Design and Analysis of RT-Ring: A Protocol for Supporting Real-Time Communications

Marco Conti, Lorenzo Donatiello, and Marco Furini

***Abstract*—Distributed applications with quality of service (QoS) requirements are more and more used in several areas (e.g., automated factory networks, embedded systems, conferencing systems). These applications produce a type of traffic with hard timing requirements, i.e., transmissions must be completed within specified deadlines. To handle these transmissions, the communication system must use *real-time* protocols to provide a communication service that is able to satisfy the QoS requirements of the distributed applications. In this paper, we propose a new real-time protocol, called RT-Ring, able to support transmissions of both real-time and generic traffic over a ring network. RT-Ring provides both network guarantees and high network resource utilization, while ensuring the compatibility with the emerging differentiated service architectures. Network guarantees are fully proved and high network utilization is highlighted by a comparative study with the FDDI protocol. This comparison shows that RT-Ring network capacities are greater than the corresponding FDDI capacities. In fact, by assuming the FDDI frames with a length equal to the RT-Ring slot size and by using the same traffic load we show that the capacities of FDDI are equal to the lower bound capacities of RT-Ring.**

***Index Terms*—Real-time protocol, quality of service (QoS) traffic, worst case analysis.**

I. Introduction

THE use of distributed applications with stringent quality of service requirements (QoS applications hereafter) is becoming more and more important in several scenarios: from automated factory networks to LANs, from MANs to the Internet. Classical data applications (e-mail, file transfer, etc.) distributed in a local and metropolitan area networks (LAN/MAN) require only a reliable transportation service but they have no other particular requirement. On the other hand, QoS applications may require some other performances guarantees from the communication service: an upper bound of the end-to-end delay and/or of the delay variability, a packet loss rate not greater than a certain threshold, a minimum guaranteed throughput, and others. For example, multimedia applications that include voice and video streams require both an upper bound of the end-to-end delay with which information is transferred from source to destination and a low (possibly zero) probability that packets violate the deadline constraints. The transmission of an alarm message needs both a delay constraint and high reliability, while some data transfer applications could require a minimum guaranteed throughput. Hereafter, we will consider two main classes of distributed applications: *non-real-time applications* (mainly classical data applications) and *real-time applications*. The latter is the subset of QoS applications for which the correctness of the application depends not only on the logical results of computation, but also on the timing properties of the system [33]. In fact, traffic generated by a real-time application is coupled with a deadline, and the communication system must rely on a network that provides transmission guarantees (i.e., the traffic deadline must be met). For example, for a remote flight control system it is not only important the correctness of the data, but it is fundamental that data arrive at the controlled airplane within a fixed delay.

In the past, the most common approach to real-time communication in the automation industry was the use of circuit-switching networks, or proprietary networks. For instance, Allen–Bradley's RIO (Remote Input/Output) Network and Control Net have been used for automated factory networks to meet application's stringent QoS requirements and deal with harsh working environments [21].

However, in the last few years, the network scenario has been changing and packet switching networks are now more common than the circuit switching networks. The main reason for this change is that packet switching networks are less costly and can achieve better utilization of the network resources than circuit switching networks. This means that, transporting real-time traffic over packet switching networks has become essential to many scenarios: from automated factories to many embedded systems, from audio/video conferencing to remote medical services.

Currently, the manufacturing automation industry has been pursuing the use of commercial off-the-shelf (COTS) network products in communicating control messages between programmable logic controllers (PLCs). Process control signals, online transaction messages, manufacturing control signals, and multimedia traffic, are other examples of real-time traffic.

The first requirement to provide a real-time communication service is on the adopted network technology. It is not possible to provide a real time service if the network technology does not offer some guarantees, such as the following [15].

1) *Upper/lower bound on the packet transfer delay:* The network must provide at least an upper bound on packet transfer delay. This allows us to handle users that want to transfer messages with temporal constraints.

Manuscript received March 20, 2001; revised June 29, 2001, September 6, 2001, and November 13, 2001. Abstract published on the Internet September 13, 2002.

M. Conti is with the National Research Council, IIT Institute, 56010 Ghezzano, Pisa, Italy (e-mail: m.conti@iit.cnr.it).

L. Donatiello is with the Computer Science Department, University of Bologna, 40127 Bologna, Italy (e-mail: donat@cs.unibo.it).

M. Furini is with the Computer Science Department, Piemonte Orientale University, 15100 Alessandria, Italy (e-mail: marco.furini@mfn.unipmn.it).

Digital Object Identifier 10.1109/TIE.2002.804969

2) *Guaranteed bandwidth:* A real-time service should be able to guarantee, on a finite time interval, to each real-time application a portion of the channel bandwidth for the transmission of its packets.

TDMA, 802.3D [23], FDDI [2], FieldBus [30], MetaRing [28], and SRP [34] are examples of proposed and commercial protocols that have been used for handling real-time traffic in LAN/MAN networks. These protocols differ from classical protocols (CSMA/CD [20], token-bus [17], token-ring [18]) as they provide transmission guarantees to the supported applications. Since transmission guarantees have to be provided when the application requests the real-time service and since transmission guarantees mean the satisfaction of the traffic timing requirements, it is clear that a station should know *a priori* whether it is able to meet the traffic deadline of the requesting application or not.

To summarize, the two fundamental characteristics that a LAN/MAN network must have to support a real time communication service are: 1) an upper bound to the network access time and 2) a bandwidth allocation scheme that, by exploiting 1), is able to reserve a portion of the network bandwidth to each real time application in order to meet its deadline constraints. For example, for FDDI, in [22] it is shown that the token rotation time is bounded (and, hence, the delay before a station can transmit a quota of packets is bounded), and in [35], starting from the protocol bounded-delay properties, a bandwidth allocation scheme for satisfying the real-time traffic is defined.

The upper bound to the network access time is a fundamental characteristic of a real-time protocol, as it represents the maximum time a station has to wait before accessing and transmitting into the network. Hence, if the upper bound value is greater than the traffic deadline, then the traffic deadline cannot be guaranteed. Hence, when designing a real-time protocol, it is fundamental to provide this bound.

As pointed out by Agrawal *et al.* [1], despite the fact that the bound is necessary, it is not sufficient for providing real-time communications. In fact, a real-time bandwidth allocation scheme is also needed. The bandwidth allocation is very important, as a wrong allocation may not satisfy the traffic requirements [1], [16]. In this paper, we focus on designing a real-time protocol and to provide an upper bound to the network access time. We don't propose any bandwidth allocation scheme as, given the protocol delay bound, it is possible to apply one of the efficient schemes present in literature (see, for example, [1] and [35]).

Real-time communication problems have also been studied in the Internet environment, but the best-effort nature of this packet switching network posed significant problems in delivering real-time services. Recently, to solve these problems, a small set of differentiated service (*diffserv*) has been introduced in the Internet. For instance, the 2-b architecture proposed by Nichols *et al.* [27] is one of such proposals and it provides three different classes of services to the Internet applications: Premium (real-time traffic whose transmission is fully guaranteed), best-effort (generic traffic), and Assured (traffic with higher priority than the best-effort traffic).

The great interest in real-time communication is due to the broad impacts that this type of communication has on many areas. For example, in the global internetworking environment, as the Internet, most end-users are connected to the global network via LANs or MANs. End-to-end QoS can be achieved only using a protocol that provides guarantees up to the end-user. For instance, consider two end-users, belonging to different LANs. End-to-end QoS is possible only if both the backbone network and LANs/MANs, through which the end-users are connected, provide bounded delays. For example, several cameras may be connected through a MAN and can be used to control different buildings located in a metropolitan environment. The MAN may also be connected to the Internet in order to transmit, or to receive, real-time streams from other networks. Remote industrial control process systems can be another example: in some period of the year the request of electricity can be very high (for instance, when a lot of people use air condition systems) and a power station could have problems in accommodating all of these requests at the same time. It is reasonable that the power station could make an agreement with customers that are willing to pay less while receiving different electricity load during the day. This process could be automated using computers connected through real-time networks: the computer at the customer side (for instance, an industrial process control system) communicates with the power station and, depending on the energy load information received, it could activate/deactivate electrical devices. Needless to say, these communications are real-time communications.

