Final version submitted to 21St Biennial Symposium on Communications, Apr.5, 2002, paper#43

A Dynamic Multi-Channel MAC for Ad Hoc LAN

Wing-Chung Hung, K.L.Eddie Law, A. Leon-Garcia
The Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto
Toronto, Ontario, Canada

Abstract-- **Applying single channel *ad hoc* MAC protocol (such as 802.11b) in multichannel environment (such as the ISM band) has two potential problems: connectivity and load balancing. Connectivity problem refers to the fact that stations in one channel cannot connect to those in the other channels. As resource is partitioned into multiple independent channels, when the load of two channels are different, resource in the underloaded channel cannot be used to offload the overloaded channels, and thus, reduces the overall system performance. The use of multichannel protocols where each station is capable of accessing to more than one channel is a way of solving the two problems. This paper proposes a multichannel MAC protocol for that purpose. The preliminary result shows that the objective is achievable while, at the same time, maintain good performance.**

Index terms—**Multichannel *ad hoc* Networks, Dynamic Private Channel, CDMA, MAC**

A. INTRODUCTION

Ad hoc network does not have a predefined topology. Multiple stations pool together and communicate without assistance of any centralized control. The ad hoc nature makes *network anywhere* possible and adds convince for local group communications. Due to the lack of centralized control, the size of an ad hoc LAN varies.

Currently, the most popular ad hoc MAC implemented is the 802.11b [1] which utilizes CSMA/CA and channel reservation via Network Allocation Vector (NAV). IEEE 802.11b is a single-channel MAC protocol which operates in the ISM band. For CSMA type MAC protocol, the system throughput decreases as the network size increases due to collisions and longer backoff periods. Our simulation shows that the same effect occurs even in a single cell environment where all stations are in the transmission range of each other.

The ISM band is a typical multichannel environment. In the band, spread spectrum is used for channellization. Each channel is identified by a code. Depends on the physical layer properties, a code refers to either a direct sequency spreading code or a frequency hopping pattern. In either case, multiple channels are allowed to co-exist in the same band.

If more than one channel is available and the network size is large, it is logical to allocate the stations to operate in different channels in order to improve system throughput. This is acceptable if communication is confined to the stations in the same channel. Otherwise, special equipment will be needed to bridge the stations in different channels to allow cross-channel communicaton. However, in ad hoc networks, such equipment is usually not available.

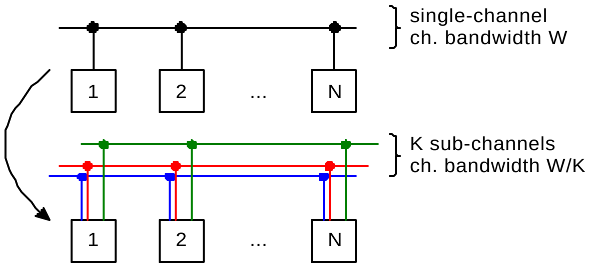

Fig.1a. Multi-channel system by dividing a fat channel into multiple thin channels

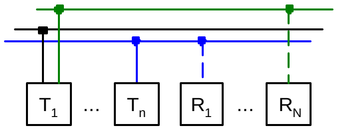

Fig.1b. Transmitter-oriented multichannel system

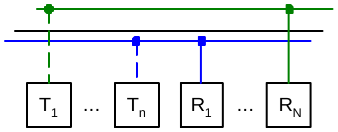

Fig.1c. Receiver-oriented multichannel system

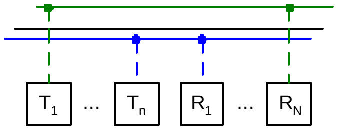

Fig.1d. Dynamic-assigned multichannel system

Besides, in many cases, it is not obvious how the network partitioning can be done because each station may have different ownership. Even if partitioning is possible, the load generated from each station varies, so does the aggregate load on each channel. Therefore, there are two potential problems when single-channel MAC is used in multi-channel environment: connectivity and load balancing. The lack of connectivity limits networking power while the unbalanced load on different channels results in lower overall system throughput.

