http://ansinet.com/itj

ITJ

ISSN 1812-5638

# INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 6 (8): 1231-1237, 2007
ISSN 1812-5638

# Implementation of Modified MAC Based Pipelining Technique for Wireless Networks

[1]V. Kumar and [2]P. Balasubramanie
[1]Department of Information Technology, Nandha Engineering College, Erode, Tamil Nadu, India
[2]Department of Computer Science and Engineering, Kongu Engineering College,
Perundurai, Tamil Nadu, India

**Abstract:** In today's modern world, people needs the fastness and efficiency in any system that we considered. So, there is a need for a mechanism which satisfied the needs of network users. Pipelining is a technique that allows the execution of many jobs to overlap in time. Overlapping is accomplished by dividing each job into stages. This study presents concepts of medium access control layer in the IEEE 802.11 Distributed Coordinated Function (DCF). Also, it modifies the medium access control layer mechanism and implementing it in the pipelining technique. The pipelining techniques used are partial pipelining technique and total pipelining technique. In the existing technique one control channel and one data channels are used. The control channel is for contention resolution and the data channel is for transmitting data and to receive acknowledgment. But in the proposed technique two contention resolution channels and one data channel is used.

**Key words:** Wireless network, MAC, pipelining, control channels, data channels

## INTRODUCTION

New Medium Access Control (MAC) protocols are constantly being developed in an attempt to approximate the ideal as closely as possible. Some protocols optimize the system throughput at the expense of fairness while others give preference to achieving a greater level of fairness in wireless networks. Given that protocols are continually being proposed, various metrics have also been developed in order to be able to evaluate the efficiency and robustness of these protocols. One common metric is channel utilization, which is defined to be the fraction of time spent on successful transmissions. Another metric is saturation throughput, which represents the maximum throughput the system can achieve under stable conditions. Both channel utilization and saturation throughput are directly affected by the amount of overhead a protocol introduces for each successful transmission and the efficiency with which collisions are resolved. Protocol fairness is also an important issue.

The function of Medium Access Control (MAC) is to enable the effective sharing of the channel among all nodes in the network. Ideally, for both wired and wireless networks, the MAC protocol should facilitate perfect channel utilization while still being fair to all flows within the network. The conflict between these goals usually does not allow the ideal to be realized in practice.

We will present some of the research that has been done in the field of Medium Access Control (MAC) for ad hoc networks. Much of the research here is motivated by the fact that the prior research done with respect to wired networks is not directly applicable to the wireless arena. This is a direct consequence of the different properties of the wired and wireless medium.

One of the earliest MAC protocols proposed was ALOHA (Yang and Vaidya, 2003a). In pure ALOHA, nodes transmit whenever they have packets to send, without regard to the current state of the medium. A positive acknowledgment scheme is used to determine if the transmission was successful. This is a common feature in most MAC protocols designed for the wireless medium since the wireless transceiver does not allow a node to transmit and listen to the channel simultaneously. If no ACK returns, a collision is assumed to have occurred and the node retransmits after a random delay. Since no form of carrier sensing is done, a transmitted packet is vulnerable to collision for an interval equal to twice its transmission time. Put differently, if a second transmission starts within a packet's vulnerable period, a collision occurs and both transmissions are unsuccessful. A throughput analysis shows that pure ALOHA utilizes at most 18% of the available channel bandwidth. A simple extension to pure ALOHA was to divide time into discrete intervals, known as slots and allow nodes to only transmit at the beginning of a slot. This extension, known as slotted ALOHA, halves the vulnerable period of a packet and is able to achieve a maximum channel utilization of 36%, i.e., double that of pure ALOHA.

