Ad Hoc Networks: A Protocol for Supporting QoS Applications

Lorenzo Donatiello[1] and Marco Furini[2]*

[1]Computer Science Department
University of Bologna
Mura Anteo Zamboni 7
40127 Bologna Italy
email: donat@cs.unibo.it

[2]Computer Science Department
University of Piemonte Orientale
Spalto Marengo 33
15100 Alessandria Italy
email: marco.furini@mfn.unipmn.it

Technical Report TR-INF-2003-01-01-UNIPMN

Abstract

A delay-bounded service in wireless ad hoc networks is challenging, as ad hoc networks do not provide any type of guarantees. Several protocols have been proposed to support applications without timing requirements in ad hoc networks, but the increasing demand of QoS applications, in ad hoc wireless environments, requires delay-bound service. The contribution of this paper is to propose a protocol that provides QoS service, by means of timing guarantees, to the supported applications in ad hoc wireless networks.

1 Introduction

A wireless ad hoc network is a mobile and multi-hop wireless network without any fixed infrastructure. It provides a flexible and low cost network solution in times of emergency or where infrastructures are not trusted or not present. For these reasons, these networks are becoming more and more used in several scenarios: from university campus to airport lounge, from conference site to coffee store.

The topology and channel conditions of these networks change with time, as users are free to move in the environment. Due to this mobility, not all nodes can directly communicate with each other, and hence nodes can be used as bridges in order to deliver data across the network. For these reasons, the design of an effective and efficient communication subsystem is challenging, as infrastructure information are not known.

Several mechanisms have been designed to handle generic communications in ad hoc networks, and two proposals, IEEE 802.11 [1] and Hyperlan [2], have been considered as wireless LAN standard for the OSI reference model. These mechanisms focused on finding a way to avoid transmission collisions and did

---

*Corresponding author

not propose any solutions to provide QoS to the supported applications [3]. In fact, in several mechanisms [1, 4, 5, 6], data can be transmitted only after the successfully end of the handshake protocol (used to avoid collisions), but the handshake protocol does not provide timing guarantees, as it suffers of collisions. An QoS enhancement of IEEE 802.11 is proposed in [3], but packet collision may occur frequently by increasing the number of mobile stations.

Due to the increasing use of applications with QoS requirements, timing guarantees can be considered as a fundamental feature for future ad hoc wireless networks. Hence, it is necessary that the underlying communication system provides a delay bounded service. In fact, while generic traffic needs only to be delivered (traffic does not have timing constraints), QoS traffic is usually coupled with timing constraints (deadlines). The presence of this delay bounded service allows the protocol to know whether it is able to meet the application timing requirements or not. In this case the protocol is also called *real-time* protocol.

Many real-time protocols have been proposed for wired networks, but their use in the wireless scenario poses some limitations to the network scenario. For instance, many protocols require central entities to provide QoS capabilities, and some others [7, 8] provide timing guarantees only in networks where hidden terminals are not present.

Since the absence of central entities and the presence of hidden terminals are key assumptions of ad hoc networks, it is necessary that the underlying protocol deals with hidden nodes and does not use any central entities.

Recently, some real-time protocols for wireless ad hoc networks have been proposed. For instance, [9, 10] use message exchange information among nodes to guarantee the contention free. Hence, they introduce large overhead. TPT (Token Passing Tree) [11] is a protocol for supporting real-time applications in indoor ad hoc networks in which terminals have low mobility and limited movement space. This protocol provides a delay bounded service and manages hidden terminals. The delay bounded service is derived from the one of the timed token protocol [12] and hence, the performances of TPT are affected from the performances of the timed token protocol. As described in [13, 14], real-time protocols based on the timed-token idea achieve low performance with respect to protocols where multiple stations can access the network at the same time.

The contribution of this paper is to propose a novel MAC protocol that provides a delay bounded service in ad hoc networks. The protocol is derived from RT-Ring, a wired real-time MAC protocol that provides timing guarantees in LAN/MAN while achieving better performance than protocols based on the timed-token mechanism [13].

As well as TPT, our protocol, named WRT-Ring (Wireless RT-Ring), is designed for indoor scenarios in which terminals have low mobility and limited movement space (airport lounge, conference site, meeting room, etc.). Throughout this paper we show that WRT-Ring can support generic and QoS applications, by providing two type of services: delay bounded and best-effort. WRT-Ring is provided with the compatibility with the emerging Diffserv architectures [15]. The analysis of WRT-Ring shows that it can provide timing guarantees and hence it can support QoS applications. The evaluation of the protocol is done through a comparison study with the TPT protocol, as TPT has been designed for the same purpose of WRT-Ring. The comparison shows that WRT-Ring can better react to the changes of the wireless environment while offering the same service.

Note that, although a MAC protocol is composed of a real-time bandwidth algorithm, in this paper we don't propose any bandwidth allocation scheme, as several studies [16, 17] have been focused on finding efficient bandwidth allocation schemes that can be implemented in WRT-Ring, using the presented WRT-Ring properties.

The remainder of this paper is organized as follows. In section 2 we present characteristics and properties of the proposed protocol. In section 3 we present a brief description of the TPT protocol and the WRT-Ring performance study. Conclusions are drawn in section 4.

