Multi-channel OLSR with a Dedicated Control Interface

Yi Qu, Chung-Horng Lung
Carleton University
1125 Colonel By Drive, Ottawa, Canada K1S 5B6
{quyi, chlung}@sce.carleton.ca

Anand Srinivasan
EION Inc.
945 Wellington Street, Ottawa, Canada K1Y 2X5
anand@eion.com

Keywords - Multi-channel, dedicated control, OLSR, MANET

Abstract - Most Mobile Ad hoc Networks (MANETs) research is based on a single channel. Nonetheless, multi-channel is an approach that can increase network capacity by transmitting traffic on different channels in an interference area. Channel assignment and channel switching, however, are two main challenges when the common communication channel is no longer available using the multi-channel scheme. We propose to separate the control and the data traffic on different interfaces of a node. The dedicated control interface of all nodes share a common channel for routing and multi-channel control. The data interface, on the other hand, uses multiple channels for data traffic. This separation makes both control and data traffic efficient. We extended the OLSR (Optimized Link State Routing) protocol to support a simple channel assignment mechanism and improved the queue-based channel switching scheme. Simulation results with NS-2 indicate significant performance improvement on throughput and end-to-end delay.

1. INTRODUCTION

Research of mobile ad hoc networks has focused on common channel networks in which each node has a single interface using the same channel. A common channel is desired for simple routing control in multi-hop networks, but radio interference on the common channel result in low throughput because only one traffic flow can be served at a time in the interference area. To overcome the interference problem, a natural approach is to use multiple channels in which nodes work on different channels in an interference area.

It is difficult to implement routing and multi-channel control in a multi-channel environment, because a common channel network does not exist for reliable and efficient control communications. Multi-channel without a common channel interface has been studied extensively, as will be described in Section 2. But there is not a prominent solution yet.

Inspired by dual-radio research by Bahl, et al. [Bahl04a] and concept of Common Channel Signaling that has been widely deployed in the wired telephone networks; we propose an approach that uses a dedicated interface running on a common channel exclusively for control. Meanwhile, two data interfaces on each node are advocated; the data interfaces using multiple channels are for data traffic. Data and control interfaces are then tuned to perform different tasks cooperatively. Our contributions in this paper include:

- Propose a multi-channel solution that can be deployed on legacy MAC protocol.
- Extend the OLSR routing protocol to support multi-channel without introducing extra overhead.
- Develop an adaptive queue-based channel switching mechanism which can be converted to per-packet switching at light load level for lowest latency.
- Improve TCP performance by sending back ACKs on the control interface.

Our simulation results from NS-2 showed up to 6 times throughput and latency improvement in random generated topologies. We have experimented various scenarios to find out the impact of traffic load, topology and number of channels etc. on performance. In cases where there is higher interference, multi-interface multi-channel (MC) can achieve higher improvement on performance than that of the traditional single interface single channel (SC) networks.

The rest of the paper is organized as follows. Section 2 presents some background and related works. Section 3 discusses our proposed multi-channel OLSR with a dedicated control interface. Section 4 demonstrates the simulation and some results. Finally, section 5 is a summary and a discussion of future research areas.

2. BACKGROUND AND RELATED WORK

Multi-channel has been studied tremendously in networks with single-interface node [So04a, Bahl04b, So04b] or multi-interface node [Wu00, Kyasanur04, Lee05], and from MAC layer [Wu00, So04a], link layer [Bahl04b] to network layer [Kyasanur04, Lee05, So04b]. Channel assignment and channel switching are two key problems that have to be dealt with in multi-channel networks. The purpose of channel assignment is to instruct a node which channel to use; thus it avoids interference with other nodes. Channel switching deals with the coordination between a sender and a receiver; they have to switch to the same channel at almost the same time before packet transmission starts.

2.1 Channel Assignment Approaches

There are mainly two channel assignment approaches [So04b]: channel assignment to flow or to node. Assigning channel to flow, used by Multi-Channel Routing Protocol (MCRP) [So04b], means that all the nodes on the route are assigned a common channel. A node receives and transmits on the same channel; therefore channel switching delay is avoided. However, this channel assignment approach couples with the routing protocol that makes both channel assignment and routing algorithm complicated. Furthermore, the route setup fails sometimes when there is no common free channel available on all nodes along the route.

In the approach of assigning channel to node, normally receiver-based channel assignment (RCA), each node is assigned a receiving channel which is unique in a certain interference area. A node is aware of its neighbors' receiving channels. The packet sending node will transmit on the receiving channel of the next hop node. This approach decouples route establishment and channel assignment. It makes routing algorithm simple and therefore is embraced by more researchers [Kyasanur04, Lee05]. But channel switching is a challenge because a node transmits and receives on different channels, and for the same reason, there is extra channel switching delay. RCA has two limitations. It requires a large number of channels to keep each node with a different channel with its neighbors. It can not completely avoid channel confliction by considering only receiver's channel uniqueness (see detail in section 3.2).