---

**Corresponding Author:** V. Kumar, Department of Information Technology, Nandha Engineering College, Erode, Tamil Nadu, India

To further increase the channel utilization, Carrier Sense Multiple Access (CSMA) protocols were developed (Yang and Vaidya, 2003b). CSMA protocols can be divided into non-persistent CSMA and p-persistent CSMA. In both categories, nodes first sense the channel before any action is taken. In non-persistent CSMA, the node transmits if the medium is idle. If the medium is sensed as busy, the node waits a random delay period before retrying. In p-persistent CSMA, the node continually senses the medium until it is idle. It then transmits in a give slot with probability p an defers transmission to the next slot with probability 1-p. If a collision occurs, the node waits a random delay before retransmitting. In the limit, 1-persistent CSMA employs the greedy principle and always transmits as soon as it senses the channel as idle.

Crow *et al.* (1997), Yan and Park (2004) and Bianchi *et al.* (2006) shows that the CSMA protocols are superior to both pure and slotted ALOHA. CSMA protocols not only achieve significantly higher throughputs, they also perform better under high network loads. CSMA is also more efficient than ALOHA as it has smaller delay times and requires a smaller number of retransmissions per successful transmission. Given these obvious benefits of incorporating carrier sensing into the MAC protocols, most of the succeeding protocols have been based on the CSMA model.

## MATERIALS AND METHODS

The concept of pipelining in MAC protocols is not a new one. Todd and Mark consider this issue for a generic MAC protocol. They divide each transmission cycle into a bandwidth-independent component, a bandwidth-dependent component and the successful transmission. The bandwidth-independent component accounts for the idle slots within a transmission cycle while the bandwidth-dependent component accounts for the collisions that occur. In terms of these components, analytically compares the capacity of the generic un-pipelined MAC protocol with that of the generic pipelined MAC protocol. Their analysis shows that the pipelined protocol cannot do any better than the un-pipelined protocol if there is no bandwidth-independent component in the transmission cycle. On the other hand, if there are no bandwidth-dependent components in the transmission cycle, pipelining does lead to a gain in performance. The performance gain decreases as the ratio of the idle overhead to the data packet size increases. Thus the benefit of pipelining will depend on how much of the transmission cycle is bandwidth-dependent and how much is bandwidth-independent.

Yang and Vaidya (2002a, b) propose Pipelined Packet Scheduling (PPS) that partially pipelines the contention resolution process over a data channel and a narrow busy tone channel. The contention resolution process is divided into two stages and both stages adopt the back-off algorithm, each with its own set of CW min and CW max values. All backlogged stations start in the first stage and decrement the back-off counter every slot, regardless of the channel status. When one node's back-off reaches 0, it transmits a busy tone to inform its neighbors that it has won the first stage. All nodes that hear this busy tone remain frozen in stage 1 until the next transmission commences. While the first stage occurs in parallel to the data transmission, the second stage begins at the end of the current transmission cycle. All nodes that enter the second stage wait for a second back-off interval before transmitting. If the channel remains idle until the end of the back-off duration, the node transmits its packet. If the channel is sensed busy before the end of the back-off interval, the node loses the contention, returns to the first stage and recon tends. If collisions occur in the second stage, the contention windows in both stages are exponentially increased. The end of the second stage is marked by a successful transmission. In this way, much of the protocol's idle overhead is effectively masked. The authors show that, at high loads, unlike IEEE 802.11, the saturation throughput remains close to the peak value. PPS also maintains the same level of fairness IEEE 802.11 displays.