Multichannel MAC protocol differents from single channel protocol in that it allows a station to access to more than just one channel. Different forms of multichannel have been proposed in the literatray. The authors in [3] and [4] show that by breaking down a channel into multiple sub-channels, the system performance will be improved for CSMA type protocol due to the reduction of normalized delay and probability of collision. The model of this approach is shown in Fig.1a. It requires each station to monitor all the channels at all time. IEEE 802.11a [2] adapts a similar model by using multiple OFDM channels to achieve total bit rate up to 54 mbps.

Another method on using multichannel is to have transmitter-receiver pair switch to the same channel when communication is required. The channel allocation mechanism can be either transmitter-oriented, receiver-oriented, or dynamically assigned [5] as shown in Fig.1b, Fig.1c and Fig.1d. The transmitter-oriented mechanism has been used in point-to-multipoint systems such as satellite TV and radio broadcasting. Cellular phone system is an example of using dynamic assignment in which different calls are assigned different channels for the mobile-and-base connection during the period of a call.

For ad hoc LAN, the Hop Reservation Multiple Access (HRMA) proposed in [6] matches transmitter-receiver dynamically. HRMA is a multichannel protocol for slow frequency hop ad hoc network. All HRMA stations hop according to a predefined hopping pattern and exchange RTS/CTS. After successful exchange of RTS/CTS, a pair of transmitter-receiver remains in a hop for further data exchange while the other stations continue to hop in the predefined pattern. It is a multichannel protocol because more than one transmitter-receiver pairs may exchange data while they stay in different hops. The protocol is applicable only in slow hopping systems. Besides, it is unfriendly to other coexisting devices that are using different hopping patterns. Due to the long dwell period while stations are exchanging data, it is likely to have a hit with those coexist devices.

In this paper, a multichannel MAC protocol, called Dynamic Private Channel (DPC), that uses dynamic channel allocation is proposed for ad hoc network. The objective of the protocol is to maintain good system performance and enhance ad hoc network connectivity. The protocol will be described in Section B and simulation results will be shown in Section C. Section D will present the concluding remarks.

B. PROTOCOL DESCRIPTION

The design of DPC is based on the assumption that there are multiple channels available. As illustrated in Fig.2, there is one multicast control channel (CCH) and multiple unicast data channels (DCHs). The CCH is shared by all stations. Therefore, transmission to CCH will be heard by all stations within the transmission range. Access to this channel is contention-based. All DCHs are either free or busy depends on whether it is being used. When a station X requests a channel for communicating with another station Y, one of the free DCHs will be assigned to the pair (X,Y) for a limited duration T_d. At the end of that duration or when (X,Y) does not need the channel anymore, the DCH will become free again. The request of a DCH is performed through the CCH and is coordinated in a distributed manner where all stations participate.

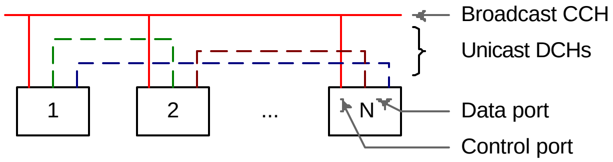

Fig.2. Multi-channel system used in DPC

Each DPC station is equipped with N_r+1 radio ports where N_r = 1. One port is the control port and is tuned to the CCH at all time while the others are the data ports which will be used dynamically for data communication using the DCHs. Each station maintains three queues: one is data queue Q_d while the others are incoming Q_i and outgoing Q_o RTS queue. When data arrives from the higher layer, it will be enqueued in the data queue. When RTS is received from CCH, it will be stored in the incoming RTS queue for further process. If a RTS is waiting to be sent out, it will be kept in outgoing RTS queue.

DPC is connection-oriented. if a station X has data packet to send to another station Y, X will initiate the setup process by sending an RTS to Y through the CCH. At the same time, station X reserves one of its data ports for communicating with Y. Before sending out the RTS, station X choose a free DCH and include the channel code in the RTS header. When Y receives the RTS, it will check if the channel chosen by X is acceptable. If so, it will return a *Reply to RTS* (RRTS) to station X with the same channel code in the header. Otherwise, Y will suggest another one and put the new channel code in the RRTS header. When the code negotiation comes to an end, both stations will tune one of their data ports to the select DCH and start exchange data packets. The data exchange begins with Y sending out CTS to X. After then, they exchange data packets back and forth in ping-pong manner. The communication ends either when they have no more data to exchange or when the reservation period T_d is expired. During this period, X and Y have the exclusive right to use that particular DCH.