2 WRT-Ring protocol

In this section we present characteristics and properties of our proposed protocol. WRT-Ring is derived from RT-Ring, a real-time protocol for wired networks that allows concurrent transmissions and achieves better network performances than protocols based on the timed-token architecture [13, 14].

Since WRT-Ring operates in wireless scenarios, the protocol is provided with characteristics and properties that are very important in this environment, such as user mobility and changes to the topology. The protocol uses CDMA mechanism [18], a mechanism that allows multiple transmissions without causing collisions. The protocol integrates two types of traffic: best-effort and real-time. The integration of these traffic is achieved using a mechanism that is shown to ensure fairness among the stations. A worst case investigation shows that WRT-Ring can provide timing guarantees to the supported stations, as it is provided with a bound on the network access time. Further, WRT-Ring is provided with the compatibility with the emerging Diffserv architecture [15]. This is an important feature, as Diffserv should handle real-time traffic in future IP-Networks.

2.1 Network Scenario

In this section we show the characteristics that a network should have in order to use WRT-Ring. We consider a wireless ad hoc network composed of several stations, say N, in which stations can communicate each other over a single hop, or through other stations (to reach hidden stations).

WRT-Ring requires the stations to form a virtual ring. For this reason, it is required that each station

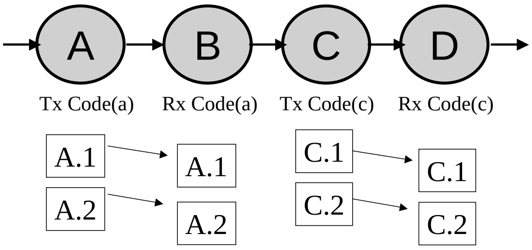

Figure 1: CDMA: concurrent transmission without collisions.

can communicate with, at least, two stations over a single hop. This is a reasonable assumption in indoor scenario, like meeting rooms, campus, etc, where the density of mobile users easily allows to create a virtual ring. The implementation of the virtual ring goes beyond the design of a MAC protocol, since routing protocols can be used for this purpose. However, we provide the protocol with routines that handle the insertion or the remotion of a station while supporting a real-time session.

As we already stated, WRT-Ring uses several characteristics of RT-Ring. One of these, is that multiple stations can access the network at the same time. While this technique can be easily implemented over wired networks, it may arise collision problems over wireless networks. For this reason, WRT-Ring uses a mechanism that can avoid collisions, such as the CDMA [18] mechanism. Briefly, this mechanism, allows to share a channel, by using a code mechanism that allows multiple transmissions without causing collisions. In this way, each station can access the ring without being aware of other stations behavior (multiple stations can access to the ring at the same time). This is done by assigning a unique code to each station, such that two stations can communicate only using the assigned code. The code is unique and can be assigned in different ways (for instance, in [19] a distributed algorithm is proposed in order to assign code to each station). The codes allow each station to only receive data encoded with a particular code and to discard data encoded with a different code [20]. The assignment of these codes goes beyond the scope of this paper. For this reason, we assume that CDMA codes are given to each station when the virtual ring is created.

Using CDMA, it is possible to have the situation depicted in Figure 1, where a small part of the ring is shown. Station A can transmit to station B, and station C can transmit to station D. Using different codes, station B can receive data from station A, even if station A and station C transmit at the same time. If CDMA would not be used, a collision between data transmitted from station A and station C happens, causing station B to receive corrupted data.

In addition to the given codes, each station is provided with a common code, which can be used to communicate with all the stations. In essence, this code represents the broadcast channel and it will not be used under normal conditions, but only when network topology changes.

The use of CDMA coupled with TDMA allows to consider the communication channel as composed of several slots. In this way, after the ring initialization, fixed-size slots continuously circulate into the ring. Each slot has a header and a data field. Among other information, the header contains a bit that indicates the status busy or empty of the slot. If the bit is set, the data field contains useful user data. In this paper we normalize all the time quantities to the slot duration, i.e., we use the slot duration as our time unit and all time quantities are expressed in number of slot duration.

2.2 Integration and Fairness mechanism

WRT-Ring is designed to support both generic and QoS applications. These applications produce two types of traffic: generic (or best-effort) and real-time traffic. Each traffic has dedicated queues inside any station: one is for the real-time traffic and the other is for the generic traffic. Real-time traffic is provided with higher priority than generic traffic.

The fairness mechanism used in WRT-Ring provides the stations with the same opportunity to access the network and provides timing guarantees to all the stations. The fairness mechanism is derived from RT-Ring, and hence we briefly summarize it in the following (readers can refer to [13, 14] for further details).

A control signal (named SAT) circulates in the ring with the same traffic direction. During each rotation, the SAT gives a predefined number of transmission authorizations to each visited station. The number of transmission authorizations is defined with two local parameters: l is used for the real-time packets and k is used for the best-effort packets. These parameters represents the maximum number of packets a station can send during a SAT rotation. Hence, during each SAT round, a station can authorize l real-time packets and k best-effort packets.

These authorizations are necessary because a station can transmit its packets only if it has collected transmission authorizations.

In particular, after each SAT departure, by exploiting the authorizations it has collected, a station i can transmit up to l_i real-time packets from its real-time queue and up to k_i non real-time packets from its non real-time queue. The authorizations for non real-time traffic must be used before the SAT returns to the station, i.e., within the SAT rotation in the ring. The authorizations for the non real-time traffic, still available when the SAT comes back at the station are not valid anymore.

