Mitigating the Exposed Node Problem in IEEE 802.11 Ad Hoc Networks

Deepanshu Shukla, Leena Chandran-Wadia and Sridhar Iyer
K. R. School of Information Technology
Indian Institute of Technology Bombay
Powai, Mumbai - 400 076
(e-mail:{deepanshu,leena,sri}@it.iitb.ac.in)
(phone: +91-22-2576 7901/02, fax: +91-22-2572 3480)

Abstract— **We describe enhancements to the IEEE 802.11 DCF MAC Protocol which enable nodes to identify themselves as exposed nodes and to opportunistically schedule concurrent transmissions whenever possible, thereby improving utilization and mitigating the exposed node problem. The algorithm makes minimal changes to the IEEE 802.11 MAC protocol. It does not require making changes to frame formats or the introduction of new frame types. We show, using simulations in GloMoSim, that it provides significant improvement in throughput.**

I. Introduction

The IEEE 802.11 Distributed Coordination Function (DCF) MAC protocol [1] continues to suffer from the hidden and exposed node problems which characterize CSMA-based protocols [2], [3]. The RTS/CTS exchange partially solves the hidden node problem [4], [5] but the exposed node problem remains unaddressed.

The 4-way handshake of the RTS/CTS/DATA/ACK exchange of the IEEE 802.11 DCF protocol requires that the roles of sender and receiver are interchanged several times between pairs of communicating nodes, so neighbours of both these nodes must remain silent during the entire exchange. This is achieved by invoking the virtual carrier sense mechanism of IEEE 802.11 [1] i.e., by having the neighbouring nodes set their Network Allocation Vector (NAV) values from the Duration field specified in either the RTS or CTS packets they hear.

In Figure 1, exposed nodes E1, E2 and E3 can hear only the RTS and DATA packets sent out from node A to node B. They cannot initiate their own transmission despite being out of range of the receiver B. Similarly, nodes R1, R2, and R3 can only hear CTS and ACK transmissions from B but they must not receive any packets (respond to any RTS). This results is an unnecessary reduction in channel utilization and throughput. The suggestion that exposed nodes such as E1 ought to be able to transmit has been made several times in literature [3], [6], but the methodology to implement it within the Distributed Coordination Function of IEEE 802.11 has so far not been available.

We propose minimal changes to the IEEE 802.11 DCF MAC protocol to enable exposed nodes to transmit as and when the opportunity presents itself. Nodes identify themselves as exposed on the basis of the *sequence* of packets that they hear e.g., a node that hears an RTS followed by DATA from the same node, within the specified time interval, can be certain that it is an exposed node. Such a node can initiate a parallel transmission by simply aligning its DATA transmission with the ongoing transmission, without invoking the RTS/CTS exchange. Details of the algorithm are described in Section III.

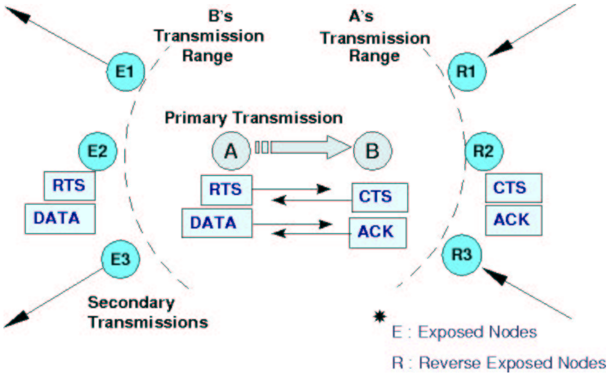


Fig. 1. Exposed nodes E1-E3 and R1-R3

We show using simulations in GloMoSim, that our enhancements provide significant improvements in throughput. Details of the simulations and the results are presented in Section IV.

II. Related Work

In a recent paper[7], the authors propose a more complex solution, the MACA-P protocol, to address the exposed node problem. In this section we describe the salient features of this protocol which envisages extensive revisions to the IEEE 802.11 DCF. It proposes several additions to the RTS/CTS frame formats and also introduces a new type of control packet called RTS'.