These simple, but realistic, examples show the benefits of having real-time protocols able to communicate with external networks. Needless to say, these benefits increase if the used protocols are able to achieve high network utilization, to provide real-time services and to be compatible with the differentiated services architectures.

The contribution of this paper is the proposal of a new real-time protocol, called RT-Ring, that provides network guarantees (i.e., we provide an upper bound to the network access time) and high network resource utilization, while ensuring the compatibility with the Differentiated Service Architecture proposed in [27]. The compatibility with these emerging architectures is an important feature of RT-Ring, as it allows RT-Ring to connect with current and future wide area networks (as the Internet2 [19]), where differentiated services architectures are used. The main motivation in developing RT-Ring was to overcome the restrictions about the effectiveness of the timed-token-like protocols. Indeed, as proved by *Conti et al.* [10], in these protocols, the presence of the ring latency may significantly reduce the utilization of the network bandwidth.

For this reason we provide RT-Ring with concurrent access and spatial reuse (as in [8], [34]), in order to increase the throughput of RT-Ring beyond the link capacity. The benefits introduced by these techniques are considerable. In fact, if we consider a slotted ring network, with N stations having uniform distribution for the traffic destination, the average distance for a packet to travel is $N/2$. This means that during one single rotation, the same slot can be used by two different stations (i.e., the spatial reuse factor is two).

RT-Ring has a unidirectional ring topology and can support both real-time and non-real-time applications. This integration is done since in most real-time systems, activities that have to

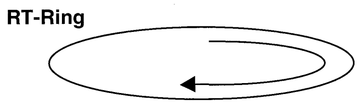

Fig. 1. RT-Ring topology.

occur in a timely fashion coexist with those that are not time critical.

Throughout the paper, we show that RT-Ring has an upper bound to the network access time and we prove the correctness of this bound.

To evaluate our proposed protocol, we compare it with the FDDI protocol. FDDI has been chosen, as it is one of the most studied real-time protocols for providing a high-speed communication subsystem for a distributed real-time system [1], [22], [25], [26], [13], [35], [16].

The comparison between these two protocols is done by investigating the protocol capacities (real-time, non-real-time, and global) achieved by RT-Ring and by FDDI. Results obtained show that the RT-Ring capacities are greater than the corresponding FDDI capacities, in sense that by assuming FDDI frames with a length equal to the RT-Ring slot size and by using the same traffic load we show that the capacities of FDDI are equal to the lower bound capacities of RT-Ring.

This paper is organized as follow. In Section II, we present the characteristics of the RT-Ring protocol. In Section III, we derive several RT-Ring properties and we prove the presence of an upper bound to the network access time. In Section IV, we evaluate RT-Ring by comparing it with the FDDI protocol. Conclusions are drawn in Section V.

II. RT-Ring Protocol

RT-Ring is designed to operate in a unidirectional slotted ring network topology, with fixed-size slots circulating into the ring (Fig. 1). Similarly to FDDI and MetaRing, the ring can be implemented with fiber-optic transmission links between adjacent stations. As in FDDI, in addition to the primary ring, a secondary ring can also be implemented for providing fault tolerance. The two rings are counterdirectional, and the secondary ring is not used under normal operating conditions.

It is assumed that, as in FDDI, token ring, MetaRing and other protocols, link and node failures are detected by some protocol. For instance, when a link or a node is detected faulty, it is removed from the network, and a new setup procedure is called. However, the handling of these events goes beyond the scope of this paper.

In this section we present the RT-Ring protocol with its basic principles: access control, fairness mechanism, integration of real-time and non-real-time traffic. Since in recent years there has been a large interest in Differentiated Services Architectures [27] that aim to handle real-time traffic in future networks, as the Internet2 [19], we provide RT-Ring with the possibility of being connected to the differentiated service architecture proposed in [27]. The mapping of the Internet Differentiated Services on RT-Ring can be done without any problems and since it has been presented in [12], we do not present it here. Readers can refer to [12] for further details on this mapping.

A. Access Control

According to the OSI reference model, the functions of a LAN/MAN network technology are grouped into several layers, e.g., Physical, Medium Access Control (MAC) and Logical Link Control layers. Hereafter, we only concentrate on the MAC layer. Since a LAN/MAN network relies on a common transmission media, the MAC protocol is in charge of managing the sharing of the transmission media. The aim of a MAC protocol is to control the interference and competition among users while optimizing overall system performance and avoiding pitfalls. The MAC protocol is thus responsible for the quality of service experienced by the LAN/MAN users and, hence, it is the critical algorithm for determining the ability of a network technology to support, in an efficient and fair way, both real-time and non-real-time traffic [10].

As said before, RT-Ring belongs to the slotted-ring family. This class includes protocols such as MetaRing [28] and Cambridge Ring [31]. In these protocols, 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. The length of a slot can be expressed in several ways: the number of bits that can be transmitted into that slot, for example, b, or the time it takes to transmit all the bits contained in a slot, for example, $t_{\rm slot}$, or (in other words) the time interval between the arrival to a station of the first and last bit related to a slot, etc. For the purpose of this paper, it is convenient to associate to a slot a time duration: $t_{\rm slot} = b/v$, where $t_{\rm slot} = b/v$ is the speed of the transmission channel (expressed in bits per second). In this way, $t_{\rm slot}$ is the length of a slot in seconds and it is a function of the channel speed and number of bits in the slot. For ease of presentation, and to provide general results that are not implementation dependent, hereafter, we normalized 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. If we wish to express these quantities in seconds we simply have to multiply their value per $t_{\rm slot}$.

As the channel is slotted, before transmitting the messages generated by higher level protocols, a segmentation procedure is applied at the transmitting side. The segmentation procedure subdivides a message into several packets, where each packet can be transmitted into a slot. At the receiving side the reverse procedure, reassembling, is applied to reconstruct the original message (before it is delivered to the higher layers) from the relieved packets. Segmentation and reassembling are normally implemented by protocols that operate on top of the network technology. Examples of these protocols are the adaptation protocols in the ATM architecture. Hereafter, we assume that the traffic arriving at an RT-Ring station for transmission is subdivided in blocks, where a block can be transmitted into one slot, and we will use the word packet and message interchangeably.

In principle, stations can transmit in all the empty slots they observe and, hence, more stations can *concurrently access* the network.

Each RT-Ring station has (at least) two local queues in which it stores packets ready for transmission: one for the real-time

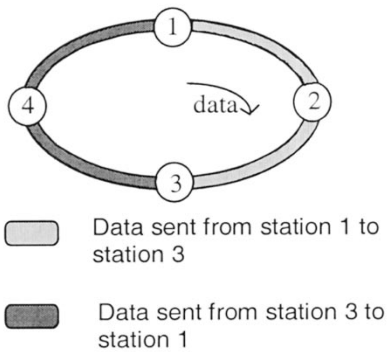

Fig. 2. Starvation scenario.

traffic and the other for non-real-time traffic. The real-time traffic has the highest transmission priority.

As we already stated, one of the main characteristics of RT-Ring is that it uses the spatial reuse policy, i.e., the packets travel on the network only from the source to the destination. This implies that the destination station changes from busy to empty the status of each slot containing packet addressed to it.

Spatial reuse is a concept used in ring networks to increase the overall aggregate bandwidth of the ring. This is possible because traffic is only passed along the ring between source and destination nodes rather than the whole ring as in earlier ring-based protocols such as token ring and FDDI.

Unfortunately, with spatial reuse policy arises (if coupled with concurrent network access) a new problem: starvation. By starvation we mean that some stations can never access the network because they are always covered by upstream traffic. In Fig. 2, we show a possible starvation scenario. The four stations in the ring has real-time traffic to transmit; Station 1 sends its traffic to Station 3, and Station 3 sends its traffic to Station 1. Due to the spatial reuse, Station 3 uses the slots previously used by Station 1. In such a scenario, Station 2 and Station 4 can never access the ring because they observe the ring as being always busy. For this reason, they are said to be in *starvation*.