In this way, during each SAT round, a station can transmit not more than k non real-time packets. To deliver real-time traffic (if any) before the non real-time traffic, WRT-Ring provides real-time traffic with higher priority than the non real-time traffic.

Note that, the SAT doesn't travel freely in the network; in fact, every time it visits a station, it can be either immediately forwarded or seized, depending on the status of the station. A station can be in two possible states: *satisfied* or *not satisfied.*

A station, say i, is said *satisfied* if it has no real-time traffic ready to be transmitted, or if between two consecutive SAT visits it has transmitted a pre-defined quota of real-time packets, denoted with l_i (one of the local parameters).

Conversely, a station, say i, is said *not-satisfied* if it has real-time traffic ready to be transmitted, and it has transmitted less than l_i packets since the last SAT visit.

When the SAT visits a not-satisfied station, the station seizes it until the station becomes satisfied. Once satisfied, the station releases the SAT, sending it to the next station.

With this policy, it is clear that every station cannot authorize more than $l + k$ packets during every SAT round. Hence, a station cannot transmit more than $l + k$ packets.

Send algorithm

1. A station can send real-time packets only if RT_PCK is not greater than l;
2. A station can send non real-time traffic only if NRT_PCK is not greater than k and the real-time buffer is empty or RT_PCK is equal to l.

After transmitting a real-time packet, RT_PCK is incremented by one, while after transmitting a non real-time packet, NRT_PCK is incremented by one.

SAT algorithm

When a station receives the SAT, it can:

1. forward the SAT if the station is satisfied, i.e. $RT\_PCK = l$ or the real-time queue is empty;
2. hold the SAT until it becomes satisfied.

After releasing the SAT, RT_PCK and NRT_PCK are cleared.

2.3 Mapping Internet Differentiated Services on WRT-Ring

Real-time protocols should be compatible with the Diffserv architectures, as this compatibility will play a fundamental role in future years. In fact, the Diffserv architectures should handle real-time traffic in future packet switching networks. For this reason, WRT-Ring is designed to interact with networks where Diffserv architectures are used. In particular, WRT-Ring can interact with the Differentiated Service Architecture proposed in [15]. To highlight the compatibility, we analyze a possible scenario where an ad hoc network is connected to a LAN (where *Diffserv* is used) (Fig. 2).

WRT-Ring can handle real-time traffic inside the ad hoc network and we show that it can manage real-time traffic transmission across the two networks. In fact, the gateway (Station G1, in Fig. 2) exactly

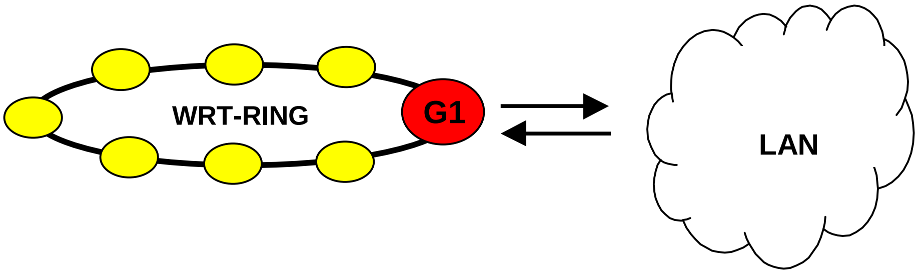

Figure 2: Interconnection with a LAN.

knows the amount of the real-time traffic sent across the two networks and hence this station doesn't differ from the other stations in the ring (as the others, it has its own real-time and generic traffic). For instance, suppose that a real-time stream has to be sent from a LAN to a ad hoc network. Before establishing a connection, the LAN asks G1 for the needed bandwidth to transmit the real-time stream towards the ad hoc network. Station G1 is controlled by WRT-Ring, hence the protocol checks whether it is able to reserve the required bandwidth to G1 or not [1]. If so, the bandwidth is allocated and the real-time service can be guaranteed. The same happens if a real-time stream has to be sent from the ad hoc network to the LAN. In this case G1 asks the Diffserv architecture if the necessary bandwidth can be guaranteed inside the LAN.

To be fully compatible with the Diffserv architecture presented in [15], we show how to implement different classes of services in WRT-Ring. In particular we show how to implement the three different classes of services introduced in [15]: *Premium* (that regards the traffic with full guarantees), *Assured* (traffic with no guarantees, but with priority higher than the best-effort traffic) and *best-effort* (traffic with no guarantees and lowest priority). Note that this implementation is very simple, and it doesn't require any modifications to the WRT-Ring protocol. In fact, any single station can decide the number of classes of services to implement. These classes are provided to its own traffic, without affecting and without being affected by the behavior of the other stations.

The implementation of the three classes of services can be done as follows.

Since l quota is the only guaranteed quota in WRT-Ring, it can be comparable to the Premium class of the Diffserv architecture proposed in [15]. The k quota is not guaranteed in RT-Ring and hence it can be comparable to the Assured and best-effort classes of the Diffserv architecture. In order to provide two classes of services, the quota k can be split into two different quotas, k_1 and k_2 (with $k_1 + k_2 = k$). In this case, k_1 represents the quota reserved to the Assured traffic and k_2 represents the quota reserved to the best-effort traffic. Note that, providing k_1 with higher priority than k_2, the network access mechanism

---

[1] The design of a bandwidth allocation algorithm for RT-Ring is outside the scope of this paper. However, by exploiting the WRT-Ring properties (see next section) it is possible to apply to WRT-Ring the algorithms developed for FDDI.

doesn't change.

