MINIMISING COLLISION TIME AND IDLE TIME IN TOKEN DCF USING SWAPPING PROCESS FOR WIRELESS NETWORK

Shailja uikey and Namrata sahayam
Department of Electronics Communication,
Branch Communication System,
Jabalpur Engg. College – Jabalpur

Abstract—IEEE 802.11 DCF is the MAC protocol currently used in wireless LANs. However, due to idle and collision times, 802.11 DCF performs poorly when it comes to channel utilization, system throughput, and channel access time. To overcome these sources of inefficiency in 802.11 DCF, we propose a distributed and dynamically adaptive MAC protocol for wireless networks, called Token-DCF with swapping process. Main focus of our approach is on reducing idle and collision times by introducing an algorithm of token passing using swapping process. In swapping process, strong data searched by DCF process and its queue length if greater than the threshold level, will focus on the minimum queue length in header list and swap it with the new neighboring node by increasing system throughput and channel access time. We simulate Token-DCF, using swapping process and 802.11 in ns-2 to measure and compare performance of MAC protocols.

I. INTRODUCTION

IEEE 802.11 defines the distributed coordination function (DCF) to share the wireless medium among multiple stations. DCF employs CSMA/CA with a binary exponential backoff algorithm to resolve channel contention. DCF specifies random backoff, which forces a station to defer its access to the channel for a random period of time. This backoff period corresponds to the number of idle slots a station has to wait before its transmission attempt. If multiple stations choose the same backoff, they will attempt to transmit at the same time and collisions will occur. Two types of overhead are associated with random access protocols. One is channel idle time (i.e., backoff time) which is the time when contending stations are waiting to transmit. Another is collision which happens when multiple stations transmit simultaneously. If there are few contending stations, idle time is the dominant overhead. If there are many contending stations, collision probability increases and becomes the main source of low channel utilization.

II. RELATED WORK

We summarize the prior work into:

1) Token-DCF: An opportunistic MAC protocol for wireless networks [1}

1) Distributed MAC protocols to improve the efficiency of 802.11 DCF [2].

ISSN: 2278 – 7798

2) Token passing MAC protocols [3]

3) Scheduling algorithms of wireless networks [4], [5], [6], [7].

The main design goal of Token-DCF is to reduce both idle time and collision time by introducing an implicit token passing algorithm. Token-DCF achieves 2X improvement in system throughput and channel access delay compared to 802.11 DCF for most network configurations [1].

Various MAC protocols have been proposed to improve the efficiency of DCF. Cali et al. modify the backoff algorithm of the IEEE 802.11 MAC protocol and derive a contention window size that maximizes network throughput [2]. The backoff window size is tuned at run-time to increase the overall throughput. In this protocol, for light and medium load conditions, where the window size defined in 802.11 DCF is sufficient for guaranteeing low collision probabilities, the standard backoff algorithm is adopted.

The Wireless Token Ring Protocol (WTRP) [3] is a token bus protocol, derived from IEEE 802.4. WTRP presents a token passing MAC protocol for wireless networks. When token passing is to be used in a WLAN, the characteristics of the wireless medium, such as connectivity loss, network partitioning and token loss, raise additional token management issues.

Longest-Queue-First scheduling (a.k.a., greedy maximal scheduling) [4] is another centralized scheduling algorithm, which has been observed to achieve throughput optimality in most practical wireless networks. LQF makes scheduling decisions based on the queue length information as follows. It starts with an empty schedule and first adds the link with the largest queue length to the schedule. It then looks for the link with the largest queue length among the remaining links. This selected link will be added to the schedule only if this addition creates a feasible schedule (i.e., the set of added links satisfies the SINR constraints). This process continues until no more links can be added to the schedule. Throughput optimal scheduling algorithms are generalized in many different directions [5], [6], [7].

