

# T-MAH: A Token Passing MAC protocol for Ad Hoc Networks<sup>1</sup>

Daniela Maniezzo <sup>‡§†</sup>, Giovanni Pau <sup>‡</sup>, Mario Gerla<sup>‡</sup>, Gianluca Mazzini <sup>†</sup>, Kung Yao<sup>§</sup>

<sup>‡</sup> UCLA Computer Science Department

<sup>§</sup> UCLA Electric Engineering Department

<sup>†</sup> Dipartimento di Elettronica - Università di Ferrara - Italy

{dmaniezzo|g.mazzini@ieee.org; gpau|gerla@cs.ucla.edu; yao@ee.ucla.edu }

**Abstract**—The Token Passing MAC protocol for Ad Hoc networks (T-MAH), discussed in this paper, is a distributed medium access protocol designed for wireless multi-hop networks. With T-MAH access scheme the network is organized in clusters (called Token Groups) with a Token Group Head as the leader of the group. In each single cluster is used a token based technique, i.e. each node in the cluster is allowed to transmit the user data only after has received a particular packet (called token) and the time slot has not been expired. The combination of a controlled MAC protocol and a hierarchical network division, reduces the number of the error due to packet collision and permits a good channel reutilization.

## I. INTRODUCTION

Wireless Ad Hoc networks have been largely deploying both in civil and military applications thanks to the availability of cheap wireless NIC's based on IEEE 802.11 [1]. The multi-hop Ad Hoc networks, based on IEEE 802.11, have poor performance with the TCP/IP protocol stack [2] due to hidden terminal problem. In this work, we propose a new MAC protocol that manages the media contention using a token based technique.

The proposed protocol organizes the network in clusters (Token Groups) and introduces the concept of Token Group Head as the leader of the group. The network flexibility and reliability are assured using a set of mechanisms in order to reduce the needs of signaling interactions between the Token Group Heads and the regular nodes. Furthermore the proposed schema can naturally support the QoS management using a pool of queues with different priorities for each Token Group and a specific field in the packet header.

The presented protocol is inspired to the IEEE 902.4 Token Bus protocol [3] and to the Wireless Token Bus Protocol (WTBP) for Ad Hoc network described in [4]. The main differences between our approach and the WTBP are in the signaling and in the network architecture.

Our protocol creates a set of clusters and manages the packet transmission in each cluster using a Token Passing protocol. We defined gateways between clusters and all signaling needed to manage the network at the MAC layer.

The WTBP, instead, proposes a protocol capable to create e manage a single logical ring, but pushes the management of whole network architecture out of the MAC protocol.

Moreover, in multi-hop networks, some nodes in the transmitter and/or receiver radio range might not be able to hear a successful Request To Send/Clear To Send message exchange, used by IEEE802.11 to reserve the shared channel.

<sup>1</sup>This work has been partially funded by the UC Core program Core01-10091 under the sponsorship of ST Microelectronics. Coresponding Author: Daniela Maniezzo UCLA Department of Computer Science, BH 3731, 420 Westwood Plaza, Los Angeles, CA 90024 - USA; dmaniezzo@ieee.org

Hence, we assume that each network node is equipped to transmit an out-of-band signal, namely Reception Busy Tone ( $BT_r$ ), when acting as receiver. The  $BT_r$  stops the receiver's neighbors from transmitting; this helps the collision avoidance at the receiver. This assumption can eventually be relaxed using an appropriate signaling schema (i.e. RTS/CTS mechanism).

Other examples of MAC protocol based on control packets sent out-of-band are: DBTMA (Dual Busy Tone Multiple Access) [5] [6], RI-BTMA (Receiver Initiated Busy Tone Multiple Access) [7] and PAMAS (Power Aware Multi-Access Protocol with Signalling) [8].

The paper is organized as follow: in section II we describe the system and the general constrains. The network operating modes, Token Group Election and the other management functionalities are depicted in sections II-A, II-B. The paper is concluded by a preliminary performance analysis in section IV ad few final considerations are given in section V.

## II. SYSTEM DESCRIPTION

The T-MAH design needs few general assumptions about the network and the nodes.