2.4 Changes to the network topology

In wireless environments, the network topology changes more frequently than in wired scenarios. For this reason, we first present how WRT-Ring behaves when network topology changes (new stations require to enter into the network, or stations leave the network) and when the SAT control signal gets lost and then we prove that WRT-Ring provides timing guarantees to the supported applications.

2.4.1 A new station requires to join the network

Due to the users mobility, it may happen that a new station asks to enter into the ring. For instance, if we consider a conference room, a late attendant may join the meeting when it is already started. In this case, the protocol should be able to manage the insertion of the requesting station without compromising the QoS guarantees provided to the stations that are already part of the ring. If the insertion may affect the guarantees offered to the supported applications, the protocol has to reject the request.

To enter the network, a requesting station has to contact a station in the ring, called *ingress* station, which manages the insertion procedure. The ingress station has to ear if new stations are asking to join the ring. This is done in a period where transmissions are not allowed. This period is called RAP (Random Access Period), is denoted with T_{rap}, and is announced with a broadcast message.

There are two phases in this period: an *earing* and an *update* phase. The earing phase is T_{ear} long and it is used to ear if new stations are asking to enter the network. The update phase is T_{update} long and it is used to update the network topology after a new station has been accepted into the network. Needless to say, $T_{rap} = T_{ear} + T_{update}$.

Even though each station in the ring may act as an ingress station [11] (there are no central entities), only one station for each SAT round can access the ring. This is done because, during the RAP, transmissions are not allowed and hence the network is idle. A mutex flag inside the SAT signal is used to ensure that only one station for each SAT round can enter the network. To ensure the fairness, after acting as ingress station, a node has to wait $S_{round}(i)$ $(\geq N)$ SAT rounds in order to enter the RAP period again.

In WRT-Ring the network topology is a ring, and hence, the requesting station may enter the network only if it can directly reach (one single hop) two consecutive stations of the virtual ring. If this happens, as we see in the following, WRT-Ring can manage the insertion of the requesting station without difficulties. Conversely, if the requesting station can reach only one station, it cannot join the network (in this case it may form another ring, but we don't present a detailed analysis of this case in this paper).

If a requesting station can reach two consecutive stations it may enter the ring between the two stations, but before entering the network, it has to wait for a permission that is given by the ingress station. After receiving the permission, the station specifies its QoS traffic requirements and the network checks if the requirements can be satisfied.

In the following we describe the algorithms for the ingress and for the requesting stations.

Ingress station: algorithm

The parameter $S_{round}(i)$ represents the number of SAT rounds that a station has to wait before entering the successive RAP period. To avoid that two or more stations are in the RAP simultaneously, the station checks a flag in the SAT control signal. This flag, called RAP_{mutex}, indicates whether the station can enter the RAP or not. If the flag is zero, the station can enter, otherwise it cannot (another station is doing its RAP). If $RAP_{mutex} = 0$ then it is set to one and the RAP begins.

After entering the RAP, the station sends a broadcast message, called $NEXT\_FREE$, which contains different information. By supposing station i in the RAP period, $NEXT\_FREE$ contains: the address of the sender (station i), the code i (to share channel i), the address of the next station (station $i+1$), the code $i+1$ (to share channel $i+1$), the number of slots for the earing phase (T_{ear}) and the maximum number of resources (e.g., bandwidth) that the network can provide.

This message reaches the requesting station, as it is transmitted in broadcast. After the transmission of this message, the ingress station waits for a new station to respond.

New Station: algorithm

Let S_{new} be the requesting station. This station keeps on checking the broadcast channel in order to get a $NEXT\_FREE$ message.

This station records on a table all the sender of the $NEXT\_FREE$ messages it gets. This is done in order to find two consecutive stations.

When the station receives another $NEXT\_FREE$ message from the same station, it means that all the other stations in the network have already entered their RAP. At this point S_{new} knows if two consecutive stations are reachable over a single hop. Note that the time that elapses between two consecutive $NEXT\_FREE$ messages is equal to $S_{round} \cdot SAT\_TIME$[2].

Let us suppose that stations S_i and S_{i+1} will be eared by the new station S_{new}. Station S_i can be the ingress station for S_{new}. Upon the reception of the next $NEXT\_FREE$ message from S_i, station S_{new} replies with a message encoded with code(i). This message contains: the address of S_{new}, the code $code_{new}$, the number of l_{new} and k_{new}.

After sending the message, the station waits for a message encoded with $code_i$ (it waits for a reply from station i). If the reply does not arrive within T_{ear} slots, S_{new} cannot join the network, and will wait for the next $NEXT\_FREE$ messages.

---

[2] SAT_TIME is derived in the next section and it represents the upper bound to the SAT rotation time.

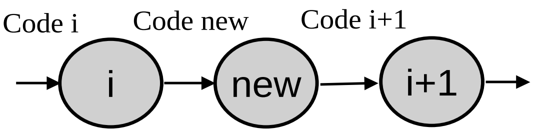

Figure 3: A new station joins the network.