The key feature of MACA-P is that it introduces a *control gap* between the RTS/CTS exchange and the subsequent DATA/ACK exchange of the first pair of communicating nodes. This control gap is then exploited by other pairs of nodes to complete their own RTS/CTS exchange and to align their data transfer with the DATA and ACK packets of the first pair.

In order to achieve this alignment, the RTS/CTS packets must be extended to contain two new time intervals the T_{DATA}

and the T_{ACK} which schedule the proposed start time of the DATA and ACK packets respectively. The RTS packet is also enhanced to carry a bit called the *inflexible* bit, which indicates whether the proposed transmission schedule can be changed by the node that receives the RTS.

When a node receives an RTS in which the inflexible bit in *not set*, it may change the proposed schedule by modifying the T_{DATA} and T_{ACK} and send the modified values back in the CTS packet. These modifications must be redistributed to neighbours of the sender who can only hear an RTS. For this MACA-P introduces a new frame, the RTS', which is always sent by the RTS sender after receiving a CTS. It contains updated T_{DATA} and T_{ACK} values. The RTS' packet also enables nodes to free the channel through cancellation messages. Other refinements to MACA-P are described in the paper [7].

III. Proposed Solution

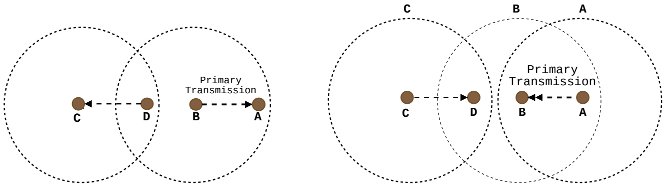
We suggest a more local and opportunistic approach: Whenever there is a data transmission in progress (called the primary transmission in the rest of the paper), an exposed node can try to squeeze in a parallel or secondary transmission for better overall throughput. As shown in Figure 2(a), once the DATA transfer from B to A begins, node D can commence DATA transmisssion to node C directly, without making use of the RTS/CTS exchange.

Both transmissions can succeed provided the ACKs returning to nodes B and D are synchronized (Figure 2(b)). For this, the start time of the secondary transmission from D must be adjusted so that it finishes at the same time as the transmission from B. This ensures that when senders B and D, which are within the transmission range of each other, switch roles to receive their respective ACKs, they can do so without interference from each other.

As mentioned earlier (Section I), a node recognises itself as an exposed node on the basis of the *sequence* of packets that it hears i.e., the RTS packet followed by the start of the DATA packet. Therefore, the size of the packet that can be sent in the secondary transmisssion must be smaller than the size of the packet currently being transmitted. The maximum size of such a packet is determined by the fact that the end of transmission must coincide with the end of transmission of the primary packet. An explicit calculation is presented in the following section. Thus, an exposed node can potentially schedule parallel transmission of a smaller sized packet contributing to the increase in channel utilization.

The proposed solution makes use of the following two observations and known facts:

- Traffic statistics show that approximately 50% of all packets on the Internet are small packets below 100 bytes in size [8].
- Several studies have shown that it is inefficient to use RTS/CTS for very small packets and that an optimum threshold is somewhere between $200 - 500$ bytes [6], [9], [10]. Many simulation studies are done with an optimal value of 250bytes.



(a) Data

(b) ACKs

Fig. 2. Secondary transmissions

The Internet traffic statistics [8] refer to the wired Internet. The statistics for the wireless Internet can be expected to show a similar spread in packet sizes. The detailed distribution of packet sizes is not of much consequence from the point of view of this work. It suffices that there will be large numbers of small packets that do not need to use RTS/CTS. These can potentially be sent in parallel transmissions by the exposed nodes.