As a special variant of the second approach, negotiating channel assignment (NCA) is an on-demand channel assignment scheme. Sender and receiver node exchange message and choose a common free channel for transmission. NCA can overcome the two limitations of RCA, but negotiating has high overhead. As we know, only some modified multi-channel MAC protocols [Wu00, So04a] are using NCA approach.

2.2 Control Communications in Multi-channel Networks

In a multi-channel network, nodes may stay in different channels. It is difficult for the channel and routing control messages to communicate between nodes in the absence of a common communication channel. Many research efforts treat control and data packets on the same interface in a multi-channel environment. They transmit control packets either on a common channel in a synchronized time window [So04a] or sending control messages on all channels [So04b, Kyasanur04]. Someone [Wu00, Bahl04a, Lee05] suggests separate control packet on a dedicated control interface.

2.2.1 Mixed Data and Control Interfaces

MAC protocol MMAC [So04a] uses a single transceiver. Before packet transmission, nodes negotiate a channel with the *Ad hoc Traffic Indication Messages* (ATIM) in a synchronized window on a common default channel. Time synchronization is difficult in MANET. MCRP [So04b] is a reactive routing protocol coupled with flow-based channel assignment. Channel usage information is carried in the Route Request (RREQ) packet, which is broadcast on all channels over the whole network. Kyasanur, et al. [Kyasanur04] use two interfaces on each node: one interface is fixed on the assigned receiving channel, another interface transmits data on multiple channels. Receiver based channel assignment is simply realized with the "Hello" message. Control messages are still transmitted on all channels. Duplicating messages on all channels causes large routing overhead and latency in effective message distribution.

2.2.2 Dedicated Control Interface

Dedicated control interface had been proposed before, but this approach is still under developed. Wu, et al. proposed a multi-channel MAC protocol [Wu00]. Each mobile node is equipped with two half-duplex transceivers. RTS/CTS with channel usage information are exchanged on the control interface to negotiate a channel to be used before each data packet transmission. Per-packet channel assignment is expensive in RTS/CTS overhead and computing cost.

Dual-radio proposed by Bahl, et al. [Bahl04a] uses a low power control radio to realize power control in MANET. It shed light on the approach of using a dedicated control interface for more efficient control mechanism.

Recently, Lee, et al. proposed Mutli-Channel DCDV (DSDV-MC) [Lee05] that uses two half-duplex interfaces: one interface is on the common channel for routing control traffic, another is for data traffic. The channel assignment is coupled with routing protocol. Routing table updates are generated if there is any change on the node channel. Routing protocol is complex and routing table is unstable due to the frequent updates. Because only one data interface is used, DSDC-MC actually is used to achieve node channel switching coordination through routing table updates whenever a channel switching occurs for transmitting or receiving. Frequent routing table updates involve a lot of routing messages broadcasted to other nodes. The main purpose of using a common channel interface is to offload the heavy routing control traffic. Lee, et al. did not present results on latency performance, but it is expected to be high because each channel switching results in a routing table update.

3. MULTI-CHANNEL OLSR WITH A DEDICATED CONTROL INTERFACE

Data traffic needs multi-channel for a non-interference environment, but effective control communication needs a common channel. It is not easy to meet both requirements at the same time. Therefore, we propose a solution to use a dedicated control interface to facilitate effective control communications, while data traffic is separated on multi-channel data interfaces. Using a dedicated common channel control interface avoids the issues in a multi-channel context which needs complicated control mechanisms. It also makes critical control packets distributed effectively on the light loaded interface without contention with data packets.

Without extra overhead, we extend the OLSR routing protocol that can proactively maintain the routing table and the receiver-based channel assignment. OLSR is only running on the control interface. We assume that data and control interfaces have the same transmission range and share the same topology; therefore, routing information learnt on the control interface can be used for data interfaces.

Beside the control interface, each node is equipped with two half-duplex data transceivers/interfaces. It not only realizes full-duplex data transmission for better throughput and latency performance, but also eases the multi-channel assignment and Tx/Rx channel switching coordination.

3.1 Receiver-based Channel Assignment

The approach of assigning channels to flows is not only complicated for coupling with the routing protocol, but also fails to resolve channel confliction along the route. For example in Fig. 1, all nodes on flow S to D are on channel 1. When B sends packets to C, its interference prevents A from receiving from S at the same channel. While in the case of assigning channels to nodes, both A and C can receive at the same time on different channels.