Further analysis shows that the performance achieved is not sensitive to the probability of busy tone detection. In fact, even when a busy tone is not transmitted, the performance of PPS does not degrade significantly. Using this fact, the authors extend the PPS algorithm in Xiao *et al.* (2006) to propose the Dual Stage Contention Resolution (DSCR) protocol. DSCR has the same protocol structure but does away with the need for the busy tone channel. In DSCR, all deferring nodes, excluding the node that just transmitted the successful packet, decrement their first stage back-off counter by a fixed value F at the end of a successful packet transmission. After this, nodes decrement their back-off counters for each idle slot sensed. Notes enter stage 2 when their back-off counter in stage 1 is less than or equal to 0. If nodes enter stage 2 at the beginning of transmission cycle, they wait a random back-off interval before attempting to transmit. If a node enters stage 2 during a transmission cycle, it immediately commences transmission. Once a transmission commences on the channel, the other contending nodes return to stage 1, double their CW sizes and re-contend. Simulation results in Yang *et al.* (2006a) show that DSCR achieves a higher performance than IEEE 802.11 and MACAW, even in the presence of hidden terminals. DSCR also uses a smaller minimum value for the contention window in stage 2. This increases the difference in CW sizes between the winning node and the rest of the network, similar to the situation seen in Yang and Vaidya (2002a, 2003a) and Yang *et al.* (2006b). As a result, DSCR performs worse than IEEE 802.11 with respect to fairness. Yang and Vaidya (2003a) and Yang *et al.* (2006a) Show that if the value of F is adaptively modified, so that nodes that have been in stage 1 longer decrease their back-off counters at a faster rate, the fairness of DSCR is significantly improved, though the performance gain is reduced.

Yang *et al.* (2006b) compare PPS, DSCR and the explicit pipelining scheme, similar to DCMAC. As the authors noted, the explicit pipelining scheme is only beneficial if both stages in the pipeline are balanced. This balance may be too difficult to achieve when pipelining IEEE 802.11 as the performance is expected to be heavily dependent on the bandwidth division, the distribution of the packet size and the number of flows. Here, we implement an explicit pipelining scheme and evaluate the exact dependence of the performance on these factors. This will enable us to conclusively state if pipelining IEEE 802.11 is as impractical as it intuitively seem.

In this study, we attempt to improve the performance achieved by masking the contention overhead via pipelining. Pipelining allows the execution of various jobs to overlap in time. This is accomplished by dividing each job into stages and as soon as the first job finishes a stage, the second job enters that stage as the first job moves on to the next stage. Figure 1 shows this concept by using a three-stage pipeline. In MAC protocols, the job is the dialog to transmit a packet and it is divided into a contention-resolution stage and a packet transmission stage.

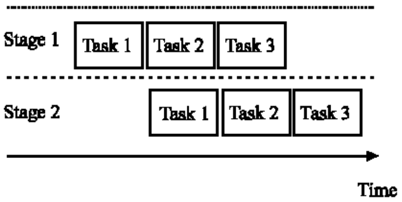

Fig. 1: Basic pipelining

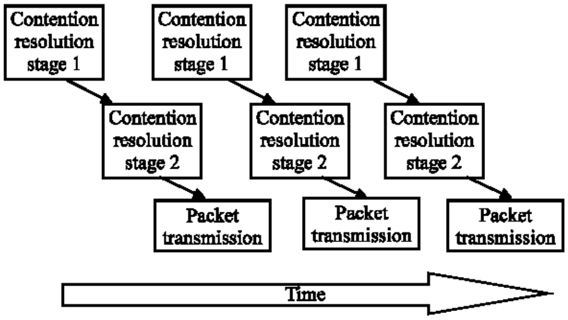

Fig. 2: Proposed pipelining technique sequence diagram

## PROPOSED TECHNIQUE

In this new pipelining technique, contention resolution stage is divided into two stages as shown in Fig. 2. When a transmission cycle begins, some stations enter stage 2, while others stay in stage 1. Only the stations in stage 3 may try to transmit a packet. Stations in stage 1 will stay in stage 1 until next transmission cycle begins. However, it is possible that no station enters the second stage at the beginning of a transmission cycle. In addition, the stations in stage 1 do not know whether there are stations in the second stage or not, unless we have blackburst or some equivalent mechanisms to indicate the status of each stage. Therefore, it is still possible for stations in stage 1 to enter stage 2 during a transmission cycle; we can call it as interference between the two stages.

The new technique differs from 802.11 DCF only in the contention resolution procedure. The other functions remain the same.