In this approach, an exposed node can make use of a "transmit opportunity" only if there is a small enough packet at the front of the queue. The IEEE 802.11 uses only a single queue and FIFO scheduling. With multiple queues as in IEEE 802.11e, or with scheduling disciplines other than FIFO, this algorithm will perform better, since small packets can be picked out more easily for parallel transmissions.

A. Description of the Algorithm

As per the IEEE 802.11 standard [1], a node which overhears an RTS and sets its NAV from the Duration field in the RTS packet also sets a timer, which we call the RESET_NAV timer, for the duration of $CTS\_TIME + 2 * SIFS\_TIME + 2 * SLOT\_TIME$. The CTS_TIME is calculated from the length of the CTS frame and the rate at which the RTS frame, used for the recent NAV update, was received. The node resets its NAV unless it overhears DATA from the RTS sender before this timer expires. We make use of this timer to initiate secondary transmissions.

The RESET_NAV timer duration is the maximum delay with which DATA transmission at the RTS sender can be heard at this node. So if the timer expires, then either the CTS was not heard by the RTS sender or the CTS was not transmitted, and nodes neighbouring the RTS sender are free to reset their NAV. If DATA transmission *does begin*, we allow the node in question to conclude that it is an exposed node and schedule its own transmission.

For a 2 Mbps radio, the value of the RESET_NAV timer is $308\mu s$. Within this time either a DATA packet will start or the primary sender will backoff for resending its RTS.

The algorithm does not try to schedule secondary transmissions when there are multiple transmissions already in progress. If the node hears a packet (RTS/CTS or DATA) from any other node during this period, the timer is cancelled and a secondary transmission is not scheduled.

B. Implementation

In practice the implementation of the above algorithm in GloMoSim [11] proceeds slightly differently. As per the

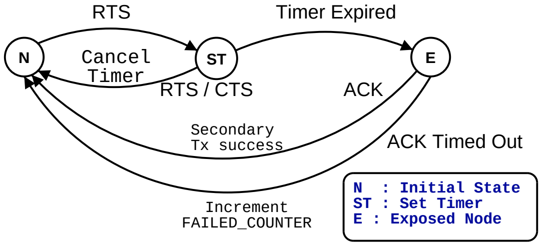


Fig. 3. State Diagram for an exposed node

standard, once the phys layer has completed the CRC check, it assembles the received data into octets and delivers them to the MAC layer one octet at a time. In GloMoSim this does not happen. Data is handed over to the MAC just once, when reception is complete. This implies that a potential exposed node cannot check whether it has started receiving DATA from the RTS sender. Instead, it has to wait for the RESET_NAV timer to expire in order to schedule its secondary transmission.

The state diagram for such a node in GloMoSim is shown in Figure 3. On reception, of an RTS frame a node in the normal state N sets its RESET_NAV Timer and transitions into the state ST. While in the ST state, if it receives RTS/CTS/DATA from any other node during this time, it cancels the timer and returns to normal mode. If, on the other hand, the RESET_NAV timer expires it identifies itself as an exposed node E, schedules a transmission and awaits an ACK. If an ACK is received then the secondary transmission is successful. If an ACK is not received, then the attempt to make a secondary transmission has failed and the node must return to normal and send the data as per standard.

For implementation of the algorithm and the simulations we have also used a separate timer called the CHECK_EXPOSED timer which has the same value as the RESET_NAV timer. This is done so as to make minimum changes to the standard implementation of the 802.11 MAC protocol in GloMoSim. The algorithm, as implemented in GloMoSim, is presented alongside.

This algorithm is an optimistic enhancement, so there may be situations when the secondary transmission fails. If this happens the exposed node will not modify its backoff counter. It will pretend that the transmission never took place (roll back) and return the DATA packet for nornal handling as per the standard. Each node also maintains a counter namely, the EXPOSED_ FAILED_ COUNTER which keeps count of the number of secondary transmissions that failed. When this exceeds a particular value (MAX_ FAILURE), the node stops attempting to send secondary transmissions. The usage of this counter is also discussed in Section IV.