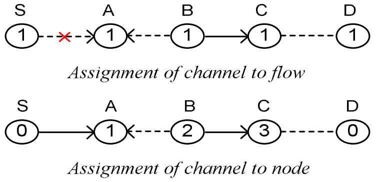

Figure 1. Channel assignment approaches

Previous works [Kyasanur04, Lee05] use receiver-based channel assignment (RCA) approach. They assign each node with a unique channel to one-hop neighbors to avoid interference like the example shown in Fig. 2(a) where B's transmission on channel 1 suspends A and C. However, we argue that a node should keep its channel uniqueness in a larger neighborhood to avoid channel confliction completely.

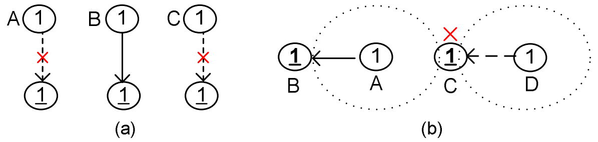

Figure 2. One (a) or two hop (b) neighbor nodes with identical channel

For example in Fig. 2(b), B and C are two-hop neighbors and are assigned the same receiving channel (ch1), then packets A→B and D→C will use the same transmission channel. They can not send at the same time otherwise packets conflict on node C. To avoid the channel confliction in this scenario, B and C should be assigned different channels.

RCA channel assignment scheme depends only on topology and considers only receiving channel uniqueness. When a certain traffic scenario applies, transmitting channel may cause channel confliction as shown in Fig. 2(b). Simply increasing the range of channel uniqueness requires a lot of different channels. At the same time, even three-hop neighbors with the same channel may not avoid channel confliction as shown in Fig. 3.

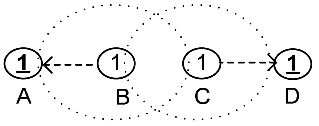

Figure 3. Three hop neighbor nodes with identical channel

On-demand NCA can overcome this problem because it considers both transmitting and receiving channels. Although NCA does not have these two limitations (section 2.1) of RCA, a new on-demand channel assignment protocol is required. It increases the control complexity and overhead. Traffic setup is slow because it must wait until the channel assignment is done after the on-demand negotiation process. RCA is dependent on topology only, so it is simple to be integrated with the OLSR to realize proactive routing control and channel assignment. Therefore, we use RCA in this paper. Node receiving channel is kept unique in two-hop range neighborhood to balance the channel confliction possibility and the total number of required channels.

3.2 Multi-channel OLSR Routing Protocol

Most multi-channel extended routing approaches [Kyasanur04, So04b] are based on on-demand protocols. We, however, adopt the proactive routing protocol to make better use of the dedicated control interface to provide quick routing provision. DSDV-MC [Lee05] is a multi-channel extension to proactive routing protocol DSDV. OLSR is possibly the best proactive MANET routing protocol for its low overhead.

Reserved	***orig_chan***	Htime	Willingness
neighbor_chan_bitmap 11 00 00 01			
Link Code	Reserved	Link Message Size	
Neighbor Interface Address			

Figure 4. New Hello message format

We extend the OLSR Hello message (Fig. 4) with two new fields *orig_chan* and *neighbor_chan_bitmap*. Four bytes of extra overhead per Hello message is introduced, but it is negligible. The field *orig_chan* specifies the node (receiving) channel of the Hello message originator. It uses part of the *Reserved* space of standard Hello message. The *neighbor_chan_bitmap* field specifies channel usage status of one-hop neighbors. Status of each channel is represented by 2 bits with the meaning defined in Table 1. The field (32-bit) can hold status for up to 16 channels.

Table 1. Definition of channel status

2bit channel status	*Definition*
00	Free channel
01	Used by one neighbor node
11	Used by multiple neighbor nodes

We extend the OLSR *neighborTuple* structure with two new fields *N_neighbor_channel* and *N_2hop_channel_bitmap*. They are corresponding to "*orig_chan*" and "*neighbor_chan_bitmap*" of the received Hello message respectively. This extension is used for the new Hello message generation and channel confliction detection (will be explained in section 3.3). When a node generates a Hello message, it sets the "*orig_chan*" field with its own node channel, and it iterates *N_neighbor_channel* field of its all *neighborTuples* to construct the "*neighbor_chan_bitmap*" field. For example, if a node has only one neighbor assigned ch_0 (i.e., *N_neighbor_channel* field of one *neighborTuple* is "0"), two neighbors assigned ch_3, and ch_1&2 are not used by any neighbors, then part of *neighbor_chan_bitmap* will be “11 00 00 01” as shown in Fig. 4.