Now, the process of setup, transfer, and termination will be described in more details.

Connection Setup

Station X will start the process of connection setup when all of the following conditions are satisfied:

1. X has data to send to Y.
2. X is currently not communicating with Y and not in the process of setting up connection with Y.
3. X has a free data port.
4. At least one free DCH is available.
5. The incoming RTS queue Q_i is empty.

The first four conditions are self explained, but the fifth condition needs some notes. A non-empty Q_i implies that some other stations, say station Z, has made a request to communicate with station X and that (Z, X) has gone through the code negotiation. Furthermore, Z already reserved one of its data ports and is waiting for X to start using the chosen DCH. In such case, station X will process the RTS in Q_i first, rather than request a connection with Y. If a data port is free, X should use it to communicate with Z by sending CTS to Z. The details of data exchange will be considered later.

If all the five conditions are satisfied, X reserves one of its data port for communicating with Y. The status of this port changes from FREE to SETTING. Next, X chose a free DCH for this communication. This means X will compute a code C_{x1} that none of its direct neighbors is using. After then, a RTS packet that contains the code is sent to Y through the CCH. Since every station hear the transmission of RTS, they learn that X intents to use C_{x1} and will avoid using the same code until its reservation expires. When Y receives the RTS, the code C_{x1} may or may not be acceptable. This is because the neighbors of X are not exactly same as the neighbors of Y. A neighbor of Y may already have reserved the same code. In that case, Y will suggest another code. This process continues until a code acceptable by both X and Y is found. The code generation and code negotiation processes will not be discussed further. For this paper, it is assumed that the code generation process is available and the first code offered is always acceptable. The mechanism of these processes will be discussed in another paper.

Station Y receives an RTS with an acceptable code. It enqueues the packet in the RTS queue Q_i and sends an RRTS to X after a short IFS. The code in the RRTS is same as the one accepted. In station Y, if there is free data port available, CTS will be sent as described above.

Since the CCH is contention based, a truncated binary exponential backoff algorithm is used for determining when to send out an RTS. However, RRTS is sent immediately after a short IFS period when RTS is received without backoff. All other stations hear the RTS will defer transmission to avoid colliding with RRTS.

When X receives the RRTS, it can detect whether the previously suggested code has been accepted and, if not, whether the new code is acceptable. If it is necessary to renegotiate the code, the exchange of RTS and RRTS continues until either both ends come to an agreement or one end decides to give up. If the code has been accepted, the connection setup process is completed. Since X initiates the RTS, one of its data ports has been reserved and is tuned to the channel indicated by the code C_{x1}. However, X cannot start sending data because Y may not have free data port at that instance. Therefore, the DCH will be blocked until Y commits a data port for communicating with X. When that happens, a CTS will be receive in the DCH and (X,Y) can start data exchange.

Data Transfer and Connection Termination

Station X has sent a RTS to Y to request for connection setup. Station Y responses with CTS to indicate its readiness for communication. After receiving CTS, X sends a data packet and waits for ACK from Y. If Y has backlogged data for X, the data will be sent even if it is not at the head of line. Otherwise, Y will reply with an ACK-only packet. In case of reception errors, a NACK-only will be replied without data payload. In case of packet loss, timeout will occur and packet will be retransmitted. The process continues until both ends indicate that there is no more data to send. To enable this notification, there is a bit in the data packet header to indicate if the sender has more data to send.

In some cases, a connection (X,Y) may occupy a DCH for too long and deny other stations the chance to communicate. To avoid starvation, a maximum reservation period is imposed. Each connection is allowed to transfer a maximum of M data packets. If a total of M data packets have been exchanged, the connection must be terminated even if there are still more data packets backlogged. Besides being able to avoid starvation, this policy also provide a hint for computing the channel reservation period T_d of a DCH which is important for determining the time for a busy DCH to become free again.

C. SIMULATIONS

In a centralized wireless network, a cell is an area covered by a base station. In an ad hoc network, the concept of a cell is different. Each ad hoc station is the center of a cell. When all stations are located within each other's cell, we called it the single-cell case. Otherwise, we call it the multi-cell case. In the single-cell case, the hidden terminal [7] [8] problem is not exist.

The single cell scenario will be studied in following simulation cases. It is assumed that error will occur only if packets are collided.