Needless to say, this is unacceptable for a real-time protocol, since each station must be able to transmit its own real-time traffic. To solve this starvation problem, a fairness algorithm has to be used. In the following, we present the fairness control mechanism we use in RT-Ring.

B. Fairness Algorithm and Integration Mechanism

As we already stated, spatial reuse and concurrent access may lead to starvation. To avoid this problem, a fairness algorithm should be used. In fact, a fairness algorithm must ensure to all stations the same opportunity to access the network. Several fairness algorithms have been proposed in the literature. Magnet [24], Orwell [14], and ATMR [29], [3], [8], [4], [5] are some of these proposals.

Briefly, fairness algorithms can be divided into two categories: *global* and *local*. Global fairness algorithms view the ring as a single shared communication resource, while local fairness algorithms view the ring as a multiplicity of communication resources (i.e., all the links between stations). As highlighted by Chen *et al.* [4], both approaches have positive and negative aspects. For this reason, we provide RT-ring with a fairness algorithm that can be considered a hybrid between local and global fairness algorithms. In fact, RT-Ring accesses the network using both local and global information.

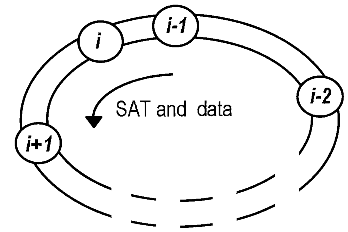

Fig. 3. Network scenario.

Global information is provided by a control signal, named SAT, that circulates in the ring in the same direction of data traffic (Fig. 3). The SAT can be represented by a bit pattern in the slot header as the token of FDDI or it can be a separate message inserted in an arbitrary position in the data packet, as for the MetaRing protocol [7]. In both cases, there is no need of using additional slots, and in this way the introduced overhead is comparable to the one of FDDI (for the token implementation) and it is equal to the one of MetaRing (for the SAT implementation). Since the implementation of a control signal has been extensively analyzed in [7], we do not present it here, but we refer the readers to [7] for further details.

Although the RT-Ring fairness mechanism uses some characteristics of the fairness mechanism used in *MetaRing* [4], [5], [7], [28], namely, the SAT mechanism, there are fundamental differences between these two mechanisms that will be discussed in the next section.

During every rotation, the SAT provides a predefined number of transmission authorizations to each station. The number of these authorizations is defined by two local parameters (l and k). 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, RT-Ring provides real-time traffic with higher priority than the non-real-time traffic.

Note that the SAT does not 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, for example, i, is said to be *satisfied* if it has no real-time traffic ready to be transmitted, or if between two consecutive SAT visits it has transmitted a predefined quota of real-time packets, denoted with l_i (one of the local parameters).

Conversely, a station, for example, i, is said to be *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.

Note that, if we denote two consecutive SAT arrivals at the same station as a *cycle*, this mechanism ensures the transmission of l real-time packets (if any) during each cycle. It also avoids the starvation problem, since after sending a maximum of $k + l$ packets, a station stops its transmission until the next SAT round. In Section III, we prove that this mechanism provides network guarantees to the applications. To better clarify this mechanism, in the following section, we present the algorithms used to handle the SAT and to control the transmission into the network.

1) Algorithms: In this section, we describe the fairness and the integration algorithms in details. Each station uses two local counters to count the transmitted packets: one for the real-time packets (RT_PCK), and one for the non-real-time packets (NRT_PCK). These counters are cleared every time the SAT leaves the station.

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.

C. Differences Between MetaRing and RT-Ring Fairness Algorithms

In this section, we highlight the differences between the fairness mechanism used in RT-Ring and the fairness mechanism used in *MetaRing* [4], [5], [7], [28]. In MetaRing, each station has two queues: one for the synchronous and one for the asynchronous traffic. Packets from the asynchronous queue are transmitted only if the synchronous queue is empty.

Whenever a station observes an empty slot, it can always transmit the synchronous traffic. Before transmitting the asynchronous packets a station must collect authorizations. Specifically, asynchronous transmissions are authorized by a control signal, called SAT (from SATisfied). Whenever a station receives the SAT signal, it performs different actions depending on its status. When a station receives the SAT, it can be in the *satisfied state* or *not-satisfied state*. A station is in the satisfied state if either between two visits of the SAT signal the station has transmitted at least l packets or its output (asynchronous) buffer is empty. When a station receives the SAT and it is satisfied it forward the SAT signal upstream without any delay. On the other hand, a not-satisfied station will hold the SAT until it is satisfied, and then it will forward the SAT signal upstream.

After a station forwards the SAT, it can send up to k ($k \geq l$)[1] additional asynchronous packets before receiving and forwarding again the SAT signal.

To avoid that asynchronous traffic may excessively delay the transmission of the synchronous traffic a mechanism is included in the protocol to disable the asynchronous traffic transmission whenever a station has a backlogged synchronous traffic. Synchronous traffic is considered to be backlogged if it has been waiting in the synchronous queue for more than a predefined time threshold $Thres$. To enable/disable the asynchronous traffic transmission, the ASYNChronous ENable (ASYNC-EN) control signal is used. A complete description of MetaRing can be found in [4], [5], [7], and [28].

On the other hand, as explained before, in RT-Ring only one signal, namely, the SAT, is used. This signal circulates in the same direction of the data traffic and controls the transmissions' authorizations for both the real-time and the non-real-time traffic.

However, each station may affect the SAT behavior only checking the number of real-time packets transmitted since the previous SAT visit (i.e., a station seizes the SAT only if the number of the *real-time* packets transmitted since the previous SAT departures is smaller than l). Hence, the SAT behavior is not affected by the non-real-time traffic.

With this mechanism, a station can transmit a real-time packet only when the real-time output buffer is not empty and if, since the previous SAT visit, it has transmitted a number of real-time packets smaller than l. Similarly, a station can transmit a non-real-time packet only when the non-real-time output buffer is not empty and if, since the previous SAT visit, it has transmitted a number of non-real-time packets smaller than k.

To summarize, the main differences between the MetaRing SAT fairness algorithm and RT-Ring are the following.

1) In MetaRing, the SAT signal only controls the transmission of non-real-time (asynchronous in the MetaRing notation) traffic. For this reason, in MetaRing, an upper bound to the network access time cannot be provided using the SAT. In RT-Ring the SAT controls both the transmission of real-time and non-real-time traffic and the SAT is used to provide an upper bound to the network access time.
2) In MetaRing, a station is always authorized to transmit its synchronous (i.e., real-time) traffic provided that "enough bandwidth" is reserved for it, see [28]. It is worth noting that in [9] it has been proved that in some cases MetaRing fails to satisfy the deadline constraints of the real-time

[1]The values of the parameters k and l may differ from station to station.

traffic. In RT-Ring a station can transmit not more than l real-time packets and not more than k non-real-time packets during each SAT round.

3) Finally, in addition to the SAT, MetaRing uses a second signal, named ASYNC-EN, to integrate real-time and non-real-time traffic. RT-Ring has only one control signal, namely, the SAT.

As in our case the transmission of the real-time traffic is controlled by the SAT, to guarantee that real-time traffic can be delivered within its deadline, for RT-Ring we have designed a SAT management algorithm to guarantee that the real-time traffic constraints are satisfied. Specifically, in this paper we formally prove: 1) an upper bound to the SAT circulation time, and by exploiting this property 2) an upper bound of the time a real-time packet waits in the station transmission queue before its transmission.

III. RT-Ring Properties

In this section, we derive some RT-Ring properties that are necessary for a real-time protocol; in particular, we prove the presence of an upper bound to the network access time. As we already stated, this bound is a fundamental requirement for a real-time protocol. In fact, it represents the maximum time a station must wait before transmitting a packet into the network.

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. By guaranteeing that the application timing constraints are satisfied assuming the upper bound of the network access time, we can guarantee (i.e., with probability 1) the timing correctness property of the application.

