



**Analysis of MetaRing: a real-time protocol for
Metropolitan Area Network**

Marco Conti Lorenzo Donatiello Marco Furini

Technical Report UBLCS-98-5

May 1998

Department of Computer Science
University of Bologna
Mura Anteo Zamboni 7
40127 Bologna (Italy)

The University of Bologna Department of Computer Science Research Technical Reports are available in gzipped PostScript format via anonymous FTP from the area `ftp.cs.unibo.it:/pub/TR/UBLCS` or via WWW at URL `http://www.cs.unibo.it/`. Plain-text abstracts organized by year are available in the directory ABSTRACTS. All local authors can be reached via e-mail at the address `last-name@cs.unibo.it`. Questions and comments should be addressed to `tr-admin@cs.unibo.it`.

Recent Titles from the UBLCS Technical Report Series

- 95-18 *Group Membership and View Synchrony in Partitionable Asynchronous Distributed Systems: Specifications*, O. Babaoglu, R. Davoli, A. Montresor, November 1995 (Revised September 1996).
- 96-1 *An Investigation on the Optimal Implementation of Processes*, C. Laneve, January 1996.
- 96-2 *Expansivity, Permutivity, and Chaos for Cellular Automata*, F. Fagnani, L. Margara, January 1996
- 96-3 *Enriched View Synchrony: A paradigm for Programming Dependable Applications in Partitionable Asynchronous Distributed Systems*, O. Babaoglu, A. Bartoli, G. Dini, February 1996 (Revised November 1996).
- 96-4 *May and Must Testing in the Join-Calculus*, C. Laneve, March 1996.
- 96-5 *The Shape of Shade: a Coordination Systems*, S. Castellani, P. Ciancarini, D. Rossi, March 1996.
- 96-6 *Engineering Formal Requirements: an Analysis and Testing Method for Z Documents*, P. Ciancarini, S. Cimato, C. Mascolo, March 1996.
- 96-7 *Using Bayesian Belief Networks for the Automated Assessment of Students' Knowledge of Geometry Problem Solving Procedures*, M. Roccetti, P. Salomoni, March 1996 (Revised March 1997).
- 96-8 *Virtual Interactions: An Investigation of the Dynamics of Sharing Graphs*, A. Asperti, C. Laneve, April 1996.
- 96-9 *Towards an Algebra of Actors*, M. Gaspari, April 1996.
- 96-10 *Mobile Petri Nets*, A. Asperti, N. Busi, May 1996.
- 96-11 *Communication Support for Critical Distributed Multimedia Applications: an Experimental Study*, F. Panziera, M. Roccetti, May 1996.
- 96-12 *A Logic Coordination Language Based on the Chemical Metaphor*, P. Ciancarini, D. Fogli, M. Gaspari, July 1996.
- 96-13 *Towards Parallelization of Concurrent Systems*, F. Corradini, R. Gorrieri, D. Marchignoli, August 1996 (Revised December 1996).
- 96-14 *The Compositional Security Checker: A Tool for the Verification of Information Flow Security Properties*, R. Focardi, R. Gorrieri, August 1996.
- 96-15 *Jada: a Coordination Toolkit for Java*, P. Ciancarini, D. Rossi, October 1996.
- 96-16 *Fault Tolerance through View Synchrony in Partitionable Asynchronous Distributed Systems*, A. Montresor, December 1996.
- 96-17 *A Tutorial on EMPA: A Theory of Concurrent Processes with Nondeterminism, Priorities, Probabilities and Time*, M. Bernardo, R. Gorrieri, December 1996 (Revised January 1997).
- 97-1 *Partitionable Group Membership: Specification and Algorithms*, O. Babaoglu, R. Davoli, A. Montresor, January 1997.
- 97-2 *A Truly Concurrent View of Linda Interprocess Communication*, N. Busi, R. Gorrieri, G. Zavattaro, February 1997.
- 97-3 *Knowledge-Level Speech Acts*, M. Gaspari, March 1997.
- 97-4 *An Algebra of Actors*, M. Gaspari, G. Zavattaro, May 1997.
- 97-5 *On the Turing Equivalence of Linda Coordination Primitives*, N. Busi, R. Gorrieri, G. Zavattaro, May 1997.
- 97-6 *A Process Algebraic View of Linda Coordination Primitives*, N. Busi, R. Gorrieri, G. Zavattaro, May 1997.
- 97-7 *Validating a Software Architecture with respect to an Architectural Style*, P. Ciancarini, W. Penzo, July 1997.
- 97-8 *System Support for Partition-Aware Network Applications*, O. Babaoglu, R. Davoli, A. Montresor, R. Segala, October 1997.
- 97-9 *Generalized Semi-Markovian Process Algebra*, M. Bravetti, M. Bernardo, R. Gorrieri, October 1997.
- 98-1 *Group Membership and View Synchrony in Partitionable Asynchronous Systems. Specification and Algorithms*, O. Babaoglu, R. Davoli, A. Montresor, April 1998.
- 98-2 *A Catalog of Architectural Styles for Mobility*, P. Ciancarini, C. Mascolo, April 1998.
- 98-3 *Comparing Three Semantics for Linda-like Languages*, N. Busi, R. Gorrieri, G. Zavattaro, May 1998.

Analysis of MetaRing: a real-time protocol for Metropolitan Area Network

Marco Conti¹

Lorenzo Donatiello²

Marco Furini²

Technical Report UBLCS-98-5

May 1998

Abstract

This paper focuses on access protocols for supporting real-time communications in a distributed system; specifically, we analyze the MetaRing protocol. MetaRing is a protocol, designed and prototyped to handle, in a local or metropolitan area environment, both periodic traffic with real-time constraints (synchronous traffic) and bursty data traffic with no delay constraints (asynchronous traffic). We show that, in the original proposal of the MetaRing protocol, the deadlines of real-time messages are not always satisfied.

Through a performance study, we solve this problem modifying MetaRing. The properties of the modified version are fully proved. These properties allow the development of a synchronous bandwidth allocation scheme. With this allocation scheme, the modified version of MetaRing, is able to support distributed real-time applications.

¹ Consiglio Nazionale delle Ricerche, Istituto CNUCE, Via S. Maria 36, 56126 Pisa, Italy

² Department of Computer Science, Mura Anteo Zamboni 7, 40127 Bologna, Italy

1 Introduction

MetaRing ([OFEK91], [OFEK94]) is a MAC (Media Access Control) protocol developed at IBM T.J. Watson Research Center. Designer's main motivation was to increase throughput of a ring-based local area network. MetaRing is able to reach a speed up to 1 Gb/s and it's compatible with the emerging standard ATM [PRYC91].

Since recent year have seen a great usage of synchronous information, also known as real-time information (information with time constraints, namely information that must be sent, by the communication protocol, within specified time windows), MetaRing is able to handle two types of traffic: synchronous (or real-time) and asynchronous (generic traffic with no time constraints).

To handle these type of traffic, MetaRing uses innovating characteristics, such as: concurrent access, spatial reuse policy, control signals that circulate in opposite way with respect to the data they control. These characteristics should yield MetaRing a protocol able to efficiently handle real-time traffic in a metropolitan environment.

In this paper we present a deeply investigation about the performance reached by the protocol, when it handles real-time traffic. We highlight ad we solve some problems of MetaRing.

The paper is organized as follow. In section 2 we present the original version of MetaRing. In section 3 we show an analysis of MetaRing and we highlight some problems. In section 4 we present properties of MetaRing obtained by our analysis and we modified the original version of MetaRing to solve problem showed in section 3. In section 5 we present properties of the modified version of MetaRing and we give the guidelines for developing a synchronous bandwidth allocation scheme. Conclusions are drawn in section 5.

2 MetaRing Protocol

MetaRing ([OFEK91], [OFEK94]) is a MAC (Media Access Control) protocol for LAN and MAN, with a speed up to Gigabit/s.

The network is based on a dual fiber-optic ring topology. Specifically (Figure 2.1) the network consists of two high-speed unidirectional rings carrying information in opposite directions. Transmission on a ring is unidirectional. The access to a ring, let us assume ring A, is managed through control information travelling on the opposite ring (i.e. ring B). MetaRing can operate under two basic access control modes: *buffer insertion mode* for variable size packets and *slotted mode* for packets with a fixed size.

The main motivation for developing MetaRing is to integrate time-constrained applications and classical data applications in a high-speed LAN environment, to utilize the network resources at the maximum extent.

