $|P_1|$ and $|P_2|$ is an odd number; otherwise, $\delta(P_1, P_2) = 0$. The value is so assigned because our algorithm prefers paths differ in lengths by an even number (refer to the discussion in Sec. IV-C.4). To reflect the signal quality perceived by nodes on P_1 and P_2, V_{qlty} is defined as $ETX(P_1) + ETX(P_2)$, where $ETX(P_i)$ is the expected transmission count of a packet along P_i [6]. Alternatively, other metrics for evaluating path quality [7], [8] can be used instead. We combine the three factors by taking their weighted average:

$$metric = w_{node}V_{node} + w_{chl}V_{chl} + w_{qlty}V_{qlty}, \qquad (4)$$

where $w_{node} + w_{chl} + w_{qlty} = 1$. The one with a lower metric is preferred.

4) Determining Superframe Patterns: Next, we need to determine the superframe pattern (TF-TF, RF-RF, TF-RF, or RF-TF) of each node. The selection will be based on the result of the route discovery. We assume that all nodes on the dual-path except the source have already determined their superframe patterns. Without loss of generality, let the gateway choose the TF-TF type. Given any dual-path, the gateway will designates one path as the *master path*, and the other as the *slave path*. The requirement to be a master path is that the superframe patterns of the gateway and the first child must match in the first part (slots s_1 to s_{2t}), and the requirement to be a slave path is that they must match in the second part (slots s_{2t+1} to s_{4t}). A "match" happens if one side uses TF and the other side uses RF. Let S be the source, G be the gateway, and (P_1, P_2) be the dual-path, such that P_1 is the master path and P_2 is the slave path. The superframe pattern of S will be selected by the following rules:

1) $|P_1| - |P_2|$ is even: We refer to Fig. 10 for ease of presentation. If $|P_1|$ is odd, the pattern of S's first part should match with that of its parent on P_1 and the pattern
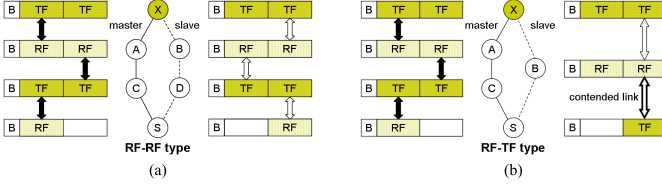
Fig. 10. The pattern selection of S when (a) $|P_1| - |P_2|$ is even and (b) $|P_1| - |P_2|$ is odd.

Fig. 11. An example where the contended link B-S on the slave path (..., X, Y, B, S) serves as a link on the master path (..., I, J, B, S, K).

of S's second part should match with that of its parent on P_2. If $|P_1|$ is even, the pattern of S's first part should match with that of its parent on P_2 and the pattern of S's second part should match with that of its parent on P_1. Hence, S chooses the RF-RF type in Fig. 10(a). This pattern selection can achieve high channel utilization. Packets on the dual paths are unlikely to interference with each other because they are separated in both the time domain and the space domain when they happen to use the same channels.

2) $|P_1| - |P_2|$ is odd: In this case, one of $|P_1|$ and $|P_2|$ is odd and the other is even. Let P be the longer path. If $|P|$ is odd, the pattern of S's first part should match with that of its parent on P_1 and the pattern of S's second part should match with that of its parent on P_2. If $|P|$ is even, the pattern of S's first part should match with that of its parent on P_2 and the pattern of S's second part should match with that of its parent on P_1. Hence, S chooses the RF-TF type in Fig. 10(b).

In case 2, packet transmission on the dual paths are also quite interference-free, except the link between S and its parent on the shorter path of P_1 and P_2, which is called the *contended link*. Because S matches with its parent on the same part as where S's parent matches with S's grandparent on that path. For example, B-S in Fig. 10(b) is a contended link. This competition may affect the end-to-end throughput of that path. So we let this happen on the shorter path. Also, the penalty is reflected by the earlier function $\delta(P_1, P_2)$ in the path metric V_{chl}.

Note that a contended link may play parts in both a master path of a dual path (P_1, P_2) and a slave path of another dual path (P'_1, P'_2). For example, in Fig. 11, B-S is a contended link in (P'_1, P'_2). If later on node S accepts a child K, which chooses the path along S as its master path, then B-S will be part of a master path in (P_1, P_2). However, the patterns of superframes of K's master path are not affected by the appearance of this contended link.