In particular we assume that the network is composed by nodes with almost the same networking capabilities, connected using radio interfaces. Moreover, as introduced previously, each node is a equipped to transmit an out-of-band signal, namely Reception Busy Tone ( $BT_r$ ) that is turned on when a node starts receiving from the radio interface. In the collision event, two different transmitters are detected, the receiver node turns off the Busy Tone  $BT_r$  and the transmission is suddenly stopped.

A T-MAH network is partitioned in clusters called Token Groups ( $TGs$ ). Each TG is identified using an unique Token Group Identifier ( $TG_{ID}$ ). The network structure allows the TGs to be overlapped by one or more nodes; each node, indeed, can be part of different TGs at the same time and a queuing pool at the MAC layer is maintained for each TG.

Any given node within a TGs knows the address of the previous and next nodes in the group logical infrastructure. The logical Token Group organization is a virtual ring where each node is allowed to transmit the user data only after the Token has been received and the Token Time Slot ( $T_{slot}$ ) has not been expired. When the  $T_{slot}$  expires, the node must forward the Token to the next node in virtual ring.

In each TG a timeout counter, namely Token Cycle ( $T_{cycle}$ ) is defined and each node within the same group knows its value that is defined as the maximum time needed by the token to complete the virtual ring once:  $T_{cycle}^n = N * T_{slot}^n$  where N is the number of nodes in the  $n^{th}$  TG. Any given node maintains a different  $T_{cycle}$  for each TG it's part of.

The transmission of all broadcast signaling packets is done using CSMA/CD with Busy Tone  $BT_r$ .

The T-MAH network operates through different phases:

- *Normal Operating*: In this phase each node just transmits user data in a token passing fashion (see II-A);
- *Token Group Head Election*: In this phase the Token Groups ( $TGs$ ), are defined and, for each  $TG$ , a Token Group Head ( $TGH$ ) is elected and a Transmission Frame is defined (see II-B);
- *Node Death Detection and Recovery*: In this phase the  $TG$  adapts itself and the Transmission Frame in the event of the death of one  $TG$ 's node member (see II-C);
- *New Node Entrance*: In this phase the  $TG$  adapts itself and the Transmission Frame to the join of a new node (see II-D).

In the following details for all phases will be given.

#### A. Normal Operating phase

In this phase the network is already partitioned in Token Groups ( $TGs$ ), and each  $TG$  has its Token Group Head ( $TGH$ ). As introduced early, each node in a  $TG$ :

- waits for the Token;
- once the Token is received and within the node Time Slot:
  - transmits the user data packet(s) if any;
  - forwards the Token to the next node in the transmission frame;

If for \*any\* reason, (i.e.  $BT_r$  from neighbors), the node is unable to complete the transmission in its Time Slot, it misses the transmission chance till the next Token Cycle. The transmission right goes to the next node in the Transmission Frame.

#### B. Token Group Head ( $TGH$ ) Election Procedure

A Token Group ( $TG$ ) contains all nodes directly connected to the Token Group Head ( $TGH$ ). The  $TGH$ 's can be or not the nodes with the largest number of directly connected neighbors.

Let consider a network with  $N$  nodes, the generic node  $i$ :

- waits (listening) a free channel for an exponential time.
- transmits a broadcast packet with TTL=1 ( $HELLOpkt$ ). The  $HELLOpkt$  contains:
  - the source node MAC address ( $MAC_i$ );
  - all known neighbor MAC addresses.

Each node that catches the  $HELLOpkt$  turns on the Reception Busy Tone and updates the  $1$ -hop connection table adding  $MAC_i$  as one-hop neighbor and as next hop to reach the  $i$ 's neighbors if they are not present in the table.

