Title	J-CAR: An efficient channel assignment and routing protocol for multi-channel multi-interface mobile ad hoc networks
Author(s)	Chiu, HS; Yeung, KL; Lui, KS
Citation	Proceedings of the Global Telecommunications Conference, 2006 (GLOBECOM 2006), San Francisco, CA, USA, 27 November - 1 December 2006
Issued Date	2006
URL	http://hdl.handle.net/10722/158549
Rights	©2006 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

J-CAR: an Efficient Channel Assignment and Routing Protocol for Multi-channel Multi-interface Mobile Ad Hoc Networks

Hon Sun Chiu, Kwan L. Yeung and King-Shan Lui
Department of Electrical and Electronic Engineering
The University of Hong Kong, Hong Kong, PRC
Email: {hschiu,kyeung,kslui}@eee.hku.hk

Abstract – **We propose an efficient joint channel assignment and routing protocol (J-CAR) for multi-channel multi-interface mobile ad hoc networks (MANETs). Aiming at overcoming the limitations of the existing channel assignment and routing algorithms, J-CAR negotiates a channel at each active link during the route setup process. It has the following major features: a) a pre-determined common control channel is used by every node for routing and channel negotiation; b) control packets for data transmission (RTS, CTS & ACK) are carried by the associated data channels; c) the spare capacity on the control channel can be used for data transmission; d) an interface is free to change its working modes between send and receive; and e) an interface can tune to any data channels for data sending or receiving at the cost of switching overhead. With J-CAR, a more flexible assignment of interfaces, channels, and the working mode of each interface can be rendered. The performance gain brought by J-CAR is substantiated by extensive simulation results.**

Keywords – **IEEE 802.11, multiple channels multiple interfaces, joint channel assignment and routing.**

I. INTRODUCTION

With the increasing popularity of mobile ad hoc networks (MANET), network capacity becomes an important issue. Previous studies [1] have shown that the capacity of a wireless network decreases with an increasing number of mobile nodes. The problem is more serious in multi-hop networks due to interference from adjacent hops on the same path as well as from neighboring paths [2]. This capacity problem can be alleviated if multiple channels are allowed in the network, e.g. IEEE 802.11a offers 12 non-overlapping channels.

Several multi-channel protocols are proposed by using a single network interface card [3-5]. However, the nodes may not be aware the existence of each other if their interfaces are tuned to different channels. To reduce the synchronization overheads, each interface has to stay on a channel for certain duration (e.g. 10ms in [3] and 100ms in [4]).

It is feasible to equip nodes with multiple IEEE 802.11 interface cards because of the cost reduction [6]. An interface can switch between different channels, at the cost of switching latency. Multiple channels and multiple interfaces are widely used in designing wireless mesh networks (WMNs). Protocols in [7-10] are designed based on the WMN properties: static network topology and stable traffic pattern. However, such long-term based protocols are not suitable for the highly dynamic MANETs.

There are also some multi-channel multi-interface protocols proposed for MANETs [11-14]. They aim at decoupling the channel assignment and routing into two separate phases, some even do not address the routing issue [11, 12]. In contrast, we believe that channel assignment and routing should be jointly considered, as that in [5]. This is due to the fact that routing decision affects the level of interference, and hence the channel assignment. On the other hand, channel assignment divides the nodes into different domains, which affects the routing decision. Decoupling them will lead to less satisfactory performance.

In this paper, an efficient joint channel assignment and routing protocol (J-CAR) is proposed for multi-channel multi-interface MANETs. Unlike existing schemes, J-CAR does not specify the role of individual interfaces for data sending or receiving, except a dedicated interface for the control channel. This extra flexibility enhances the system performance by effectively using the interfaces. In J-CAR, a route is set up only if there are data packets to send. Integrated with the route setup procedure, channel is negotiated hop by hop. This reduces the number of required channels, and also the level of interference.

