A Multi-Channel Medium Access Control Protocol for Multicast in Mobile Ad-hoc Network

Yao Zhao, Yong Xiang, Leiming Xu, Meilin Shi
Tsinghua University of Computer Science and Technology
Beijing, China
Email: zyao, xlming, shi@csnet4.cs.tsinghua.edu.cn

***Abstract*—In mobile ad-hoc network, some multi-channel MAC protocols utilize multiple channels to reduce the collision of wireless transmission and thus get high throughput. These multi-channel MAC protocols aim at improving the performance of unicast communication, and multicast data are generally transmitted as broadcast. So multicast can't benefit from the multi-channel technique. This paper proposes a multi-channel media access control protocol for multicast(MCMAC) which uses multi-channel techniques to improve multicast performance. To improve the reliability of MAC layer, we extended a reliable multi-channel MAC protocol for multicast (RMCMAC) based on MCMAC. Taking ODMRP[2] as an example of multicast routing protocols, we evaluate the multicast performance of MCMAC and RMCMAC for ad hoc networks via detailed simulation.**

Keyword: MAC; Multicast; Routing; Ad-hoc; Multi-Channel

I. Introduction

A mobile ad-hoc network is a group of wireless mobile nodes which self-organize into a network in order to communicate. Such networks can operate without fixed infrastructure or configuration. Because the nodes are dynamically linked in free ways, the most prominent feature of ad-hoc networks is frequently changing and undetermined topology of the network besides their nature of broadcast. What's more, the limited energy, low bandwidth and unreliable communication are vital factors affecting the performance.

With the development of network technologies and new applications, multicast has become a significant networking service. In mobile ad-hoc networks, multicast communication also holds an important position. Such applications as disaster discovery, search and rescue, and automated battlefields are typical examples of where ad-hoc networks are deployed. There are some typical multicast protocols of mobile ad-hoc networks such as MAODV, ADMRP, AMRIS, AMRoute, ODMRP(On-Demand Multicast Routing Protocol)[2] and CAMP. ODMRP protocol is a mesh-based on-demand multicast routing protocol with high performance among them.

Now wireless networks that employ several parallel multiple-access channels are considered. Generally, a multi-channel MAC protocol makes use of a half-duplex terminal which can work on different channels. At a time, the terminal can transmit or receive on one channel. Although the bandwidth of a channel is not increased, the capacity of the network improves because simultaneous communications can take place on different channels in the same space. Another important advantage is that the network can increase or decrease its capacity by adding or deleting channels. Much work has been done on MAC layer to utilize the multi-channel techniques, i.e. [6], [8], [10].

Traditional, multicast data are transmitted the same way as broadcast data in both single channel and multi-channel MAC protocols. To integrate multi-channel technique into multicast communication and thus improve it performance, we extend the single channel IEEE 802.11 DCF[4] for multicast with multiple channels, which comes to the MCMAC protocol. To provide reliable multicast data transmission for some special applications, we develop the RMCMAC protocol based on MCMAC with retransmission and link-break detection. In addition, with little modification MCMAC and RMCMAC can both provide multi-channel support for unicast.

The remainder of the paper is organized as follows. Section 2 illustrates the MCMAC and RMCMAC protocols in detail. Section 3 describes the simulation model and methodology followed by simulation results and analysis. Concluding remarks are made in Section 4.

II. MCMAC Protocol

MCMAC protocol extends from IEEE 802.11 MAC, an existing standard with highly accepted commercial status. MCMAC keeps most basic algorithms such as CSMA/CA scheme and RTS/CTS/ACK dialogue. The originality of MCMAC is how to use the multi-channel technique to distribute multicast transmission to different channels.

A. Utilization of multiple channels

Some schemes of using multi-channel are provided in [6], and we adopt the common-transmitter-based mechanism. First, we define a channel as the common channel, on which broadcast data and RTS/CTS control frames are transmitted. Each transmitter dynamically chooses a data channel, and multicast data are transmitted on this traffic channel.

When the backoff timer timeouts and the common channel is sensed idle, it will send a RTS frame in the common channel(See Figure 1). The data channel information is embedded in the RTS. After sending the RTS, the sender switches to its data channel, never expecting a CTS. After a definite time(SIFS + channel switch duration), the transmitter broadcast the multicast data frame in this channel. Then it resets to the common channel immediately after the transmission.

On receiving the RTS, some receivers get to know the coming multicast packet is for themselves. If a receiver wants to receive the packet, it switches to the traffic channel and then waits for the data. After receiving, these receivers change back to the common channel, without replying ACK.