Our OLSR is now able to support multi-channel routing with above extensions. When a node is to send packets, it looks up the routing table to find the next hop, and then reads the *N_neighbor_channel* filed of the corresponding *neighborTuple* to identify the transmission channel.

3.3 Channel Assignment Algorithm

We choose receiver-based channel assignment approach as explained in section 3.1. A node initiates with a random channel on start up. It updates its node channel when detects channel confliction in processing the received Hello messages. For example, node B (at ch_b) receives a Hello message from its neighbor A. One-hop neighbor channel confliction occurs if ch_b equals to *orig_chan* within the Hello message. Two-hop neighbor channel confliction exists if ch_b status in *neighbor_chan_bitmap* of this Hello message is “11”. It means, besides B, node A has other one-hop neighbor(s) on ch_b too.

On detecting channel confliction in processing the Hello message, a node randomly selects a free channel which is not used by its one and two-hop neighbors. A node finds all free channels by iterating all *neighborTuples.* A free channel is not used by any *neighborTuple* as *N_neighbor_channel* and its status is “00” in *N_2hop_channel_bitmap* field of all *neighborTuples.* If no free channel is available, the node stays at the current channel. If a node selects a new channel, it will not switch its receiving interface to the new channel until it has sent out a Hello message to notify its neighbors about its new channel. In this way, we minimize the lag between node channel switching and neighbor’s *neighborTuple* update; hence it decreases the possibility that packets are sent on the old channel, but the next hop node had switched to a new receiving channel.

3.4 Transmitting Channel Switching

We use two data interfaces on each node as [Kyasanur04] does. With dedicated Tx and Rx interfaces, sender and receiver nodes avoid Rx/Tx coordination problem that is a challenge for single data interface node.

The Tx interface of a node switches to the receiving channel of its next hop node. Wireless interfaces need a few hundred microseconds in channel switching [Kyasanur04]. It is expensive if Tx channel switching is performed on per-packet basis. Queue based channel switching was proposed before in [Kyasanur04], we have improved it by adaptively adjusting channel switching frequency with respect to the traffic load level, thus to keep latency as low as possible.

The adaptive adjustment is described as follows. Each Tx channel is associated with a queue. A packet to be transmitted on a channel is enqueued in the corresponding queue first, but not sent out until the Tx interface switches to this channel. When the Tx interface has worked on a channel for a predefined maximum time (QmaxTime in seconds) or the current queue is empty, the node gives other channels/queues a chance to be served. The queue that holds the oldest packet in terms of arrival time is selected, and then the Tx interface switches to that channel. The node will continue transmitting packets on the current channel if no other queues have packets waiting. If all queues are empty, the routing agent changes to the IDLE state and it will switch to the corresponding channel where it receives a new outgoing packet.

QmaxTime is the upper bound for each cycle, but Tx interface does not stay on a channel up to that time if the queue becomes empty first, therefore the channel switches more frequently (could even converge to per-packet switch) at lower traffic load. This is an adaptive process to increase switch frequency (decrease latency) as long as the switching overhead is affordable at low traffic load level.

3.5 Effective TCP with Control Interface

TCP traffic flow is composed of forwarding TCP data packets and backward ACK packets. If we treat ACKs like other data packets, it may experience a long latency waiting in the Tx queue. For example in Fig. 5, after B receive TCP packet from A, it need send ACK packet (e.g., ACK1) back to A on A’s Rx channel (ch1), if at the same B is transmitting CBR packets on ch2 to node D, ACK1 has to wait in the queue of ch1. Alternatively, if we send the ACK (e.g., ACK2) back on the same interface/channel where the TCP packet was received, ACK could be dropped if the TCP sender node (A) has switched its Tx interface to another channel (ch2) for another CBR traffic to node C.

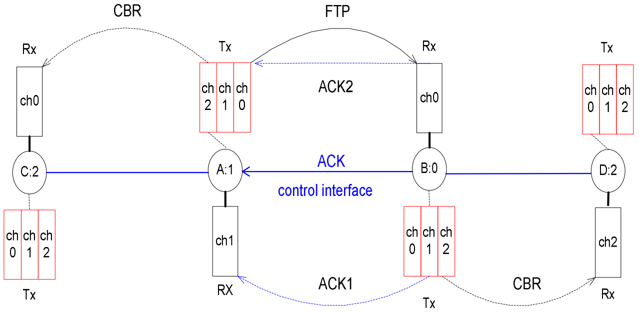

Figure 5. Sending TCP ACKs on the control interface

For efficient ACK/TCP packet transmission in multi-channel environment, our solution is to transfer the ACKs on the dedicated control interface. Fast ACK transmission is assured on the light loaded common control channel. Thus we can expect enhanced TCP traffic. ACK packets are small (40 bytes) unicast packets, so this mechanism is scalable even there are many TCP flows using the common channel for their ACK transmission.