**Contention resolution stage 1:** A back-off counter, named bc1, is associated with the first contention resolution stage. bc1 is chosen to be uniformly distributed over the interval (0, CW1). CW1 is the contention window for stage 1 and has a minimum value CW1 min and maximum value CW1max. The initial value of CW1 is CW1 min. The back-off counter bc1 is counted down in two ways:

- Recall that the beginning of a transmission cycle is marked by the end of a successful packet transmission, which a station could determine using its NAV. At the beginning of a transmission cycle, each active station reduces bc1 by K, except for the station which just finished transmitting a packet. Since a station will reset CW1 to CW1 min after a successful transmission, if it immediately reduces bc1 by K, this station will have more chances than other stations to enter stage 2.
- During a transmission cycle, each active station reduces its bc1 by 1 after each idle slot.

Whenever a station's bc1 becomes less than or equal to 0, this station enters stage 2. If a station's bc1 is counted down to zero during a transmission cycle (i.e., by above the second method for decrementing bc1 by 1), we say this station interferes with the second stage, as explained at the beginning of this section. With a suitable choice of K, interference only happens to a small fraction of stations entering stage 2. Now assume that there are M stations that enter stage 2 in a transmission cycle. As seen later, among the M stations, one station will eventually win channel access. The remaining M-1 stations in stage 2 will return back to the first stage, double CW1 and regenerate bc1. The winning station will finish its packet transmission, then return back to the first stage, reset CW1 to CW1min and regenerate bc1. Those stations which do not enter the second stage will not participate in channel contention and hence, will not have their CW1 value changed.

**Contention resolution stage 2:** A back-off counter, named bc2, is associated with the second contention resolution stage. If a station enters stage 2 during a transmission cycle, it will set the initial value of bc2 to 0. On the other hand, if a station enters stage 2 at the beginning of a transmission cycle (i.e., bc1 is less than or equal to zero after reducing by K), it will generate an initial value for bc2 uniformly distributed over the interval (0, CW2). CW2 is contention window for the second stage. It has minimum value CW2 min and maximum value CW2 max. The initial value of CW2 is CW2 min. After the channel has been idle for DIFS duration, each station in stage 2 decreases bc2 by 1 after each slot. When bc2 reaches zero and the channel is idle, the station will begin its transmission. Before the bc2 of a station reaches zero, if a frame sent by some other station is successfully received or overheard (e.g., RTS or CTS frames in the case of the RTS/CTS access method), the former station will return to stage 1 and double its CW1. When a collision3 happens, the colliding stations will double their CW2 and generate a new bc2 value from the interval (0, CW2). They will remain in the second stage and repeat the above channel contention procedure for stage 2 until someone wins the channel.

**Packet transmission:** In packet transmission stage data and acknowledgment transmission takes place. Data is transmitted from source to destination and in the same way as the part of the confirmation of data transmission acknowledgment will be send from destination to source.

## PERFORMANCE OF THE PROPOSED TECHNIQUE

This protocol currently assumes that all nodes on a single channel sense the same channel conditions, especially since the network is operating as a WLAN. While this assumption may hold for an isolated WLAN, the presence of other WLANs or interference sources could result in different nodes sensing different channel conditions within the same wireless LAN. This impacts the transmissions on both the control and data channels. On the control channel, a node may not reply to an RTS if it senses the channel to be busy, regardless of whether the packet was correctly received. This could again result in the sender's CW exponentially growing unnecessarily. Nodes would also freeze their back-off counters when they sense the channel to be busy. So if the interference source continually affects only a fixed set of nodes, then the rest of the node have a higher probability of winning the contention. This is true assuming that the destination of the unaffected node also does not sense the interference.

On the data channel, differing channel conditions could cause either the sender or receiver to enter the busy-wait loop while the other node either commences transmission or expects an incoming transmission. If the sender enters the busy-wait loop, the receiver could time out and return to the control channel before the sender is able to commence transmission. As such, the data transmission will be unsuccessful and the sender will have to return to the control channel and initiate the retransmission process. If the receiver enters the busy-wait loop, it will still be able to receive the sender's transmission. Depending on how strong the interference is, the packet may or may not have been corrupted. If the packet is not corrupted, then the transmission will be successful as no carrier sensing is done before the transmission of an ACK. If the packet was corrupted, the receiver does not reply with an ACK and returns to the control channel. Consequently, it is still possible for the transmission to be unsuccessful and result in retransmission.