The procedure is repeated until the channel is detected free for a time interval  $T_{max}$  and then, if the MAC address of the node  $i$  is the greatest within all neighbors directly connected, the node  $i$ :

- elects itself  $TGH$  ( $TGH$  Election Procedure MODE 1);
- defines the Token Group Identifier ( $TG_{ID}$ );
- defines the transmission order between the  $TG$  nodes (from the lower to the Greater MAC address in MODE A or MODE B, see section II-B.1)

If  $i$  detects no  $BT_r$ , it transmits a broadcast packet, called  $TGHpkt$  containing the following information:

- $TG_{ID}$ ;

- $MAC_i$  i.e.  $TGH$  MAC address;
- transmission order between the nodes within the same  $TG$ .

Otherwise, if the  $i$  MAC address is not the greatest (it's part of at least another group), and  $i$  has directly connected neighbor(s) NOT in  $i$   $TG$ (s), and the named neighbor(s) has a Lower MAC address (i.e. the neighbor(s) waits  $i$  election as  $TGH$ ), than  $i$  transmits a broadcast packet called  $NoTGHpkt$ . This advices other neighbors that the node  $i$  must not be considered in the  $TGH$  election.

Among the remaining  $i$  neighbor nodes the  $TGH$  election procedure MODE 2 is started. This procedure is the same to MODE 1 but the  $MAC_i$  is excluded to the election (because it has sent the  $NoTGHpkt$ ).

The difference between MODE 1 and MODE 2 is that in the first one the  $TGH$ 's MAC address is the greatest MAC address of the nodes in the same  $TG$ ; in the second one, the  $TGH$  has a MAC address smaller than one of the nodes in its  $TGH$ .

In the  $TGH$  election procedure two other basics parameters of the T-MAH network are defined: the Transmission Frame and the Time Slot.

1) *Transmission Frame Definition in MODE A and MODE B*: As introduced previously, when the  $HELLOpkt$  exchange is finished, the  $TGH$ :

- builds the transmission frame (between the  $TG$  nodes), listing the directly connected neighbors by increasing the MAC address.
- Using the information from  $1$ -hop connection table (built in with the HELLO procedure), the  $TGH$  verifies the full connectivity of the transmission order.

If there are all direct links needed to complete the transmission order, the transmission order is defined as MODE A.

If there are direct links missing, the  $TGH$  puts itself between two subsequent nodes not directly connected (transmission order in MODE B). Note: the  $TGH$  is directly connected with all nodes in the  $TG$ .

The Token Group Head in MODE B has at least two timeout counters for its group since it is present twice in the transmission frame.

2) *Transmission Frame Time Slot Definition*: Each  $T_{cycle}$  is spited in slots. The slot length is  $T_{slot} = T_{cycle}/n$  where  $n$  is the # of established transmission frame ( $n = N$  in MODE A,  $n > N$  in MODE B).

Each node has right to talk for 1 slot (of  $T_{slot}$  size) in each  $T_{cycle}$ . The  $TGH$  is allowed to talk more than  $1Slot/T_{cycle}$  if its a MODE B frame.

The node currently owning the token ( $Token - Owner$ ) must forward the token to the next node in the transmission frame before its slot time ( $T_{slot}$ ) elapses.

#### C. Node Death Detection and Recovery

The Token Group Header of group  $x$ ,  $TGH_x$ , keeps a counter for each node part of the  $TG_x$ . The node  $i$  counter is increased by one any time the  $TGH_x$  does not detect transmission activity by node  $i$  in its time slot. The  $TGH_x$  resets the counter anytime it detects a node  $i$  transmission in its slot.

If the node  $i$  counter reaches a pre-defined maximum value  $N_{max}$ , the  $TGH_x$  declares that the node  $i$  no longer present in the  $TG_x$

and it eliminates  $i$  from the  $1$ -hop connection table.

The  $TGH_x$ :

- transmits (during its Time Slot) a broadcast packet  $STOPpkt_x$ ; this stops the normal operation of the  $TG_x$ ;
- rebuilds the Transmission Frame using the updated  $1$ -hop connection table;
- transmits a broadcast packet with the new Transmission Frame to all members of the  $TG(x)$ .

The transmission restarts with the new frame.

The node  $i$ , next to the  $TGH_x$  in the Transmission Frame, counts how many consecutive times its timeout  $TO_x(i)$  expires. If  $TO_x(i)$  expires more than a  $N_{max}$  time, the node  $i$  starts a new election procedure sending an  $HELLOpkt$  that triggers the election procedure.