4. SIMULATION AND RESULTS

We have conducted simulation with NS-2.28 [NS2] and an OLSR implementation [OOLSR]. Experiments were designed to compare multi-channel (MC) and single-channel (SC) performance with respect to various factors. We have also investigated the behaviors of queue-based channel switching and TCP performance in MC. Unless otherwise specified, default scenario setting is described below.

- Wireless interface: 802.11b, data rate of 11Mbps, 11 channels, NS-2 default transmission range.
- Topology: 5 random topologies (s1~s5) generated by NS with below characteristics - 50 nodes randomly located in a 700m x 700m area, random way-point mobility model with the (max, min) speed of (15, 5) m/s and pause of 30s.
- Traffic flow: CBR flows between random node pairs at rate 200 packet/s; UDP packets with size of 512 bytes; simulation duration of 110s; and traffic start at 10s.
- Queue-based channel switching options: queue length: 30, QmaxTime: 10ms, channel switching delay: 200 microseconds.

4.1 MC vs. SC Performance Comparison

When higher interference degrades the performance of single channel (SC) networks, the multi-channel (MC) networks should not be impacted as much as SC networks. The network interference level depends on the topology and the traffic pattern. We first experimented with a certain number of traffic flows on different topologies, and then we changed the number of flows and traffic rate for different network interference levels.

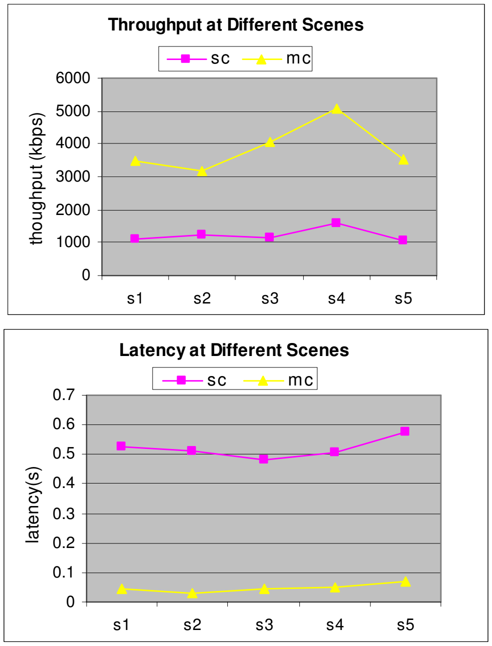

Figure 6. MC vs. SC at different scenes of topology

If we use "ns-random 0", every time running get different result because the random channel selection get different channel assignment for same topology and traffic/connection pattern, the confliction status is completely different, the effect is just same as using a new topology in every time run.

Fig. 6 shows the results of 10 CBR flows in 5 topology scenes. In this set of scenarios, MC achieves 3~4 times higher throughput than SC and only about 10% of the latency of SC. We believe the throughput fluctuation in MC is caused by the topology difference in *"flow intersecting degree"*. If there are more flows intersecting at a node, the node has higher *"flow intersecting degree"*. For example in Fig. 7(a), two flows (A-B-E & D-B-C) intersect at B. in Fig.7(b), nodes F and B forward only one flow each. Fig. 7(a) has higher *"flow intersecting degree"* than that of Fig. 7(b). The intersecting node B in (a) can only forward packets for one flow (to E or C) at a time, and at B's Rx interface, it can only receive packet from either A or D at a time, thus it becomes the bottleneck and decreases the throughput. In contrast, all 4 links (A-F, F-E, D-B and B-C) can transmit simultaneously on different channels. Therefore, to further increase the throughput in MC, it is for the routing protocol to discover node disjoint routes.

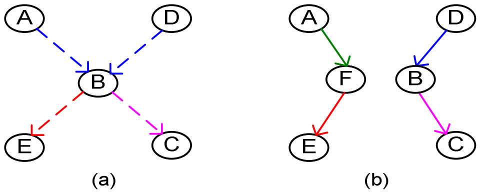

Figure 7. Avoid intersecting node in MC

The throughput of SC is not as sensitive to the flow intersecting degree as MC which has larger throughput fluctuating curve. In SC, if two flow paths have nodes in the interference area, only one node can serve its flow traffic, this is the same effect as flow intersecting in MC. For example, in Fig.8 (a) although two flows have no intersecting node, node F and node B can not send/receive packet at the same time. The intersecting point (node) in MC is enlarged to node interference area in SC. Two flows are not "intersecting" only when they have no node in another flow's interference area as demonstrated in Fig.8 (b). So, SC's *"flow intersecting degree"* is not sensitive to topology changes.