The rest of the paper is organized as follows. Some related work is reviewed in the next section. Section III discusses the protocol design of J-CAR. Its performance is evaluated in Section IV. Finally, we conclude the paper in Section V.

II. RELATED WORK

A. Single-interface protocols

Multi-channel protocols in [3-5] assume a single wireless interface per node. Synchronization is required in SSCH [3] and MMAC [4]. In SSCH, each node maintains its own channel hopping sequence. These sequences have to be synchronized for communication. While in MMAC, periodic channel negotiation window (ATIM) is used for nodes to negotiate channels. However, the bandwidth on data channels during the ATIM window period is wasted.

ECA-AODV [5] combines channel assignment and routing together. However, limited by the single interface, a node is required to switch frequently between the control and data channels. For multi-hop paths, the aggregate switching delay can be significant.

B. Multiple-interface protocols for Wireless Mesh networks

A common approach adopted by protocols in [7-9] is that, based on a given traffic profile, routing is performed first and then followed by channel assignment. In [7], a load balanced shortest path tree rooted at the gateway is designed. Channels are then assigned to each link based on the aggregate loading. The algorithms in [8, 9] do not employ a tree-based routing structure. They first set up initial tentative routes for link load estimation, and based on it channels are assigned. A scheduling algorithm is also proposed in [8] to obtain interference-free links.

While all protocols in [7-9] are centralized, a distributed mechanism is proposed in [10], called Hyacinth. Upon receiving the ADVERTISE messages, each node makes its own join decision based on the cost of the paths connecting to the gateway. The parent node replies the request by an ACCEPT message, with the information about which channel to use. Eventually a tree is formed and rooted at the gateway.

C. Multiple-interface protocols for Mobile Ad hoc networks

1-4244-0357-X/06/$20.00 ©2006 IEEE

Some multi-channel multi-interface protocols [11-14] are proposed for MANETs, under the assumption that the number of interfaces per node is less than the number of non-overlapping channels. DCA [11] assumes two interfaces per node. One interface is used to exchange control packets, including routing, channel negotiation and RTS/CTS/ACK packets, in a dedicated control channel. The remaining interface transmits data in the negotiated data channel. This can saturate the control channel easily and thus underutilizes the data channels. Although MTMA [12] allows more than two interfaces, the control channel congestion problem deteriorates.

In PCAM [13] and MCR [14], the receiving channel of each node is assigned in advance, thus channel negotiation is not needed. The control channel congestion problem is also solved by moving RTS/CTS/ACK packets to the associated data channels. But other limitations exist. As each interface is assigned with a fixed role of send or receive [13, 14], the utilization is limited. E.g. if a node only receives data, its send interface is wasted. The receiving channels are pre-assigned to all the nodes [13, 14] (or *node-based channel pre-assignment*), even if there is no data to send/receive, power and bandwidth are still consumed for maintaining the channel assignment due to the mobility of the nodes. Finally, without a dedicated control channel [14], the cost of broadcasting is high, as control packets must be duplicated and broadcasted in every data channel.

III. JOINT CHANNEL ASSIGNMENT AND ROUTING (J-CAR) PROTOCOL

For multi-channel multi-interface MANETs, J-CAR is designed to provide a more flexible utilization of interfaces and channels. In J-CAR, a channel is selected by an active node/ link (i.e. a node has data to send) during the route setup process. Among the multiple interfaces a node has, one of them is assigned with the pre-defined common control channel, called *control interface*. The control interface is responsible for routing and channel negotiation (detailed later). The control packets for coordinating data packet transmission in 802.11, RTS, CTS & ACK, are moved to the associated data channels. So the load on the control channel is expected to be light.

The spare capacity can be used to send data packets to increase throughput. But the control channel should be protected from congestion. To provide a tradeoff, we associate a data sending probability p with the control channel. When p=0, the control channel is not used to carry any data packets. When p=1, the control channel and other data channels have equal chance to be selected. Note that if a node has only a single interface, then data packets must be carried by the control interface and via the control channel. (The impact of using different values of p is studied in Section IV.)