If timeout  $TO_x(i)$  expires consecutively for  $N_{max}$  times, node  $i$  states itself as no longer part of the group  $x$ ; then,  $i$  stops any transmission in the  $TG_x$  and waits the election/new node procedure for the new group; moreover, the timeout  $TO_x(i)$  is set to  $KT_{cycle}$  (see II-D). If  $TO_x(i)$  expires, the node  $i$  elects itself  $TGH$  since it has not received the invitation to enter in a new group.

#### D. New Node Entrance

Every  $K$  transmission cycle, a time slot of  $T_{n-in}$  is reserved for the procedure that allows new nodes to enter in the  $TG_x$ :

- the  $TGH_x$  transmits an  $HELLOpkt$  to invite the new nodes (if present) to join the  $TG_x$ ;
- a new node  $p$  receives the  $HELLOpkt$  and answers, using CSMA/CD, with an  $HELLOpkt$  that contains its MAC address and the  $TGH_n$  MAC address;
- the  $TGH_x$  receives the reply of the new node  $p$  and transmits the  $STOPpkt$ , stopping the current transmission and triggering an  $HELLO$  procedure to redefine the Transmission Frame; no  $TGH$  election is performed;
- the  $TGH_x$ , rebuilds the Transmission Frame taking into account the new node;
- the  $TGH_x$ , sends the new transmission frame and the network restarts the regular operations.

If the new node  $p$  is out of the  $TGH_x$  transmission range, it waits  $KT$  and transmits an  $HELLOpkt$  (using CSMA/CD+ $BT_r$ ). The  $HELLO$  procedure starts within the nodes in the node  $p$  transmission range.

A given node replies only to an  $HELLOpkt$  by an unknown node. Each node part of a given  $TG$  is enabled to transmit an  $HELLOpkt$  only if its Token-Owner.

### III. EXAMPLES

#### A. TGH Election Procedure: MODE A

Let consider the network depicted in figure 1, and the  $HELLO$  packets transmission sequence shown in figure 2.

Node 6 waits for a free channel for an exponential time, then starts transmitting an  $HELLOpkt$  containing its  $MAC$  address. The  $HELLOpkt$  is received by all one hop neighbors (nodes 3,4,5). Nodes [3, 4, 5]:

- set the Busy Tone  $BT_r$  ON for the whole receiving phase (note: this will deny nodes 1 or 2 to transmit);
- create an entry in the  $1$ -hop connection table with the information of a direct link with the node 6.

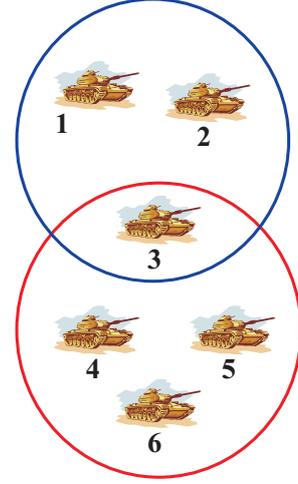


Fig. 1. Network Layout - MODE A

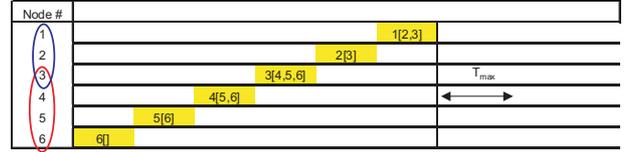


Fig. 2. Election Procedure: Message Exchange originate a MODE A Election

As shown in figure 2, the  $HELLOpkt$  of node 5 follows the node 6 (note: the node 5  $HELLOpkt$  contains its  $MAC$  and the information about the direct link 5-6).

When the node 5  $HELLOpkt$  is received by nodes 3,4,6, the nodes[3, 4, 6]:

- turn ON the Reception Busy Tone  $BT_r$  for the whole receiving phase;
- create an entry in the  $1$ -hop connection table with the information of a direct link with the node 5, moreover the information about the connectivity to node 6 through node 5 is recorded (if not already present).