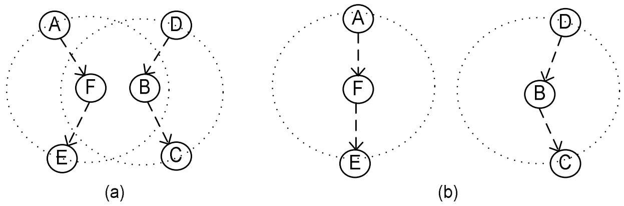

Figure 8. Node interference in SC

Fig.9 is the result of varying the number of traffic flows in a certain topology scene. A connection pattern with greater number of flows is formed by adding new flows between random selected nodes. MC has almost linear increase throughput and slightly impacted latency. In contrast, SC has fast increasing latency, and the throughput is saturated at a low level. More traffic flows increase the network contention level in SC, which results in longer packet back-off when nodes detect busy media before transmitting attempt.

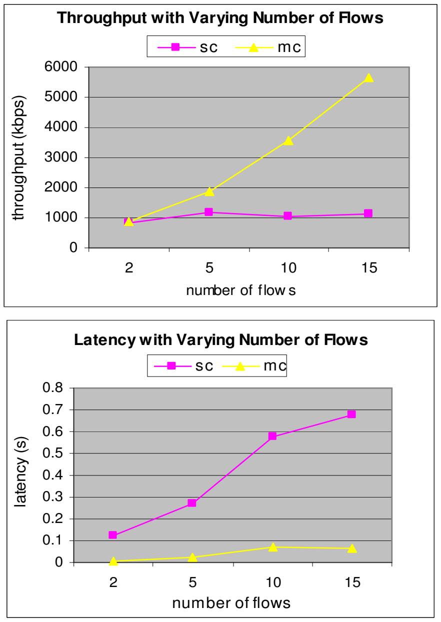

Figure 9. Performance with varying number of flows

Fig. 10 shows the result of varying the flow rate of the 5 CBR flows on the same topology. We can see that SC has saturated load at rate of about 100 packet/s, while MC does not saturate even at 350 packet/s. The slowly increasing latency curve in MC is due to queue delay as load increasing. MC latency is very low (<4ms) at light load (<100pps) because of the adaptive nature of queue-based channel switching.

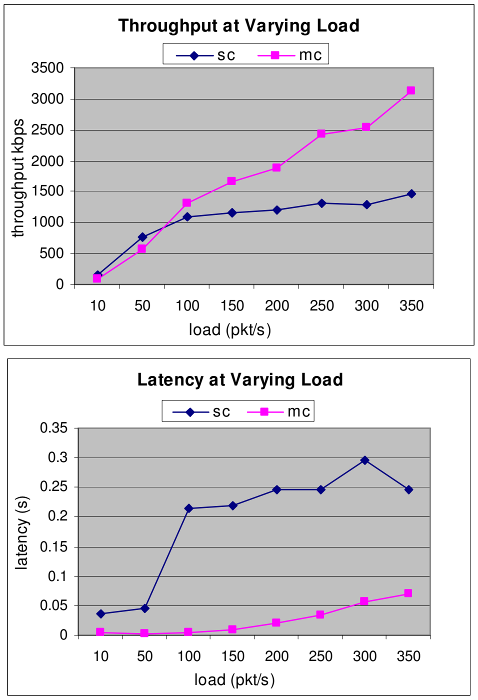

Figure 10. Performance with varying flow rate

Topology, connection pattern (number of flows) and traffic load (flow rate) are three main factors determining the network interference level for SC. The effect of these factors on MC, however, is quite different. As the above three sets of experiments indicate: the performance of MC improves more at higher interference level. Up to 5.5 times higher throughput was obtained in our simulation, we can expect more improvement for higher loads, more flows in the low *"flow intersecting degree"* topology.

4.2 Queue-based Channel Switching Performance

When a node forwards two traffic flows on different channels, its Tx interface need to switch between two channels. In random topology, it is not easy to tell which node is at flow intersection. We use a specific topology to investigate channel switching performance. One source node S has two one-hop flows to D1 and D2 (i.e., D1← S →D2). Four sets of test are conducted: Qmaxtime of 10ms, channel switching delay of 100/200/300 microseconds; Qmaxtime of 2ms, channel switching delay of 300 microseconds (300µs* in Fig.11). Each set exercises with varying traffic load, but queue length is fixed at 30.