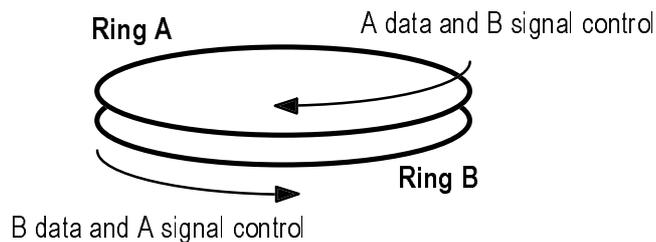


Figure 2.1 Dual ring topology

The integration of traffics with different Quality of Service (QoS) constraints is achieved by providing a connection-oriented and connectionless service to the synchronous (real-time) and asynchronous (no real-time) traffic, respectively. Specifically, resources are reserved (if available) for a real-time application in the connection set-up phase. Furthermore, a set of protocol mechanisms has been designed which ensure that non-reserved asynchronous traffic will not prevent the synchronous-traffic access.

An efficient utilization of the network resources is achieved via the spatial reuse and the concurrent access to the network. Spatial reuse means that packets are removed from the network at their destination. A concurrent access is obtained by enabling stations to transmit in all the empty slots they observe.

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 always can transmit the synchronous traffic. Before transmitting the asynchronous packets a station must collect an authorization. Specifically, asynchronous transmissions on a ring A are authorized by a control signal, called SAT (from SATisfied), travelling on the ring B. When a station receives the SAT signal in ring B, it performs different actions depending on its status. A station, when it receives the SAT on ring B, can be in the *satisfied* state or *not-satisfied* state. A station is said to be satisfied if between two visits of the SAT signal the station has transmitted, on the ring A, at least l packets or its output (asynchronous) buffer is empty.

A station that is satisfied, when receives the SAT, it forwards the SAT signal up-stream without any delay. On the other hand, a not-satisfied station will hold the SAT until it is satisfied, and only then it will forward the SAT signal up-stream.

After a station forwards a SAT, it can send up to k ($k \geq l$) additional asynchronous packets, before receiving and forwarding again the SAT signal. SAT and send packets algorithms are shown in Figure 2.1 and in Figure 2.3.

MetaRing has a mechanism for integrating two types of traffic over the full-duplex ring: synchronous or real-time traffic that requires connection or reservations set-up, a given bandwidth and bounded delay, and asynchronous traffic with no real-time constraints that can use the remainder of the bandwidth in a fair manner.

The mechanism is based on two control messages SAT and ASYNC-EN which circulate in the opposite direction to the data traffic they regulate. The ASYNC-EN (ASYNCronous ENable) is used for enabling the integration of the asynchronous traffic, and the SAT is used for ensuring fairness of the asynchronous traffic. A complete description of MetaRing can be found in [OFEK91], [OFEK94], [CIDO93].

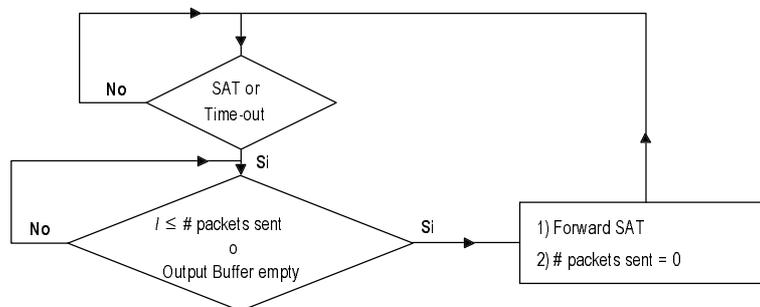


Figure 2.2 Forward SAT algorithm.

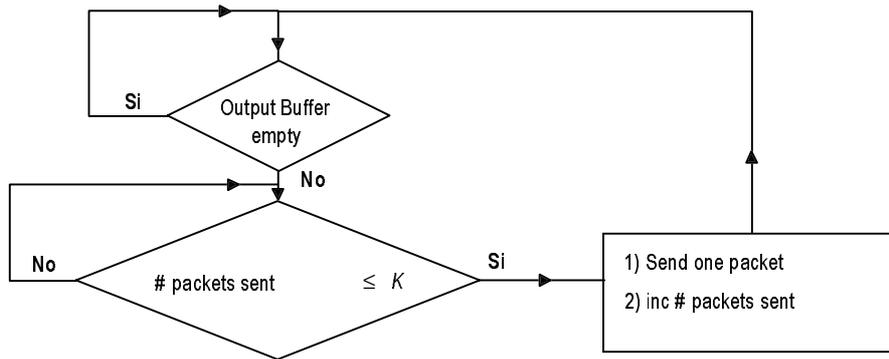


Figure 2.3 Send packets algorithm. Analysis of MetaRing

In this section we study the MetaRing performance; more precisely our studies are focused on the performance reached when MetaRing handles real-time traffic. In particular we analyze if MetaRing is able to satisfy the deadlines of the traffic it handles.

It's known that a first protocol requirement, in a real-time area, is the presence of a known upper bound to the network access time (i.e. the transfer delay must have a known upper bound: deterministic or statistic [FERR90]).

Since MetaRing's designers state that MetaRing provides this upper bound, it implies that the protocol can be used, without problems, in a Real-Time environment. Unfortunately this implication is not true; in fact in this section we show, through a performance analysis, that it exists at least one scenario in which the synchronous traffic waiting time exceeds the designers' upper bound.

In [OFEK91] designers state:

Theorem

For a given node j , let ρ denote the amount of synchronous bandwidth reserved (through node j) by the other network nodes, and let ε the synchronous bandwidth reserved by node j , then if $\rho + \varepsilon < 1$, the waiting time of node j synchronous traffic is less or equal to T_{max} , where:

$$(2.1) \quad T_{max} = \frac{Thres + (r + 2) \cdot Tring}{1 - \rho}.$$

$Tring$ is the rotation time, $Thres$ indicates a threshold beyond that an enqueued packet must be sent as soon as possible, r is the number of round that the ASYNC-EN makes in a particular state. A complete description and the proof on this theorem can be found in [OFEK91].

□

In the following, we will show a scenario in which, even-though, the conditions of theorem are satisfied, the synchronous-traffic waiting time at node j exceeds T_{max} . The scenario we consider is a slotted ring with S slots and N stations.

Let P_i the interarrival time (i.e. the period) of the station i real-time traffic. Furthermore we assume that:

1. the deadline D_i of the station i packets coincided with the traffic period (i.e. $D_i = P_i$);
2. each real-time message can be transmitted in a slot.

Hence, according to these hypotheses, the synchronous bandwidth reserved by a station i is $\rho_i = 1/P_i$.

According to theorem, let $\varepsilon=1/P_j$, $\rho = \sum_{i=1}^N \frac{1}{P_i}$, then if the following conditions hold

$$(2.2) \begin{cases} D_j = P_i \geq T_{\max} \\ \rho + \varepsilon = \sum_{i=1}^N \frac{1}{P_i} + \varepsilon < 1 \end{cases} \quad \forall i \in 1..N,$$

station j synchronous-packets should never miss their deadlines. This is not true. For example, let us consider a ring with $N=100$ stations, $S=10$ slots in the ring, $Thres=10$ slots, $r=0$ and suppose that stations generate synchronous messages according to the following laws:

$$(2.3) \begin{cases} P_1 = t_o + S + 75 \cdot j \quad \forall j \in \aleph \\ P_i = t_o + 200 \cdot j \quad \forall j \in \aleph, \forall i \in 2..100 \end{cases}$$

In this configuration, if we consider the station 1 (i.e. $j=1$), according to (2.1)-(2.3) we have $\rho = \sum_{i=2}^N \frac{1}{P_i} = 0.495$, $\varepsilon = \frac{1}{P_1} \approx 0.013$, and the upper bound for its synchronous traffic is $T_{\max} = \frac{30}{0.505} < 59.5$ slots. T_{\max} is less than the station 1 deadline, and hence, there should be no problem to handle the station 1 traffic in this MetaRing configuration.

The example presented in Figure 2.1 shows that, in some case, the station 1 deadline is missed. In this example all the network nodes are transmitting to a gateway station (station G) which does not transmit any packet. Furthermore let us assume that slots circulate clockwise. Therefore stations from N to 2 observe the empty slots before station 1.

As shown in Figure 2.2, at time t_0 , a synchronous message is queued for transmission at the station 2, 3, ..., N . When a message arrives at station 1, at time t_0+10 , all slots in the network are used by other stations to transmit their messages. Up to time t_0+99 all the slots observed by station 1 are busy and so its transmission cannot occur before this time. To satisfy its deadline, the station 1 traffic must be transmitted before $t_0+S+75=t_0+85$ slots, and this means that, in the scenario presented in Figure 2.1 and Figure 2.2 the station 1 traffic deadline is missed.

This means that MetaRing can't be used to handle real-time traffic. In next section we're modifying the original version of MetaRing to solve these real-time problems.