Besides the control interface, other interfaces are for data packets, or *data interfaces*. An active data interface can work in either *send mode* or *receive mode*. To save energy, an inactive data interface can be turned off, or in *sleep mode*. In either send or receive mode, an interface is allowed to exchange *data-related control packets* (RTS, CTS & ACK) for coordinating data packet transmission. The send mode and receive mode differ as follows. In send mode, an interface is allowed to send *data packets* on *multiple* channels according to where the packet goes. As can be seen later on, J-CAR tries to use different channels to different downstream nodes. It is worthwhile to note that an interface in send mode is prohibited from receiving data packets.

In receive mode, an interface works on a *single* data channel. Although both *receiving* and *sending* data packets are allowed, configuring a receive mode interface to *send* packets is a last resort. We do so only if there are no idle or send mode interfaces at a node (more discussion in Section III.C). To minimize the *sender-receiver synchronization overhead*, the data sending must use the selected receiving channel. This is to avoid the situation that while an interface is sending on *another* channel, its upstream node does not know when it will be available for receiving again. The overhead involved in synchronizing the sender-receiver pair can be very high. (This also explains why a send mode interface is not used to receive data.) By restricting receiving and sending on the same channel, the status of the receive mode interface is always known by the upstream node/sender; this accredits to the broadcast nature of RTS/CTS/ACK packets.

Note that the control interface is not assigned with any working mode, as it always operates in the pre-determined common control channel.

A. An example

Before presenting the J-CAR protocol, let us preview its potential gain by an example. Fig. 1 shows a network consisting of five nodes. Each node has four interfaces, labeled as I_{ctrl}, I_1, I_2, and I_3. There are two active connections, A to D and B to E, intersect at node C. Consider the path A-C-D. With J-CAR, data channel 1 (ch1) is selected by link A-C, and occupying I_1 at both nodes. The two nodes communicate by having I_1 at node A in send mode and I_1 at node C in receive mode. Similarly, ch2 is selected by link C-D, occupying I_2 at C and I_1 at D. We can see that different channels are selected by the two links. This minimizes the mutual interference.

Next consider the second path B-C-E. With J-CAR, node B switches its I_1 to ch3 (send mode), and node C switches its remaining idle interface I_3 to ch3 (receive mode). In order not to interfere with the data receiving at I_1 and I_3, and assume I_{ctrl} is not preferred (with probability 1-p), node C selects I_2, and ch4 is selected for link C-E. Since I_2 at node C is in send mode, it can switch between ch2 and ch4 for data sending, at the cost of switching delay. Finally, node E switches its I_1 to ch4 (receive mode) for receiving. It should be noted that node C is receiving data using two interfaces. Compared with schemes that fix a particular interface for receiving in a pre-assigned channel [13, 14], the throughput performance is significantly improved due to the reduced packet collision probability.

B. J-CAR Protocol Details

Though the implementation of J-CAR may vary with the routing protocol used, the main idea is the same. Without loss

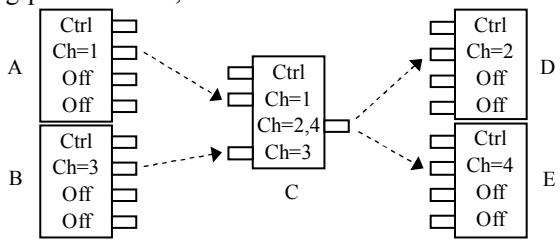

Fig. 1 Channel assignment and routing for two connections

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

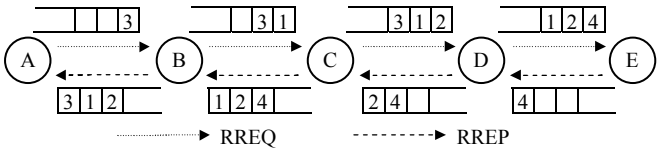

Fig. 2. Route setup with channel negotiation using J-CAR

of generality, we consider the integration of J-CAR and AODV [15] routing protocol below.