C. Comparison with MACA-P

Apart from the differences in the complexity levels of the two approaches, it is relevant to note that the storage requirements of MACA-P are large. Each node is required to maintain the state of its neighbouring nodes, by overhearing the RTS/CTS exchanges from its neighbours. The single NAV

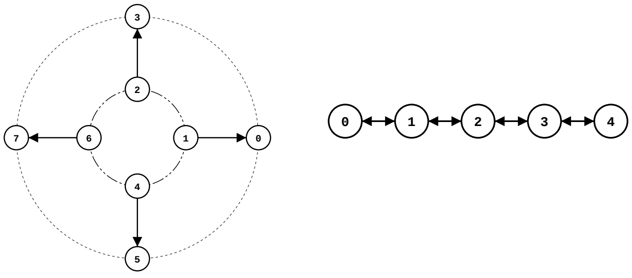
Algorithm 1 Exposed Node Algorithm

```
CASE : RTS HEARD FROM RADIO
  if ANY of the following conditions are satisified
    • Packet belongs to THIS node
    • There are no DATA packets to send
    • Node status is "Exposed Timer SET"
    • Node status is "Waiting for Secondary ACK"
    • $EXPOSED\_FAILED\_COUNTER$ has exceeded $MAX\_FAILURE$
  then
    Cancel all Timers
    Cancel all scheduled data transmissions, if any
    Handle RTS as per standards
    {In case a node hears multiple RTS it resets it status and does not proceed with modified Algorithm}
  else
    $Size_T \leftarrow$ Size of on going Transmission
    $Size_Q \leftarrow$ Size of packet at head of queue
    $T_{DATA} \leftarrow$ Time needed to transmit data
    $T_{START} \leftarrow hdr.duration - T_{DATA} - ACK\_duration - SIFS$
    {hdr.duration is the duration field in the RTS by which NAV is set}
    {$T_{START}$ is the actual time at which the Exposed data packet will be transmitted}
    if $Size_Q < Size_T$ AND $T_{START}$ is positive then
      set CHECK_EXPOSED Timer
      Set Node status as "Exposed Timer Set"
    else
      Handle RTS as per standards
      {The Data exceeded size limits }
    end if
  end if

CASE : TIMER EXPIRED
  if CHECK_EXPOSED Timer AND Node status is "Exposed Timer Set" then
    Set Node status as "Waiting for Secondary ACK"
    Set ACK_TIMEOUT timer
    Schedule $DATA$ transmission at $T_{START}$
  end if
  if ACK_TIMEOUT Timer AND Node status is "Waiting for Secondary ACK" then
    Increment the $EXP\_FAILED\_COUNTER$
    {The secondary transmission failed}
    Set WAIT_FOR_DIFS Timer
    {After DIFS period the node will attempt to transmit the data packet normally.}
  else
    Handle Timer as per standards
  end if

CASE : CTS OR DATA HEARD FROM RADIO
  Cancel ALL Timers
  RESET node status as specified in standards
  Handle packet as per standards

CASE : ACK HEARD FROM RADIO
  if Packet belongs to THIS node AND node status is "Waiting for Secondary ACK" then
    Reset $EXPOSED\_FAILED\_COUNTER$ to $ZERO$
    {Secondary transmission successful}
  end if
  Cancel ALL Timers
  RESET node status as specified in standards
  Handle ACK as per standards
```



(a) Ring

(b) String

Fig. 4. Simulation Topologies

value which normally needed to be stored by each node is now replaced by a table.

When a node wants to send a data packet, it uses this table to check that there is no neighbour in the table which is labeled a recepient. Similarly if a node receives an RTS it needs to consult the table to ensure that none of its neighbours is a transmitter before it can respond with a CTS. In addition the NAV allows a node to find out if there is a transmission already scheduled and to align its own transmission to it.

Throughput improvements from our algorithm are compared to those in MACA-P in Section IV-A.