This process is very simple, and the main difference to single channel IEEE 802.11 MAC is that multicast data are transmitted and received on a data channel designated by senders. Only MAC control frames and broadcast data are delivered on the common channel. Multicast data deliveries are distributed in several data channels, which makes it possible for senders to transmit data simultaneously in the same region.

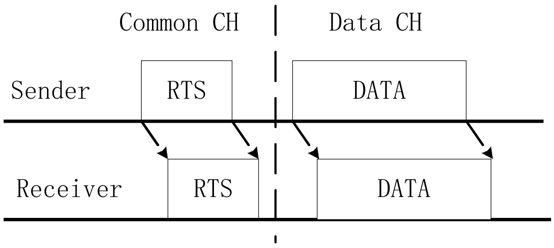


Figure 1 Time diagram of MCMAC

B. *Detection of Duplicate Packets*

In ad-hoc networks, multicast routing protocols usually set a sequence ID(SEQ_ID) in a multicast packet other than the RPF(Reverse Path Forwarding) check to avoid resending a duplicate packet.

In MCMAC, if a node switches to a data channel to receive a duplicate frame, it is waste of chance to receive or send new data. If no measure is taken in MCMAC, the performance degrades much, especially cooperating with mesh-based multicast routing protocols. So we decide to add the duplicate examination to MCMAC. When MCMAC receives a packet from the upper routing layer, it requires a sequence number of the packet at the same time. This sequence number is defined by multicast routing protocols. For example, most protocols can simply use the combination of the source address and the SEQ_ID as the unique sequence number(SEQ_NUM). For each multicast group, MCMAC caches a certain number of SEQ_NUMs of received or transmitted packets recently.

The SEQ_NUM of a multicast packet is appended to the RTS. When a receiver gets the RTS, it can tell if the coming data packet is duplicate by checking the SEQ_NUM field in the RTS. If duplicate, the receiver won't switch its working channel and keeps on listening on the common channel. Thus MCMAC saves much bandwidth and improves it performance.

By this procedure, MCMAC does rely on routing protocols to some extent, and it fulfills part of functions of the routing layer. But this little cost brings high improvement of the performance, especially when the upper multicast routing protocols are mesh-based. In fact, the interface provided by MCMAC for SEQ_NUM is so simple and universal that most multicast routing protocols can fit its requirement effortlessly.

Figure 2 depicts the finite state machine(FSM) of MCMAC. Without CTS and ACK control messages, the FSM of MCMAC is very simple. It has only three states: IDLE, WF_DATA(waiting for data) and TR_DATA(sending data).

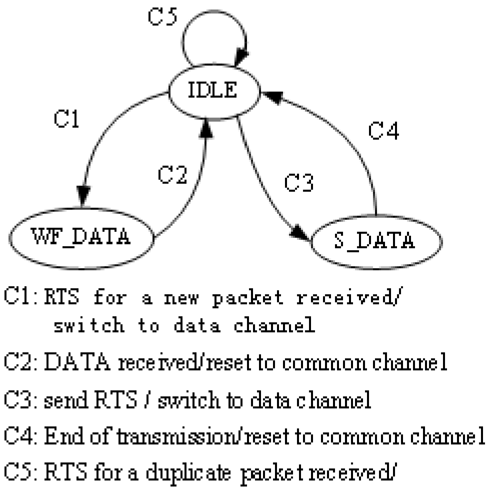

C5
IDLE
C1
C4
C2
C3
WF_DATA
S_DATA
C1: RTS for a new packet received/ switch to data channel
C2: DATA received/reset to common channel
C3: send RTS / switch to data channel
C4: End of transmission/reset to common channel
C5: RTS for a duplicate packet received/

Figure 2 The finite state of MCMAC

C. *Deal with limited channels*

Ideally, we have enough channels and each node owns a unique data channel. In this condition, there is no collision of data transmission, and the hidden terminal problem and exposed terminal problem are inexistent in data transmission. Collisions only take place in the common channel for broadcast and MAC control messages. In fact, the resource of channel is so limited that usually all nodes share some data channels. The problem is how to fully utilize these channels and reduce collisions to the least extent.

The collisions occur only at receivers, and it may be safe for two close senders to send simultaneously to different receivers. But in multicast, there are several receivers around one sender, and it's better to avoid simultaneous transmissions of close senders in the same channel.