III. TOKEN-DCF DESIGN

In Token-DCF, when a station transmits on the channel, it might give a privilege (i.e., a token) to one of its neighbors. When a transmission ends, the privileged station, starts transmitting after a short period of time, namely SIFS (Short Inter Frame Space). Non-privileged stations follow the backoff procedure of 802.11 to access the channel.

Token-DCF is fully distributed and does not require any centralized point of coordination. In Token-DCF, queue length of a station is included in the MAC header of the transmitted packets and is overheard by the neighboring stations. Each station keeps track of queue length of its neighbors. Queue length information is used in the scheduling component of the protocol, where a neighbor of the transmitting station is selected as the privileged station. No extra control packet is needed for giving a privilege to a station. Instead, the next privileged station (i.e., the scheduled station) is specified in the MAC header of data packets being transmitted on the channel. This probability is adjusted based on the accuracy of the neighbors' traffic estimation. Token-DCF is an opportunistic MAC protocol which behaves similar to 802.11 DCF when packets are not overheard by the neighboring stations. However, when the opportunistic overhearing is feasible, we eliminate the backoff procedure of 802.11 DCF to improve efficiency.

A. Overview

Token-DCF is implemented in the MAC layer of the protocol stack. Scheduling information is embedded in the MAC header of data packets and is transferred to the neighboring stations via overhearing. Each station maintains queue length of the neighboring stations. These queue lengths are then used in the scheduling phase to select the privileged station for the next transmission. Transmitting station announces the privileged station in the privileged field of the MAC header of the data packets it transmits. By overhearing these packets, the privileged station is informed that it has a higher priority for the next transmission. When a transmission ends, the privileged station can start transmitting after SIFS if the channel is sensed idle. If opportunistic overhearing does not work, i.e., token is not received by the next privileged station, Token-DCF operates similar to 802.11 DCF. But when the next privileged station overhears the token, it can transmit on the channel without going to the backoff procedure.

B. Reducing idle time

The scheduling algorithm of Token-DCF determines which neighbor is chosen as the privileged station. When a transmission ends, the privileged station starts transmitting after SIFS, if the channel is sensed idle. Non-privileged stations follow the backoff procedure of IEEE 802.11 to access the wireless medium. Backoff mechanism of 802.11 DCF is shown in Figure 1. In this mechanism, after a transmission ends, the station senses the channel after DIFS interval and if the channel is sensed idle, it waits for a random backoff time.

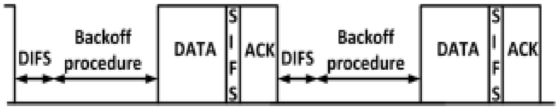


Fig. 1: Access method of IEEE 802.11 DCF

Channel access method of our protocol is shown in Figure 2. In Token-DCF, when the channel becomes idle, the privileged station, if there is any, starts transmitting on the channel immediately, and non-privileged stations have to defer backoff count down till when transmission of the privileged station finishes. This process of giving a privilege to one of the neighbors of the transmitting station repeats in each transmission. Whenever a privileged station transmits on the channel, the idle time of the channel is limited to SIFS. On the other hand, in IEEE 802.11 protocol, the channel idle time between two consecutive transmissions is equal to DIFS plus random backoff duration. Furthermore, since the privileged station immediately transmits after waiting an idle duration of SIFS, while all other stations should wait for at least a longer DIFS, the transmission of the privileged station will not collide with other transmissions.