IV. Simulation and Results

The proposed changes in the IEEE 802.11 MAC protocol were implemented in Glomosim [11] and simulations for various topologies carried out. The topology shown in Figure 4(a) is also studied with MACA-P. It consists of an inner ring and an outer ring, with each node on the inner ring sending packets one-hop to its corresponding receiver on the outer ring.

Traffic consists of CBR flows of packet size 1024 bytes between nodes 1 and 0 and nodes 6 and 7, Flows between the other two pairs contain smaller packets of size 512 bytes. The packet rate was sufficiently high to keep the network saturated. Default Glomosim parameters were used for the simulation [12]:

- Frequency (2.4 GHz), Pathloss Model (Two-ray) , Noise Calculation (Cumulative - AccNoise) , Radio Rx Sensitivity (-91.0 dBm), SNR-Threshold (10.0), Radio-RX-Threshold(-81.0 dBm), Transmit power (-15dBm), Bandwidth (2Mbps).

With these parameters, the transmission range is 376 meters and interference range 637 meters. The distance from node 1 to each of its two neighbours in the inner ring, nodes 2 and 4, as well as to node 0 on the outer ring is 350m. Thus when node 1 sends data to node 0, nodes 2 and 4 are within its transmission range and are therefore exposed. As per our algorithm they can send smaller packets to their respective outer nodes in parallel. When the number of nodes in the simulation was varied, care was taken to ensure that roughly half the flows had packets of each size, to aid the comparison.

The results for average throughput is compared with IEEE 802.11 in Figure 5(a). The improvement in throughput is presented as a percentage increase in Figure 5(b). The improvement (152% increase) compares favourably with the 200% increase reported in MACA-P using NS-2 [13], (discussed further in Section IV-A). The latter graph also shows the contribution to increase in throughput from packets of each size. As expected, the increase in throughput is due to a much larger number of smaller sized packets getting through in parallel transmissions.

For larger numbers of inner nodes the percentage of large packets that are delivered (1024 bytes) reduces drastically. This is due to the fact that GloMoSim implements the cumulative noise model and power capture [12]. Both factors contribute to reduction in throughput. As more secondary transmissions take place, the background noise increases substantially hindering packet reception. The power capture behaviour of GloMoSim is such that a stronger signal is captured *any time* during the reception of a packet. So it appears that some 1024 byte packets are lost when stronger secondary transmissions are started in the neighbourhood.

Similar studies were done for the string topology of Figure 4(b). The traffic flow is multihop, from node 0 to node N-1 and from node N to node 1 in an N+1 node string. Packet sizes for the flows in one direction is 1024 bytes and 512 bytes for flows in the reverse direction. The algorithm provides sizeable improvement (176% increase) for a string length of 4 Nodes. The drop in improvement for larger number of nodes can be attributed to a couple of reasons. The throughput itself drops drastically with string length [14] and there is also the interference from many more parallel transmissions.

Since our algorithm is opportunistic, the improvement it provides depends largely on two factors, the opportunities that exposed nodes receive and the interference that is generated due to the parallel transmissions. To get a flavour for the contributions of these two factors in random networks, we have simulated a network consisting of different numbers of nodes moving in a 2000m x 1500m area. Roughly a third of the nodes are sources. Traffic flows are CBR, with each flow having different packet rates selected randomly between 10 packets/second to 200 packets/second. The routing protocol is AODV.

The packet sizes for each flow is also different, taking values from among 256, 512, 768 and 1024 bytes respectively. About half the flows consisted of large-sized packets. The results for the improvement in throughput is shown in Figure 7 for 20, 40 and 60 nodes respectively, for different pause times. Nodes move according to the Random-Waypoint [15] model with speeds ranging from 1 m/s to 5 m/s. The simulatin time is 400s.

Two trends are visible in the results. The improvement increases with reducing mobility. It also increases with increase in the number of nodes reflected in these simulations as an increase in "transmit opportunities". More simulations will have to be done to understand the balance between increased opportunity and increased interference in this topology. The overall improvement in throughput is about 20% in this simulation.