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 to the real-time application [6], [36].

In the following, we derive the upper bound to the network access time in three steps: first we derive an upper bound to the SAT rotation time (since a station can transmit only if it has received authorizations from the SAT) and then we generalize the SAT-bound result by providing a bound to n SAT rotations. The latter value is useful both for deriving a bound on the waiting time of a packet in the network queue, and to implement real-time bandwidth allocation schemes (see, for example, [1] and [35]). Finally, using the SAT bounds, we derive an upper bound to the network access delay.

In the following analysis we consider a slotted ring with S circulating slots and N stations; the time factor is normalized to the slot unit (i.e., one time unit is equal to one slot) and packet size is not greater than the slot size. In order to compute our analysis, we also assume that the network is free from hardware or software failures.

A. 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 will be used to derive the upper bound to the network access time. First, we note that SAT_TIME is affected by three possible components, as we explain in the following.

1) First is the number of the stations, denoted with N, present in the ring. Specifically, in the following theorems and lemmas, the impact of the number of stations on the SAT rotation time is represented by a summation on all the station numbers, say j ($1 \leq j \leq N$), of the maximum number of packets that each station can transmit, i.e., l_j real-time packets and up to k_j non-real-time packets.
2) Next is 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. In fact, S represents the SAT rotation time when the SAT signal freely travels into the network.
3) Last is the time the SAT is held at the not-satisfied stations.

To compute the latter quantity it is useful to introduce the following definitions and propositions.

Definition: We define as $SAT\_Cycle_{a,\,i}^{(n)}$ ($SAT\_Cycle_{d,\,i}^{(n)}$) the time interval between the $(n-1)$th and nth arrival (departure) of the SAT at (from) station i.

Proposition 1: When a not-satisfied station i holds the SAT during the nth visit of the SAT at this station, the busy slots it observes containing only packets whose transmissions have been authorized during $SAT\_Cycle_{d,\,i}^{(n)}$.

Proof: Let us prove this proposition by contradiction. We assume that after the SAT there is a packet transmitted by a station j whose transmission was authorized during $SAT\_Cycle_{a,\,i}^{(n-x)}$ ($x = 1, 2, \ldots$). This is clearly not possible because if the packet is a real-time packet then the station j is not satisfied when it receives the SAT during $SAT\_Cycle_{a,\,i}^{(n)}$ and therefore it will transmit that packet before releasing the SAT and, hence, the packet arrives at station i before the SAT.

On the other hand, authorizations for non-real-time packets can be used for packet transmissions only inside the SAT round in which a station gets the authorizations. In fact, authorizations for non-real-time packets are lost when the SAT comes back to the station. $\diamond$

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:

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

Proof: First, we focus on the delay that a station can add to the SAT rotation time. Let us denote with $A_i^{(n)}$, the number of packets whose transmission is authorized at station i during the nth visit of the SAT.

Station i, at the nth SAT visit, can add a delay, denoted with $Del_i^{(n)}$, to the SAT rotation time equal to the number of busy slots it observes while holding the SAT. By taking into account Proposition 1, the station i delay at the nth SAT visit is lower or equal to

$$Del_i^{(n)} \leq l_i^{(n-1)} + A_{i+1}^{(n-1)} + A_{i+2}^{(n-1)} + \cdots + A_N^{(n-1)} + A_1^{(n)} + A_2^{(n)} \cdots + A_{i-1}^{(n)}$$

where $l_i^{(n-1)}$ is the maximum number of packets that station i has to transmit while holding the SAT at the nth visit, i.e., the number of real-time packets whose transmissions were authorized at the $(n-1)$th SAT visit at station i. The other quantities represent the maximum number of busy slots that station i may observe while holding the SAT during the nth visit.

Similarly, station $i+1$, at the nth SAT visit, can add a delay to the SAT rotation time lower or equal to

$$Del_{i+1}^{(n)} \leq l_{i+1}^{(n-1)} + A_{i+2}^{(n-1)} + A_{i+3}^{(n-1)} + \cdots + A_N^{(n-1)} + A_1^{(n)} + A_2^{(n)} \cdots + A_{i-1}^{(n)}.$$

Finally, station $i-1$, at the $(n+1)$th SAT visit, can add a delay to the SAT rotation time equal to

$$Del_{i-1}^{(n+1)} \leq l_{i-1}^{(n)} + A_i^{(n)} + A_{i+1}^{(n)} + \cdots + A_N^{(n)} + A_1^{(n+1)} + A_2^{(n+1)} \cdots + A_{i-2}^{(n+1)}.$$

By summing the delays upper bound and by counting each packet only once (a busy slot can cause a delay in one station only), we have

$$\begin{aligned} &Del_i^{(n)} + Del_{i+1}^{(n)} + \cdots + Del_N^{(n)} \\ &\quad + Del_1^{(n+1)} + \cdots + Del_{i-1}^{(n+1)} \\ &\leq l_i^{(n-1)} + A_{i+1}^{(n-1)} + A_{i+2}^{(n-1)} + \cdots \\ &\quad + A_N^{(n-1)} + A_1^{(n)} + A_2^{(n)} \cdots + A_N^{(n)} \\ &\quad + A_1^{(n+1)} + A_2^{(n+1)} \cdots + A_{i-2}^{(n+1)} \end{aligned}$$

that is equal to

$$2 \cdot \sum_{j=1}^{N} (l_j + k_j) - k_i - (l_{i-1} + k_{i-1}).$$

By considering the times it takes the SAT to complete one rotation (S time units), (1) holds.

By applying the same line of reasoning, we can prove that the maximum time that elapses between the nth SAT departure from station i and the $(n+1)$th SAT departure from station $i+1$ is

$$S + l_{i+1}^{(n-1)} + A_{i+2}^{(n-1)} + \cdots + A_N^{(n-1)} + A_1^{(n)} + A_2^{(n)} \cdots + \cdots A_N^{(n)} + A_1^{(n+1)} + A_2^{(n+1)} \cdots + A_{i-1}^{(n+1)} + l_i^{(n+1)}$$

that is equal to

$$S + 2 \cdot \sum_{j=1}^{N} (l_j + k_j) - k_i - k_{i-1}.$$

Therefore, we can say that between two consecutive arrivals (departures) of the SAT from the same station, SAT_TIME has an upper bound, and (1) holds. ◇

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 two consecutive SAT arrivals at the same station has an upper bound equal to

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

Proof: It follows from the previous theorem ◇

Proposition 3: The number of circulating slots (i.e., S) represents the ring latency, as it is the time necessary for the SAT to perform one complete rotation when no traffic is present (i.e., $l = k = 0$ for each station in the ring). ◇

Proof: It immediately follows from (1) when no packet transmission is authorized, i.e., $l = k = 0$. ◇

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

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

Proof: A formal proof follows from Theorem 1.

As the complete proof is simple but quite long, hereafter we will just summarize it. Each SAT cycle may introduce both a delay equal to the SAT rotation, i.e., S, plus the delay due to the SAT holding time. The latter is determined by the number of busy slots observed by a station while holding the SAT. This signal, during each rotation, can give up to A_i authorizations to each station i. Considering n SAT rotations (from p to $n+p$), the maximum number of authorizations is given by: $n \cdot \sum_{j=1}^{N} A_j$. Some of the authorized packets cannot be transmitted in the same round they obtained the authorizations for. For instance, packets that have received the authorizations in the $(p-1)$th cycle $(A^{(p-1)})$ may be transmitted in the next rotation. Hence, the number of transmissions in n rotations can be $(n+1) \cdot \sum_{j=1}^{N} A_j = \sum_{j=1}^{N} (k_j + l_j)$.

Based on the previous considerations, the upper bound to n consecutive SAT arrivals is given by (3). ◇

Proposition 4: 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

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

Proof: Follows from Theorem 2. ◇

Proposition 5: The average SAT rotation time $E[SAT\_TIME]$ is equal to

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

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

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

B. 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.

Proposition 6: The maximum time that elapses between the authorization and the transmission of a real-time packet is equal to $\max\{SAT\_Cycle_d\}$, i.e., the interval between two consecutive departures of the SAT from the same station.