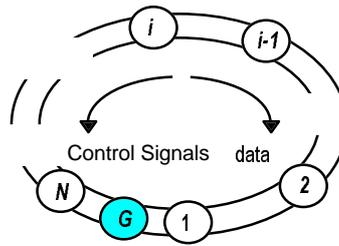


Figure 2.1 Scenario studied: station G is the gateway.

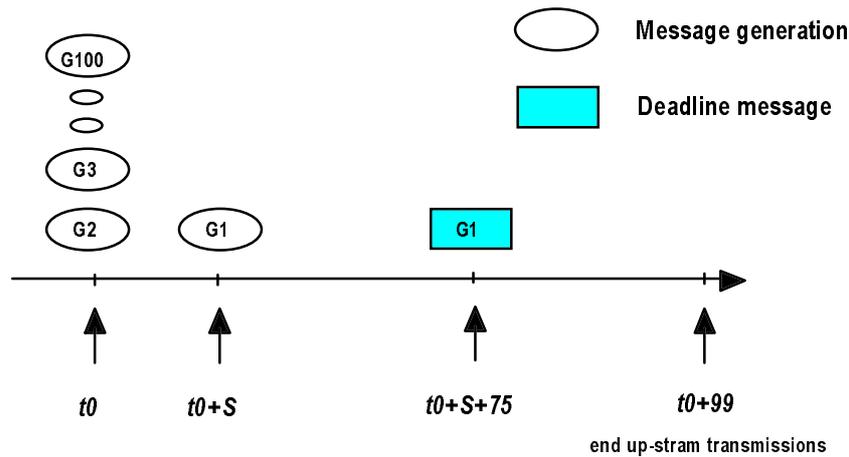


Figure 2.2 Deadline of message G1 is not meet.

3 MetaRing properties

In this section we present some properties of MetaRing, obtained by a deeply investigation of the protocol. In particular we are going to solve problems highlighted in the previous section. The solution to these problems allows MetaRing to handle real-time traffic.

The study of MetaRing has shown that one control signal (SAT) has a similar function to the one of the FDDI token.

It is known that FDDI [ANSI87] can handle real-time transmissions because the token provides an upper bound to the network access time and each time a station receives the token it can always transmit a fixed quota of synchronous traffic (see e.g. [AGRA94], [MALC94]). MetaRing, likewise, can handle the real-time traffic transmission, only if the control signal has a known upper bound to the network access time.

Hence the goal is to show the presence of an upper bound to the control signal round trip time. That's why we investigate if the SAT round trip has a time upper bound; if so the signal can control the real-time traffic yielding MetaRing to be able to support distributed real-time applications.

In this way we introduce a change with respect to the original definition of the protocol. In fact we are trying to control the real-time traffic by the SAT signal control. In the original version this signal only controls the asynchronous traffic.

Furthermore we have to note that in MetaRing the SAT control signal circulates in opposite way with respect to the data it controls, while in FDDI data and control signal (the token) have the same direction. However SAT circulation doesn't affect protocol architecture and so it's possible to implement both solutions.

To understand which solution can be the best one (i.e. has a lower value for the upper bound to the network access time), we analyze both solutions.

Initially we have considered a MetaRing approach and then an FDDI approach. A comparison between these approaches shows that with FDDI approach is possible to reach a lower value of the upper bound to the SAT rotation time.

3.1 SAT goes in opposite data circulation

This section shows results obtained when the SAT control signal circulates in opposite way with respect to the data it controls.

Particularly we present a property of MetaRing that show the presence of the upper bound to the SAT rotation time; the proof of this property, together with other properties, can be found in [FUR195] and in appendix A of this paper.

In the following we consider a slotted ring with N stations and S slots (Figure 3.1).

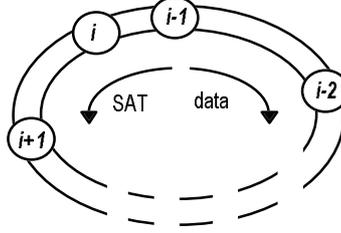


Figure 3.1 Network scenario.

Before showing the upper bound to the SAT rotation time, we need to introduce two definitions:

Primary delay: is the delay caused to a station i by a slot that has never caused delay to others stations. In this delay we also include the slot transmitted while a station holds the SAT.

Secondary delay: is the delay caused to a station i , by a slot that has already caused delay to some others station.

□

Lemma 3.1

When a station receives the SAT signal, the maximum secondary delay is less or equal to $S - \delta_{i,prec(i)}$, where $prec(i)$ denotes the index of the last station who has held the SAT.

If, in the last round, no station has held the SAT, we have $prec(i)=i$.

The value $\delta_{i,prec(i)}$ denotes the time it takes the SAT to move from the station $prec(i)$ to the station i . In the worst case $prec(i)=i$ and so $\delta_{i,prec(i)} = S$.

□

Theorem 3.1

Let SAT_TIME_i be the time elapsed between two consecutive visits of the SAT signal at the same station i . SAT_TIME_i has an upper bound and the following holds:

$$(3.1) \quad SAT_TIME_i \leq S + (S - \delta_{i,prec(i)} + k_i + l_i) + \sum_{\substack{j \neq i \\ j \in T}} (S - \delta_{j,prec(j)} + k_j + l_j) + \sum_{\substack{j \neq i \\ j \notin T}} 2 \cdot k_j$$

where T is the set containing all the index stations that hold the SAT. The index is given to any station in according to the SAT rotation.

□

Proposition 3.1

Equation (3.1) can be simplified if we consider a scenario in which all the stations can send, in the ring, the same quota of traffic, such as: if $S - \delta_{i,i-1} \geq k_i - l_i$, $l_i = l_j$ and $k_i = k_j$ for each station j and each station i , then the upper bound is equal to:

$$(3.2) \quad N \cdot (S + k + l).$$

□

The upper bound obtained in Theorem 3.1 shows that is possible to have an upper bound to the network access time, if the ring traffic is controlled by the SAT control signal.

In this way it is possible to implement a synchronous bandwidth allocation scheme that uses SAT control signal to handle real-time traffic. In fact at every SAT round, each station i can send, with certainty, l_i packets. If we give a priority to the synchronous traffic, these l_i packets can be seen as a synchronous quota reserved to station i .

Hence there is a change to the original version of MetaRing; in that version the SAT control signal only controls the asynchronous traffic, while in the modified version the SAT is also used to control the synchronous traffic.

The modified version solves problems highlighted in section 3. Note that, in this new version, the SAT controls both synchronous and asynchronous traffic, and so there is no need of having a second signal (ASYNC-EN) like in the original version of the protocol.

3.1.1 Studies of real scenario

In this section we present a study of a real-scenario (see appendix B for more details). This study confirms the presence of the upper bound described by equation (3.1). In fact it was proved (in a ring with N stations and S slots) that the follow (T_{max} represents the maximum synchronous traffic waiting time) holds:

$$T_{max} = S + (N - 1) \cdot l + (N - 1) \left[(N - 2) \cdot \frac{S}{N} + k \right].$$

Particularly, if $N \gg 1$ we have:

$$T_{max} \cong S + (N - 1) \cdot l + (N - 2) \cdot S + (N - 1)k \Rightarrow T_{max} \cong N \cdot (S + l + k),$$

that coincide with equation (3.2).

Moreover, it's easy to prove that the following proposition holds.

Proposition 3.2

The maximum time elapsed between two consecutive visits of the SAT at the same station belong to $[T_{max}, SAT_MAX)$ range.

Analytically:

$SAT_TIME \in [T_{max}, SAT_MAX)$ where:

$$T_{max} = S + (N - 1) \cdot l + (N - 1) \left[(N - 2) \cdot \frac{S}{N} + k \right]$$

$$SAT_MAX_i = S + (S - \delta_{i,prec(i)} + k_i + l_i) + \sum_{\substack{j \neq i \\ j \in T}} (S - \delta_{j,prec(j)} + k_j + l_j) + \sum_{\substack{j \neq i \\ j \notin T}} 2 \cdot k_j$$

The proof can be found in [FURI95] and in the Appendix B of this paper. □

3.2 SAT goes in same data direction

In this section we analyze MetaRing with the SAT control signal that circulates in the same way with respect to the data it controls (like the FDDI token).

Our goal is finding a new upper bound with a lower value than the one of equation (3.1). In fact, this equation depends from the number (S) of the slots present in the ring and, in a heavy manner, from the number (N) of the ring stations (with synchronous traffic).

In the remaining of this section we only present the properties of MetaRing that show the existence of an upper bound, called SAT_TIME , to the time elapsed between two consecutive visits of the SAT at the same station. The proofs of these properties, together with other properties of MetaRing, can be found in [FURI95] and in the Appendix C of this paper.

In the following we consider a slotted ring with N stations and S slots (Figure 3.2).