To minimize interference, different channels should be used within the k-hop neighborhood, where k (typically 2 or 3) defines the interference range. In J-CAR, the routing and channel negotiation are conducted in parallel, as explained by the example in Fig. 2. Assume there are 5 non-overlapped channels (including control), 3 interfaces per node and k=2. When node A initiates a route setup to node E, it broadcasts a RREQ (route request) according to AODV [15]. Extended by J-CAR, a channel (ch3) is proposed by node A for connecting to B. Its identity (ch3) is stored in a *channel list*, and piggybacked onto the RREQ. The channel list has a length of k+1 for storing the proposed channels within the interference range. When node B receives the RREQ, it checks the availability of the channel proposed by the pervious hop[1] according to its local *channel_usage_table*.

Channel_usage_table is defined as

$$usage\{i, j\} = \begin{cases} 0 & \text{if channel } j \text{ is not used by its } i\text{-hop neighbors} \\ 1 & \text{otherwise} \end{cases},$$

where $i \in [0, k+1]$ and $j \in [0, num\_channel-1]$. In particular, $usage\{0, j\}$ shows the channels being used by the local node. *Channel_usage_table* is updated by periodically broadcasting its 0^{th} to k^{th} rows to its immediate neighbors, which update the 1^{st} to $(k+1)^{th}$ row at the receiving node. With AODV, this information is piggybacked in the HELLO packet as a bitmap.

A channel (j) proposed by a pervious hop is acceptable to the local node if it is free within the interference range (k hops), or $usage\{i, j\}=0$ for all $i \in [0, k]$. (Otherwise, a *channel conflict* occurs and we will address this issue in the next section.) If the receiving node is not the final destination, it proposes a channel for the next hop based on the received channel list and its own *channel_usage_table*. Again, any free channel within the k-hop neighborhood can be selected. If not possible, we relax the constraint to find a free channel that is free within the (k–1)-hop neighborhood. When the 0-hop neighborhood is reached, i.e. all channels are being used by *the local node*, then the channel that is shared by the least number of immediate neighbors will be selected. By gradually relaxing the constraint, we take advantages of the fact that the interference strength reduces with distance.

Continue with the example in Fig. 2. Upon receiving the RREQ, node B marks ch3 as unavailable. Since the path has not yet been confirmed, ch3 is still indicated as free in its *channel_usage_table*. Then it chooses a free channel ch1 for using at the next hop. The RREQ is updated and forwarded. The same operation is performed at nodes C and D.

When node E receives the RREQ, it checks the availability of the proposed channel and sends a RREP (request response) to node A through the reverse path. J-CAR confirms the channel selection by piggybacking the *confirmed* channel list in the RREP, which also updates the corresponding entries in the *channel_usage_table*. When node D receives the RREP, it records the *confirmed* ch4 and the route to node E in its routing table. Then node D confirms ch2 and forwards the updated RREP to node C. Likewise, the RREP is delivered back to node A. The path from A to E, with *different* channels selected within the interference range of any participating node, is formed.

Note that all the routing packets above are transmitted in the control channel. Since RREQ is delivered by broadcasting, it may reach the destination via multiple paths. If a path is not confirmed (by receiving a RREP), the records of the reverse route and the corresponding proposed channel list are flushed when the timer expires (as defined in AODV [15]).

The advantages of channel negotiation for each *active link* over the node-based channel pre-assignment [13, 14] can be seen from the example in Fig. 3. Suppose there are two data channels, and four mobile nodes within the transmission range of each other. Fig. 3a shows a possible scenario of a node-based channel pre-assignment, in which ch1 is assigned to A and C, and ch2 to B and D. If there are two connections established from A to B and from C to D respectively, both connections use ch2. The performance is equivalent to a single channel network. If J-CAR is used (Fig. 3b) and assume ch1 is selected by the A-B connection. In setting up the C-D connection, ch2 will be chosen as both C and D know that ch1 has been engaged. The interference is minimized.