2.4.2 A station leaves the ring

As well as a station can enter the ring, a station can leave the ring. The exit can be communicated by the leaving station or not. In fact, a station may leave the ring on purpose, or it may be forced to leave the ring (no more batteries, out of the reachable zone, etc.). If a station wants to leave the network it can use the SAT signal to inform all the ring stations that is leaving the ring. For instance, if station i wants to leave the ring, it informs station $i+1$, which behaves as if the SAT would have been lost. As we better explain in the following, it sends the SAT_REC message instead of the SAT message. If the SAT_REC message arrives again at station $i+1$, the ring is always on even without station i.

If station i leaves the network without notifying it, the network activates the SAT recovery procedure, as if the SAT has been lost between station $i-1$ and station i.

2.5 SAT Loss

The purpose of the SAT control signal is to provide a number of transmission authorizations to each station during each SAT round. Since the SAT can be lost, the protocol must provides algorithms to find out that the SAT has been lost and has to recover from that situation. If the SAT control signal gets lost, the guarantees to the supported QoS applications cannot be guaranteed anymore. For this reason, it is fundamental that the network finds out, as soon as possible, that the SAT is no longer traveling into the network. WRT-Ring uses the following mechanism to find out that the SAT got lost.

Each station i has a local timer, named SAT_TIMER_i, and if the SAT does not come back within SAT_TIME slots, the SAT is considered lost and the recovery procedure is activated.

SAT Recovery Let us suppose that station $i+1$ finds out the SAT loss. This station generates a new SAT control signal, named SAT_REC. This particular signal is generated in order to inform all the other station in the network that the SAT has been lost. The SAT_REC contains information about the network address of the supposed failed station (in this case, station i) and the CDMA Code $i+1$, in order to allow communication with station $i+1$. In fact, station i was the only station that could communicate with station $i+1$.

When a station receives a SAT_REC it considers this signal as a normal SAT only if the successive station is not i. Station $i-1$ behaves differently: it does not send the signal with the code i, but it sends it with the code $i+1$. Note that station $i-1$ is trying to re-establish the previous network scenario, by simply cutting out station i.

If station $i+1$ receives the SAT_REC within the SAT_TIME_{i+1}, it means that the ring has been re-established, and it substitutes the SAT_REC with the SAT signal. It is to note that the transmission quota assigned to station i (l_i and k_i) can be re-assigned to all the other station.

If station $i+1$ does not receive the SAT_REC within SAT_TIME_{i+1}, the previous ring is no longer valid (for instance, station $i-1$ could be too far to directly reach station $i+1$). If this happened, station $i+1$ broadcast a message, notifying that the network has been lost and a new procedure to form a ring must take place.

If there are no hidden terminals, as in [24, 11], the recovery procedure cannot fail as station $i-1$ can always directly reach station $i+1$.

2.6 Bound to the network access time

A real-time protocol must provide a bound to the network access time in order to support applications that require a delay bounded service. In this section we prove the presence of this bound by first showing that the SAT rotation time is bounded and then, using this bound, we derive the upper bound to the network access time.

Since the network access time depends on the traffic condition (hence it is impossible to know its value ahead of time), it is important for the protocol to know the maximum value it can assume under all traffic patterns. This can be achieved with a *worst-case analysis* that provides the upper bound to the network access time in the worst-case scenario. Even though the worst case scenario may not be realistic or happens with a very low probability, it is the only way to derive the upper bound to network access time, and hence to provide guarantees (i.e., with probability 1) to the real-time application [22, 23].

The following properties are very similar to the RT-Ring properties. The only difference is the presence of the T_{rap} as it represents the number of slots that are used to ear if some stations are requiring to enter the network. Since the proofs are the same of RT-Ring, we don't report these proofs in this paper, but we refer the readers to [13] for details.

2.6.1 Upper bound to the SAT rotation time

In this section we derive an upper bound to the SAT rotation time, i.e., the time interval between consecutive arrivals (departures) of the SAT from the same station, denoted with SAT_TIME. This bound is important since it represents the longest time a cycle (i.e. two consecutive SAT arrivals at the same station) can be, and it is used to derive the upper bound to the network access time. First, we note that SAT_TIME is affected by three possible components: the number of the stations, denoted with N, present in the ring; the time it takes to the SAT for traveling, without being stopped at any station, across the ring (by using the slot time as time unit, this time quantity cannot be greater than S); the time the SAT is held at the not-satisfied stations.

Theorem 1 *Let SAT_TIME_i be the time elapsed between two consecutive SAT arrivals (departures) at the same station i. SAT_TIME_i has an upper bound and the following holds:*

$$SAT\_TIME_i < S + T_{rap} + 2 \cdot \sum_{j=1}^{N} (l_j + k_j) \qquad \text{for all } i = 1, \ldots, N. \tag{1}$$

Proof The proof is equal to the one of RT-Ring. For this reason, we refer the readers to [13]. ◇