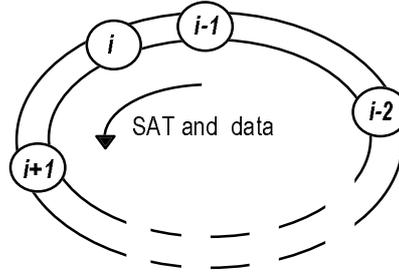


Figure 3.2 Network scenario.

Theorem 3.2

Let SAT_TIME_i be the time elapsed between two consecutive visits of the SAT at the same station i . SAT_TIME_i has an upper bound and the following holds:

$$(3.3) \quad SAT_TIME_i \leq S + l_i + l_{i-1} + \sum_{\substack{j=1 \\ j \neq i-1}}^N 2 \cdot k_j - k_i,$$

The index is given to any station in according to the SAT rotation. □

Proposition 3.3

If all the ring stations have the parameters l and k with the same value ($l_i=l_j$ and $k_i=k_j$ for each station j and each station i) then the maximum time elapsed between two consecutive visits of the SAT at the same station has an upper bound equal to:

$$(3.4) \quad S + k \cdot (2 \cdot N - 3) + 2 \cdot l.$$

□

As in the previous section, the upper bound obtained by Theorem 3.2 shows that it is possible to have an upper bound to the network access time if the ring traffic is controlled by the SAT signal.

The reason why this second analysis was made, is for finding a new upper bound whose value was lower than the one showed in the previous section. To do that we changed the SAT rotation way.

A comparison between these upper bound will be made in next section.

3.3 Comparison between the two SAT directions

In the previous sections we defined the equations of the upper bound to the SAT rotation time (equations (3.1) and (3.3)). Now, to compare these equations, we analyze a ring network with 150 slots (S). Each station in the ring have the same parameters ($l=150$ and $k=150$).

These values let us to use equations (3.2) and (3.4). Initially, case A, the SAT goes in opposite direction with respect to the data it controls, and in a second analysis, case B, we change the SAT circulation way (same way of the data it controls).

As we can note in Figure 3.3 the upper bound has a lower value when the SAT circulates with the same direction of the data it controls, than when the SAT circulates in the opposite direction of the data it controls. This mean that an other innovating characteristic of MetaRing can be dropped. In fact it's recommended that data and SAT circulate in the same direction (in the original version, the SAT goes in opposite way with respect to the data it controls).

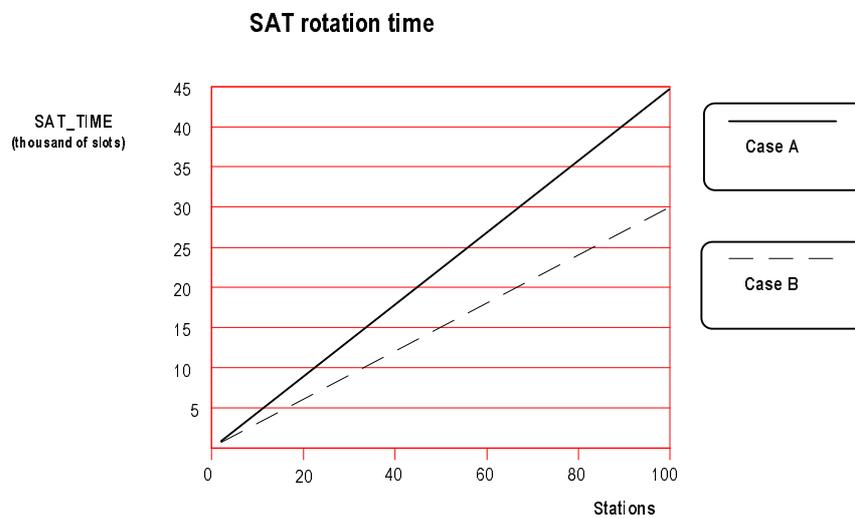


Figure 3.3 Maximum time elapsed between two consecutive SAT arrivals at the same station.

4 Synchronous Bandwidth Allocation Scheme

The objective of a bandwidth allocation scheme is to assign, at each station and in a particular time window, a quota of the available bandwidth for the station traffic transmissions.

For this reason the definition of the allocation scheme is very important for the real-time protocol [CHEN92]. In fact, a wrong synchronous allocation could lead a station unable to satisfy the deadline of its own synchronous traffic.

In previous sections we've shown that the original version of MetaRing is not able to handle real-time traffic. However, with the upper bound (Theorem 3.1 or Theorem 3.2) showed in last section, is possible to define a synchronous allocation scheme that allows to the modified version of MetaRing to correctly manage the real-time traffic.

As we're going to show, the upper bound is not sufficient; that's why we're showing some new properties (proved in Appendix D) of the modified version of MetaRing. Since results obtained in the previous section, we consider data and control signal with the same circulation way.

These properties, together with the upper bound (Theorem 3.2) allows the development of a synchronous bandwidth allocation scheme.

Since the aim of this paper is not the definition of that scheme, we only present some allocation scheme present in the literature, and we show guidelines to develop one of those, in the modified version of MetaRing.

4.1 Allocation schemes

In this section we show some allocation scheme present in the literature. Before showing the schemes, it is important to note that in [MALC94] the real-time traffic present in all the station i can be characterized as follow:

- C_i = length message (in slots);
- D_i = deadline message (in slots);
- P_i = interarrival message (in slots).

Moreover we can suppose, without loss of generality, that the deadline of every message is equal to the interarrival time (i.e. $D_i=P_i$).

These notations allow us to better understand the following scheme.

4.1.1 Full Length

With this scheme, the synchronous bandwidth allocated to a station is equal to its total time required for transmitting its synchronous messages.

This scheme attempts to transmit a synchronous message in a single turn rather than spitting it into chunks and distributing its transmission over its period P_i . Although the synchronous bandwidth allocated is sufficient, the worst case achievable utilization is zero [AGRA94].

4.1.2 Proportional

With this scheme, the synchronous bandwidth allocated to a station is proportional to the ratio of C_i and P_i . Intuitively speaking, this scheme divides the transmission of its message into as many parts as the number of times the token is expected to arrive at station i within its period P_i . The worst case achievable utilization of this scheme can asymptotically approach 0% [AGRA94].

4.1.3 Equal Portion

In this scheme the usable bandwidth is divided equally among the n stations for allocating their synchronous capacities. The worst case achievable utilization of this scheme is approximately 0% [AGRA94].

4.1.4 Normalized Proportional

Whit this scheme, the synchronous bandwidth is allocated according to the normalized load of the synchronous message on a station. This scheme uses both local and global information. The worst case achievable utilization is equal to 33% of the available ring utilization [AGRA94].

4.1.5 MCA (Minimum Capacity Allocation)

This scheme [CHEN92a] always assign the minimum capacity requested by stations. It works under the assumption $D_i = P_i$. In [CHEN92b] it is showed that the one obtained by the normalized proportional scheme.

4.1.6 EMCA (Enhanced MCA)

This scheme uses an upper bound between n consecutive arrivals of the token at the same station. In [ZHAN94] EMCA is defined as optimum allocation scheme. □

All these allocation schemes don't use exclusively the upper bound to the token rotation time, but they also use other parameters that strictly depend by the protocol they are embedded to. In fact, all these protocols have to satisfied three constraints, called protocol, deadline and buffer constraint:

Protocol constraint: it assures that the total synchronous bandwidth allocated is not greater that the synchronous bandwidth available;

Deadline constraint: it assures that each synchronous message is sent within its deadline;

Buffer constraint: it assures that the buffer dimension in each station can contain the maximum number of synchronous messages that can be queued; in other words it assures that synchronous messages will not lost by overflow.

To satisfy these constraints, we need to know, at least, the synchronous bandwidth available, the network access time and the buffer size.

Furthermore, it is to note that an allocation scheme presented (EMCA) needs one more parameter; in fact it uses an upper bound between n consecutive token arrivals at the same station. In fact, using n token rotations, this upper bound allows a better allocation resources [CHEN92].

With the following properties we give all the necessary parameters for the synchronous bandwidth allocation schemes presented above. Having these parameters it will be easy to implement one of the previous allocation schemes. In fact our intent is to give guidelines to implement it in the modified version of MetaRing.

A first consideration lead us to consider only the first two requirements (protocol and deadline), because these are fundamental to correctly handle real-time traffic, while the last one (buffer) is not consider critical (this problem can be solved with an adequate sizing resource).

Protocol constraint tells us that the sum of the synchronous bandwidth allocated to each node can not be greater than SAT_TIME (that is the amount of bandwidth available between two consecutive SAT arrivals).

To define deadline constraint we need to know the time elapsed between message arrival and message transmission.

In fact, while in the token passing protocols, the maximum token rotation time equals the maximum time that a message can spend in the output queue before being transmitted, in MetaRing this is not true; the SAT arrival doesn't mean transmission, but only permission for transmission. The packets with permission will be transmitted as soon as possible (when the station sees empty slots).