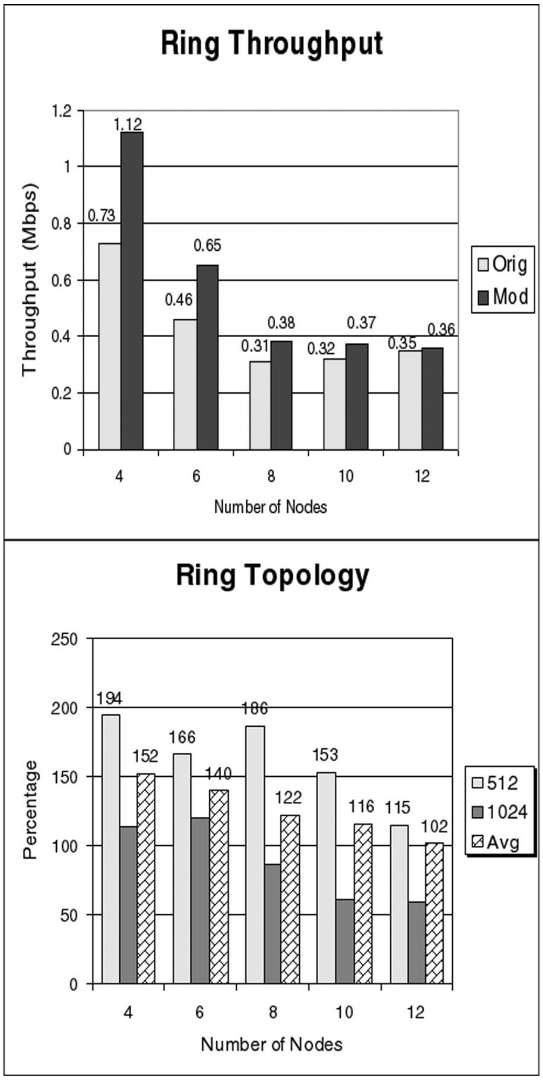


Fig. 5. Average Throughput and its breakup with respect to packet sizes

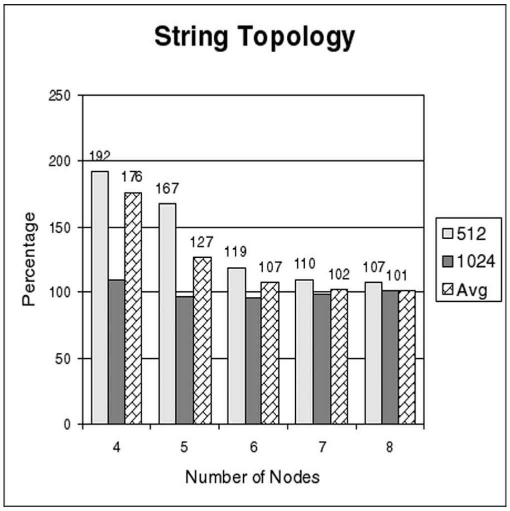


Fig. 6. Improvement in average throughput for a String of Nodes

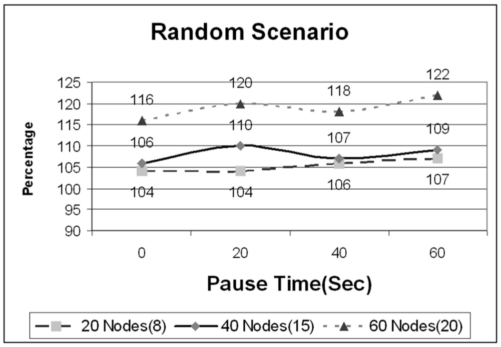


Fig. 7. Throughput improvement for randomly placed mobile nodes

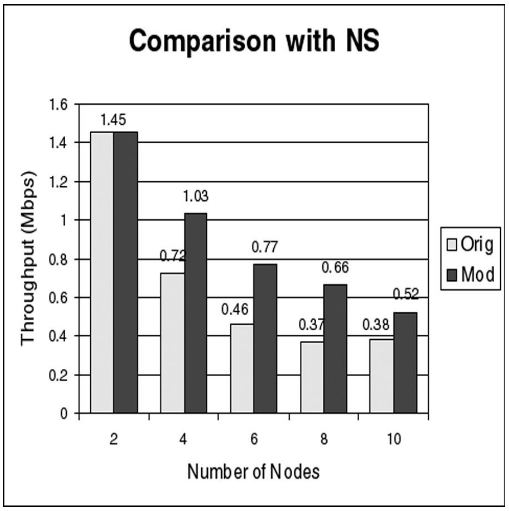


Fig. 8. Average throughput in the Ring Toplogy

In dense networks the interference effect can be expected to cause a drop in throughput as either primary or secondary transmissions fail. This effect has already been seen in the results in Figure 5(b). We have made a refinement in the algorithm, to reduce secondary transmissions in such situations. The EXPOSED_FAILED_COUNTER (discussed in SectionIII-B) keeps count of the number of failed secondary transmissions. If this increases beyond a certain preset number (MAX_FAILURE), the node stops further attempts to transmit secondary packets. In effect the algorithm is not invoked if the algorithm fails to execute successfully on previous occasions. In our simulations we set the value of MAX_FAILURE to be between 2 and 5.

A. *Comparison with MACA-P*

To try to separate the effect of interference, we have also measured throughput in the ring topology of Figure 4(a) with GloMoSim parameters chosen to correspond [12] with that of NS-2 [13]. The latter implements the No noise model which reduces interference and improves throughput, but it does not implement power capture. The results for throughput are shown in Figure 8. The improvement for 4 nodes is about 170%. The authors of MACA-P also present similar results using NS-2 in the same topology. They obtain throughput

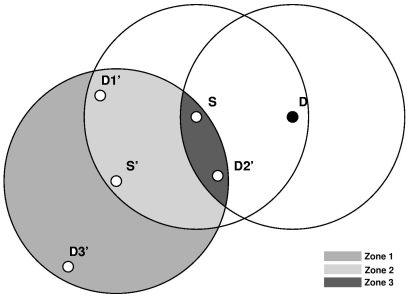


Fig. 9. Interference effect

improvement of over 200% in some cases. Some details of their topology are different from our simulations. For e.g., in their ring topology, all the inner nodes are exposed when one of them is transmitting whereas in our simulations only the immediate neighbours are exposed, so there are fewer parallel simulations.