Proposition 1 *If $l_i = l_j$ and $k_i = k_j$ for each station j and each station i, then the maximum time elapsed between two consecutive SAT arrivals at the same station has an upper bound equal to:*

$$S + T_{rap} + 2 \cdot N \cdot (l + k) \tag{2}$$

Proof It follows from the previous Theorem. ◇

Theorem 2 *Let $SAT\_TIME_i[n]$ be the time elapsed between n consecutive SAT arrivals at the same station i. The following holds:*

$$SAT\_TIME_i[n] \leq n \cdot S + n \cdot T_{rap} + (n+1) \cdot \sum_{j=1}^{N} (l_j + k_j) \tag{3}$$

Proof The proof is equal to the one of RT-Ring. For this reason, we refer the readers to [13]. ◇

Proposition 2 *If $l_i = l_j$ and $k_i = k_j$ for each station j and each station i, then the maximum time elapsed between n consecutive SAT visits at the same station has an upper bound equal to:*

$$n \cdot S + n \cdot T_{rap} + (n+1) \cdot N \cdot (l + k) \tag{4}$$

Proof Follows from Theorem 2. ◇

Proposition 3 *The average SAT rotation time, $E[SAT\_TIME]$, is equal to:*

$$S + T_{rap} + \sum_{j=1}^{N} (l_j + k_j) \tag{5}$$

Proof The bound on average SAT rotation time, is derived as follows:

$$E[SAT\_TIME_i] \leq \lim_{n \to \infty} \frac{SAT\_TIME_i[n]}{n} = S + T_{rap} + \sum_{j=1}^{N} (l_j + k_j)$$

2.6.2 Upper Bound to the Network Access Time

In this section we use the upper bound to the SAT rotation time, obtained in the previous section, to derive an upper bound to the network access time. As happens in RT-Ring, the upper bound is affected by the SAT_TIME, and the following holds. The proof of this Theorem is not presented as it is equal to the one presented in RT-Ring. Readers can refer to [13] for the proof.

Theorem 3 *Let us consider a tagged real-time packet that is inserted in the station* i *queue for transmission and denote with* x *the number of real-time packets already present in the station* i *queue when the tagged packet arrives Let* T^i_{Wait} *be the time that this tagged packet has to wait before being transmitted. The following holds:*

$$T^i_{Wait} \leq SAT\_TIME[\lceil \frac{x+1}{l_i} \rceil + 1] \tag{6}$$

where $\lceil x \rceil$ *indicates the small integer greater or equal than* x. ◇

3 Efficiency of RT-Ring

In this section, we compare WRT-Ring with TPT (Token Passing Tree) protocol [11]. TPT is a protocol designed to support QoS applications in indoor environments in which users have low mobility and limited movement space. Since WRT-Ring has been designed with the same objective we present a study comparison between these two protocols. In particular we investigate the bounds to the average control signal rotation time and the reaction time to the loss of the control signal. These bounds are in fact very critical for a real-time protocol, as the average rotation time is used by bandwidth algorithms to efficiently allocate the bandwidth, and the reaction time allows the protocol to find out network problems and hence to recover from those problems or to inform the supported applications that the provided guarantees are no longer available.

Before going into the analytical comparison of the two protocols, we briefly review the main characteristics of TPT.

3.1 Token Passing Tree

The Token Passing Tree protocol [11] (TPT), has been designed to support real-time traffic over indoor ad hoc networks. It is based on the timed token MAC protocol [12] and its network access bound is straightly derived from the bound of the timed-token protocol.

In TPT the underlying ad hoc network is organized as a tree. A token travels into the network and provides transmission authorizations to the stations. A station that has to transmit, waits for the token, seizes it, transmits its traffic and then releases the token. Only the station that holds the token can transmit. At the end of its transmission period, the token is transmitted using a depth-first algorithm to reach all the station connected in the tree.

3.1.1 A new station

In order to include new stations into the network TPT periodically stops the transmissions using a flag in the token. When this flag is activated, the receiving station has to wait a T_{rap} period before starting its transmissions. If the requesting station can ear the token, it knows that the random access period is about to start. In this case, it tries with an handshake mechanism to enter the network. The network checks if the requesting station can be handled by the network and if accepted, it joins the tree considering the accepting station as the parent node in the tree (readers can find details in [11]).

3.1.2 Upper bound to the network access time

TPT is derived from the timed-token protocol and hence the protocol guarantees that the average token rotation time is equal to the Target Token Rotation Time (TTRT) and that the network access time cannot be greater than $2 \cdot TTRT$.

During the start-up phase, each station proposes its maximum access time to access the network in order to guarantee the considered applications. If we denote the maximum delay proposed by a station i with D_i, it follows that the network must provide guarantees that the network access time is not greater than $D = \min(D_i) \forall$ station i present in the tree.

Each station i can reserve a time quantity $H_{e,i}$ to transmit its real-time traffic. This means that, if we have N stations in the network, the sum of the reserved time cannot be greater than $TTRT$.

The bound becomes

$$\sum_{i=1}^{N} H_{e,i} + 2 \cdot (N-1) \cdot (T_{proc} + T_{prop}) + T_{rap} \leq D/2, \tag{7}$$

where T_{proc} is the time necessary to transmit a token message, T_{prop} is the time necessary for the token to move from one station to the successive.

3.1.3 Token loss

TPT uses a timer mechanism to find out a token loss. A timer is provided locally to each station, and it is initialized to the maximum token rotation time $(2 \cdot TTRT)$ every time the token departs from the station. When a station observes that this timer is expired, it puts into the network a new token. If the new token comes back to the sender station, it means that the tree is still valid, otherwise the tree is considered lost. At this point the station sends a broadcast message informing all the earing stations that the tree is no longer valid. A build-tree procedure has to be activated in order to form a new tree.