Moreover the original version of MetaRing doesn't give any guarantee to the synchronous traffic: the only guarantee it gives to each station, is the transmission of l packets (even only asynchronous packets) at every SAT rotation.

That's why we must give priority to the synchronous traffic; in this way, the synchronous traffic can be transmitted before the asynchronous one.

In next section, we present some new properties of the modified version of MetaRing. These properties allow to establish how satisfied the constraints to correctly handle real-time traffic.

Constraints and properties describe here allow the development of a synchronous bandwidth allocation scheme, that yield the modified version of MetaRing as a protocol able to correctly handle the real-time traffic.

4.2 Properties

Following properties establish the upper bound to the time elapsed between n consecutive visits of the SAT at the same station. These properties are fundamental for the implementation of the EMCA synchronous bandwidth allocation scheme. The proof of these properties can be found in Appendix D of this paper.

Theorem 4.1

Let $SAT_TIME_i[n]$ be the time elapsed between n consecutive visits of the SAT at the same station. $SAT_TIME_i[n]$ has an upper bound and the following holds:

$$SAT_TIME_i[n] \leq n \cdot S + l_i + l_{i-1} + (n+1) \cdot \sum_{j=1}^N k_j$$

□

Proposition 4.1

If all the ring stations have the parameters l and k with the same value ($l_i=l_j$ and $k_i=k_j$ for each station j and each station i) then the maximum time elapsed between n consecutive visits of the SAT at the same station has an upper bound equal to:

$$n \cdot S + 2 \cdot l + (n+1) \cdot N \cdot k .$$

□

As we know, the previous properties are necessary, but not sufficient for the respect of the real-time traffic deadline. For this reason we present the following properties (proved in Appendix D) that establish the upper bound to the network access time. This upper bound is used to verify if the protocol can meet the real-time traffic deadline (deadline greater or equal to the network access time). With the following properties it will be available all the parameters needed to implement one of the allocation schemes presented in section 5.1. In this way the modified version of MetaRing, can be integrated with one allocation scheme, and hence it can be efficiently used in whichever real-time environments.

Lemma 4.1

Let us consider the first synchronous packet in the output queue of station i and let $T_{wait_per}^i$ be the time that this packet has to wait before receiving the permission of transmission. The following holds:

$$T_{wait_per}^i = S + l_{i-1} + \sum_{\substack{j=1 \\ j \neq i-1}}^N 2 \cdot k_j - k_i$$

□

Proposition 4.2

Let us suppose that each synchronous message of station i is divided into l_i packets. Let $T_{wait_per}^i(n)$ be the time elapsed in the synchronous queue by the n -th message before receiving the permission of transmission. The following holds:

$$T_{wait_per}^i(n) \leq SAT_TIME_i[n]$$

□

Lemma 4.2

Let us consider station i and let $T_{wait_tx}^i$ be the time elapsed between the permission and the complete transmission of the first group of l_i synchronous packets in the queue. The following holds:

$$T_{wait_tx}^i \leq S + l_i + l_{i-1} + \sum_{\substack{j=1 \\ j \neq i-1 \\ j \neq i}}^N 2 \cdot k_j + k_{i-1}$$

□

Proposition 4.3

If the first group of l_i synchronous packets in the queue of station i have waited for a time equal to $T_{wait_per}^i$, before having the permission of transmission, the maximum time elapsed between the permission and the complete transmissions is equal to $l_i + 2 \cdot k_{i-1}$.

□

Lemma 4.3

Let us consider the first synchronous message in the output queue of station i (divided into l_i packets) and let T_{wait}^i be the time elapsed between its arrival in the first queue position and its complete transmission. The following holds:

$$T_{wait}^i \leq S + l_i + l_{i-1} - k_i + 2 \cdot \sum_{j=1}^N k_j$$

□

Theorem 4.2

Let us suppose that each synchronous message of station i is divided into l_i packets. Let $T_{wait}^i(n)$ be the time elapsed between its arrival in the queue and its complete transmission. The following holds:

$$T_{wait}^i(n) \leq (n+2) \cdot S + l_i + l_{i-1} + (n+3) \cdot \sum_{j=1}^N k_j$$

□

5 Conclusions

In this work we've presented a study of a MAC protocol, named MetaRing, developed for supporting the real-time traffic transmissions in a distributed environment. MetaRing was developed to overcome the inefficiency problems caused by using the real-time protocol, designed for LAN (like FDDI), in a metropolitan area.

Our studies have shown that, despite the intentions' designers, MetaRing is not able to support the transmission of the real-time traffic in whichever scenario. In fact we've presented a scenario in which MetaRing doesn't meet all the deadlines of the real-time traffic present in the network; indeed the synchronous traffic waiting time exceeds the designers' upper bound to the network access time.

For this reason we've deeply analyzed the protocol for finding a solution to this problem. This solution doesn't require new technology, because it uses only MetaRing technology. Moreover we preserve the compatibility with ATM standard, and we use some of the innovating characteristic presented by MetaRing, like concurrent access and spatial reuse policy.

Our studies have shown that the traffic can be controlled by only one control signal. This signal can circulate both in the same and in opposite way whit respect to the data it controls. However, for having a lower value of the upper bound to the SAT rotation time, it's recommended that data and SAT signal circulate in the same direction.

To handle real-time traffic, an access control protocol must have a synchronous bandwidth allocation scheme. Since the aim of this paper was the analysis of MetaRing and not the definition of an allocation

scheme, we haven't deeply investigated this issue. However, we have showed some bandwidth allocation scheme present in the literature. To implement one of these some new parameters are needed. That's why we've investigated the modified version of MetaRing, to find out properties from which those parameters are deducted. These properties are presented and they are fully proved. In this way it is easy to implement a synchronous bandwidth allocation scheme.

In conclusion, properties described in this paper allow the modified version of MetaRing to be used in any real-time scenario with certainty that all the deadlines will be meet.

Appendix A

Lemma A.1

The maximum number of busy slots that a station i can see, when it holds the SAT control signal, is equal to:

$$S + \sum_{\substack{j=1 \\ j \neq i}}^N k_j.$$

Proof

We consider the scenario presented in Figure A.1 and we suppose that the SAT signal leaves the station $i-1$ and goes towards station i .

Station $i-1$ releases the SAT signal when it is satisfied, such as when it has transmitted a fixed quota of packets. Moreover, after releasing the SAT, station $i-1$ can send up to k_{i-1} packets before blocking itself.

All the others network station j ($j \neq \{i, i-1\}$), can have up to k_j packets to transmit. The worst case is presented when, while the SAT is going to the station i from the station $i-1$, all the slots in the ring are marked busy. This situation can happen if the quota (l) that make the station satisfied is greater than the number of the slot in the ring (S).

When a station i receives the SAT, it sees up to S busy slots, and it sees k_{i-1} slots which contain packets sent by station $i-1$. This mean that we do not consider the quota l_{i-1} , but only the number S . In the worst case all the j stations (from $i-2$ to $i+1$) will transmit their own k_j packets.

Hence, station i sees its first empty slot after a maximum number of slots equal to:

$$S + \sum_{\substack{j=1 \\ j \neq i}}^N k_j .$$

□

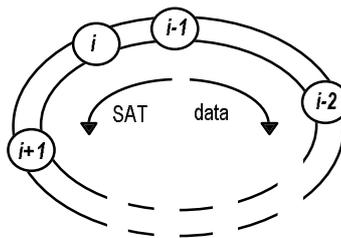


Figure A.1 Network scenario.

Proposition A.1

A station i can hold the SAT signal for a maximum time equal to: $S + l_i + \sum_{\substack{j=1 \\ j \neq i}}^N k_j$ slots.

Proof

From Lemma A.1 we know that the maximum number of busy slots that a station i can see, while it is holding the SAT signal, is equal to $S + \sum_{\substack{j=1 \\ j \neq i}}^N k_j$.

Station i leaves the SAT signal when it is satisfied, such as when it has transmitted l_i packets. In conclusion a station i can hold the SAT signal for a maximum time equal to $S + l_i + \sum_{\substack{j=1 \\ j \neq i}}^N k_j$ slots. □

Proposition A.2

If a station i holds the SAT, then the maximum number of packets that a station i has transmitted in the previous cycle is equal to l_i .

Proof

Station i holds the SAT signal if it is not satisfied, namely it has sent a number of packets less than l_i .

Station i holds the SAT until the number of packets transmitted reached the quota l_i , and then it releases the SAT signal. So the maximum number of packets transmitted in the previous cycle is equal to l_i .

In Figure A.2 we present the situation of a generic station i .

At time a station i releases the SAT signal. From time b to time c , the station i transmits a part of its l_i quota.