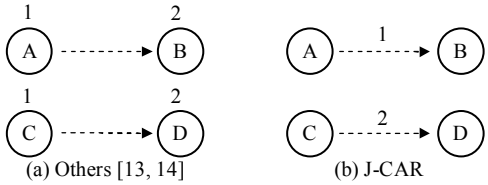

Fig. 3. Comparing channel assignment schemes

C. Handling Channel Conflict

There are cases that the proposed channel (in RREQ) cannot be approved, we call it a *channel conflict*. With J-CAR, a node is responsible to select a data channel for the pervious hop in case of channel conflict. There are two possible causes for channel conflict. The first one is illustrated by the example in Fig. 4, which is due to the absence of idle interfaces. In Fig. 4, node F is setting up a route to H, which crosses an active route A-B-C-D-E. Assume k=2 and 3 interfaces per node. With J-CAR, node F proposes ch4, which is approved by G. Then G finds that channels 1, 3, 4 and 5 are busy (within k hops), so it proposes ch2. This causes a *channel conflict* in node B, as all its 3 interfaces are respectively occupied by ch0 (control channel), ch5 (receive mode) and ch1 (send mode). Since a *send mode* interface is not allowed for receiving, node B can only receive in ch0 or ch5. To solve this conflict, node B selects ch5 for node G. It further selects ch6 for the next hop and broadcasts the updated RREQ.

[1] Assume node A proposes channel ch3, which is free as seen by node B but not by node C. This *must* due to the fact that ch3 is used in node C's interference range, not covered by that of node B, e.g. ch3 is used by node E. Since nodes A and E are outside each other's interference range, node C does not need to check the second entry ch3 in its received list. In fact, checking the first entry in the list is sufficient.

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

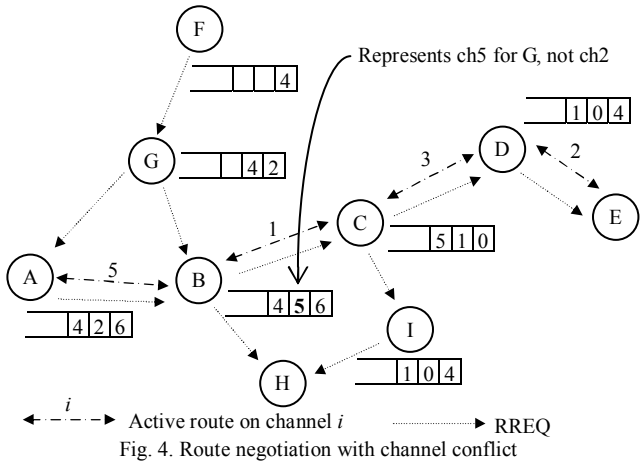

Fig. 4. Route negotiation with channel conflict

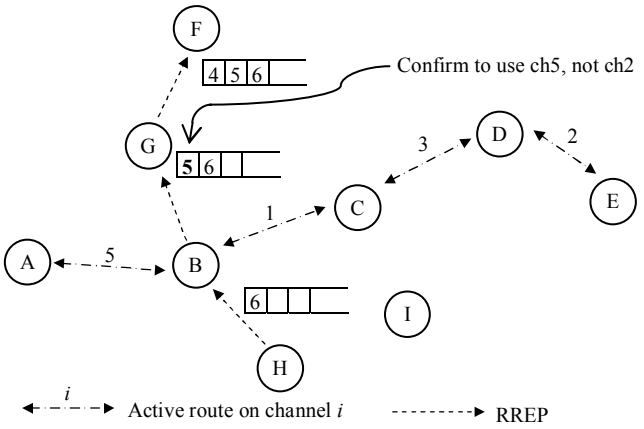

Fig. 5. Route reply confirms the channel selection

When node G receives the RREP (Fig. 5), it notices that the confirmed channel is ch5, instead of its proposed ch2. Since a send mode interface is allowed to switch between channels, ch5 is accepted. On the other hand, if all data interfaces of node G are in receive mode, its proposed ch2 cannot be changed. In this case, it must set the *force_bit* in the updated RREQ header to 1, indicating that the proposed channel must be honored. If node B cannot support ch2, it simply drops the RREQ packet, and the route may be established in another path, e.g. F-G-A-B-H.