Fig.11 indicates, the every low latency (<3ms) at light load (<200pps) prove the effectiveness of adaptive mechanism of the queue-based channel switching – node does not stay on a channel for the whole Qmaxtime (10ms) if no more packets are in the queue. At high load, compared with queue buffer delay of Qmaxtime 10ms, latency due to channel switching is small. It is reasonable because queue-based channel switching decrease channel switching frequency and few hundred microseconds channel switching delay is also relatively small. Faster channel switching with smaller Qmaxtime (2ms) increases latency, but not significantly.

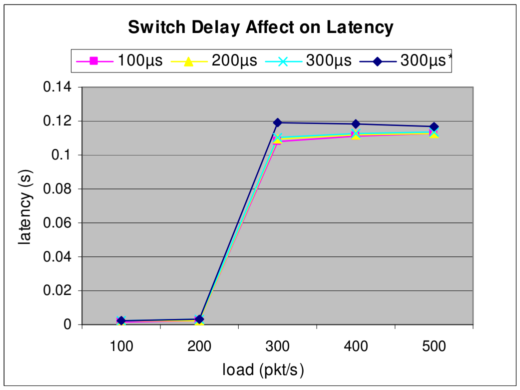

Figure11. Channel switching delay & Qmaxtime with latency

4.3 Number of Channels and MC

RCA need a large number of channels to maintain an interference free multi-channel network. Dense network requires more channels. Furthermore, we can only use non-overlap channels simultaneously in the interference area. 802.11b/g has 3 non-overlap channels in 2.4GHz band; 802.11a has 12 in 5.2/5.8 GHz band. The total number of available channels is an important factor for a network at certain node density.

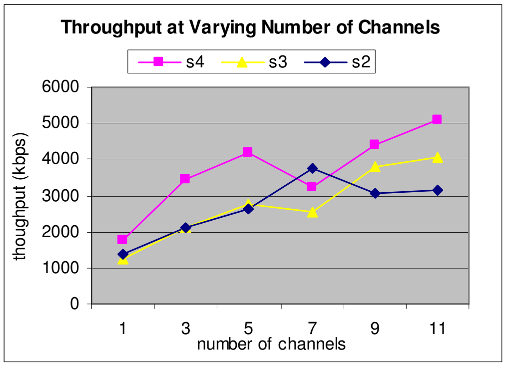

Figure 12. Impact of varying number of channels

Previous work [Lee05] claimed network throughput increases linearly with the total number of channels. We think that might be the case when the number of channels is equal to the number of data interfaces. We simulated 10 traffic flows with different number of channels and on three scenes (s2-s4). Fig.12 indicates that throughput can increase at beginning when number of channels is small, but will stop increasing after the number of channels is enough to keep all nodes in an interference area assigned a unique channel. [at number of channel is 7, S3 and S4 throughput drop down strangely. Did I trurn random on? If so every time running result is different (because channel assignment is completely different which is same effect as changing a new topo scenario). So to make comparable of the result of running on different number of channels, turn off random by set ns-random a non-0 value]

4.4 TCP Performance

Finally, we conducted experiments to study TCP performance using MC. We had a 10-flow connection pattern including 5 CBR/UDP flows and 5 FTP/TCP flows. Each flow is between a pair of randomly selected nodes. TCP and UDP flows have the same packet size of 512 bytes and the same rate of 200 packet/s. We experimented on 5 topology scenes on MC and SC networks. The throughput, as shown in Fig.13 & 14 for both TCP and UDP flows, increases significantly at most scenes. The results also reveal that TCP has even more performance improvement than UDP in MC. TCP not only benefits from MC as UDP does, but also gets further performance improvement by making use of control interface for ACK packets transmission.

