

*A MAC protocol  
for  
Multi frequency Physical Layer*

Santashil PalChaudhuri (*santa@cs.rice.edu*)  
Rajnish Kumar (*rajnish@cs.rice.edu*),  
Amit Kumar Saha (*amsaha@cs.rice.edu*),

January 23, 2003

**Abstract**

Existing MAC protocols for wireless LAN systems assume that a particular node can operate on only one frequency and that most/all of the nodes operate on the same frequency. We propose a MAC protocol for use in an ad hoc network of mobile nodes using a wireless LAN system that defines multiple independent frequency channels. Each node can switch quickly from one channel to another but can operate on one channel at a time. We simulate the proposed protocol by modifying the wireless extension<sup>1</sup> to *ns-2*<sup>2</sup> simulator. Our simulations show that the proposed protocol, though simple, is capable of much better performance in the presence of *multiple* independent channels than IEEE 802.11 which assumes a single frequency channel for all nodes. As expected, the proposed protocol works as well as IEEE 802.11 in the presence of a *single* channel.

## 1 Introduction

Present MAC layer protocols for wireless communication, such as IEEE 802.11 [1] assume that all nodes operate on a single frequency. Such protocols hence do not take advantage of such wireless LAN systems that define multiple independent frequency channels. Present hardware allows a node to quickly switch from one channel to another but can only operate on one channel at a time. While using a channel, the node can send or receive packets on that channel normally, but can't send or receive any packets on any of the other channels.

We propose a MAC protocol suitable for use in an ad hoc network of mobile nodes using this multi-channel wireless LAN system. There are no base stations or access points to help coordinate which nodes use which channels at a time. Instead, each node is free to switch channels any time it needs to. In order to transmit a packet from one node to another, the sender and receiver need to be on the same channel during the time that the packet is transmitted (and received). The proposed MAC protocol allows two nodes to *rendezvous* like this (to meet on the same channel at the same time to transfer a packet from one to another), in addition to the normal job of MAC protocol in allowing the multiple nodes to contend for access to the different channels at different times, attempting to avoid collisions.

---

<sup>1</sup>The wireless extension was implemented at CMU as part of the CMU MoNARCH project, 1999

<sup>2</sup>The original version was a part of the VINT project, UC Berkeley, USC/ISI and Xerox PARC, 1998

We are not worried about how routing in the ad hoc network is done. Our aim is only to define a MAC protocol that allows one ad hoc network node to transfer a packet to a neighbor node (presumably, the neighbor node that the routing protocol has decided is the next hop for that packet, but the choice of which neighbor node is not our concern).

In the remainder of the report, we will first summarize the existing IEEE 802.11 MAC protocol specification, introduce our proposed protocol, and finally we will see the simulation results that compare the proposed protocol to existing one.

## 2 IEEE 802.11 for wireless LAN

The IEEE 802.11 standard draft describes a *Medium Access Control (MAC) and Physical Layer specification for wireless connectivity for fixed, portable and moving stations within a local area*. It supports two different MAC schemes for use of the medium to transport asynchronous and time-bounded services. One is distributed coordination function (DCF), is similar to traditional legacy packet network supporting best effort delivery of data. The other MAC scheme supported is point coordination function (PCF) where medium access is regulated by a central access manager. Since our proposed protocol does not assume or utilize the presence of a central access manager, we will see only the DCF scheme in details below.

The MAC sub layer is responsible for the channel allocation procedures, protocol data unit (PDU) addressing, frame formatting, error checking, fragmentation and reassembly. Under DCF scheme, the transmission medium operates exclusively in contention mode, requiring all stations to contend for access to the channel for each packet to be transmitted. Since it is assumed that stations are unable to listen to the channel for collisions while transmitting, the DCF scheme is based on carrier sense multiple access with collision avoidance (CSMA/CA) instead of CSMA/CD (Collision detection). Carrier sensing is done at both the air interface and at the MAC sub layer.

A source station performs virtual carrier sensing by sending MPDU duration information in the header of request to send (RTS), clear to send (CTS), and data frames. This duration information indicates the amount of time (in microseconds) after the end of the present frame the channel will be utilized to complete the successful transmission of the data. Stations listening to the channel adjust their network allocation vector (NAV), which indicates the amount of time that must elapse until the channel can be sampled again for idle status. Priority access to the wireless medium is controlled through the use of inter frame space (IFS) time intervals between the transmission of frames. There are two types of IFS used in DCF: short IFS (SIFS) and DCF-IFS (DIFS). SIFS being shorter than DIFS, thus stations only required to wait for SIFS have higher priority to access the shared channel over those stations required to wait for DIFS amount of time.