5) Packet Forwarding Rule: With dual-path routing, our system needs to inject packets to both paths to exploit communication parallelism. Below, we summarize our packet forwarding rule. When a source node or a gateway generates a sequence of packets, we will alternately mark them as to be sent along the master path or along the slave path. For each packet, we will compute a value $P = M \oplus E \oplus D \oplus C$, where

$$M = \begin{cases} 0 & \text{if the pkt is to be sent along the master path;} \\ 1 & \text{if the pkt is to be sent along the slave path;} \end{cases} \quad (5)$$
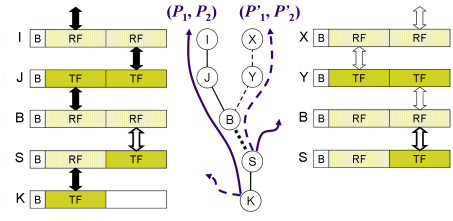
$$E = \begin{cases} 0 & \text{if the hop count to gw along the intended path is even;} \\ 1 & \text{if the hop count to gw along the intended path is odd;} \end{cases} \quad (6)$$

$$D = \begin{cases} 0 & \text{if the pkt is issued by a gateway;} \\ 1 & \text{if the pkt is issued by a source;} \end{cases} \quad (7)$$

$$C = \begin{cases} 0 & \text{if the pkt is to be transmitted to a non-contended link;} \\ 1 & \text{if the pkt is to be transmitted to a contended link.} \end{cases} \quad (8)$$

If $P = 0$, the packet will be forwarded to the first part unicast queues; otherwise, the packet will be sent to the second part unicast queues (refer to Fig. 6).

For a relayed packet, it is alternated between the first and the second parts except when it passes through a contended link. Specifically, if a packet is received from a contended link, it is enqueued to the same part of unicast queues; otherwise, it is enqueued to a different part of unicast queues from its original one. For example, in Fig. 10(b), when node B receives a packet from S in the second part, it enqueues the packet to the same second part, but when C receives a packet from S, it enqueues the packet to the different part.

6) Route Maintenance: Faulty links are detected by nodes' periodical HELLO messages. Losing a predefined number of HELLOs is an indication of a fault of link. When a node discovers a faulty link, it will propagate a Gateway ERRor (GERR) message to all its successors which use this link. Each successor will initiate a new gateway discovery procedure to find a new dual-path. Before new paths are found, the other (non-broken) paths can still be used for communication. Therefore, JMM is also quite resilient to failure.

V. PERFORMANCE EVALUATION

To evaluate the performance of the proposed protocol, we have implemented a JMM module in the NCTUns network simulator 2.0 [22]. The JMM module has a link layer, a routing layer, and some FIFO queues. The MAC layer is the IEEE 802.11a without using RTS/CTS. Data rate is 54 Mbps. Each node has a transmission range of 250 meters and an interference range of 550 meters. The default parameters used are shown in Table II.

A. Comparison of SCSP, SCMP, MCSP, and MCMP Routing

We first compare the performance of SCSP, SCMP, MCSP and MCMP routing. Network topologies as shown in Fig. 12 are tested. Assuming $H = 200$ meters, $V = 300$ meters, and five available channels, we vary the number of hops from the gateway to the destination and observe the end-to-end throughput. Continuous 512-byte packets are injected from the

Parameter	Default	Meanings
l	20 ms	the slot size
t	4	the number of slots in a quarter of a superfame
α	0.2	the weight between actual and smooth traffic
$Threshold_h$	2	the high threshold of adjusting the T/R ratio
$Threshold_l$	0.5	the low threshold of adjusting the T/R ratio
w_{node}	0.74	the weight of parameter V_{node}
w_{chl}	0.18	the weight of parameter V_{chl}
w_{qlty}	0.08	the weight of parameter V_{qlty}
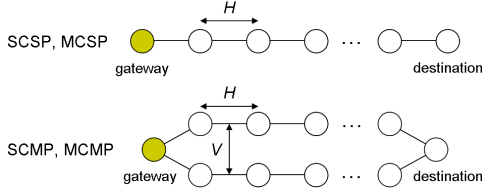
Fig. 12.   Single-path and dual-path topologies used in our simulation.

Fig. 14.   Aggregate gateway throughput vs. traffic load under different numbers of traffic sources.

gateway to the destination. SCSP routing uses IEEE 802.11 MAC and AODV. SCMP routing uses the multi-path routing protocol AODVM. MCMP routing uses our JMM protocol. MCSP routing also uses our JMM protocol but it only employs a single path routing. The results are shown in Fig. 13(a). As can be observed, in SCSP and SCMP routing the end-to-end throughputs decrease dramatically as the number of hops increases. The SCMP routing is only slightly better than the SCSP routing since the two parallel paths still seriously interfere with each other. On the other hand, the throughputs of MCSP and MCMP remain relatively constant since newly added nodes will not interfere with existing nodes. The throughput of MCMP is about twice the throughput of MCSP. This demonstrates the advantage of our superframe structure in avoiding temporal and spatial interferences.