In MCMAC, each node keeps a table of currently used channels, with the time until when the current use expected to expire. On receiving others' RTS, a node updates its table. When a node wants to send multicast data, it randomly selects an unused channel as its data channel after searching the table. If all channels are in use, the frame is delayed until there are free channels and then backs off for a shot interval. This algorithm can greatly depress the problems caused by hidden terminals and exposed terminal, but it can't solve the two problems thoroughly.

D. *RMCMAC protocol*

As mentioned above, MCMAC doesn't completely address the hidden terminal problem and exposed terminal problem when channel resources are limited. The exposed terminal problem only takes effect when all channels are marked busy. The hidden terminals may lead to collisions when selecting a using channel. And the wireless channel is unreliable that data transmission may fail. Here we propose the Reliable Mult-Channel MAC protocol for Multicast(RMCMAC), which adopts the RTS/CTS/ACK dialogue and retransmission mechanism to enhance the reliability of MAC layer. Figure 3 is the time diagram of RMCMAC.

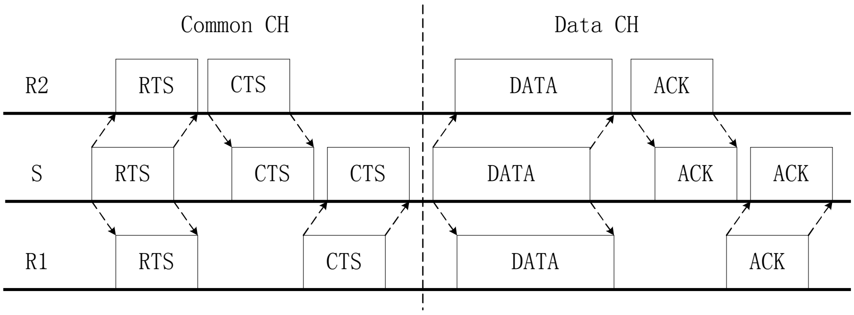


Figure 3 Time diagram of RMCMAC

Unlike unicast, a data packet may be received by more than one downstreams in multicast. A RTS may cause many CTS replies. These CTS messages will collide with each other if they are not designated an order. So the sender should appoint a sequence in the RTS. RMCMAC requires the upper multicast routing protocols to tell it the downstream set maintained by them. For a tree-based routing protocol, it just needs to inform RMCMAC of its downstream set when sending a multicast packet down. As for a mesh-based multicast protocol, it has a tree backbone which extends to the mesh. RMCMAC tries to improve the reliability of the tree backbone.

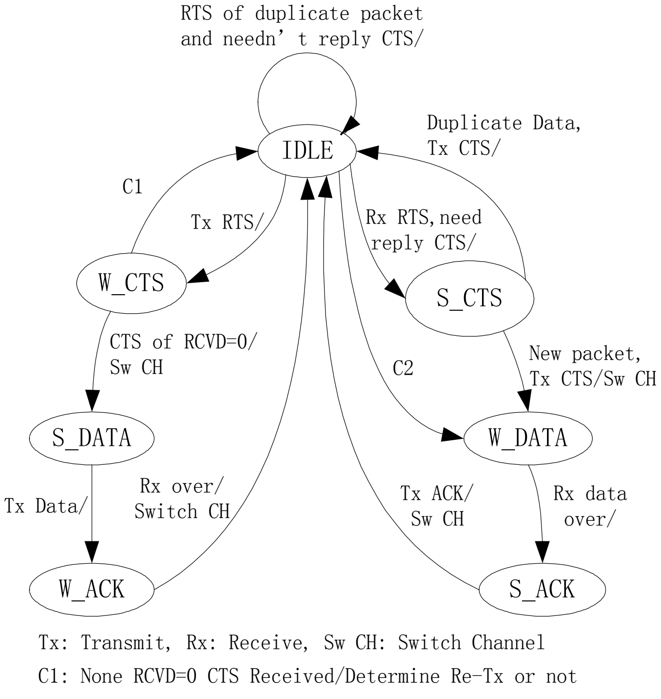
The RTS of RMCMAC is extended again to include all the downstream addresses with an arbitrary order. When these downstream nodes receive the RTS, they calculate out the reply time and schedule their CTS by its sequence in the downstream list(SIFS × n + (n-1) × CTS duration, n is the CTS reply order). Besides, other receivers of this multicast group can also switch to the traffic channel to receive data. But they need not reply CTS and ACK after transmission. Thus mesh-based multicast routing protocols also benefit from RMCMAC.