If a station wants to transmit data, it starts sensing the channel for being idle or busy. When the channel is sensed to be idle, the station waits for DIFS and again senses the channel. If the channel is still found to be idle, station transmits the RTS and waits for CTS from the destination station. Upon receipt of the correct packet, receiving station waits for SIFS and sends the CTS back. Correctness of the packet is checked using checksum bits stored in any MPDU header. The source station, after receiving CTS understands that receiver is ready to receive the data, and so after waiting for SIFS, it starts sending data as MPDU. The destination station acknowledges every data packet received by sending acknowledgment (ACK) packet. During this packet exchange between source and destination, all other stations hear-

ing the channel keep on adjusting their NAV based on the duration field value of the data frames.

If large MSDUs are handed down from link payer to MAC, MSDU need to be fragmented to increase the transmission throughput. To determine whether to perform fragmentation, the MSDU size is compared to a threshold value. If the MSDU is fragmented, all fragments are sent sequentially and channel is not released till all fragments are transmitted successfully. ACK packet are sent for each fragment.

The collision avoidance of CSMA/CA is performed through a random back off procedure. If a station with a frame to transmit initially senses the channel to be busy, then the station waits until the channel becomes idle for a DIFS period, and then computes a random back off time. After waiting for the computed back off time, it tries to transmit the data. In case of collision at that time, the back off time is increased and this procedure continues.

### 3 The proposed protocol

Under multiple channel wireless LAN environment, the functionality to be supported at the MAC layer can be divided into two parts : first making the source and destination stations rendezvous at same channel, and second, making them able to exchange data packets at that channel. While the first part is what we are going to describe below, we can use IEEE 802.11 MAC protocol to handle the functionalities corresponding to the other part.

The basic idea is to associate every station with a particular channel. The station keeps on hearing on the associated channel when ever the station is free. This mapping between station and associated channel id is known to every station. Whenever a station wants to transmit data to some other station, it finds out the channel at which the destination would be hearing, and switched itself temporarily to that channel. Now, both source and destination stations are at same channel and ready to use IEEE 802.11 MAC sub layer protocol to access the common channel and exchange data.

The association may be static or dynamic. Static association means the mapping between station and channel id does not change. Under dynamic association, the association can change depending upon the traffic load at different channels. In case of dynamic association, we need to define how the change in association is propagated to other stations. Also the traffic load parameters, based upon which decision about association change could be made by the stations, are to be identified. We have focused only on static association in our experiments.

For our experiments, we defined the static association between stations and channels as a modulo function. If there are  $k$  channels available, a station  $A$  with MAC address  $MAC_a$  will be mapped to the channel whose id is given by the modulo function :  $MAC_a$  modulo  $k$ .

When ever a station is not involved in data transmission with other stations, it will hear at its associated channel. When the station, say  $A$  decides to transmit data to station  $B$ , it switches to the channel associated with the destination station. Now station  $A$  will be hearing at the same channel as station  $B$ , and so they can follow IEEE 802.11 MAC protocol to exchange data in RTS/CTS/DATA/ACK manner. Once ACK packet is received by station  $A$ , it switches back to its associated channel.

The protocol, being so simple, is very easy to implement. It requires very little modification over IEEE 802.11 MAC protocol. If MAC addresses are randomly chosen, using modulo function for channel assignment will be fairly even. Unless dynamic association of channel is implemented, this protocol will lead to uneven utilization of channels. Even if some channel is

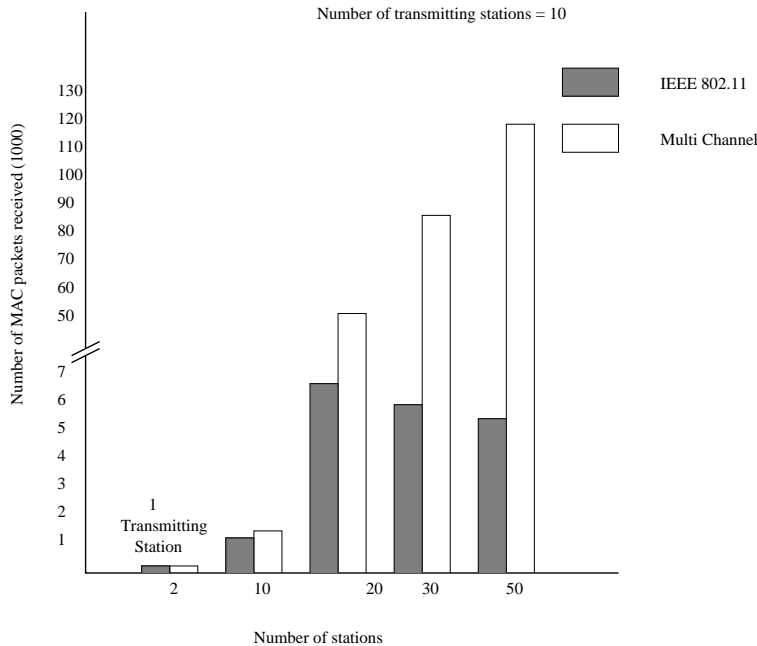


Figure 1: Performance for 10 transmitting stations ( MAC packets )