For SCMP and MCMP routing, we further vary the distance V between the two parallel paths. As shown in Fig. 13(b), as V decreases, the average end-to-end throughput of SCMP drops significantly due to higher and higher contention between the two paths. JMM achieves more than three time the throughput of SCMP routing as V reduces to below 400 meters. The throughput of JMM is quite insensitive to the value of V, which demonstrates the advantage of our JMM protocol in distributing packets to two parallel paths on which the transmissions are well interleaved.
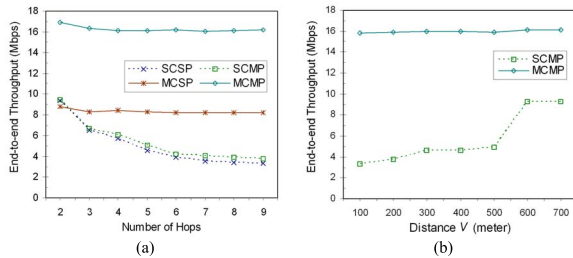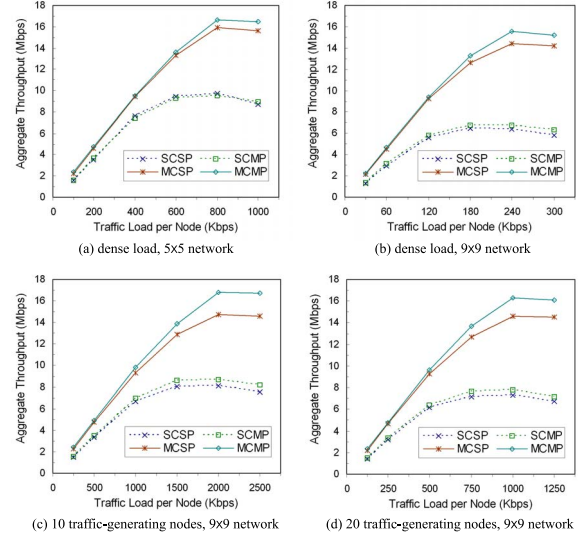

Fig. 13.   (a) Average end-to-end throughput vs. number of hops, and (b) average end-to-end throughput vs. distance V (path length = 6)

B. Impact of Traffic Load

To study JMM's performance for different traffic loads, we simulate a stationary 5×5 and 9×9 grid networks with only one gateway located in the center of the grid. Neighboring nodes are uniformly separated by 200 meters. Two different traffic loads are simulated: a dense load where each node in the grid generates even CBR (Constant Bit Rate) traffics towards the gateway, and a sparse load where only a few random-chosen nodes generate traffic. In both simulations, we gradually increase the traffic load of each flow and measure the gateway's throughputs, as shown in Figure 14. JMM outperforms SCSP and SCMP by over 100%, and outperforms MCSP by 10-20% depending on the traffic load. The amount of improvement is less significant in the dense load case. Because every node is transmitting and thus it is hard to see the advantage of multi-path routing. Our saturated throughput is close to the upper bound 19.5 Mbps (the maximum throughput between only two nodes after considering all MAC and PHY overheads). Note that this also includes JMM's overheads of broadcast slots and channel switching latency.

C. Impact of Slot Size l on JMM Protocol

Above simulations have fixed the slot size l to 20 msec. The length of l can influence the performance of JMM. Longer l may result in increased end-to-end delay as well as the buffer requirement at each node. On the other hand, if the length of l is too short, the channel switching overhead becomes considerable and degrades the system performance. To study this impact, aggregate throughput is measured using different l under 5×5 and 9×9 grid networks as shown in Fig. 15. In packet sizes of 256, 512, and 1024 bytes, we see consistent higher network throughputs as l increases from 5 to 30 msec, due to less channel switching overhead. However, this is at the cost of higher end-to-end delays. We recommend $l = 20$ msec from our experience.
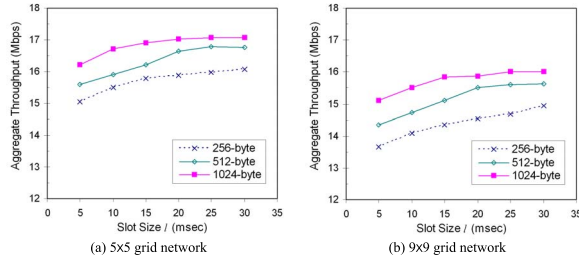
(a) 5x5 grid network   (b) 9x9 grid network

Fig. 15. Aggregate network throughput vs. slot size l.