Another possible cause of channel conflict happens when the proposed channel is occupied by some node within the receiving node's interference range. In this case, the receiving node selects another channel for its previous hop, by using all the rows in its local *channel_usage_table*, i.e. all $i \in [0, k+1]$. The additional $(k+1)$-hop channel information contains the channel occupancy of the pervious hop's k-hop neighborhood.

IV. Performance Evaluations

In this section, we evaluate the performance of J-CAR by simulations using ns-2. J-CAR is compared with PCAM [13] because its architecture is similar to ours, with multiple interfaces and a dedicated broadcast channel (i.e. our control channel), which can be used for data transmission as well. MCR [14] is not compared because it does not have a control channel, which has been shown to be inefficient.

The parameters of PCAM are set according to [13]: the signal capture threshold 8dB, the effective transmission range 200m and the carrier sense range 550m. While the traditional values, 10db, 250m and 550m respectively, are used in J-CAR. As PCAM is designed with 3 interfaces per node in mind, we also use 3 interfaces in J-CAR. In all simulations, we set k=2; the channel bandwidth is assumed to be 1Mbps; all data flows carry CBR traffic with packet size of 512 bytes; and the channel switching latency is set to 100μs. Nodes are randomly placed in the simulated area. Non-overlapping sender-receiver pairs are randomly selected. AODV protocol is used for routing. Each simulation result is obtained by averaging the results over 10 random topologies.

A. Single-hop topology

Our first simulation consists of 4 channels, 8 data flows and 16 nodes within a 200x200m^2 area. Fig. 6 shows the aggregate throughput performance (of the 8 flows) with different packet arrival rates, in which 'J-CAR=0' stands for our J-CAR protocol without using the control channel for data packets. As PCAM allows data transmission in the broadcast channel, setting p=0 makes J-CAR performs slightly poorer than PCAM. On the other hand, J-CAR performs better than PCAM when p>0. At packet arrival rate of 200 packets/sec, J-CAR gives a throughput gain of 7.6%, 8.8%, and 8.1% for p=0.3, 0.5 and 0.8 respectively.

This performance gain is mainly contributed by the channel negotiation mechanism. In PCAM, receiving channels are pre-assigned to all nodes, uniform distribution of data flows among multiple channels is not guaranteed (Fig. 3a). With J-CAR, the 8 data flows are evenly distributed over the 4 channels. This minimizes the co-channel interference.