The previous procedure ( $HELLO$  procedure) is repeated for all network nodes.

At the end of the  $HELLO$  procedure, the node with the higher  $MAC$  address value (we are supposing an unique identifier) within the nodes directly connected (in the considered network the node 6):

- builds the Transmission Frame of the Token Group (uniquely identified by the a  $TG_{ID}$ ) listing all the neighbors in the future transmission order (i.e. 3 – 4 – 5 – 6);
- verify the connectivity between each couple of adjacent nodes in the transmission frame using the  $1$ -hop connection table; in the examples the links between 3 – 4, 4 – 5, 5 – 6, 6 – 3 are present: a MODE A transmission frame has been successfully generated;
- waits a free channel for an exponential time  $T_{max}$  and transmits the  $TGHpkt$  packet (note: in the considered network the time  $T_{max}$  starts after the node 1 ends its  $HELLOpkt$  trans-

mission since node 6 is in the range of the node 3  $BT_r$ ).// The  $TGHpkt$  contains the following information:

- $TG_{ID6}$ : the  $TG$  Identifier (usually the MAC address of the  $TGH$ , but here, for the sake of simplicity, we consider the label  $TG_{ID6}$ );
- $MAC_6$ : the  $TGH$  MAC address;
- $[3 - 4 - 5 - 6]$ : the transmission frame.

All  $TGH$  neighbors will receive the  $TGHpkt$  and record the information memorized in it.

The other node in the networks act as follow:

- node 3 detects that nodes 1 and 2 are its neighbors but they are not part of the  $TG$  led by node 6; then it transmits a  $NoTGHpkt$  to point out that it can't be the  $TGH$ ;
- the  $NoTGHpkt$  is received by node 2 that:
  - elects itself as  $TGH$ ;
  - builds the Transmission Frame for the  $TG$  that contains the nodes 1 - 2 - 3 (MODE A);
  - checks the transmission frame validity using the  $1$ -hop connection table;
  - transmits the  $TGHpkt$  to the neighbors.

Note: that node 3 is part of 2 different  $TG$ .

### B. $TGH$ Election Procedure: MODE B

Let now consider the network depicted in figure 3. The  $HELLO$  packet transmission sequence is shown in figure 4.

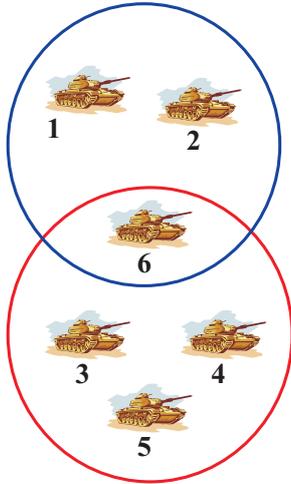


Fig. 3. Network Layout

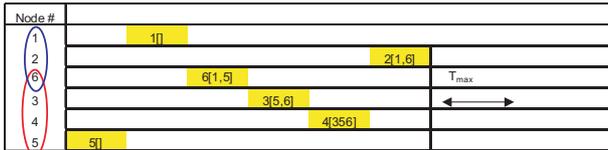


Fig. 4. Election Procedure: Message Exchange originate a MODE 2

After detecting the free channel for an exponential time, node 5 transmits in broadcast the  $HELLOpkt$  (containing the its MAC

address). The  $HELLOpkt$  is received by all neighbors in its radio range (nodes: 3,4,6). These nodes:

- turn on the  $BT_r$  (Busy Tone for the whole receiving process);
- add an entry in the  $1$ -hop connection table (adding node 5).

During the node 5 transmission, the node 1 detects node 6's Busy Tone. When the  $BT_r(6)$  is turned off, node 1 listens a free channel for an exponential time and transmits its  $HELLOpkt$ . The  $HELLOpkt$  of node 1 is received by nodes 2 and 6 and the  $1$ -hop connection table is updated according to.

The process is repeated for all network nodes following the time frame shown in figure 4.// In table I are shown the  $1$ -hop connection table of node 6 at the end of the  $HELLO$  procedure.