VI. CONCLUSIONS

We have shown that multi-path routing, when being harmonized with multi-channel capability, has great potential to achieve good performance for WMNs. We then design the JMM protocol which combines multi-channel link layer and multi-path routing to offer this benefit. Dividing the time into slots, JMM coordinates channel usage among slots using a receiver-based channel assignment and schedules transmissions along dual paths. In the route discovery phase of JMM, we propose a GREQ forwarding strategy to reduce broadcast overhead. In addition, we define a new routing metric which explicitly accounts for the disjointness between paths and interference among links. According to this metric, it is easy to select two maximally disjoint paths with less interference. Our simulation results show that JMM yields a significant end-to-end throughput improvement in WMNs as compared to single-channel scenarios. In summary, JMM efficiently increases the performance by decomposing contending traffic over different channels, different time, and different paths.

ACKNOWLEDGMENT

REFERENCES

[1] IEEE Standard 802.11-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[2] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou. A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks. In *International Conference on Broadband Networks (Broadnets)*, October 2004.

[3] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Elsevier Computer Networks Journal*, March 2005.

[4] P. Bahl, R. Chandra, and J. Dunagan. SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, September 2004.

[5] J. Chen and Y.-D. Chen. AMNP: Ad Hoc Multichannel Negotiation Protocol for Multihop Mobile Wireless Networks. In *Proceedings of the IEEE Interneational Conference on Communications (ICC)*, June 2004.

[6] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris. A High-Throughput Path Metric for Multi-Hop Wireless Routing. *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, September 2003.

[7] R. Draves, J. Padhye, and B. Zill. Comparison of Routing Metrics for Static Multi-Hop Wireless Networks. *Proceedings of the Special Interest Group on Data Communication (SIGCOMM)*, Auguest 2004.

[8] R. Draves, J. Padhye, and B. Zill. Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, September 2004.

[9] M. Gerla, R. Bagrodia, L. Zhang, K. Tang, and L. Wang. TCP over Wireless Multi-hop Protocols: Simulation and Experiments. In *Proceedings of the IEEE Interneational Conference on Communications (ICC)*, June 1999.

[10] M. X. Gong and S. F. Midkiff. Distributed Channel Assignment Protocols: A Cross-Layer Approach. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, March 2005.

[11] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. *Mobile Computing*, pages 153–181, 1996.

[12] P. Kyasanur and N. H. Vaidya. Routing and Interface Assignment in Multi-Channel Multi-Interface Wireless Networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, March 2005.

[13] S.-J. Lee and M. Gerla. SMR: Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks. In *Proceedings of the IEEE Interneational Conference on Communications (ICC)*, June 2001.

[14] J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris. Capacity of Ad Hoc Wireless Networks. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, July 2001.

[15] M. K. Marina and S. R. Das. On-Demand Multipath Distance Vector Routing for Ad Hoc Networks. In *Proceedings of the International Conference for Network Protocols (ICNP)*, November 2001.

[16] V. D. Park and M. S. Corson. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In *Conference on Computer Communications (Infocom)*, April 1997.

[17] C. E. Perkins and E. M. Royer. Ad-Hoc On Demand Distance Vector Routing. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, February 1999.

[18] A. Raniwala and T. Chiueh. Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network. In *Conference on Computer Communications (Infocom)*, March 2005.

[19] J. So and N. H. Vaidya. Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver. In *Proceedings of the ACM Interatioal Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, May 2004.

[20] Y.-C. Tseng, S.-L. Wu, W.-H. Liao, and C.-M. Chao. Location Awareness in Ad Hoc Wireless Mobile Networks. *IEEE Computer*, pages 46–52, June 2001.

[21] A. Valera, W. Seah, and S. Rao. Cooperative Packet Caching and Shortest Multipath Routing In Mobile Ad hoc Networks. In *Conference on Computer Communications (Infocom)*, March 2003.

[22] S.-Y. Wang and Y.-B. Lin. NCTUns Network Simulation and Emulation for Wireless Resource Management. *Wiley Wireless Communications and Mobile Computing*, pages 899–916, December 2005.

[23] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu. A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks. In *International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN)*, December 2000.

[24] S.-L. Wu, Y.-C. Tseng, and J.-P. Sheu. Intelligent Medium Access for Mobile Ad Hoc Networks with Busy Tones and Power Control. *IEEE Journal on Selected Areas in Communications*, pages 1647–1657, September 2000.

[25] S. Xu and T. Saadawi. Does the IEEE 802.11 MAC protocol work well in multihop ad hoc networks? *IEEE Communications Magazine*, pages 130–137, June 2001.

[26] Q. Xue and A. Ganz. Temporal Topologies in Multi-channel Multihop Wireless Access Networks. In *International Conference on Broadband Networks (Broadnets)*, October 2005.

[27] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi. A Framework for Reliable Routing in Mobile Ad Hoc Networks. In *Conference on Computer Communications (Infocom)*, June 2001.

[28] J. Zhu and S. Roy. 802.11 Mesh Networks with Two Radio Access Points. In *Proceedings of the IEEE Interneational Conference on Communications (ICC)*, May 2005.