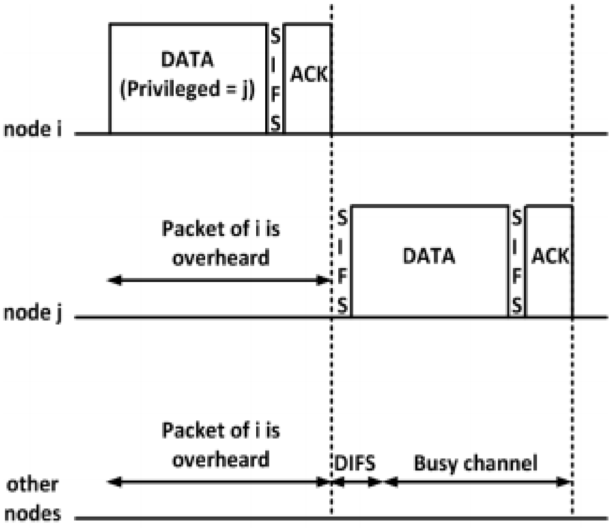


Fig. 2: Access method of Token-DCF protocol

C. Protocol details

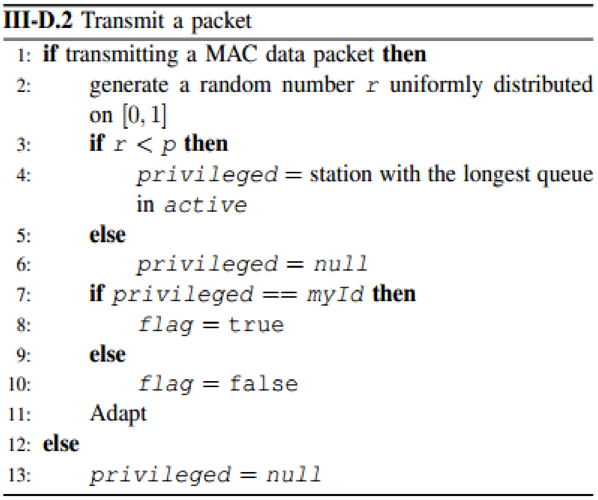
The detailed Token-DCF is presented in this section. Procedure III-D.1 sets the initial values of Token-DCF parameters. *p,* the probability of giving a privilege, is initially set to zero and changes during the execution. *active* denotes the set of neighbors of a station that has transmitted on the channel during the current scheduling period and the transmission is overheard by the station. The station itself, *myId*, is also included in the set *active.* When a station transmits, it might give a privilege to one of the stations in the set *active.* By including *myId* in the set *active*, a station might choose itself as the *privileged*. Each station keeps track of the transmissions on the channel by overhearing of the packets.

III-D.1 Initialization at station $myId$

```
$p = 0$
$active = \{myId\}$
$nFail = 0$
$nSuccess = 0$
$flag = false$
call Initialization after $period$
```

Procedure III-D.2 is executed right before a packet is transmitted on the channel. If the packet is a MAC data packet, the station might give a privilege to one of its neighbors.

In III-D.2, *privileged* is the station in the set active with the longest queue. Another example of scheduling algorithms is the one in which *privileged* is chosen uniformly at random from the set of stations in *active* with non-zero queue length. This policy achieves fairness among the network stations. Many other queue based scheduling algorithms are presented in the literature that can be incorporated in Token-DCF protocol [5], [6].



III-D.2 Transmit a packet

```
if transmitting a MAC data packet then
    generate a random number $r$ uniformly distributed on $[0, 1]$
    if $r < p$ then
        $privileged$ = station with the longest queue in $active$
    else
        $privileged = null$
    if $privileged == myId$ then
        $flag = true$
    else
        $flag = false$
    Adapt
else
    $privileged = null$
```

D. Swapping process

Next neighboring node is searched [strong data] by DCF process. If its queue length is greater than the threshold level, collision will occur and the data in the node will be lost. We will focus on the minimum queue length in header list and swap it with the new neighboring node and add their information like node id and queue length in the header list. Then again privileged process will start, and continuity of the process will remain same.

IV. EVALUATION

In the example considered, the threshold level of each node is assumed to be 200 bytes, i.e., each node can contain 200 bytes of data. Suppose we run DCF process for 60 nodes. Then Source node searches for the node with the largest queue length, i.e., the node with largest data, in its header list. Then checks the total receives size of data and calculates the throughput.