Proof: The SAT provides authorizations when it leaves a station. If the station catches enough empty slots before the next SAT arrival, then the station is able to transmit its authorized packets. Otherwise, the station, at the next SAT arrival, will hold the SAT and will complete the transmission of the authorized packets before releasing the SAT. ◇

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:

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

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

Proof: The authorization for being transmitted, is given by the SAT signal: up to l_i real-time authorizations every time the SAT leaves a station.

The packet, say P_{x+1}, arrives at the queue, where P_1, P_2, P_3, …, $P_x$packets are already present in the output queue. P_1 is the packet that will be transmitted first.

The packet P_{x+1} will receive the authorization only after $P_1, \ldots, P_x$ have been authorized. The authorizations to $x+1$ packets are provided in $\lceil (x+1)/l_i \rceil$ SAT rounds.

Proposition 6 states that an authorized packet has to wait no more than one SAT round for being transmitted. Hence, (6) holds. ◇

C. Real-Time Allocation Bandwidth

In order to provide real-time communications, a real-time bandwidth allocation scheme is essential as well as the properties we just described. In fact, a wrong allocation may lead the protocol to violate the timing requirements of the traffic [1], [16]. In this paper, we do not propose any bandwidth allocation scheme, as several studies have been focused on finding efficient bandwidth allocation schemes. For instance, Agrawal *et al.* [1] and Zhang and Burns [35] propose efficient schemes to allocate the bandwidth over an FDDI network. These schemes exploit the relationship between the reserved bandwidth in a cycle (e.g., l_i in RT-Ring) and the packet waiting time. These and other schemes [12] can use the previous properties, in order to efficiently allocate the bandwidth inside RT-Ring.

IV. Efficiency of RT-Ring

LANs and MANs rely on a common transmission medium, hence, the transmissions of the network stations must be coordinated by the MAC protocol. This coordination can be achieved by means of control information that is carried explicitly by control messages traveling along the medium (e.g., Token, ACK messages), or can be provided implicitly by the medium itself using the carrier sensing to identify the channel being either active or idle. Control messages, or message retransmissions due to collision, remove channel bandwidth from that available for successful message transmission. Therefore, the fraction of channel bandwidth used by successfully transmitted messages gives a good indication of the overheads required by the MAC protocol to perform its coordination task among stations. This fraction is known as the utilization of the channel, and the maximum value it can attain is known as the capacity of the MAC protocol [10].

As our interest is to measure the maximum fraction of the channel bandwidth that can be used to deliver the user data, the efficiency analysis is performed in asymptotic conditions, i.e., all network stations always have segments to transmit (see [10], [32], and the references therein).

In this section, we analyze the RT-Ring capacity comparing it with the FDDI protocol [2], as FDDI is a well-investigated protocol and it is suitable for providing a high-speed communication subsystem for a distributed real-time system (see, for example, [1], [22], [25], [26], [13], [35], and [16]).

Before going into the analytical comparison of the two protocols, we highlight the main difference between FDDI and RT-Ring. This difference lies in the network access, which is concurrent in RT-Ring and sequential in FDDI. The concurrent network access mechanism, coupled with spatial reuse policy, allows RT-Ring to increase the throughput beyond the link capacity. As we already stated, if N stations are present, each of them with full load (i.e., always traffic to transmit), under uniform destination distribution, the average distance for a packet to travel is $N/2$ hops, producing a spatial reuse factor of two (i.e., the same slot can be used twice during one round trip).

It is to note that the spatial reuse gain depends on the traffic addressing. To make our comparison the most general as possible, we will perform the comparison by assuming that in RT-Ring the spatial reuse never occurs (i.e., each busy slot must complete one ring rotation in order to reach the destination station). This implies that the RT-Ring capacities, which we compute in the following, are the lower bounds of the RT-Ring capacities under normal traffic addressing.

In the following, we first analyze the real-time and the non-real-time (usually referred to as synchronous and asynchronous in FDDI studies) capacities, and then we analyze the global capacity (i.e., when both real-time and non-real-time traffic circulate in the network) achieved by both protocols.

Note that, since FDDI does not use a slotted ring, hereafter to compare it with RT-Ring we normalize the FDDI capacity with respect to the slot unit. The comparison is performed by considering the number of transmitted bits without considering the impact of the overheads due to headers. For this reason and since we are interested in how long a station can transmit, we can assume without loss of generality that FDDI frames have a constant length that is equal to the RT-Ring slot size.

Specifically, the relevant quantities of FDDI are as follows [10].

- $TTRT$: This is a network parameter that defines the target time for a token rotation. It is expressed in time units (let us assume seconds).
- $Ring\_Latency$: This is the time it takes to the token to complete a ring rotation in a idle ring, i.e., when no traffic is transmitted. It is expressed in time units (let us assume seconds);
- F: This is the frame transmission time, i.e., the number of bits in an FDDI frame divided by the channel speed.

As we have stated before, for comparison purposes we assume that an FDDI frame is equal to an RT-Ring packet, this means that $F = t_{\mathrm{slot}}$, i.e., the slot unit used throughout this paper. Hence, $a = Ring\_Latency/F$ is the ring latency normalized to the slot duration. Therefore, if the rings of FDDI and RT-Ring have the same length: $a = S$. It is known from the FDDI literature (see [10] and the references therein) that $TTRT - Ring\_Latency$ is the maximum amount of time the FDDI stations can be transmitting during a token rotation. Hence, $h = (TTRT - Ring\_Latency)/F$ is the maximum number of packets transmitted by the FDDI stations during a token rotation.

The above relationships will be used in the following to compare the FDDI and RT-Ring capacities, where the RT-Ring and the FDDI capacities are derived using the same pattern traffic.

A. Real-Time Capacity

The real-time capacity, ρ^{RT}, is computed by assuming that every station has always real-time traffic to transmit and zero non-real-time traffic. To compute the real-time capacity we need to introduce the following results.

Proposition 7: Let us focus on a station j that receives the nth SAT visit, if all stations are always satisfied before this time instant (i.e., at the SAT visits before this time instant), then the busy slots observed by station j during $SAT\_Cycle_{a,j}^{(n)}$ correspond to packets whose transmission has been authorized during $SAT\_Cycle_{a,j}^{(n-1)}$.

Proof: If a station is satisfied, all its transmission occur in the time interval between the departure of the SAT that authorizes these transmissions and the next SAT arrival at this station. In these conditions, a busy slot never overtakes the SAT that has authorized its transmission. For this reason, a packet whose transmission (i.e., the corresponding busy slot) is observed by a station j between the $(n-1)$th arrival and the nth arrival must have been authorized between the $(n-1)$th and $(n-2)$th departure of the SAT from station j, i.e., packets whose transmission has been authorized during $SAT\_Cycle_{a,j}^{(n-1)}$. In fact, these packets will be observed by station j only after the $(n-1)$th SAT visit and before the nth SAT visit. ◇

Lemma 1: In a network with N active stations transmitting only real-time traffic with $S \geq \sum_{j=1}^{N} l_j$, by assuming asymptotic conditions, the following holds.

1) $SAT\_TIME = S$.
2) $\rho^{RT} = (\sum_{j=1}^{N} l_j)/S$.

Proof: Let us prove point 1) by contradiction. Let j be the first station that receives a SAT with a delay greater than S (i.e., $SAT\_TIME_j > S$). Let us assume that this first delayed SAT cycle occurs when the SAT arrives for the nth time at station j (i.e., $SAT\_Cycle_{a,j}^{(n)} > S$).