At time d the SAT come back to the station i . The station i is not satisfied, and so it holds the SAT.

The SAT will be released when the station i is satisfied (i.e. when it sends a number of packets equal to l_i); this happen at time e . □

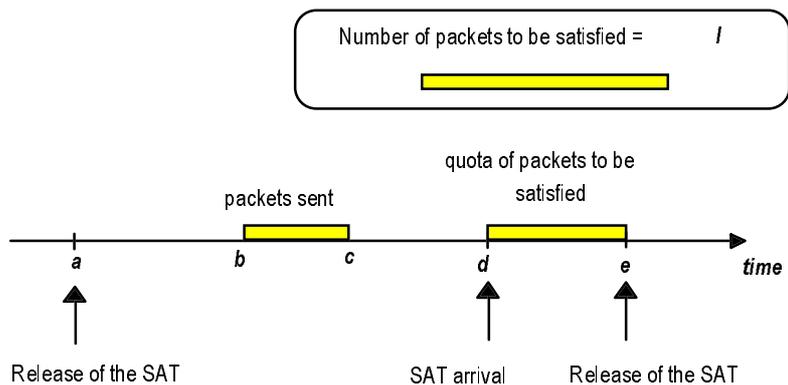


Figure A.2 Transmissions when a station holds the SAT.

Lemma A.2

If a station i holds the SAT for a time equal to $S + l_i + \sum_{\substack{j=1 \\ j \neq i}}^N k_j$, then the SAT will come back after S slots.

Proof

If a station i holds the SAT for $S + l_i + \sum_{\substack{j=1 \\ j \neq i}}^N k_j$ slots, it means that all the stations in the ring are satisfied

because station i has transmitted l_i packets and all the others stations have sent their k packets.

Hence any station won't hold the SAT signal, and it will come back to station i after S slots.

□

Lemma A.3

During the time elapsed between two consecutive arrivals of the SAT signal at station i , the number of packets that this station can send is not greater than $k_i + l_i$.

Proof

To be satisfied a station i must send l_i packets. In the worst case this packets are transmitted when the station holds the SAT.

When a station i releases the SAT, it can send up to k_i new packets, and so when the SAT come back to the station, the number of packets sent is not greater than $k_i + l_i$.

□

Lemma A.4

Let SAT_TIME be the time elapsed between two consecutive arrivals of the SAT signal in the same station; this time has an upper bound and we have:

$$SAT_TIME < \sum_{i=1}^N \left(S + l_i + \sum_{\substack{j=1 \\ j \neq i}}^N k_j \right)$$

Proof

- Proposition A.1 says that a station can hold the SAT signal for a maximum time equal to $S + l_i + \sum_{\substack{j=1 \\ j \neq i}}^N k_j$

and so if we consider all the stations in the ring we have:

$$SAT_TIME \leq \sum_{i=1}^N \left(S + l_i + \sum_{\substack{j=1 \\ j \neq i}}^N k_j \right)$$

- If a station holds the SAT for a time equal to $S + \sum_{\substack{j=1 \\ j \neq i}}^N k_j$ (Lemma A.1) then all the stations present in the ring (with the exception of station i) are satisfied (Lemma A.2) and so it is impossible for the SAT to be held, in the same round, in more than one station, and so:

$$SAT_TIME < \sum_{i=1}^N \left(S + l_i + \sum_{\substack{j=1 \\ j \neq i}}^N k_j \right)$$

□

Lemma A.5

The maximum number of busy slots that a station i can observe from the start to the next arrival of SAT has an upper bound equal to: $\sum_{\substack{j=1 \\ j \neq i}}^N 2 \cdot k_j$.

Proof

We consider a station i ; when the SAT leaves station i , it can send up to k_i packets before blocking itself. However these slots don't have to be considered because, by the spatial reuse, these slots, for the station i , are empty.

All the others station send their k packets, so they use $\sum_{\substack{j=1 \\ j \neq i}}^N k_j$ slots.

At this point all the stations are blocked, but when the SAT transits, these stations can send others k packets.

This means that the maximum number of busy slot that a station can observe is given by $\sum_{\substack{j=1 \\ j \neq i}}^N 2 \cdot k_j$.

□

Proposition A.3

If all the ring stations have the parameters l and k with the same value ($l_i=l_j$ and $k_i=k_j$ for each station j and each station i) the maximum number of busy slot that a station can observe is equal to $2 \cdot k \cdot (N - 1)$.

Proof

Trivial and therefore omitted. □

Lemma 3.1

When a station receives the SAT signal, the maximum secondary delay is less or equal to $S - \delta_{i,prec(i)}$, where $prec(i)$ denotes the index of the last station who has held the SAT.

If, in the last round, no station has held the SAT, we have $prec(i)=i$.

The value $\delta_{i,prec(i)}$ denotes the time it takes the SAT to move from the station $prec(i)$ to the station i . In the worst case $prec(i)=i$ and so $\delta_{i,prec(i)} = S$.

Proof

If a station i receives the SAT at time t_o , the busy slots observed by station i at time $t_o + S - \delta_{i,prec(i)}$ can't have caused primary delay in the others stations. □

Lemma A.6

If a station doesn't hold the SAT, it doesn't introduce additional delay to the SAT propagation time.

Proof

Trivial and therefore omitted. □

Lemma A.7

The maximum delay that a station can add to the SAT time propagation is less or equal to $S + l_i + x_i$, where x_i is the primary delay experienced from station i .

Proof

It follows from the previous lemma. □

Lemma A.8

Let t_o denote the SAT releasing time from station i , and let t_1 the next SAT departure time at the same station i . In $[t_o, t_1]$ the maximum slots marked busy by stations are:

$$(k_i + l_i) + \sum_{\substack{j \neq i \\ j \in T}} (k_j + l_j) + \sum_{\substack{j \neq i \\ j \notin T}} 2 \cdot k_j ,$$

where T is the set which contain the stations that hold the SAT.

Proof

The first term belong to the station i : if at time t_1 the station is satisfied, it doesn't hold the SAT signal and so in $[t_o, t_1]$ it can send up to k_i packets, but, if it holds the SAT, its transmissions can be $k_i + l_i$.

Likewise, if a station $j \neq i$ in $[t_o, t_1]$ holds the SAT, it sends l_j packets before receiving the SAT and k_j after releasing the SAT.

All the stations that don't hold the SAT give the last term. □

Theorem 3.1

Let SAT_TIME_i be the time elapsed between two consecutive visits of the SAT signal in the same station i . SAT_TIME_i has an upper bound and the following holds:

$$SAT_TIME_i \leq S + (S - \delta_{i,prec(i)} + k_i + l_i) + \sum_{\substack{j \neq i \\ j \in T}} (S - \delta_{j,prec(j)} + k_j + l_j) + \sum_{\substack{j \neq i \\ j \notin T}} 2 \cdot k_j$$

where T is the set which contain all the index stations that hold the SAT. The index is given to any station in according to the SAT rotation.

Proof

The first term is the SAT propagation time.

The term $\sum_{j \in T} (S - \delta_{j,prec(j)})$ is the secondary delay bound, and the term $k_i + l_i + \sum_{\substack{j \neq i \\ j \in T}} k_j + l_j + \sum_{\substack{j \neq i \\ j \notin T}} 2 \cdot k_j$ is

the primary delay bound. □

Proposition 3.1

Equation (3.1) can be simplified if we consider a scenario in which all the stations can send, in the ring, the same quota of traffic, such as: if $S - \delta_{i,i-1} \geq k_i - l_i$, $l_i = l_j$ and $k_i = k_j$ for each station j and each station i , then we have:

$$SAT_TIME_i \leq N \cdot (S + k + l) .$$

Proof

Trivial and therefore omitted. □

Appendix B

Let us consider a particular scenario: a slotted ring with S circulating slots and 4 stations equally spaced. It is known that with spatial reuse policy, packets are removed from ring by their destinations. However, using a broadcast approach, destinations matches sources; this means that each packet travels once all over the ring.

Initially ring contains no data and the SAT control signal is in station S_1 (Figure B.1).

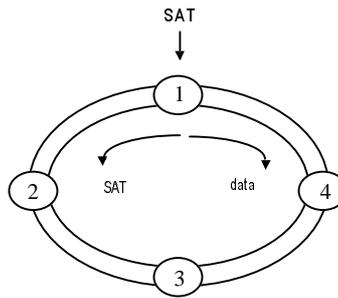


Figure B.1 Startup scenario

Furthermore we suppose that each station behaves like follow:

1. it has no data to deliver;
2. it generates real-time traffic (in asymptotic manner) at time $t-\epsilon$ if t is the SAT time arrival.

Using a worst case approach, we suppose that the satisfied quota l fills the ring (i.e. $l \geq S$); this mean that station S_1 releases SAT (sending it to station S_2) when the ring is full of its own packets (Figure B.2).