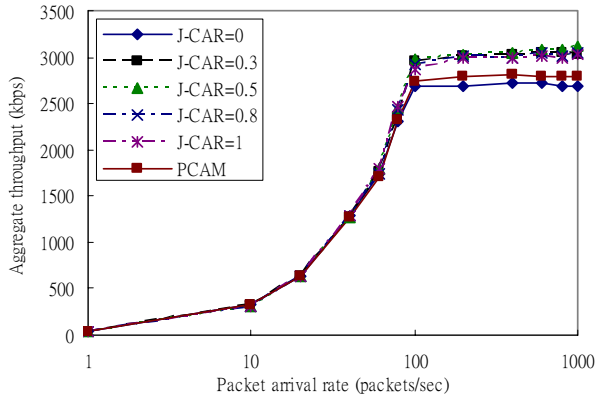

Fig. 6. Throughput vs packet arrival rate;16 nodes, 8 flows, and 4 channels

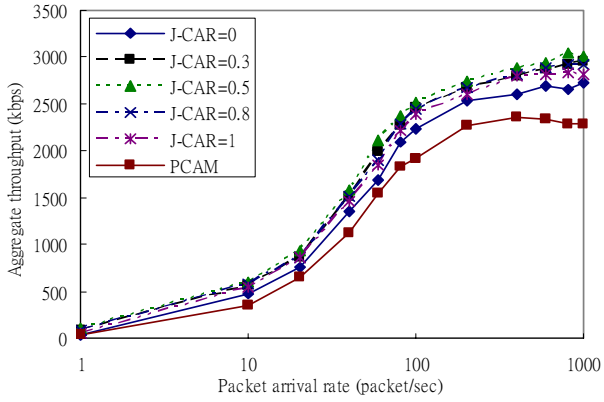

Fig. 7. Throughput vs packet arrival rate; 50 nodes, 25 flows, and 12 channels

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

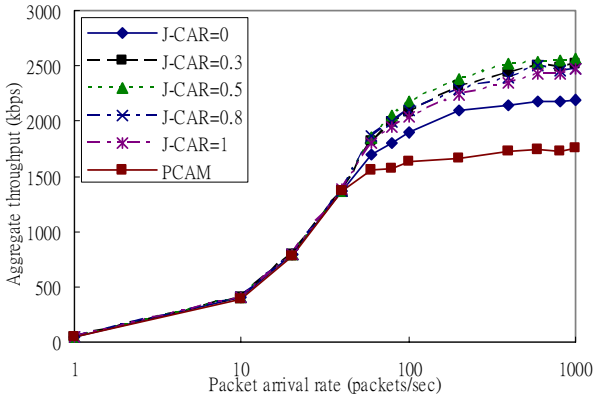

Fig. 8. Throughput vs. packet arrival rate; 100 nodes, 10 flows, and 12 channels

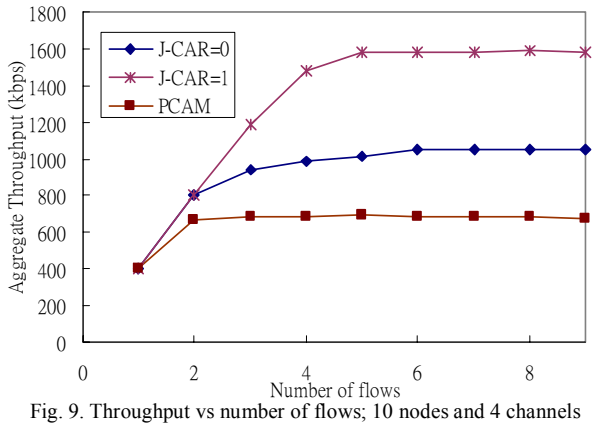

Fig. 9. Throughput vs number of flows; 10 nodes and 4 channels

B. Multi-hop topology

To study the performance in multi-hop networks, we assume there are 12 channels with 50 nodes and 25 flows randomly placed in a 700x700m^2 area. The aggregate throughput performance is given by Fig. 7. J-CAR outperforms PCAM by 33% when p=0.5, at 1000 packets/sec. Even when p=0, J-CAR performs better than PCAM by 19% at the same packet arrival rate. This shows that the node-based channel pre-assignment adversely affects the performance more seriously in multi-hop transmission.

Another simulation is conducted in a denser multi-hop wireless network, with the same settings except that there are 100 nodes and 10 simultaneous flows. The simulation results are given in Fig. 8. As expected, PCAM performs even worse in this case, due to many nodes using the same channel as their primary receiving channel. At the packet arrival rate of 1000 packets/sec, J-CAR with p=0.5 outperforms PCAM by 46%.