free, stations might not be able to use that channel. Also, if MAC addresses are not randomly chosen, the channel assignment would not be fair to all channels.

## 4 Results

We modified the wireless extension to ns-2 simulator [2] to simulate the proposed protocol. We simulated various scenarios by varying the number of stations in a *fixed* wireless LAN. We also varied the number of transmitting stations for each scenario. The number of channels present in each scenario was taken to be 10. The routing protocol used was dynamic source routing(DSR), though any other routing protocol would have served the purpose. The fact that we are using only fixed stations in the scenarios does not affect the performance of the MAC protocol. The stations were transmitting data using constant bit rate (CBR) flow.

Following were the scenarios for which we ran the simulation :

- Two stations,with one transmitting station (Fig.1 and Fig.2) : For this scenario, the proposed protocol performed exactly similar to the IEEE 802.11 MAC protocol. This is expected as there is only one channel that can be used at any time for data transmission.
- Ten stations, all stations trying to transmit (Fig.1 and Fig.2): Here, we found that the proposed protocol performed marginally better than the IEEE 802.11 MAC protocol. The improvement can be explained considering the availability of multiple channels for parallel transmission. However, the gain is only marginal because of all the stations trying to transmit leading to increase in contention for channels.
- Fifty stations, with 10, 20 and 30 transmitting stations (Fig.3 and Fig.4) : The proposed protocol performed much better than IEEE 802.11 in all cases. However, the gain in performance decreases with the increase in the number of transmitting stations. This

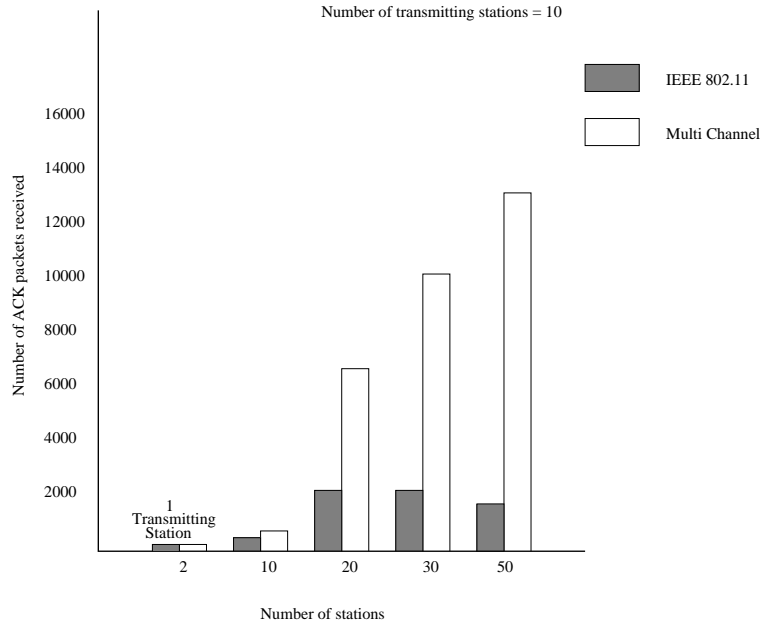


Figure 2: Performance for 10 transmitting stations ( ACK packets )