This can also be extended to multi-hop networks. The spatial distribution of the various flows generally results in varying channel conditions. For the case of the WLAN, the interference was a result of an external source situated at close proximity and affected a certain subset of the nodes. In a multi-hop network, the flows themselves act as the sources of interference for other flows. As such, the channel conditions across the network are constantly varying and dependent on the state of each flow. This also gives rise to the hidden-terminal problem which results from the sender's inability to detect the channel conditions at the receiver. Thus, the severity of the performance loss would be worse in multi-hop networks. Thus, differing channel conditions could seriously degrade the throughput achieved by increasing the duration of the average contention-resolution cycle and by increasing the number of unsuccessful transmissions on the data channel.

The general performance characteristics for this protocol are similar to that of the disjoint flows. The throughput peaks at a certain bandwidth division and as the packet size increases, this optimum bandwidth division occurs at smaller control channel bandwidth.

Regardless of the number of flows present in the system, the aggregate throughput the system achieves also tends to converge as the control bandwidth increases. while the general performance characteristics of both scenario are the same, the absolute throughput achieved by this protocol in the AP flow scenario is significantly smaller than that of the disjoint flows and is always worse than IEEE 802.11 DCF. This is due to the presence of the access point, which is the common receiver across all flows in the network. Given that this protocol freezes a node's back-off counter if its destination is known to be on the data channel, the contention-resolution stage is frozen during every data transmission cycle. This effectively sequentializes the contention-resolution stage and the data transmission stage. Consequently, only one channel is in use at any instant. This bandwidth wastage reduces the throughput that the system is able to achieve. Furthermore, since the data channel bandwidth is always less than the total available bandwidth, this protocol will not be able to match the performance of IEEE 802.11 DCF in the AP scenario. The this protocol throughput in this scenario can be improved by using two transceivers at the access point, which will allow pipelining to proceed while it is receiving on the data channel.

Unlike the disjoint flows and the AP flows, random flows are more realistic as they account for the possibility that a node could be both a sender as well as a receiver. This interdependence will affect the aggregate throughput as deafness becomes an important factor. While the performance curves for the random flows follow the same trends as those of the disjoint and AP flows, the throughput achieved by this protocol for random flows lies within the range bounded by the throughput of the disjoint flows and that of the AP flows. As described earlier, the random flows were picked such that the each node in the network is a source of exactly one flow and its destination is randomly selected from the remaining nodes in the network. For a small number of flows, most of the flows become interdependent. Two flows in a network are independent if there are no common nodes between them; otherwise they are interdependent. Consequently, the contention resolution and data transmission stages become heavily sequentialized. The network becomes very similar to that of the AP flows and hence the performance of this protocol is worse than IEEE 802.11 DCF for a small number of flows (Fig. 3).

As the number of flows increases, there are a larger number of flows that are independent of each other. With a greater number of independent flows, the impact of pipelining becomes more obvious. When one flow is on the data channel, all the flows that are dependent on this flow will freeze their back off counters and not participate in the current round of contention. Flows that are independent of the nodes currently on the data channel proceed with the contention-resolution on the control channel. There is a high likelihood that, by the end of the current data transmission cycle, one of the independent flows would have won the contention-resolution on the control channel and would be ready to switch to the data channel to start transmitting. Thus, the presence of independent flows within a network causes the random flow scenario to behave like the disjoint flow scenario, thus resulting in an increase in the system throughput. This is evident from the sharp increase in system throughput from a four-flow scenario to an eight-flow scenario. As the number of flows continues to increase, the throughput gain that results from the presence of independent flows within the network is slightly offset by an increase in the number of collisions. Despite this, the presence of independent flows within a network still contributes to a significant performance gain. The performance of a network with 128 flows is always better than that with two or four flows, regardless of the data size.