The replied CTS also include the sender's data channel number, which shows this channel will be busy for a certain period. This information is useful to maintain the table of used channels, and helps to reduce the collision caused by hidden terminals. What's more, a RCVD flag is added to the CTS message. When a recipient finds that it gets a RTS for a duplicate packet, it won't switch channel to receive the data packet. It then replies a CTS with the RCVD flag set and stays in the common channel.

If the sender receives CTS messages with RCVD flag unset, it will switch to its transmitter channel to send data. Otherwise the data frame will be retransmitted next time. After the sender transmits the multicast data, the receivers reply ACK messages following the sequence appointed in the RTS. At last the sender gets to know which downstream nodes received the data correctly by received ACKs and CTSs with the flag RCVD=1. If not all the downstreams received the data packet, the sender will retransmit the data for these downstreams(The retransmission is limited, and in RMCMAC the max time is 3).

RMCMAC uses the downstream set of the multicast tree(or mesh backbone). But in ad-hoc networks with frequently changing topology, the multicast tree(or backbone) needs an interim to adapt to a new topology. In this transient state, RMCMAC may get wrong information from upper protocols. For example, if the downstream set consists of a node that has left this region, the sender can't receive its replies of CTS or ACK. Then there may be many useless retransmissions for leaving nodes. So RMCMAC carries the detection of broken links, which helps to avoid the bad effect from upper routing protocols' fault.

RMCMAC maintain a broken-link table, which records those unreachable nodes with a failing time for each node. When RMCMAC has to discard a multicast packet after 3 times retransmission, it will add this node into the broken -link table or increase the failing time by 1 if it is already in the table. If the failing time of a node is more than a threshold, RMCMAC takes this node for unreachable. When a transmitter makes a RTS, it still includes the addresses of unreachable downstreams. But when checking whether all downstreams feed back with CTS or ACK, these unreachable nodes are not taken into account. So senders won't retransmit data for unreachable nodes. If a broken link is connected again, the unreachable nodes will reply CTS and ACK again. Then the sender will remove this reconnected node from the broken-link table. Besides, if not refreshed, any entry in the broken-link table will expire after some time.



Tx: Transmit, Rx: Receive, Sw CH: Switch Channel

C1: None RCVD=0 CTS Received/Determine Re-Tx or not

C2: Rx RTS, needn' t reply CTS/Sw CH

Figure 4 The finite state of RMCMAC

Figure 4 depicts the finite state machine of RMCMAC, which is much more complex than that of MCMAC. The FSM comprises seven states: IDLE, W_CTS(waiting for CTS), S_CTS(sending CTS), S_DATA (sending data), W_DATA (waiting for data), W_ACK(waiting for ACK) and S_ACK (sending ACK). General error handles are to reset to the common channel, and some are omitted in Figure 4.

E. Interfaces between MAC and multicast routing protocols

MCMAC and RMCMAC require some information from upper multicast routing protocols, and define some simple and universal interfaces to collaborate with them. We take ODMPR[2] as an example to show how to use these interfaces.

1) *Map the multicast address to a MAC address:*

   In ODMRP, when a node joins group G or the forwarding group of G, it calls this interface to tell the MAC layer to map a MAC address for group G. When a node leaves group G or its forwarding group, it call this interface to remove the MAC address for G. This interface is the same as that in wireline network.

2) *Sequence* number *interface*

   When ODMRP sends a multicast packet down to the MAC layer, it calls this interface to transfer a long number including both the source address and sequence ID of the packet as the SEQ_NUM. MCMAC and RMCMAC execute the duplicate examination through the SEQ_NUM.

3) *Multicast downstreams interface*

   RMCMAC needs the downstream set of the multicast tree to ensure the reliability. ODMRP is a mesh-based protocol, but it has a backbone of shortest path trees(SPT). Now OMDRP records the SPTs and sends downstream information to RMCMAC when there are packets to send down.

From above, we can see that ODMRP needs only a litter amelioration to cooperate with the two multi-channel MAC protocols for multicast and benefits from the great throughput improvement. These interfaces are so simple and universal that other multicast routing protocols also can fit the requests without difficulty. Besides, the interfaces are optional. If the upper routing protocols don't use these interfaces, they just lose some profits.

III. Performance Evaluation

In the simulation, we choose IEEE 802.11 DCF, MCMAC and RMCMAC as MAC protocols, and ODMRP as the multicast routing protocol. These experiments try to show the advantage of multi-channel based MAC protocols.