[THROUGHPUT= NUMBER OF PACKETS RECEIVED / NUMBER OF PACKETS GENERATED]

Once found, the node with largest queue length is granted privilege. Now privilege process is run for this node. The data contained inside the node is distributed among each selected node. The amount of data to be distributed is determined by taking the average of the bytes of data inside the privileged node. The average is calculated by dividing the total bytes of data, inside that node, by total number of selected node.

[Average = Total bytes of data/ Total no. of selected node]

Then again we check the total receives size of data and calculate the throughput. The performance is better than the simple DCF process. So, privileged process is improving the efficiency of simple DCF process.

The selected node, among which the data is distributed, is the nodes which have minimum bytes of data. Reason for doing this is that if data is add to nodes which already have enough data, it may exceed the threshold level, and thus collision of node may occur. To prevent this, we select a predefined number of nodes, in our example nodes are selected by sorting out of 60 nodes on the basis of byte of data they contain, the first 30 nodes, having least data, are taken.

The average data calculated to distribute, from the privileged node is distributed

ISSN: 2278 – 7798

among these selected nodes. Due to this the probability of the queue length of nodes exceeding their threshold level is minimum byte. The number of times the privilege process is run is limited. Here is limited to 5 times.

Every time the average bytes of data are calculated, it is checked that any selected nodes exceeds its threshold level if the data is added to it. If it does exceed, the process is stopped and addition of bytes is avoided.

After running the privilege process for limited number of time, the result is checked and throughput is calculated.

In special cases if a new neighboring node is encountered which has queue length greater than threshold level, then swapping process is used. In this process the bytes of data is the new neighboring node is checked. If the bytes are above the threshold level then the data equal to the threshold level is added to the new neighboring node and the remaining part of bytes of data is swapped with the data of the node with minimum queue length. After running the swapping process for the limited number of time, the result is checked and throughput is calculated. After completing the swapping process, then again privilege process is repeated.

Figures 3 plot the performance parameters in a single-hop network. The size of the network is 150mx150m and all flows are single-hop.

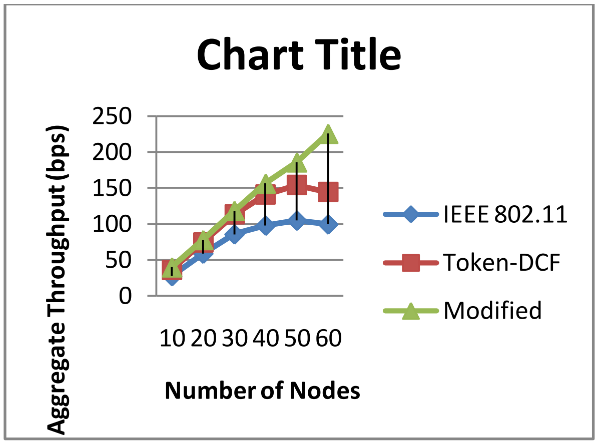


Fig. 3: Aggregate throughput (area=150mx150m)

The aggregate throughput of 802.11 DCF and Token-DCF modified token DCF is presented in Figure 3. As can be seen, throughput gain obtained by modified Token-DCF using swapping process compared to Token DCF and IEEE 802.11.

Figure 4 shows the packets loss of the three protocols. Packets loss is defined as the number of packets loss in modified Token DCF (Swapping process) as compared to Token DCF and IEEE 802.11 Mac protocol.