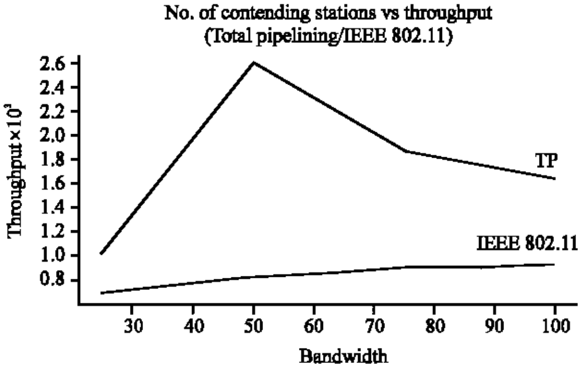

Fig. 3: Performance of partial total technique

The performance of this protocol degrades below that of IEEE 802.11 faster for random flows than it does for disjoint flows. This is indicative of the impact the interdependence and deafness have on the overall system throughput. For a packet size of 750 bytes, the performance of DMCAC for an eight-flow network becomes worse than that of IEEE 802.11 as the control bandwidth becomes larger than 5 Mbps. In contrast, this protocol performed better than IEEE 802.11 up to a control bandwidth of 5.5 Mbps for an eight-flow network in the disjoint flow scenario. This implies that the performance gain is more sensitive to the bandwidth division in the random flow scenario than in the disjoint flow scenario (Fig. 4).

From the above discussion, it can be inferred that, for a random network topology, the benefit of this protocol is

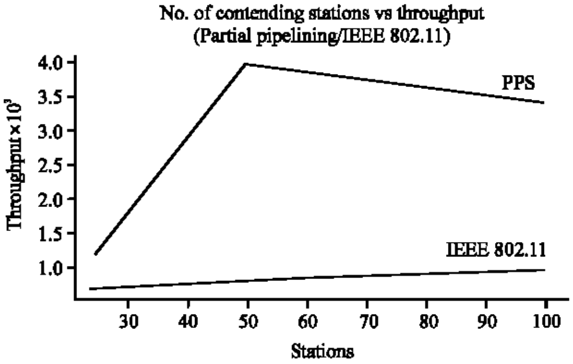

Fig. 4: Performance of partial pipelining technique

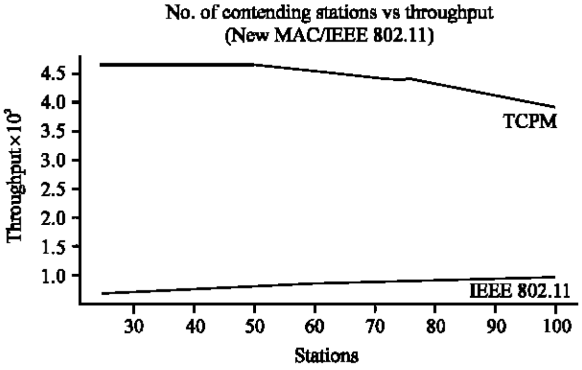

Fig. 5: Performance of proposed pipelining technique

dependent on the number of flows and the packet size distribution as well as the bandwidth allocation between the two channels. Despite the fact that the maximum throughput occurs at a certain bandwidth division, the decrease in the aggregate throughput for a bandwidth division around the optimum is not very significant. This gradual degradation implies that, given the packet and flow distributions, a certain bandwidth division can be chosen such that the system throughput across all number of flows is close to the peak throughput. Hence, the improvement of this protocol over IEEE 802.11 is more sensitive to the number of flows in the network and to the packet size distribution than it is to the bandwidth division (Fig. 5).