Case 1: 802.11b

The first simulation case will show the saturation throughput of IEEE 802.11b when the network size increases. Simulation paramenters are shown in Table 1. All stations in the network will generate traffic. To simulate differe network load, the average inter-arrival time is changed and the packet length is kept the same. The throughput is normalized to the channel capacity.

Table 1. 802.11b simulation attributes

Parameter	**Value**
Packet length (bytes)	Constant(500)
Packet inter-arrival time	Exponential
Destination address	Random
RTS/CTS	Always
Channel Capacity	1 Mbps
PHY Property	PHSS

Fig.3 shows the result when the network size increases from 2 to 20 stations. For each network size, the offered load is gradually increased until the network is overloaded. From the result, the maximum throughput is obtained. The figure shows that the best performance is achieved when there are only two stations in the network and decreases as the network grows. This indicates that when the aggregate load in a network is heavy, a small network will yield a higher throughput. From this observation, the need to partition the network is clear.

The point that is not so obvious is when and how to do the partition. Regarding this, multichannel protocols clearly have the administrative advantage of distributing stations to different channels automatically.

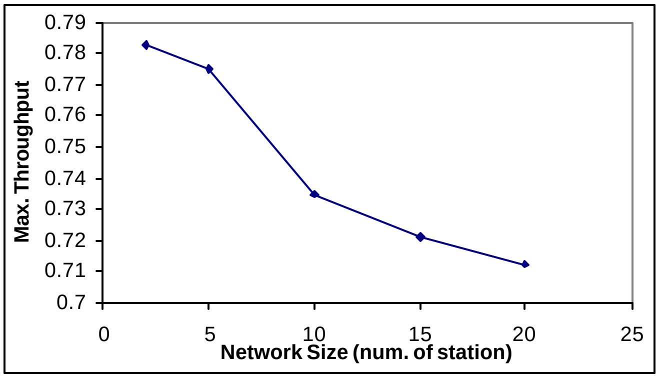

Fig.3. IEEE 802.11b maximum channel throughput as the network grows from 2 to 20 stations

Cast 2: DPC

The simulation parameters for DPC is shown in Table 2.

Table 2. DPC simulation attributes

Parameter	Value
Network Size	20 stations
Packet length (bytes)	Constant(500)
Packet inter-arrival time	Exponential
Destination address	Random
Max. Connection Burst Size (M)	5 packets
Max. Data Connections Allowed	Infinite, 1-6
Number of Data Radio Ports per Station	1
Channel Capacity Per Channel	1 Mbps
PHY Property	PHSS (Same as those for 802.11b)

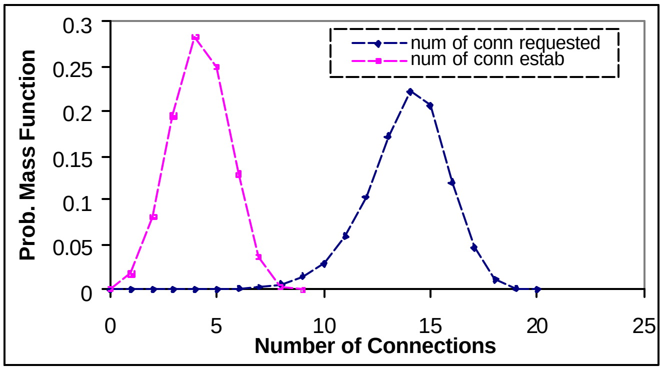

Fig.4. The PMF of number of connections being requested and established. Network size is 20 stations.

First, assume that the number of channels available is infinite and the network is overloaded. The result in Fig.4 shows the probability mass function (PMF) of the number of connections being requested and the number of connections being established. It can be seen that the number of connections established is much less than those being requested. This is due to connections blocked by busy stations. To illustrate that, consider the following scenario:

- A is communicating with B, thus both A and B are busy.
- C sends RTS to B. C is waiting for CTS from B, but B is busy. Therefore, C is blocked by B.
- D sends RTS to C and is blocked by C.

There are three connections being requested which are (A,B), (C,B) and (D,C), but only one is estabished which is (A,B). Fig.4 indicates that most of the time, the number of established connections are 3, 4 or 5. Besides showing the negative impact of blocking, the result conveys a more important message that the network probably just need about 4 connects to achieve good resource utilization.