ISSN: 2278 – 7798

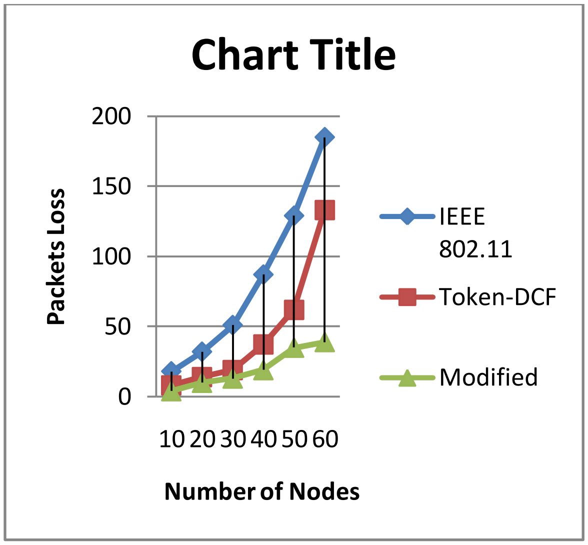


Fig. 4: packets loss (area=150mx150m)

Figure 5 shows the average access delay of the three protocols. Access delay is defined as the delay between the time a packet arrives at the MAC layer and the time the source of the packet receives acknowledgment from the destination. Access delay of a packet consists of the waiting time before transmitting on the channel and the time spent in packet retransmissions. In Figure 5, access delay is smaller in modified Token-DCF. Furthermore, many retransmissions are avoided because of reduced collision frequency.

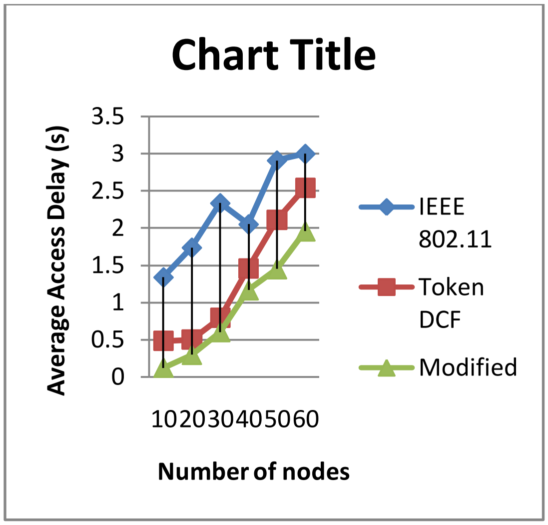


Fig. 5: Average access delay (area=150mx15O)

V. CONCLUSION

In this paper, we presented the design and performance evaluation of Token-DCF using swapping process. Token-DCF is a distributed MAC protocol that uses an opportunistic overhearing mechanism to schedule network stations for transmission on the channel. The main design goal of Token-DCF using swapping process is to reduce both idle time and collision time by introducing an algorithm of token passing using swapping process. Thus we concluded that swapping process improves efficiency, reduces collision and packet loss and increases throughput.

ISSN: 2278 – 7798

VI. REFERENCES

[1] Token DCF: An opportunistic MAC protocol for wireless networks. 2013 IEEE.

[2] F. Cali, M. Conti, and E. Gregori, “Dynamic tuning of the ieee 802.11 protocol to achieve a theoretical throughput limit,” IEEE/ACM Trans. On Networking, vol. 8, no. 6, December 2000.

[3] M. Ergen, D. Lee, A. Puri, P. Varaiya, R. Attias, R. Sengupta, S. Tripakis, “Wireless token ring protocol,” Proceedings of the Eighth IEEE Symposium on Computers and Communications (ISCC 2003) pp. 710-715, 2003.

[4] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” IEEE Transactions on Automatic Control, 37(12): pp. 1936-1948, 1992.

[5] A. Dimakis and J. Walrand. “Sufficient conditions for stability of longest-queue-first scheduling: Second-order properties using fluid lim-its.” Advances in Applied Probability, 38(2):505521, 2006.

[6] S. Shakkottai, R. Srikant, and A. Stolyar. “Pathwise optimality of the exponential scheduling rule for wireless channels,” Advances in Applied Probability, 36:10211045, 2004.

[7] A. Eryilmaz, R. Srikant, and J. Perkins. “Stable scheduling policies for fading wireless channels,” In Proceedings of IEEE International Symposium on Information Theory, 2003