Station j is the first station to observe a delayed SAT. This means that when the SAT arrives at station $j-1$ for the nth time, station $j-1$ is not satisfied and it seizes the SAT. If station $j-1$ is not satisfied at the nth SAT visit, this implies that it has not been able to transmit the packets authorized at $(n-1)$th SAT visit. This occurs if after the $(n-1)$th SAT departure, station $j-1$ observes more than $S-l_{j-1}$ busy slots. Note that the busy slots it observes between the $(n-1)$th and nth SAT visit are only those packets whose transmission was authorized before the $(n-1)$th SAT visit at station $j-1$.[2]

Furthermore, as station $j-1$ at the nth SAT visit is the first not-satisfied station, it follows from Proposition 7 that the busy slots observed by station $j-1$ during $SAT\_Cycle_{a,j-1}^{(n)}$ only contain packets whose transmissions were authorized during $SAT\_Cycle_{a,j-1}^{(n-1)}$, that is,

$$A_j^{(n-2)} + \cdots + A_N^{(n-2)} + A_1^{(n-1)} + A_2^{(n-1)} \cdots + A_{j-2}^{(n-2)} \equiv \tilde{A}.$$

By definition, the above quantity is not greater than $\sum_{i=1}^{N} l_i - l_{j-1}$. Hence, $S - \tilde{A} \geq l_{j-1}$. This implies that during the nth cycle station $j-1$ must observe at least l_{j-1} empty slots and hence it must be satisfied at the nth SAT visit. From this the absurd follows and, hence, point 1) is proved.

The prove of point 2) immediately follows from point 1). In asymptotic conditions, each station always transmits the entire quota of real time packets and, hence, the number of packets transmitted in each SAT rotation is equal to $\sum_{j=1}^{N} l_j$. Hence,

$$\rho^{RT} = \frac{\sum_{j=1}^{N} l_j}{SAT\_TIME} = \frac{\sum_{j=1}^{N} l_j}{S}.$$

◇

Lemma 2: In a network with N active stations transmitting only real-time traffic, by assuming asymptotic conditions, the

[2]If a packet transmission is authorized after the $(n-1)$th SAT departure from station $j-1$, the busy slot containing this packet cannot arrive at station $j-1$ before the nth SAT visit. Remember that (by assumption) station $j-1$ at the nth SAT visit is the first not-satisfied station and, hence, before that time the SAT freely circulates in the ring.

real-time protocol capacity ρ^{RT} satisfies the following relationship:

$$\frac{1}{1+S\Big/\sum\limits_{j=1}^{N} l_j} \le \rho^{RT} \le 1. \tag{7}$$

Proof: First, we prove the upper bound to the protocol capacity. Let us consider a network in which the number of circulating slots S is equal to $\sum_{j=1}^{N} l_j$. By applying Lemma 1, it follows that $\rho^{RT} = 1$.

To derive the lower bound on the protocol capacity we focus on the worst case characterized by SAT rotation cycles with average length equal to the upper bound value [(5)].

Under the assumption that all the stations operate in asymptotic conditions, the number of packets transmitted during each SAT rotation is constant (i.e., l_i for each station i). Hence, we compute the ratio between $\sum_{j=1}^{N} l_j$ and $E[SAT\_TIME_i]$, which is the upper bound on the average cycle length when only synchronous traffic is present in the network, whose value is given by Proposition 5 with $k_j = 0$ for each j. This ratio leads to (7). ⋄

It is easy to verify that the real-time capacity of RT-Ring is greater or equal than the FDDI real-time capacity (in the literature known as synchronous capacity). In fact, as described in [10], the real-time capacity of FDDI is equal to $(TTRT - Ring\_Latency)/TTRT$ and can be expressed (in order to compare it with the RT-Ring capacity), as follows.

By exploiting the notations introduced at the end of Section IV, and assuming $F = t_{\rm slot}$, with some algebraic manipulations, the FDDI capacity can be written as: $\rho^{RT}_{FDDI} = 1/(1+a/h)$, where $a = S$, and h corresponds to the maximum number of real-time packets that can be transmitted by the FDDI station in a token rotation, i.e., $h = \sum_{j=1}^{N} l_j$. Hence, the real-time protocol capacity of FDDI is equal to the lower bound of the RT-Ring real-time protocol capacity [(7)].

B. Non-Real-Time Capacity

To compute the non-real-time capacity, ρ^{not_RT}, we assume that every station always has non-real-time traffic to transmit (*asymptotic conditions*) while no real-time packets are ready for transmission. Under these hypotheses, the non-real-time capacity is defined by Lemmas 3 and 4.

Lemma 3: In a network with N active stations transmitting only non-real-time traffic, by assuming asymptotic conditions, the following holds.

1) $SAT\_TIME = S$.
2)

$$\rho^{not\_RT} = \frac{\min\left(\sum\limits_{j=1}^{N} k_j,\, S\right)}{SAT\_TIME} = \frac{\min\left(\sum\limits_{j=1}^{N} k_j,\, S\right)}{S}.$$

Furthermore, if $\sum_{j=1}^{N} k_j \le S$ then

3) each station j has a throughput equal to k_j/S.

Proof: The proof of point 1) is immediate as with no real-time traffic, all the stations are always satisfied. To prove point 2), it is sufficient to note (see Proposition 7) that the busy slots observed by a station during the nth SAT cycle contain packets that have been authorized by the SAT between the $(n-2)$th and $(n-1)$th visit at that station: $\sum_{j=1}^{N} k_j$. Furthermore, there can be no more than S busy slots between two consecutive SAT arrivals at a station. Hence, the number of transmissions in an SAT cycle is lower bounded[3] by $\min(\sum_{j=1}^{N} k_j,\, S)$. The proof of point 3) immediately follows from point 2). ⋄

Lemma 4: In a network with N active stations transmitting only non-real-time traffic, by assuming asymptotic conditions, the protocol capacity, ρ^{not_RT}, satisfies the following relationship

$$\frac{1}{S/\min\left(\sum\limits_{j=1}^{N} k_j,\, S\right)} \le \rho^{not\_RT} \le 1. \tag{8}$$

Proof: First, we prove the upper bound to the non-real-time protocol capacity. If no real-time traffic is present, the SAT travels freely into the network. Hence, it gives k new authorizations to each station every S slot time. Let us consider a network where $\sum_{j=1}^{N} k_j = S$. By applying Lemma 3, it follows that $\rho^{not\_RT} = 1$.

The lower bound of ρ^{not_RT} can be derived as follows.

Since, no real-time traffic is present, the average SAT rotation [(5)] is upper bounded by $S + \sum_{j=1}^{N} k_j$, while according to point 2i) of Lemma 3 $\min(\sum_{j=1}^{N} k_j,\, S)$ is the amount of non-real-time traffic transmitted during an SAT rotation. Hence, the percentage of time the channel is used to transmit non-real-time traffic is greater or equal to the lower bound of Equation (8). ⋄

Once again, as described in [10], the non-real-time (asynchronous) capacity of FDDI is equal to $(N \cdot (TTRT - Ring\_Latency))/(N \cdot TTRT + Ring\_Latency)$ and, to compare it with the RT-Ring capacity, can be expressed as follows.

Again, by exploiting the notations introduced at the end of Section IV, and assuming $F = t_{\rm slot}$, with some algebraic manipulations, the non-real-time capacity of FDDI is $\rho^{not\_RT}_{FDDI} = 1/(1+a/h)$, where $a = S$, and h corresponds to the maximum number of non-real-time packets that can be transmitted by the FDDI station in a token rotation, i.e., $h = \sum_{j=1}^{N} k_j$.

For the capacity comparison, we have to distinguish two cases.

1) $S \le \sum_{j=1}^{N} k_j$. In this case, the RT-Ring capacity is optimal, i.e., 1, hence, it is always much better than that of FDDI.
2) $S > \sum_{j=1}^{N} k_j$. In this case, the RT-Ring capacity is $\rho^{not\_RT} = 1/(S/\sum_{j=1}^{N} k_j)$.

By noting that $\sum_{j=1}^{N} k_j$ is the maximum number of non-real-time packets transmitted during an SAT cycle, this quantity corresponds to h. By remembering that $S = a$, it follows that $\rho^{not\_RT} = 1/(a/h) > 1/(1+a/h) = \rho^{not\_RT}_{FDDI}$. Hence, also in this case the RT-Ring capacity is better than that of FDDI.