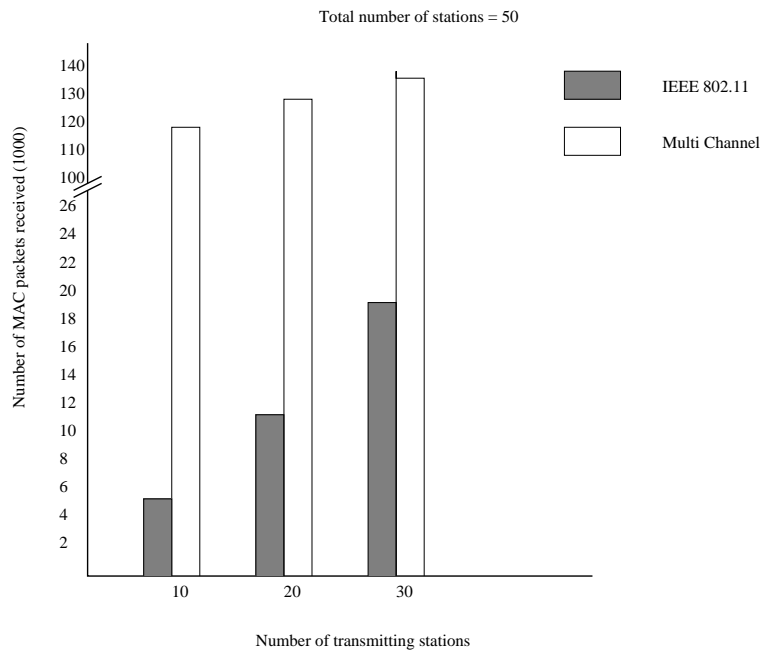


Figure 3: Performance for 50 total stations ( MAC packets )

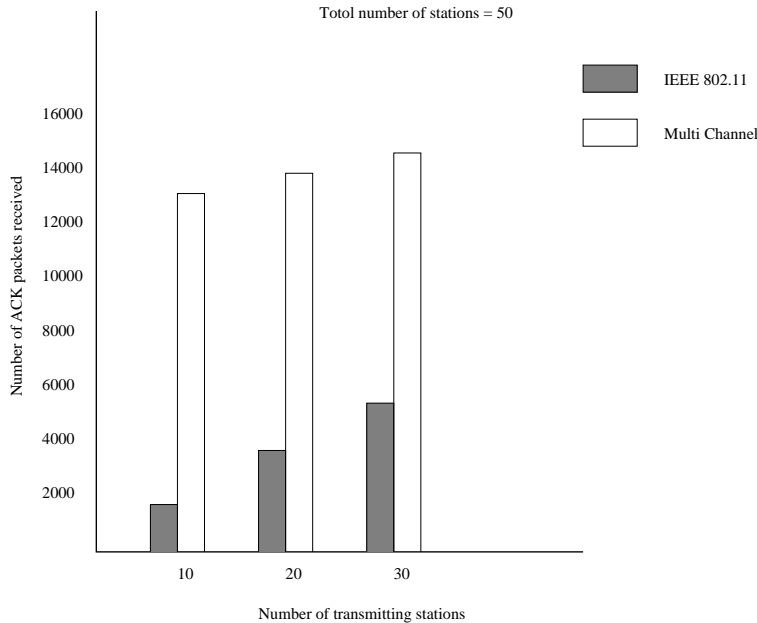


Figure 4: Performance for 50 total stations ( ACK packets )

can be explained considering the fact that increase in number of transmitters leads to increase in contention for channel access.

Similar results were found for other scenarios consisting of 20 and 30 stations. We can also see that relative performance increased with the increase in number of stations in different scenarios.

From a naive analysis of this protocol one would expect a performance of  $k$  times where  $k$  is the number of independent channels. However because of the presence of multiple channels the neighboring stations do not receive RTSs meant for other stations and hence they do not update their NAVs and are able to transmit. This leads to a better performance gain.

## 5 Conclusion and future work

In this paper we proposed a naive MAC protocol for use in a multi-channel wireless LAN. The motive behind this protocol was simply to test if at all there is any performance benefit from having multiple channels. There are many simplifying assumptions in the design of this protocol so that the protocol was easy to simulate and test. However as the simulations prove, even this naive protocol ( under certain assumptions ) is able to exploit the inherent increase in the number of channels, quite well.

We sketched out another protocol in which a transmitting station coordinates with the receiving station as to which channel the transmitter should use. The transmitting station sends a list of channels free with it and the receiving station responds by picking one from this list and the data transfer takes place on that channel. The mutual agreement of channels take place on a separate control channel of which there are a fixed number. However this protocol is in a nascent stage. Besides it would take substantial effort to implement a simulation code for it whereas the protocol that has been simulated was extremely fast and easy to simulate.

In the near future we hope to finalize the design for the protocol described above and run simulations for it in order to compare it with the naive protocol presented in this paper.

The simulation code can be downloaded from the following site :

<http://www.cs.rice.edu/santa/524/README>

## References

- [1] Crow, Widjaja, Kim and Sakai ; *IEEE 802.11 Wireless Local Area Networks*
- [2] ns-2 Documentation ; <http://www.isi.edu/nsnam/ns/doc/index.html>
- [3] <http://www.monarch.cs.cmu.edu>