As the number of channels available in reality is always limited, the next simulation case will limit the the number of channels available. The result is shown in Fig.5. The throughput is normalized to the total capacity of all data channels. For example, for 4 channels, it is normalized to 4 mbps.

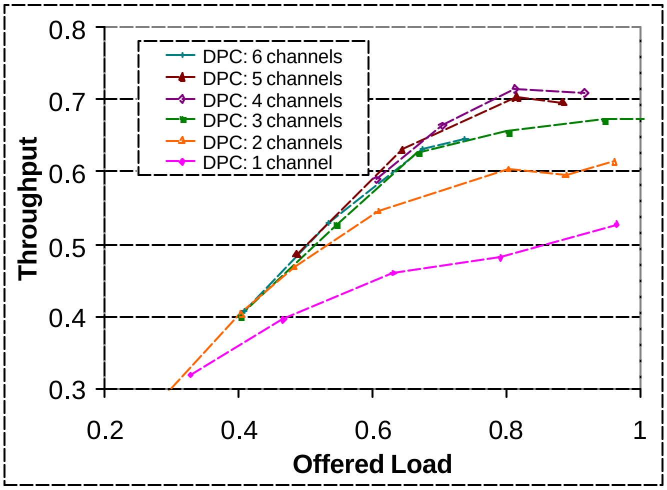

Fig.5. The throughput of DPC as offered load increases.

The throughput increases as number of channels increases from 1 to 4. This is because each additional channel is being utilized better as hinted by the result shown in Fig.4. When 4 channels are available , the maximum through will be 71.5%. Further increase of channels will actually reduce the average channel utilization.

This preliminary result gives two signals. First, multichannel will have good throughput with proper protocol design. Second, the blocking effect in DPC has a great impact on the performance. This indicates that the protocol should be improved in a way of reducing blocking. There are two approach for improvements. One is to modify the terminating procedure so that surrounding stations will get notice both when a connection is setup and when it is terminated. Another way is to implement a different scheduling mechanism so that a station will send connection request to a station that has smaller probability of blocking. Currently, a request is always send to the receiver of the head-of-line data packet.

D. REMARKS

Compare to single-channel MAC protocol, the use of multichannel MAC protocol has the advantage of providing cross-channel connectivity and better use of resources through load balancing. In the proposed protocol, these are done by dynamically choosing one of the free data channels when data exchange is needed. Our study demonstrates the merit of applying multichannel protocol in multichannel environment. However, we acknowledge that improvement is needed in the protocol design of DPC.

REFERENCE

1. "IEEE Std 802.11: Wireless LAM Medium Access Control (MAC) and Physical Layer (PHY) Specifications". IEEE, 1999.
2. IEEE 802.11a: Wireless LAM Medium Access Control (MAC) and Physical Layer (PHY) Specifications – Amendment 1: High-speed Physical Layer in the 5 GHz band". IEEE, 1999.
3. Marco A. Marsan, Daniele Roffinella. "Multichannel Local Area Network Protocols". IEEE JSAC, vol. SAC-1, no.5, November 1983. pp.885-897.
4. A. Nasipuri, J. Zhuang, and S.R. Das. "A Multichannel CSMA MAC Protocol for Multihop Wireless Network". IEEE Wireless Communications and Networking Conference (WCNC'99), 1999.
5. Elvino S. Sousa, and John. A. Silvester. "Spreading Code Protocol for Distributed Spread-Spectrum Packet Radio Networks". IEEE Transactions on Communications, Vol.36 No.3, March 1988. Page 272-281.
6. Zhenyu Tang and J.J. Garcia-Luna-Aceves. "Hop-Reservation Multiple Access (HRMA) for Ad-Hoc Networks". INFOCOM '99.
7. L. Kleinrock and F.A.Tobagi, "Packet Switching in Radio channels: Part II – The Hidden Terminal Problem in Carrier Sense Multiple-Access Modes and the Busy-Tone Solution ". IEEE Trans. Communications. Vol.23, no.12, 1975. Pp.1417-1433.
8. Jing. Deng and Zygmunt J. Hass, "Dual Busy Tone Multiple Access (DBTMA): A New Medium Access Control for Packet Radio Networks". IEEE International Conference on Universal Personal Communications (ICUPC), May 1982.