After  $HELLO$  procedure, the node with the highest MAC address within the nodes directly connected (in this case node 6):

- elects itself as  $TGH$ ;
- builds the transmission frame of the Token Group: 1 - 2 - 3 - 4 - 5 - 6 listing the nodes from the Lower to the Higher MAC address;
- verifies (using the  $1$ -hop connection table) if all adjacent nodes in the transmission frame are directly connected. In the considered network the following links are present: 1 - 2, 3 - 4, 4 - 5, 5 - 6 and 6 - 1 but the link 2 - 3 is not present;
- builds a Transmission Frame using the MODE B:  $[1 - 2 - 6 - 3 - 4 - 5 - 6]$ ;
- listens for a free channel for an exponential time  $T_{max}$  after node 2;
- transmits a  $TGHpkt$  containing:
  - $TG_{ID6}$ : the token Group Identifier (usually the  $TGH$  MAC address but for the sake of the simplicity in the considered network we use the label  $TG_{ID6}$ );
  - $MAC_6$ : the  $TGH$  MAC address;
  - $[1 - 2 - 6 - 3 - 4 - 5 - 6]$ : the Transmission Frame.

The  $TGHpkt(6)$  is received by all node 6 neighbors.// Note that in at the end of the  $HELLO$  procedure there is only one Token Group.

TABLE I  
MODE B: 1 HOP CONNECTION MATRIX

S/D	1	2	3	4	5	6
1	-	2	6	6	6	6
2	1	-	6	6	6	6
3	6	6	-	4	5	6
4	6	6	3	-	5	6
5	6	6	3	4	-	6
6	1	2	3	4	5	-

## IV. PERFORMANCE EVALUATION

In this section we present some preliminary performance evaluation of T-MAH protocol for the single cluster case.

### A. Token Cycle Time

We defined the Token Cycle time  $T_{cycle}$  as the time needed to allow all nodes in a given cluster to transmit once.

We can split the  $T_{cycle}$  in  $T_{ctrl}^i$ , defined as the amount of time needed by the node  $i$  to transfer the control to the next node in the

virtual ring, and  $T_{data}^i$ , defined as the amount of time needed by the node  $i$  to transmit the user data.// Note that  $T_{ctrl}^i$  can be considered as the sum of different factors such as the token transmission time, the signal propagation time, and the time needed to synchronize the stations.

Let consider the following system of equations:  $E[T_{ctrl}^i] = E[T_{ctrl}]$  for each node  $i$ , where  $E[.]$  is the statistical average. We assume a inter-arrival packet time distribution as Poisson. Moreover given a single node  $i$ , we define  $\lambda$  as the packet arrival rate in that node. The average number of packets arrived at  $i$  for a given Time Cycle is computed as  $\lambda E[T_{cycle}]$ .

In a T-MAH network composed of a single Token Group, only one node can access the channel per time, so, considering only the uniform traffic case, it will be able to get an average bandwidth of  $C/N$  where  $C$  is the channel capacity in bit/s and  $N$  is the number of the cluster's nodes.

The throughput  $G$  is given by:

$$G = \lambda E[P_{size}] \frac{N}{C} \quad (1)$$

where  $E[P_{size}]$  is defined as the average packet size.

The average service time is given by:

$$E[T_{data}^i] = \frac{\lambda E[T_{cycle}] E[P_{size}]}{C} = \frac{G}{N} E[T_{cycle}] \quad (2)$$

The average Token Cycle time,  $T_{cycle}$ , can be computed as follows:

$$\begin{aligned} E[T_{cycle}] &= \sum_{i=1}^N E[T_{ctrl}^i] + \sum_{i=1}^N E[T_{data}^i] = \\ &= N E[T_{ctrl}] + G E[T_{cycle}] \end{aligned}$$

and, solving the previous equation, it is equal to:

$$E[T_{cycle}] = \frac{N E[T_{ctrl}]}{1 - G} \quad (3)$$

### B. Queueing Time

The Queueing Time main componets are the following:

- $T_{Q1}$ : time that elapses from the packet arrival time at the given node to the token arrival time at the same node;
- $T_{Q2}$ : time elapsed from the token reception at the given node to the packet transmission.