SAT control signal and stations state are illustrated in Figure B.3.

In the following, we analyze, step-by-step, all the ring situations.

When station S_1 is satisfied, it sends, to station S_2 , the SAT, and it starts to transmit others k packets towards station S_4 .

When SAT arrives at station S_2 , the first of the k packets sent by S_1 is arrived at station S_4 ; this packet will arrive at station S_2 after a time equal to: $(N - 2) \cdot \frac{S}{N}$, where N indicates the number of stations. This mean that station S_2 sees the first empty slot only after the k packets of station S_1 . Analytically, after $k + (N - 2) \cdot \frac{S}{N}$ slots.

It is to note that, we've supposed that station S_2 has generated its packets just before receiving the SAT signal.

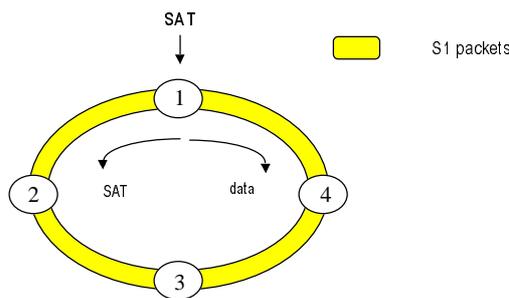


Figure B.2 Ring is full of station S_1 packets

After seeing the first empty slot, station S_2 transmits l packets, then it sends the SAT to the station S_3 and it starts to transmit his new k packets towards station S_1 .

When the SAT arrives at station S_3 , this station starts to generate messages, and when S_3 receives the SAT, the first of the k packets, sent by station S_2 towards station S_1 , is at station S_1 . This mean that station S_3 will see this packet after a time equal to $(N-2) \cdot \frac{S}{N}$ time units, and the first empty slot will be seen after a time equal to: $k + (N-2) \cdot \frac{S}{N}$ slots.

Station S_3 holds the SAT to transmit its l packets, after that it sends the SAT to station S_4 and it starts to transmit its k packets toward station S_2 .

Station S_4 starts to generate packets just a time before receiving the SAT. When the SAT arrives at station S_4 the first of the k packets, sent from station S_3 toward station S_2 , is at station S_2 , and it will arrive at station S_4 after a time equal to: $(N-2) \cdot \frac{S}{N}$. Also in this case, station S_4 sees its first empty slot after a time equal to $k + (N-2) \cdot \frac{S}{N}$.

Station S_4 transmits its l packets, and then it sends the SAT signal to station S_1 and it starts to transmit its new k packets towards station S_3 .

After a time equal to the slot-distance between station S_4 and station S_1 the SAT signal will come back to station S_1 .

Station S_1 is now satisfied and so it immediately releases the SAT. This signal will be at station S_2 , after a time equal to $\frac{S}{N}$, and so, time elapsed between two consecutive arrivals of the SAT at the same station S_2 is equal to:

$$T_{\max} = S + (N-1) \cdot l + (N-1) \left[(N-2) \cdot \frac{S}{N} + k \right].$$

Particularly, if $N \gg 1$ we have:

$$T_{\max} \cong S + (N-1) \cdot l + (N-2) \cdot S + (N-1)k \Rightarrow T_{\max} \cong N \cdot (S + l + k).$$

T_{\max} value coincide with the one showed in Proposition 3.1.

Looking at

Figure **B.3** it is easy to note that the worst case is realized when there is no concurrent access to the ring and so each station is able to transmit $l+k$ packets. This situation can be obtained with the supposition that if t is the SAT arrival time, traffic is generated at time $t-\varepsilon$, with $\varepsilon \in (0, T_{\text{slot}})$, where T_{slot} is the temporal slot length.

This reasoning lead us to the following property.

Proposition 3.2

The maximum time elapsed between two consecutive visits of the SAT at the same station belong to $[T_{\max}, SAT_MAX)$ range.

Analytically:

$SAT_TIME \in [T_{\max}, SAT_MAX)$ where:

$$T_{\max} = S + (N - 1) \cdot l + (N - 1) \left[(N - 2) \cdot \frac{S}{N} + k \right]$$

$$SAT_MAX_i = S + (S - \delta_{i,prec(i)} + k_i + l_i) + \sum_{\substack{j \neq i \\ j \in T}} (S - \delta_{j,prec(j)} + k_j + l_j) + \sum_{\substack{j \neq i \\ j \notin T}} 2 \cdot k_j$$

□

Proof

From Theorem 3.1 we know that:

$$SAT_TIME_i \leq S + (S - \delta_{i,prec(i)} + k_i + l_i) + \sum_{\substack{j \neq i \\ j \in T}} (S - \delta_{j,prec(j)} + k_j + l_j) + \sum_{\substack{j \neq i \\ j \notin T}} 2 \cdot k_j$$

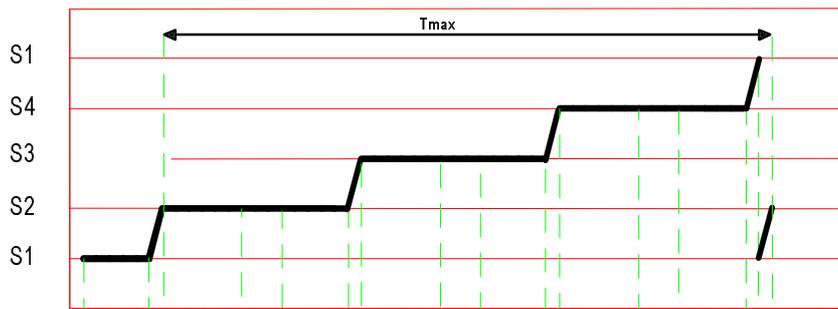
and we know that $SAT_TIME \geq T_{\max}$, where

$$T_{\max} = S + (N - 1) \cdot l + (N - 1) \left[(N - 2) \cdot \frac{S}{N} + k \right].$$

This means that: $SAT_TIME \in [T_{\max}, SAT_MAX)$

□

SAT position



Transmitting stations

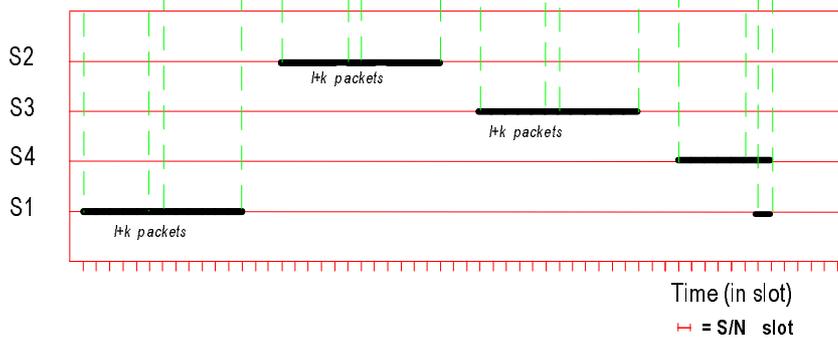


Figure B.3 Ring situation

Appendix C

Lemma C.1

The maximum number of busy slots that a station i can see, when it holds the SAT control signal, is equal to:

$$\sum_{\substack{j=1 \\ j \neq i}}^N k_j$$

Proof

We consider the scenario in Figure C.1 and we suppose that the SAT signal leaves the station $i-1$ and goes towards station i . Station $i-1$ releases the SAT when it is satisfied such as when it has sent $0 \leq x_{i-1} \leq k_{i-1}$ packets. Moreover, after releasing the SAT, the station $i-1$ can send up to k_{i-1} packets before blocking itself.

Station i sees the SAT at the end of the x_{i-1} slots and so we don't consider the quota x_{i-1} .

All the other stations j with $i+1 \leq j \leq i-2$ can have up to k_j packets to transmit, and so at the end of the k_{i-1} slots there could be all the others k_j packets. Therefore station i will see the first empty slot after a

maximum numbers of slots equal to: $\sum_{\substack{j=1 \\ j \neq i}}^N k_j$ slots.

□

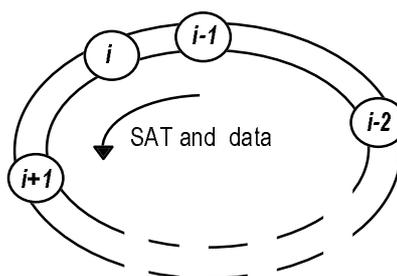


Figure C.1 Network scenario.

Proposition C.1

A station i can hold the SAT signal for a maximum time equal to: $l_i + \sum_{\substack{j=1 \\ j \neq i}}^N k_j$ slots.