C. Usage of multiple interfaces

In this simulation, we assume there are 4 channels and 10 nodes are randomly placed within the transmission range of each other. One node is selected to be a receiver, and the remaining nodes send CBR traffic to it at 100 packets/sec each. Fig. 9 shows the aggregate throughput performance of different protocols with varying number of simultaneous flows. As PCAM only allows a node to receive data packets by its primary interface, the performance is only as good as a single-channel single-interface scheme. In contrast, J-CAR uses two interfaces to receive data when p=0; and when p=1, J-CAR uses all three interfaces for receiving, which gives a 130% performance improvement when there are 5 or more flows.

V. Conclusion

In this paper, we proposed a joint channel assignment and routing protocol (J-CAR) for multi-channel multi-interface wireless ad hoc networks. In J-CAR a channel is negotiated by each active link during the route setup process. unlike existing protocols, J-CAR has the following major features: a) a pre-determined control channel is used by every node for routing and channel negotiation; b) control packets for data transmission are carried by the negotiated data channels; c) the spare capacity on the control channel can be utilized for data transmission; d) an interface is free to change its working modes between send and receive; and e) an interface is free in tuning to any data channels for data sending or receiving at the cost of switching delay. Through extensive simulations, we showed that J-CAR performs much better than the existing protocol PCAM.

Acknowledgment

We would like to acknowledge Dr. S. H. G. Chan and Dr. J. He for providing the ns-2 simulation codes for their LCA protocol [7].

References

[1] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Trans. on Information Theory*, vol.46, issue 2, March 2000, pp.388-404.

[2] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of Interference on Multi-hop Wireless Network Performance," *ACM MobiCom'03*.

[3] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks," *ACM MobiCom'04*, pp. 216-230.

[4] J. So, and N. Vaidya, "Multi-Channel MAC For Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using a Single Transceiver," *ACM MobiHoc'04*, pp. 222-233.

[5] M. X. Gong, S. F. Midkiff, and S. Mao, "Design Principles for Distributed Channel Assignment in Wireless Ad Hoc Networks," *IEEE ICC'05*, vol. 5, 16-20 May 2005, pp. 3401-3406.

[6] P. Bahl, A. Adya, J. Padhye, and A. Wolman, "Reconsidering Wireless System with Multiple Radios," *ACM Computing Communication Review*, July 2004.

[7] J. He, J. Chen, and S. H. G. Chan, "Extending WLAN Coverage Using Infrastructureless Access Points," *IEEE HPSR'05*, pp. 162-166.

[8] M. Alicherry, R. Bhatia, and Li (Erran) Li, "Joint Channel Assignment and Routing for Throughput Optimization in Multi-radio Wireless Mesh Networks," *ACM MobiCom'05*, pp. 58-72.

[9] A. Raniwala, K. Gopalan, and T. Chiueh, "Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks," *ACM MC2R*, vol. 8, no. 2, April 2004.

[10] A. Raniwala and T. Chiueh, "Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network," *IEEE Infocom 2005*, vol. 3, 13-17 March 2005, pp. 2223-2234.

[11] 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," *The Computer Journal*, vol. 45, 2002, pp. 101-110.

[12] C. Xu, G. Lui, W. Cheng, Z. Yang, "Multi-Transceiver Multiple Access (MTMA) for Mobile Wireless Ad Hoc Networks," *IEEE ICC'05*, vol. 5, 16-20 May 2005, pp. 2932-2936.

[13] J. S. Pathmasuntharam, A. Das, and A. K. Gupta, "Primary Channel Assignment based MAC (PCAM) – A Multi-Channel MAC Protocol for Multi-Hop Wireless Networks," *IEEE WCNC'04*, pp. 1110-1115.

[14] P. Kyasanur and N. H. Vaidya, "Routing and Interface Assignment in Multi-Channel Multi-Interface Wireless Networks," *IEEE WCNC'05*, vol. 4, 13-17 March 2005, pp. 2051-2056.

[15] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," *Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, Feb. 1999, pp. 90-100.

1-4244-0357-X/06/$20.00 ©2006 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.