## CONCLUSIONS

This research describes and evaluates the MAC protocol with many channels that incorporates and explicit pipelining structure. This technique adapts the IEEE 802.11 DCF protocol by splitting the transmission dialog into a contention-resolution stage and a packet transmission stage. The protocol divides the available bandwidth between two channels, namely a control channel and a data channel. The contention-resolution stage occurs over the control channel and involves the RTS/CTS exchange to win the access rights to transmit the data packet. Upon winning the contention-resolution, the DATA/ACK exchange is done on the data channel. By allowing the contention-resolution for the next data packet to occur in parallel with the current data transmission, a significant portion of the contention overhead is masked and thus results in a throughput gain.

Here simulated the protocol within an isolated WLAN while varying the packet size, bandwidth division and number of flows. As expected, there is an optimum bandwidth division such that the duration of the contention-resolution stage balances the duration of the data transmission stage and the aggregate throughput is max at this bandwidth division. The control bandwidth at which, this optimum bandwidth division occurs increases with the number of flows. Furthermore, as the bandwidth on the data channel decreases, the aggregate throughput that can be achieved tends to converge to a single value, regardless of the number of flows and the packet size. These trends hold across all type of flow topologies. This technique results in a substantial performance improvement over IEEE 802.11 DCF for disjoint flows, but it performs significantly worse for the AP flow scenario.

The random flow scenario is a middle point between these two scenarios and accounts for the common receiver problem as well as the impact of deafness on the overall performance. Form the results, it is evident that the performance is more sensitive to the distribution of packet size and to the number of flows than it is to the bandwidth division. The optimum bandwidth division for all number of flows occurs within a range of 1Mbps of each other. Also, a slight deviation from the optimum bandwidth allocation does not significantly degrade the throughput of the system. Hence, if the packet size distribution is known, a fixed bandwidth division can be selected such that the system throughput across all number of flows is relatively close to the peak value.

The next step would be to attempt to extend the pipelining protocol to the multi-hop scenario and evaluate the actual impact differing channel conditions have on the aggregate throughput. Another possible direction for future work would be to outfit each node with two transceivers. The evaluation in the thesis was done under the assumption that each node only has one transceiver and is only able to send or receive on one channel at any point in time. As such, dependent flows become sequential zed since contention-resolution cannot proceed while the common node is on the data channel. With two transceivers, nodes will be able to send and receive on both channels simultaneously. Consequently, a node will be able to participate in the contention-resolution process on the control channel while it is involved in the current transmission on the data channel.

## REFERENCES

Bianchi, G., L. Fratta and M. Oliveri, 2006. Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 Wireless LANs. Politecnico di Milano, Dip. Elettronica.

Crow, B.P., I. Widjaja, J.G. Kim and P.T. Sakai, 1997. IEEE 802.11 Wireless Local Area Networks. IEEE Commun. Magazine, pp: 116-126.

Xiao, Y., H. Chen and M. Guizani, 2006. Performance evaluation of pipeline paging under paging delay constraint for wireless systems. IEEE Trans. Mobile Comput., 5 (1): 64-76.

Yan, Y. and D.S. Park, 2004. Design of MAC Protocols with Less Collision Fast Resolution for WLAN. IEEE, pp: 571-573.

Yang, X. and N.H. Vaidya, 2002a. DSCR: A more stable wireless MAC protocol. Technical Report, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.

Yang, X. and N.H. Vaidya, 2002b. Pipelined packet scheduling in wireless LANs. Technical Report, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.

Yang, X. and N.H. Vaidya, 2003a. DSCR-A wireless MAC protocol using implicit pipelining. ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc) Poster Session.

Yang, X. and N.H. Vaidya, 2003b. Explicit and implicit pipelining for wireless medium access control (invited paper). Proceeding IEEE Semiann. Vehicular Technol. Conference.

Yang, X., Member and N.H. Vaidya, 2006a. A wireless MAC protocol using implicit pipelining. IEEE Trans. Mobile Comput., 5: 3.

Yang, X., N.H. Vaidya and P. Ravichandran, 2006b. Split-Channel pipelined packet scheduling for wireless networks. IEEE Trans. Mobile Comput., 5 (3): 240-257.