A. Simulation Environment and Methodology

The simulations of ODMRP and three kinds of MAC protocols are all implemented in ns2.1b9[9]. Our simulation models a network of 50 mobile hosts placed randomly within a 1200m×1200m area. Radio propagation range is 250 meters and carrier sense range is 550 meters. We use the two ray ground propagation model in our experiments. The channel capacity is 2 Mbit/sec. There is a little temporal partition of the network and the average number of neighbors for each node is 7.62. Each simulation executes for 400 seconds of simulation time. Multiple runs with different random seed number are conducted for each scenario and collected data is averaged over those runs.

The multicast data streams are CBR streams with jitters. The size of data packet is 512 bytes. The multicast group size is set constant at twenty and the number of senders is five. The multicast sources are selected from all 50 nodes randomly and most of them act as receivers at the same time. Receivers join one multicast group at the beginning of the simulation and never leave the group during the simulation. The simulation scenarios are generated by the Setdest tool of ns2.1b9. Nodes randomly select a destination and move with a predefined average or constant speed.

We use the packet delivery ratio of the application layer as the metric to compare the multicast performance of different MAC protocols. Packet delivery ratio is the ratio of the number of multicast data packets delivered to the destinations versus the number of data packets supposed to be received. This number presents the effectiveness of the cooperation of the MAC protocol and the routing protocol.

B. Simulation Result

1) *MCMAC vs IEEE 802.11 DCF*

In this experiment, we want to compare the throughput of MCMAC and IEEE 802.11 DCF. The mobile speed of node is 2m/s in the scenarios. The multicast traffic load varies from light to heavy. The packet delivery ratio shows each protocol's performance.

In Figure 5, MCMAC-k means that MAC layer only occupies k data channels to transmit multicast data, and the legend MCMAC means that MAC layer can use unlimited channels ideally. Figure 5 shows that with the increasing traffic load, the packet delivery ratio of IEEE 802.11 DCF drops rapidly. While the product of the load and packet delivery ratio increases lowly, which reflect the capacity of network to some extent. The ideal MCMAC avoids the collisions of data transmission, and hardly suffer from hidden terminals and exposed terminals. So it has a quite high performance, and the packet delivery ratio is even above 90% with a heavy load(100pks/s). The performance of MCMAC-k improves as the number k increases, as we expected. When k is large, the advantage of additional channels becomes little. For example, the packet delivery ratio of MCMAC-8 is only a little higher than that of MCMAC-6, and here ideal MCMAC is the same as MCMAC-50. For a certain load, we should choose a suitable number of k to get efficient channel usage and high performance.

2) *Reliability of RMCMAC*

In this experiment, we try to show the reliability of RMCMAC. To demonstrate that RMCMAC is not affected by upper routing protocols with broken-link detection, we use scenarios with rapidly moving nodes. In the scenarios, the average mobile speed of nodes is 5m/s and maximum speed is 10m/s. The multicast traffic load is relatively light(30pkt/s). The channel is not reliable, and its error probability for the delivery of each long data packet ranges from 0 to 0.4. But the error probability for the delivery of a short frame such as RTS, CTS and ACK is quite low. Although mesh-based ODMRP uses redundant transmission to improve its reliability, the performance of MCMAC drops heavily as the channel becomes more unreliable(See Figure 6). RMCMAC uses retransmission mechanism to enhance its reliability, and keeps its high packet delivery ratio even in tough environments.