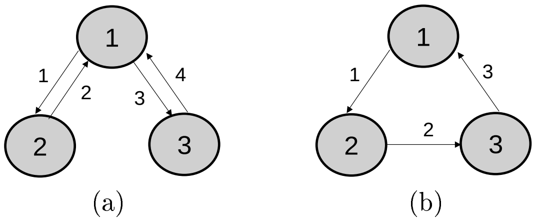

Figure 4: (a)Token-passing Tree. (b)WRT-Ring

3.2 Comparison study between WRT-Ring and TPT

The main difference between WRT-Ring and TPT is the method to access the network. Although there is a signal that travels into the network and controls the station transmissions (the token in TPT and the SAT in WRT-Ring), in TPT the signal allows the transmission only to the station that holds the token, while in WRT-Ring the SAT provides only transmission authorizations and, hence, several stations can access the network at the same time (this is possible using the CDMA technique).

Both protocols have been derived from a wired real-time protocol: the token-passing scheme for TPT and RT-Ring for WRT-Ring. In [13] it has been shown that if several stations can access the network at the same time, the protocol may achieve higher network capacities than protocols based on the token-passing scheme. In the following we do not compare such network capacities because this comparison is done in [13].

TPT has no channel contention as only one station at time can access the network and the authorization to access the network is given by the token, while WRT-Ring uses CDMA to avoid possible collisions, since several stations can access the network at the same time.

3.2.1 Hops to visit the whole network

In Fig. 4(a) we show a simple tree structure, with 3 nodes. Station 1 is the root of this structure and the token travels into the network as depicted by the number. To visit all the station and to come back at the original position, the token must travel from station 1 to station 2, then from station 2 back to station 1, then from station 1 to station 3 and than back to station 1. It is also necessary that the root (Station 1) has to directly reach both station 2 and station 3.

In Fig. 4(b) we show a simple ring structure. In this case, to visit all the station and to come back at the original position, the SAT must travel from station 1 to station 2, then from station 2 back to station 3, then from station 3 to station 1. In this case all the stations in the network has to see at least two stations in order to create the ring.

If we consider N stations, in TPT the token needs to travel $2 \cdot (N-1)$ links to complete one round, while the SAT, in WRT-Ring, needs to travel only N links to complete one round. As we can see in the following, this difference affects the performance of the protocols.

3.3 Bound comparison

Let us consider the round trip time of the signal control: token in TPT and SAT in WRT-Ring, respectively. The bound in TPT is given by equation (7) and the bound in WRT-Ring is given by equation (3). To compare the bounds, we consider the same scenario. This means that the two protocols handle the same applications with the same traffic load. Hence, the term $\sum_{j=1}^{N}(l_j + k_j)$ and the term $\sum_{i=1}^{N} H_{e,i}$ (i.e., the bandwidth reserved in WRT-Ring and in TPT, respectively) can be supposed equal.

If we do not consider the traffic, to perform one round trip in TPT, the token needs $2 \cdot (N-1) \cdot (T_{proc} + T_{prop}) + T_{rap}$ time units, where $T_{proc} + T_{prop}$ is the time necessary to transmit the control signal. By considering $T_{proc} + T_{prop}$ the time necessary to transmit the SAT in WRT-Ring, it follows that the SAT needs $N \cdot (T_{proc} + T_{prop}) + T_{rap}$ time units to complete one round-trip.

It is easy to note that the token needs more time to complete one round trip with respect to the SAT rotation time. This means that, by considering the same network scenario (i.e., the same number of stations and the same applications), WRT-Ring is able to support applications with more stringent QoS timing requirements that TPT.

Another interesting difference between TPT and WRT-Ring happens when the control signal gets lost. In TPT when a station is down, the current network topology is considered broken and a new tree must be created. Conversely, in WRT-Ring, when a station is down, a mechanism tries to cut out this broken station by connecting the previous and the following station (w.r.t. the broken one). Only if this is not possible, a new ring must be created.

Since in wireless environments the control signal can be frequently lost, it is important for the protocol to react as soon as possible to recover the situation. Both protocols use a timer that waits for the maximum token rotation time to find out that the control signal got lost. The maximum round trip time in WRT-Ring is given by SAT_TIME, while in TPT is given by $D = 2 \cdot TTRT$. By comparing these two bounds it is easy to observe that $SAT\_TIME < D$.

This means that if the control signal gets lost, WRT-Ring can react in a shorter time than TPT. Hence, WRT-Ring can better support the QoS applications, as it able to re-establish the previous guarantees or to inform the applications that the service is no longer available.

4 Conclusion

In this paper we proposed a new protocol, WRT-Ring, for supporting QoS applications over wireless ad hoc networks. The protocol is derived from RT-Ring, a real-time protocol for wired networks [13].

We proved that WRT-Ring is a protocol that provides timing guarantees to the applications and hence is a protocol that can support QoS applications over ad hoc networks. Since the importance of being compatible with the Diffserv architectures, we provided our protocol with the compatibility with these architectures and we evaluated it through a comparison analysis with the TPT protocol.