[3]Here, we are not considering the spatial reuse policy.

C. Global Capacity

Since RT-Ring can operate with both types of traffic, we now compute the protocol capacity, ρ, assuming both asymptotic real-time and non-real-time traffic conditions.

Proposition 8: In a network with N active stations transmitting both real-time and non-real-time traffic, by assuming asymptotic conditions, the global protocol capacity, ρ, satisfies the following relationship:

$$\rho \geq \frac{\sum_{j=1}^{N} l_j + E\left[\sum_{j=1}^{N} T_j^{not\_RT}\right]}{S + \sum_{j=1}^{N} l_j + E\left[\sum_{j=1}^{N} T_j^{not\_RT}\right]} \tag{9}$$

where $E[T_j^{not\_RT}]$ is the average number of non-real-time packets transmitted by station j between two consecutive SAT arrivals. Needless to say that $E[T_j^{not\_RT}] \leq k_j$.

Proof: In asymptotic conditions each station j transmits the entire quota of real-time packets, l_j. In addition, (by definition) in average each station j will transmit $E[T_j^{not\_RT}]$ non-real-time packets. Hence, the numerator of (9) is the average number of packets transmitted in each SAT cycle. The second step of the proof corresponds to prove that the denominator of (9) represents an upper bound to the average length of a SAT cycle. This can be easily proved by following the line of reasoning used to prove the SAT_TIME and the $E[SAT\_TIME]$ in Theorem 1 and in Proposition 5, respectively. In fact, it is sufficient to repeat those proofs by replacing k_j (i.e., the maximum number of non-real-time transmissions of station j during an SAT cycle) with $T_j^{not\_RT}$ (i.e., the real number of non-real-time transmissions of station j during an SAT cycle). ⋄

Proposition 9: In a network with N active stations transmitting both real-time and non-real-time traffic and with $\sum_{j=1}^{N}(l_j + k_j) \leq S$, by assuming asymptotic conditions, the following holds.

1) $SAT\_TIME = S$.
2)

$$\rho = \frac{\sum_{j=1}^{N}(l_j + k_j)}{SAT\_TIME} = \frac{\sum_{j=1}^{N}(l_j + k_j)}{S}.$$

Proof: The proof of point 1) is obtained by extending the proof of point 1) of Lemma 1, taking into consideration that in each visit the SAT authorizes a station j to transmit k_j packets in addition to l_j packets. From that proof, it also results that in each cycle a station j observes at least $l_j + k_j$ empty slots and, hence, it will transmit all its authorized packets (note that stations operate in asymptotic conditions and, hence, they always have packets ready for transmission). Hence, the numerator of 2) is the number of transmissions performed by the stations for each SAT cycle while the denominator of 2) is the SAT-cycle length. ⋄

Lemma 5: In a network with N active stations transmitting both real-time and non-real-time traffic, by assuming asymptotic conditions, the protocol capacity ρ satisfies the following relationship:

$$\frac{1}{1 + S \Big/ \left(\sum_{j=1}^{N} l_j + E\left[\sum_{j=1}^{N} T_j^{not\_RT}\right]\right)} \leq \rho \leq 1. \tag{10}$$

Proof: The upper bound of (10) follows from Proposition 9, with $\sum_{j=1}^{N}(l_j + k_j) = S$, while the lower bound is given by Proposition 8. ⋄

To compare the global capacity of RT-Ring with the global capacity of FDDI, we use the notations introduced at the end of Section IV. Further, we notice that in [10] the global capacity of FDDI can be expressed as $\rho_{FDDI} \leq 1/(1 + a/h)$, where h corresponds to the real-time frames that each station can transmit during a token rotation. In fact, in [10] it is shown that the upper bound to the FDDI capacity is obtained when the available bandwidth is completely used by real-time traffic.

Based on the previous consideration and since the average number of frames transmitted during a token rotation is equal to $\sum_{j=1}^{N} l_j + E[\sum_{j=1}^{N} T_j^{not\_RT}]$ and a is equal to S, it follows that

$$\rho_{FDDI} \leq 1 \Bigg/ \left(1 + S \Bigg/ \left(\sum_{j=1}^{N} l_j + E\left[\sum_{j=1}^{N} T_j^{not\_RT}\right]\right)\right).$$

Hence, the global capacity of FDDI is equal or lower than the lower bound of the RT-Ring global capacity [(10)].

V. Conclusion

Real-time traffic over packet switching networks has become essential to support QoS distributed applications that are more and more used in different scenarios: from automated factories to many embedded systems, from LANs to the Internet.

Several timed-token protocols can support real-time distributed applications, but they give rise to efficiency problems when the network dimension increases. In particular, the network bandwidth utilization is significantly reduced.

To overcome these efficiency problems, we designed a new real-time protocol, named RT-Ring, that can support both real-time and legacy (non-real-time) data applications over packet switching ring networks.

Our proposed protocol, RT-Ring, is provided with concurrent network access and with spatial reuse policy. These characteristics allow the protocol to achieve high network utilization. RT-Ring also provides transmission guarantees (proved with a worst case analysis) to real-time traffic. The worst case analysis has been used as it is the only way to derive the upper bound to the network access time and hence to provide guarantees to the real-time application [6], [36]. An interesting study would consist of a simulation analysis that exploits the bounds (obtained through the worst case analysis) using some typical traffic scenario. We are considering this study for future work.

Further, since connection among networks is very important, RT-Ring has the possibility of being connected with the emerging Differentiated Service Architecture [27]. However, since in [12] we already described how to connect RT-Ring with WANs where *diffserv* architectures are used, we referred the readers to [12] for further details.

We did not propose any bandwidth allocation scheme, as RT-Ring can use one of the efficient allocation schemes present in the literature (see, for example, [1] and [35]), but we presented some protocol properties, to facilitate the selection of one of the bandwidth allocation scheme present in the literature.

We then evaluated the performance aspects of RT-Ring: we compared its capacities (real-time, non-real-time, and global) with the corresponding FDDI protocol capacities and we proved that RT-Ring achieves protocol capacities higher than the FDDI protocol.

Real-time guarantees, better performance than FDDI, and compatibility with the Differentiated Service Architecture are characteristics which make RT-Ring a candidate for a protocol that is worth implementing in order to support QoS networking applications with timing requirements.

ACKNOWLEDGMENT

The authors wish to thank the referees for their helpful comments that contributed to significantly improve the paper quality.

REFERENCES

[1] G. Agrawal, B. Chen, W. Zhao, and S. Davari, “Guaranteeing synchronous message deadlines with the timed token medium access control protocol,” *IEEE Trans. Comput.*, vol. 43, pp. 327–339, Mar. 1994.

[2] *FDDI Token Ring Media Access Control*, Draft proposed, ASC X3T9.5, 1987.

[3] B. R. Bellur and G. H. Sasaki, “A SAT-based network access scheme for fairness in high speed networks,” *IEEE/ACM Trans. Networking*, vol. 5, pp. 371–381, June 1997.

[4] J. S. C. Chen, I. Cidon, and Y. Ofek, “A local fairness algorithm for gigabit LAN’s/MAN’s with spatial reuse,” *IEEE J. Select. Areas Commun.*, vol. 11, pp. 1183–1192, Oct. 1993.

[5] J. Chen, I. Cidon, and Y. Ofek, “A local fairness algorithm for MetaRing and its performance study,” in *Proc. GLOBECOM ’92*, vol. 3, 1992, pp. 1635–1641.

[6] D. Chen, V. C. S. Lee, and E. Chan, “On the ability of the FDDI-M protocol to support real-time traffic,” in *Proc. Fifth Int. Conf. Real-Time Computing Systems and Applications*, Oct. 1998, pp. 51–57.

[7] I. Cidon and Y. Ofek, “MetaRing—A full-duplex ring with fairness and spatial reuse,” *IEEE Trans. Commun.*, vol. 41, pp. 110–120, Jan. 1993.