The performance of both MACA-P and our protocol depends very strongly on the capture capability of the radios. As shown in Figure 9 the location of the destination of the secondary transmissions will also have an important bearing on its success. Destinations of the type D2' will not receive their DATA while destinations of type D3' will most likely do so. With destinations of the type D1'it is apriori unclear as to whether secondary transmissions will succeed.

The interference range (determined by Radio_Rx_Sensitivity) of these radios is much larger than the transmission range (determined by Radio_Rx_Threshold). Nodes within this interference range only sense the channel to be busy through the physical carrier sense and not the NAV. These will continue to remain exposed. This is a physical limitation and cannot be overcome.

B. Limitations of the algorithm

This algorithm does not enable nodes of the type R1, R2, R3 in Figure 1 to receive data in parallel. So these continue to behave like they would in IEEE 802.11. Figure 2(b) illustrates this well. The primary transmission in this case is from node A to node B and node D is the exposed node. In this topology shown in the figure, node C has no information about timing synchronization or the size of primary data transmission. In such a case this algorithm can not enable node D to participate in concurrent transmissions.

While it is true that MACA-P is a fairly complex protocol, it does handle more situations than our protocol. In the above example, MACA-P can sometimes enable node D to participate in secondary transmissions. If node C happens to send an RTS to node D during the control gap of the A to B communication, then node D can indicate the revised timings it would like to receive data in so as to align with the primary transmission. If that is acceptable to node C, then the C to D communication can proceed in parallel. In general, if either node in the secondary pair has heard the RTS/CTS exchange of the primary pair and the secondary sender has initiated communciation in the control gap, then MACA-P will allow the transmission but our algorithm will not.