Results showed that WRT-Ring can react to the network changes in shorter time than TPT, while providing the same bounded delay service. Further, since protocols based on the RT-Ring architecture have better performance than protocols based on the timed-token mechanism [13], WRT-Ring is an efficient and effective protocol for supporting QoS applications in ad hoc wireless networks.

References

[1] IEEE 802.11. Draft standard for Wireless LAN. 1996

[2] G. Anastasi, L. Lenzini, E. Mingozzi, *Stability and Performance Analysis of HYPERLAN*, in proceedings of IEEE INFOCOMM'98, pp. 134-141, San Francisco, CA, USA, 1998.

[3] M. Ogawa, I. Shimojima, T. Hattori, *CoS Guarantee Control for Wireless LAN*, in Proceedings of VTC 2002, Birmingham, AL, USA, pp. 50-54, 2002.

[4] P. Karn: *MACA-A new channel access method for packet radio*, in Proceedings of ARRL/CRRL Amateur Radio Ninth Computer Networking Conference, 1990.

[5] F. Talucci, M. Gerla: *MACA-BI (MACA By Invitation) A Wireless MAC Protocol for High Speed Ad Hoc Networking*, The 6th IEEE International Conference on Universal Personal Communications, pp. 193-917, 1997.

[6] C.L.Fullmer, J.J. Garcia Luna Aceves: *Floor Acquisition Multiple Access (FRMA) for Packet-Radio Networks*, in Proceedings of ACM SIGCOMM, 1994.

[7] J.L. Sobrinho, A.S. Krishnakumar: *Real-Time traffic over the IEEE 802.11 Medium Access Control Layer*, Bell Labs Technical Journal, Vol. 1, No. 2, pp. 172-187, 1996

[8] M.J. Markowsk, A.S. Sethi: *Fully Distributed Wireless Transmission of Heterogeneous Real-Time Data*, Vehicular Technology Conference, Vol. 2, pp. 1439-1442, 1198.

[9] C.R. Lin, M. Gerla: *Asynchronous Multimedia Radio Network*, IEEE Infocomm 97, March 1997.

[10] A. Muir, J. Garcia Luna Aceves: *Supporting real-time multimedia traffic in a wireless LAN*, in Proceedings of SPIE multimedia computing and Networking, pp. 41-45, 1997

[11] R. Jianqiang, J. Shengming, H. Dajiang, *A Token Passing Tree MAC Scheme for Wireless Ad Hoc Networks to Support Real-Time Traffic*, in proceedings of the International Workshop Mobile and Wireless Communications Networks (MWCN 2000), Paris, May 2000, pp.47-57.

[12] N. Malcom, W. Zhao, *The Timed-Token protocol for real-time communications*, IEEE Computer, Vol. 27, No. 1, January 1994, pp.35-41.

[13] M. Conti, L. Donatiello, M.Furini, *Design and Analysis of RT-Ring: a protocol for supporting real-time communications*, IEEE Transaction of Industrial Electronics, to appear, December 2002.

[14] M., Conti, L., Donatiello, M., Furini, *Design and Analysis of RT-Ring: a protocol for supporting real-time communications*, in Proceedings of the 3rd IEEE Workshop on Factory Communications (WFCS 2000), pp. 91-98, Porto, September 2000.

[15] Nichols, K., Jacobson, V., Zhang, L. : *A Two-bit Differentiated Services Architecture for the Internet*, Internet Draft, 1998.

[16] G. Agrawal, B. Chen, W. Zhao, S. Davari, *Guaranteeing synchronous message deadlines with the timed token medium access control protocol*, IEEE Transaction on Computer, vol. 43, n. 3, pp. 327-339, March 1994.

[17] S., Zhang, A., Burns, *An Optimal Synchronous Bandwidth Allocation Scheme for guaranteeing Synchronous Message Deadline with the Timed-Token MAC protocol*, IEEE/ACM Transaction on Networking, vol. 3, n. 6, pp. 729-741, December 1995.

[18] R. L. Chunhung, L. Jain-Shing, *QoS Routing in Ad Hoc Wireless Networks*, IEEE Journal, August 1999.

[19] L.Hu, *Distributed Code Assignments for CDMA packets radio networks*, IEEE/ACM Transaction on Networking, Vol. 1 No. 6, December 1993, pp. 668-677.

[20] R.L. Chunhung, L. Jain-Shing, *QoS Routing in Ad Hoc Wireless Networks*, IEEE Journal on Selected Areas in Communications, Vol. 17 n.8, pp. 1426-1438.

[21] L. Hu, *Distributed code assignments for CDMA packet radio networks*, IEEE/ACM Trans. Networking, December 1993.

[22] D. Chen, V.C.S. Lee, E. Chan, *On the ability of the FDDI-M protocol to support real-time traffic*, in Proceedings of the fifth International Conference on Real-Time Computing Systems and Applications, 1998, pp. 51-57, October 1998.

[23] S. Zhang, E.S. Lee, *The worst-case scenario for transmission of synchronous traffic in an FDDI network*, in Proceedings of Local Computer Networks 1999 (LCN'99), pp. 192-201, October 1999.

[24] M.J. Markowski, A.S. Sethi, *Fully Distributed Wireless MAC Transmission of Real-Time Data*, in proceedings of fourth IEEE Real-Time Technology and Applications Symposium, 1998, pp. 49-57.