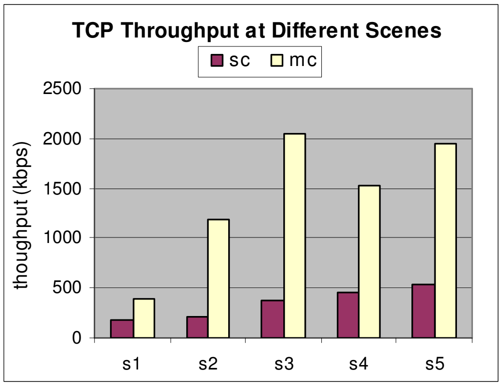

Figure 13. TCP throughput compare between MC and SC

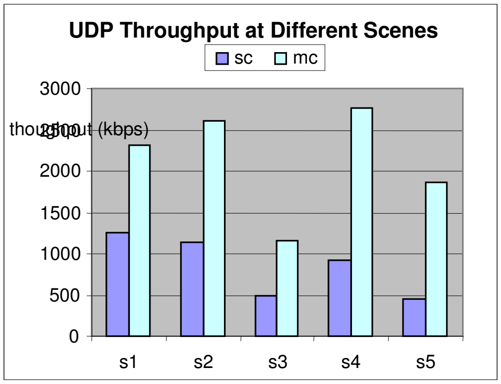

Figure 14. UDP throughput compare between MC and SC

5. CONCLUSIONS AND DISCUSSION

In this paper, we proposed an architecture using a dedicated control interface in multi-channel multi-interface mobile ad hoc networks. This architecture enables simple and efficient multi-channel assignment and routing control. We designed a simple extension to the OLSR routing protocol, embedded channel information in the Hello message. Thus, multi-channel assignment is realized without much overhead. It is effective to transmit control messages on the common channel (provided by the control interface) instead of sending them on all channels. Simulation indicated that this architecture and our multi-channel based OLSR can greatly improve performance over single channel networks, especially in more contentious environment. The control interface can also be used to transfer backward the ACK packets for TCP flows. The performance improvement for TCP flows using this scheme is even higher than that of UDP traffic.

We are working on some related issues. First, RCA has two limitations that it can not avoid channel confliction completely and requires a large number of channels. Currently, on-demand Negotiation-based Channel Assignment (NCA) seems to be the only solution. It is now only used by multi-channel approach on MAC layer. Our proposed architecture makes on-demand NCA possible on the network layer rather than modifying the MAC protocol. Dedicated control interface and OLSR proactive routing can facilitate the on-demand negotiation message exchange for initial channel assignment. OLSR can be extended to support proactive channel maintenance, thus to decrease NCA message overhead further.

Second, in our study, we have an assumption that control and data interfaces have the same transmission range. Actually, two heterogeneous interfaces normally have mismatched transmission ranges. We believe that if the control interface has shorter but comparable transmission range than the data interface, it will provide short hop route as long as the network is still connected for the control interface. Many researchers believe routing over more short hops outperforms smaller number of long hops, while Haenggi et al. disputes this claim [Haenggi05]. We are investigating the effect of radio range mismatch in dedicated control interface networks.

6. ACKNOWLEDGEMENT

The authors wish to acknowledge CITO (a part of OCE Inc.), Precarn and EION Inc. for partially funding this research. In particular, the authors wish to thank Mr. Bharat Rudra, Mr. Derek Best and Dr. Kalai Kalaichelvan for supporting this research.

7. REFERENCES

[Bahl04a] P. Bahl, A. Adya, J. Padhye, and A. Wolman, 2004, "Reconsidering Wireless Systems with Multiple Radios." *ACM Computing Communication Review*, Vol. 34, No. 5, October: pp. 39-46

[Bahl04b] V. Bahl, R. Chandra, and J. Dunagan, 2004, "SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks." *Proc. of MobiCom'04*, pp. 216-230

[Haenggi05] M. Haenggi, D. Puccinelli, 2005, "Routing in ad hoc networks: a case for long hops", *IEEE Communications Magazine*, October: pp. 93-101.

[Kyasanur04] P. Kyasanur, N.H. Vaidya, 2005, "Routing and Interface Assignment in Multi-channel Multi-interface Wireless Networks." *Proc. of Wireless Communications and Networking Conference,* Vol. 4, pp.2051-2056

[Lee05] U. Lee, S. F. Midkiff, J. S. Park, 2005, "A Proactive Routing Protocol for Multi-Channel Wireless Ad-hoc Networks (DSDV-MC)." *Proc. of the International Conference on Information Technology: Coding and Computing,* Vol. 2, April, pp. 710-715.

[NS2] Network Simulator ns-2, www.isi.edu/nsnam/ns.

[OLSR] Optimized Link State Routing Protocol (OLSR) RFC 3626, http://www.ietf.org/rfc/rfc3626.txt

[OOLSR] oolsr-0.99.15, an OLSR implementation, http://hipercom.inria.fr/OOLSR

[So04a] J. So and N. Vaidya, 2004, "Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver." *Proc. ACM Intl. Symp. On Mobile Ad Hoc Networking and Computing (MobiHoc)*, May 2004, pp. 222-233.

[So04b] J. So and N. H. Vaidya, 2004, "A Routing Protocol for Utilizing Multiple Channels in Multi-Hop Wireless Networks with a Single Transceiver." Tech. Report, Dept. of Computer Science, University of Illinois at Urbana-Champaign, October 2004.

[Wu00] S.L Wu, C.Y Lin, Y.C Tseng, and J.P Sheu, 2000, "A New Multi-channel MAC Protocol with On-demand Channel Assignment for Multi-hop Mobile Ad Hoc Networks." *Proc. of the International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN)*, Dec. 2000, pp.232-237