V. Conclusion

We have presented simple enchancements to the IEEE 802.11 DCF MAC protocol which allow concurrent transmission to occur whenever possible, reducing the loss in throughput due to the exposed node problem. While it has long been recognized that exposed nodes should be free to transmit, we have presented an explicit algorithm within the IEEE 802.11 framework. It makes use of the observed fact that traffic on the Internet has a large number of small packets, and the accepted fact that it is inefficient to use RTS/CTS for such small packets.

The algorithm is local and opportunistic and will therefore give varying levels of improvement. However in most of the scenarios studied, the algorithm gave a sufficiently good performance. The algorithm also learns from past failures so that there is no drastic drop in overall throughput in dense networks.

References

[1] IEEE Computer Society LAN MAN Standards Committee:. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. In *ANSI/IEEE Std. 802.11, 1999 Edition, The Institute of Electrical and Electronic Engineers*, New York, 1999.

[2] 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 Transaction on Communications*, COM-23(12):1417–1433, 1975.

[3] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. *MACAW: A media Access Protocol for Wireless LANs*. In *Proc. ACM SIGCOMM'94*, pages 212–215, London,UK, August 1994.

[4] P. Karn. *MACA - a new channel access method for packet radio*. In *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pages 134–140, September 1990.

[5] C.L. Fullmer and J.J. Garcia-Luna-Aceves. *Solutions to Hidden Terminal Problems in Wireless Networks*. In *SIGCOMM 97*, pages 39–49, September 1997.

[6] J. Weinmiller, H. Woesner, and A. Wolisz. Analyzing and tuning the distributed coordination function in the ieee 802.11 dfwmac draft standard. In *Proc. of MASCOT '96*, San Jose, California, February 1996.

[7] A. Acharya, A. Mishra, and S. Bansal. *MACA-P: A MAC for Concurrent Transmissions in Multi-hop Wireless Networks, IBM Research Report RC22528, July 18, 2002*. In *Proc. IEEE PerCom*, 2003.

[8] K. Claffy, G. Miller, and K. Thompson. *The nature of the beast: Recent traffic measurements from an Internet backbone*. In *INET98*, Geneva, Switzerland, July 1998.

[9] J. G. Kim B. P. Crow, I. Widjaja and P. T. Sakai. *IEEE 802.11 wireless Local Area Networks*. *IEEE Communications Magazine*, pages 116–126, 1997.

[10] G. Bianchi. *Performance Analysis of the IEEE 802.11 Distributed Coordination Function*. *IEEE Journal on Selected Areas in Communications*, pages 535–547, 2000.

[11] M. Takai, L. Bajaj, R. Ahuja, R. Bagrodia, and M. Gerla. *GloMoSim: A Scalable Network Simulation Environment*. In *Technical Report 990027*, UCLA, 1999. http://pcl.cs.ucla.edu/projects/glomosim/.

[12] Mineo Takai, Jay Martin, and Rajive Bagrodia. *Effects of Wireless Physical Layer Modeling in Mobile Ad Hoc Networks*. In *Proc. ACM Mobihoc'01*, Long Beach, California, Oct. 2001.

[13] Kevin Fall and Kannan Varadhan eds. *NS notes and documentation*, http://www.isi.edu/nsnam/ns/.

[14] Jinyang Li, Charles Blake, Douglas S. J. De Couto, Hu Imm Lee, and Robert Morris. *Capacity of Ad Hoc wireless networks*. In *Proc. of ACM Mobicom'01*, pages 61–69, Rome, Italy, July 2001.

[15] J. Broch, D.A. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva. *A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols*. In *Proc. of ACM Mobicom'98*, pages 85–97, Dallas, Texas, Oct. 1998.