[8] I. Cidon, L. Georgiadis, R. Guerin, and Y. Shavitt, “Improved fairness algorithms for rings with spatial reuse,” *IEEE/ACM Trans. Networking*, vol. 5, pp. 190–204, Apr. 1997.

[9] M. Conti, L. Donatiello, and M. Furini, “MetaRing+: An enhancement of the MetaRing access protocol for supporting real-time applications,” in *Proc. Eighth Euromicro Workshop Real-Time Systems*, June 1996, pp. 218–223.

[10] M. Conti, E. Gregori, and L. Lenzini, *Metropolitan Area Networks (MAN’s): Architectures, Protocols and Performance Evaluation*. London, U.K.: Springer-Verlag, 1997.

[11] M. Conti, L. Donatiello, and M. Furini, “Analysis of MetaRing: A real-time protocol for metropolitan area network,” Dept. Comput. Sci., Univ. Bologna, Bologna, Italy, Tech. Rep. UBLCS-98-5, 1998.

[12] ——, “Design and analysis of RT-Ring: A protocol for supporting real-time communications,” in *Proc. 3rd IEEE Workshop Factory Communications (WFCS 2000)*, Porto, Portugal, Sept. 2000, pp. 91–98.

[13] D. Dikeman and W. Bux, “Analysis and tuning of the FDDI media access control protocol,” *IEEE J. Select. Areas Commun.*, vol. 6, pp. 997–1010, July 1988.

[14] R. M. Falconer and J. Ladams, “Orwell: A protocol for an integrated services local network,” *Br. Telecom Technol. J.*, vol. 3, no. 4, pp. 27–35, Oct. 1985.

[15] D. Ferrari, “Client requirements for real time communication services,” *IEEE Commun. Mag.*, vol. 28, pp. 76–90, Nov. 1990.

[16] M. Hamdaoui and P. Ramanathan, “Deferring real-time traffic for improved nonreal-time communication in FDDI networks,” *IEEE Trans. Comput.*, vol. 44, pp. 1420–1428, Dec. 1995.

[17] *IEEE Standards for Local Area Network—Token-Passing Bus Access Method*, IEEE 802.4, 1984.

[18] *IEEE Standards for Local Area Network—Token-Ring Aaccess Method*, IEEE 802.5, 1986.

[19] Internet 2 project [Online]. Available: http://www.internet2.edu.

[20] L. Kelinrock and F. A. Tobagi, “Packet switching in radio channels: Part 1—Carrier sense multiple access modes and their throughput-delay characteristics,” *IEEE Trans. Commun.*, vol. COM-23, pp. 1400–1416, Dec. 1975.

[21] S. K. Kweon, K. G. Shin, and Q. Zheng, “Statistical real-time communication over ethernet for manufacturing automated systems,” in *Proc. IEEE Real-Time Technology and Applications Symp.*, June 1999, pp. 192–202.

[22] M. J. Johnson and K. C. Sevcik, “Cycle time properties of the FDDI token ring protocol,” *IEEE Trans. Software Eng.*, vol. SE-13, pp. 376–385, Mar. 1987.

[23] G. Le Lann, “The 802.3D Protocol: A variation on the IEEE 802.3 standard for real-time LANs,” INRIA, Sophia Antipolis, France, Tech. Rep., July 1987.

[24] A. A. Lazar, A. T. Temple, and R. Gidron, “MAGNET II: A metropolitan area network based on asynchronous time sharing,” *IEEE J. Select. Areas Commun.*, vol. 8, pp. 1582–1594, Oct. 1990.

[25] N. Malcom and W. Zhao, “The timed-token protocol for real-time communications,” *IEEE Computer*, vol. 27, pp. 35–41, Jan. 1994.

[26] ——, “Hard real-time communication in multiple access networks,” *Real-Time Syst.*, vol. 8, no. 1, pp. 35–77, 1995.

[27] K. Nichols, V. Jacobson, and L. Zhang, “A two-bit differentiated services architecture for the Internet,” Internet Draft, 1998.

[28] Y. Ofek, K. Sohraby, and H.-T. Wu, “Integration of synchronous and asynchronous traffic on the MetaRing and its performance study,” *IEEE/ACM Trans. Networking*, vol. 5, pp. 111–121, Feb. 1997.

[29] H. Ohnishi, N. Morita, and S. Suzuki, “ATM ring protocol and performance,” in *Proc. Infocom’89*, 1989, pp. 394–398.

[30] K. G. Shin and C. C. Chou, “Design and evaluation of real-time communications for FieldBus-based manufacturing systems,” *IEEE Trans. Robot. Automat.*, vol. 12, pp. 357–367, June 1996.

[31] W. Stallings, “Local networks,” *ACM Comput. Surv.*, vol. 16, no. 1, pp. 3–41, Mar. 1984.

[32] ——, *Local & Metropolitan Area Networks*. Upper Saddle River, NJ: Prentice-Hall, 1997.

[33] J. A. Stankovic and K. Ramamritham, *Advances in Real-Time Systems*. Los Alamitos, CA: IEEE Comput. Soc. Press, 1993.

[34] *The Cisco SRP MAC Layer Protocol*, RFC 2892, Aug. 2000.

[35] S. Zhang and A. Burns, “An optimal synchronous bandwidth allocation scheme for guaranteeing synchronous message deadline with the timed-token MAC protocol,” *IEEE/ACM Trans. Networking*, vol. 3, pp. 729–741, Dec. 1995.

[36] S. Zhang and E. S. Lee, “The worst-case scenario for transmission of synchronous traffic in an FDDI network,” in *Proc. Local Computer Networks 1999 (LCN’99)*, Oct. 1999, pp. 192–201.

Marco Conti received the Laurea degree in computer science from the University of Pisa, Pisa, Italy, in 1987.

In 1987, he joined the Networks and Distributed Systems Department of CNUCE, Pisa, Italy, an institute of the Italian National Research Council (CNR). He is currently a Senior Researcher with the National Research Council, IIT Institute, Pisa, Italy. His current research interests include Internet architecture and protocols, wireless networks, mobile computing, multimedia systems, and QoS in packet switching networks. He is the coauthor of *Metropolitan Area Networks* (London, U.K.: Springer, 1997). He served as Coordinator for two minitracks (“Mobile *ad-hoc* networking” and “QoS in Web Services”) at the HICSS-34 conference. He is the Technical Program Chair of the second IFIP-TC6 Networking Conference “Networking 2002.” He served as a Co-Guest Editor for journal special issues: *Cluster Computing Journal* (“Mobile *Ad-Hoc* Networking”) and *ACM/Kluwer Mobile Networks and Applications Journal* (“Mobile *Ad-Hoc* Networks”). He is a member of IFIP WG 6.3 and WG 6.8.

Mr. Conti served as a Co-Guest Editor for the IEEE TRANSACTIONS ON COMPUTERS Special Issue on “Quality of Service Issues in Internet Web Services.”

Lorenzo Donatiello received the Laurea degree in computer science from the University of Pisa, Pisa, Italy.

From 1984 to 1990, he was with the Department of Computer Science, University of Pisa. Since November 1990, he has been a Professor of Computer Science at the University of Bologna, Bologna, Italy. From March 1983 to April 1984 and from August 1986 to October 1986, he was a Visiting Scientist at the IBM T. J. Watson Research Center, Yorktown Heights, NY. His research interests include performance models of computer and communication systems, software architecture, parallel and distributed simulation, and wireless networks. He is a member of IFIP WG 7.3 and a member of the International Board of Editors of *Performance Evaluation*.

Marco Furini received the Laurea degree and the Ph.D. degree in computer science from the University of Bologna, Bologna, Italy, in 1995 and 2001, respectively.

He is currently a faculty member of the Computer Science Department, Piemonte Orientale University, Alessandria, Italy. From August 1998 to May 1999, he visited the Network Research Group in the Department of Computer Science, University of Massachusetts. His current research interests include QoS architectures and protocols in packet switching networks, multimedia systems, and *ad-hoc* wireless networks.