In order to evaluate  $T_{Q1}$ , we consider a node that does not holds the token. From the eq. (2), we know that any node is out of service for  $E[T_{cycle}] - \frac{G}{N} E[T_{cycle}]$  time, on the average. Follows that, considering the case of an uniform arrival distribution, the  $T_{Q1}$  can be computed as follows:

$$E[T_{Q1}] = \frac{1}{2} \left(1 - \frac{G}{N}\right) E[T_{cycle}] = \frac{(N - G) E[T_{ctrl}]}{2(1 - G)}$$

In order to compute  $T_{Q2}$  we can consider 0 the time needed to transfer the control ( $E[T_{ctrl}] = 0$ ) since considered in  $E[T_{Q1}]$ . Under this condition, we can consider the whole cluster as M/G/1 system with the queue distributed within the N cluster's nodes; so:

$$E[T_{Q2}] = \frac{E[P_{size}^2]}{2(1 - G) E[P_{size}]} \frac{G}{C} \quad (4)$$

We can sum together the (4) and (5) obtaining the queuing time as follow:

$$E[T_Q] = \frac{(N - G) E[T_{ctrl}]}{2(1 - G)} + \frac{E[P_{size}]^2}{2(1 - G) E[P_{size}]} \frac{G}{C} \quad (5)$$

### C. Access Time

The acces time is the sum of the Queuing time, the packet transmission time ( $T_{PKT}$ ) and the propagation time ( $\tau_P$ ):

$$T_{acc} = E[T_Q] + E[T_{PKT}] + \tau_P \quad (6)$$

Considering, for the sake of the simplicity, the case of fixed packet size the access time is computed as follow:

$$T_{acc} = \frac{(N - G) E[T_{ctrl}]}{2(1 - G)} + \frac{G}{2(1 - G)C} + E[T_{PKT}] + \tau_P \quad (7)$$

## V. CONCLUSION AND FUTURE WORK

In this paper a novel medium access scheme for wireless ad hoc networks, T-MAH, is presented. The proposed schema uses a token passing mechanism to realize a wireless token bus infrastructure. T-MAH, moreover, realizes the network clustering using a clustering mechanism built in schema. We started the study of the T-MAH performances via an analytical study a simple case showing some preliminary analytical results.

We are now working on the simulation model of T-MAH in order start a comparative analysis with the main Wireless mac protocols and to investigate issues the T-Mah scalability.

## REFERENCES

- [1] IEEE 802.11 Working Group, Part 11: "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", ANSI/IEEE Std. 802.11, Sept. 1999.
- [2] S. Xu, T. Saadawi, "Revealing TCP Incompatibility Problem in 802.11-based Wireless Multi-hop Networks", Proc. of IEEE GlobeCom'01, Nov. 2001.
- [3] ISO International Standard Organization, ISO/IEC8802-4:1990 - ANSI/IEEE Std. 802.4-1990.
- [4] D. Lee, R. Attias, A. Puri, R. Sengupta, S. Tripakis, P. Varaiya, "A Token-Ring Medium-Access-Control with Quality of Service Guarantees for Wireless Ad-Hoc Networks", in Proceedings of ACM Mobicom '01, also Univeristy Of California Berkeley Technical Report.
- [5] Z. J. Haas, J. Deng, "Dual Busy Tone Multiple Access (DBTMA)- Performance Evaluation", in Proc. of IEEE VTC'99, Houston, TX, May 17-21, 1999.
- [6] J. Deng, Z. J. Haas, "Dual Busy Tone Multiple Access (DBTMA): A New Medium access Control for Packet Radio Networks Performance", in Proc. of IEEE ICUPC'98, Italy 1998.
- [7] C. S. Wu, V. O. K. Li, "Receiver Initiated Busy Tone Multiple Access in Packet Radio Networks", in Proc. of ACM SIGCOMM'88.
- [8] S. Singh, C. S. Raghavendra, "PAMAS-Power Aware Multi-Access Protocol with Signalling for Ad Hoc Networks", Computer Communication Review, July 1998.