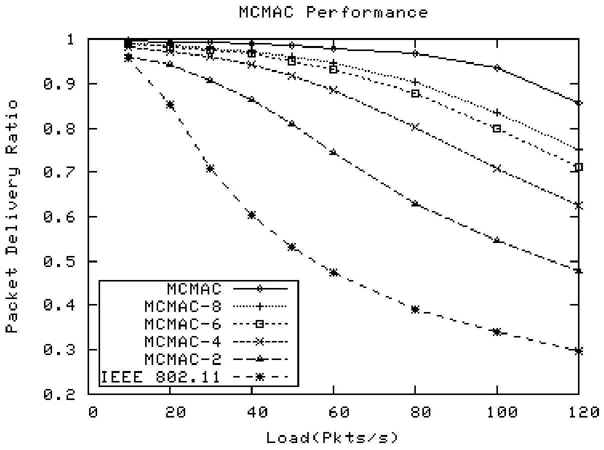


Figure 5 Packet delivery ratio as a function of multicast load

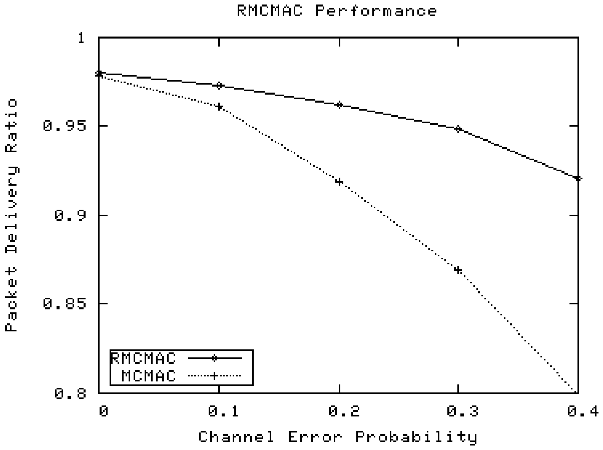


Figure 6 Packet delivery ratio as a function of channel reliability

Relying on the retransmission mechanism, RMCMAC brings much more data frames than MCMAC. What's more, it has additional control overhead such as CTS and ACK. So RMCMAC has worse performance in conditions with heavy loads. Most multicast data are audio and video streams, which may not require much high reliability but high throughput. Only a few applications demand reliable multicast transmission. Thus we imagine that MCMAC and RMCMAC work together as the MAC protocol, and upper layer can dynamically choose a service type according to its requirement or the condition of the channels at present.

IV. Conclusion

Traditional single-channel distributed MAC protocols put their emphasis on the dialog between senders and receivers to solve the hidden terminal problem and exposed terminal problem.

We propose the multi-channel based MCMAC and RMCMAC protocols, which aim at improving the throughput of multicast communication. With some universal interfaces, the MAC protocols and multicast routing protocols can cooperate well and still keep either's independence. In MCMAC and RMCMAC, we don't mention unicast. But in fact, unicast is a kind of special multicast. MCMAC and RMCMAC can be a multi-channel MAC protocol for both unicast and multicast with little modification.

For multi-channel MAC protocols, several pair of data transmissions can take place at the same time. This situation may cause the fairness problem if some streams depress others. It will be our relate work to involve the fairness problem. Besides, the capacity of the multi-channel ad hoc networks for multicast and the more efficient multi-channel access scheme will also be our research focuses.

Acknowledgment

Thanks for the support from National Natural Science Foundation of China under Grant No. 60273010 and National Hi-Tech R&D Program of China under Grant No. 2002AA123022.

References:

[1] S. Corson and J. Macker, "Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501, January 1999

[2] S.-J. Lee, W. Su, and M. Gerla , "On-demand multicast routing protocol in Multihop Wireless Mo-bile Networks" , *ACM/Kluwer Mobile Networks and Applications, 2000.*

[3] Sung-Ju Lee, William Su, and Mario Gerla. "On-demand multicast routing protocol (ODMRP) for ad hoc networks", Internet Draft, draft-ietf-manet-odmrp-02.txt, January 2000.

[4] IEEE Standard for Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) Specifications, IEEE Standard 802.11, 1997

[5] Z. J. Haas and J. Deng, "Dual Busy Tone Multiple Access (DBTMA) - A Multiple Access Control Scheme for Ad Hoc Networks," IEEE Transactions on Communications, vol. 50, no. 6, pp. 975-985, June 2002

[6] Mario Joa-Ng, I-Tai Lu, Spread Spectrum Medium Access Protocol with Collision Avoidance in Mobile Ad-hoc Wireless Networks, INFOCOM 1999, 776-783

[7] Jiandong Li, Zygmunt J. Haas, and Min Sheng, Capacity Evaluation of Multi-Channel Multi-Hop Ad Hoc Networks,ICPWC 2002, New Delhi, India, Dec 2002

[8] J. Li, Z.J. Haas, M. Sheng, and Y. Chen, Performance Evaluation of Modified IEEE 802.11 MAC for Multi-Channel Multi-hop Ad Hoc Network, Advanced Information Networking and Applications (AINA) conference, Xidian University, Xian, China, March 27-29, 2003

[9] The Network Simulator - ns-2, Available from http://www.isi.edu/nsnam/ns/

[10] Y.-C. Tseng, S.-L. Wu, and J.-P. Sheu, A Multi-Channel MAC Protocol with Power Control for Multi-Hop Mobile Ad Hoc Networks, WNMC 2001