Proof

From Lemma C.1 we know that the maximum number of busy slots that a station i can see, while it's holding the SAT, is equal to $\sum_{\substack{j=1 \\ j \neq i}}^N k_j$. At this value we have to add the quota of packets able to satisfy the station i (i.e. the l_i packets).

Hence a station i can hold the SAT signal for a maximum time equal to $l_i + \sum_{\substack{j=1 \\ j \neq i}}^N k_j$ slots.

□

Proposition C.2

If a station i holds the SAT, then the maximum number of packets that a station i has transmitted in the previous cycle is equal to l_i .

Proof

Proof is analogous to the one presented in Proposition A.2.

□

Lemma C.2

If a station i holds the SAT for a time equal to $l_i + \sum_{\substack{j=1 \\ j \neq i}}^N k_j$, then the SAT will come back after S slots.

Proof

Proof is analogous to the one presented in Lemma A.2.

□

Lemma C.3

During the time elapsed between two consecutive arrivals of the SAT signal at station i , the number of packets that this station can send is not greater than $k_i + l_i$.

Proof

Proof is analogous to the one presented in Lemma A.3.

□

Lemma C.4

Let SAT_TIME be the time elapsed between two consecutive arrivals of the SAT signal at the same station; this time has an upper bound and the following holds:

$$SAT_TIME < \sum_{i=1}^N \left(l_i + \sum_{\substack{j=1 \\ j \neq i}}^N k_j \right).$$

Proof

- Proposition C.1 says that a station can hold the SAT signal for a maximum time equal to $l_i + \sum_{\substack{j=1 \\ j \neq i}}^N k_j$, and

so if we consider all the stations in the ring we have:

$$SAT_TIME \leq \sum_{i=1}^N \left(l_i + \sum_{\substack{j=1 \\ j \neq i}}^N k_j \right).$$

- If a station i holds the SAT for a time equal to $\sum_{\substack{j=1 \\ j \neq i}}^N k_j$ (Lemma C.1) then all the stations present in the ring (with the exception of station i) are satisfied (Lemma C.2) and so it is impossible for the SAT to be held, in the same round, in more than in one station, and so:

$$SAT_TIME < \sum_{i=1}^N \left(l_i + \sum_{\substack{j=1 \\ j \neq i}}^N k_j \right).$$

□

Lemma C.5

The maximum number of busy slots that a station i can observe between the release to the next arrival of the SAT signal, has an upper bound equal to:

$$\sum_{\substack{j=1 \\ j \neq i}}^N 2 \cdot k_j - k_{i-1}.$$

Proof

We consider a station i . When the SAT leaves station i , it can send up to k_i packets before blocking itself. However these slots don't have to be considered because, by the spatial reuse, this slots, for the station i , are empty.

All the other stations send their k packets, so they use $\sum_{\substack{j=1 \\ j \neq i}}^N k_j$ slots.

At this point all the stations are blocked, but when the SAT transits, these stations can send others k packets.

Station $i-1$ sends its own k packets after releasing the SAT; this mean that we don't have to consider the term k_{i-1} .

In conclusion the maximum number of busy slot that a station can observe is given by $\sum_{\substack{j=1 \\ j \neq i}}^N 2 \cdot k_j - k_{i-1}$.

□

Proposition C.3

If all the stations present in the ring have the parameters l and k with the same value ($l_i=l_j$ and $k_i=k_j$ for each station j and each station i) then the maximum number of busy slots that a station i can observe from the start to the next arrival of SAT, has an upper bound equal to $k \cdot (2 \cdot N - 3)$.

□

Theorem 3.2

Let SAT_TIME_i be the time elapsed between two consecutive visits of the SAT at the same station i . SAT_TIME_i has an upper bound and the following holds:

$$SAT_TIME_i \leq S + l_i + l_{i-1} + \sum_{\substack{j=1 \\ j \neq i-1}}^N 2 \cdot k_j - k_i .$$

The index is given to any station in according to the SAT rotation.

Proof

If a station is not satisfied (i.e. it was not able to transmit its l_i packets), when it sees the SAT, it seizes the SAT.

The slots which contain the packets of the station j can produce a delay only to the station k (the station that holds the SAT).

If station j holds the SAT then the maximum delay it adds to the SAT propagation time is given by $l_j + k_j$. So, if all the ring stations hold the SAT, the SAT circulation has an upper bound and we have:

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

If a station h doesn't hold the SAT, this station can send $2 \cdot k_h$ packets and we have:

$$SAT_TIME^* \leq S + \sum_{\substack{j=1 \\ j \neq h}}^N (l_j + k_j) + 2 \cdot k_h .$$

If $SAT_TIME^* > SAT_TIME$ means that the station h (who didn't hold the SAT) has increased the SAT propagation time; we have this situation when $k_h > l_h$.

Let $T = \{i\} \cup \{h | k_h = l_h\}$ the set of stations that increase the SAT propagation time when they hold the SAT signal. The ordering of the set T is given in respect with the SAT circulation.

Note that the station i is always present in T ; this happens because if the SAT signal arrives to this station and this station is satisfied, the number of slots to be considered is k_i ; for this reason the T set always contain the index of the station i .

After this considerations the upper bound can be rewrite as:

$$SAT_TIME \leq S + \sum_{j \in T} (l_j + k_j) + \sum_{\substack{j=1 \\ j \notin T}}^N 2 \cdot k_j .$$

Now, we can do an other consideration: when the SAT moves from the station, whose index is the last of the T set, towards the station i , all the stations visited by SAT don't introduce a delay of $2 \cdot k$ to the SAT propagation time, but they introduce only a delay of k slots. At this point we have:

$$SAT_TIME \leq S + \sum_{j \in T} (l_j + k_j) + \sum_{\substack{j=1 \\ j \notin T}}^N 2 \cdot k_j - \sum_{j=Last(T)}^{i-1} k_j ,$$

where $j = Last(T)$ if $T = \{i, \dots, j\}$.

The right side of the above equation has an high value when the last term is very low; this happen when $j=i-1$. Otherwise if $T = \{i\}$ then $Last(T) = i$ (i.e. $j=i$), and so the last term would assume its higher value (sum of all the k quantity) leading the minimum value to the right side of the equations.

Moreover the term $\sum_{\substack{j=1 \\ j \notin T}}^N 2 \cdot k_j$ has a low value if the cardinality of T is little.

After this consideration we can affirm that we have the greater SAT propagation time when the T set is so composed: $T = \{i, i-1\}$, and we have:

$$SAT_TIME_i \leq S + l_i + l_{i-1} + \sum_{\substack{j=1 \\ j \neq i-1}}^N 2 \cdot k_j - k_i$$

□

Proposition 3.3

If all the ring stations have the parameters l and k with the same value ($l_i=l_j$ and $k_i=k_j$ for each station j and each station i) then the maximum time elapsed between two consecutive visits of the SAT at the same station has an upper bound equal to: $S + k \cdot (2 \cdot N - 3) + 2 \cdot l$.

Proof

Trivial and therefore omitted.

□

Appendix D

Lemma D.1

Let $SAT_TIME[n]$ be the time elapsed between n consecutive visits of the SAT at the same station. $SAT_TIME[n]$ has an upper bound given by:

$$SAT_TIME[n] < n \cdot \left(S + l_i + l_{i-1} + \sum_{\substack{j=1 \\ j \neq i-1}}^N 2 \cdot k_j - k_i \right)$$

Proof

Theorem 3.2 shows the upper bound to the time elapsed between two consecutive (such as one SAT rotation) visits of SAT at the same station. If we consider n rotations, we have:

$$SAT_TIME[n] \leq n \cdot \left(S + l_i + l_{i-1} + \sum_{\substack{j=1 \\ j \neq i-1}}^N 2 \cdot k_j - k_i \right)$$

It is to note that the maximum number of transmitted packets, in one rotation, is obtained when the transmission is equal to $2k$ (such as, when in one rotation we transmit also the quota of the previous rotation). For this reason we can't have two consecutive maximum time rotation. That's why we have:

$$SAT_TIME[n] < n \cdot \left(S + l_i + l_{i-1} + \sum_{\substack{j=1 \\ j \neq i-1}}^N 2 \cdot k_j - k_i \right)$$

□

Theorem 4.1

Let $SAT_TIME_i[n]$ be the time elapsed between n consecutive visits of the SAT at the same station. $SAT_TIME_i[n]$ has an upper bound and the following holds:

$$SAT_TIME_i[n] \leq n \cdot S + l_i + l_{i-1} + (n+1) \cdot \sum_{j=1}^N k_j$$

Proof

It is known that (Proposition C.2) if station i holds the SAT, then in the previous cycle it has transmitted less than l_i packets. This mean that if station i doesn't hold the SAT, then in the previous cycle it has transmitted not less than l_i packets (and not more than k_i).

The SAT, in each rotation, gives the permission to send k_i packets at each station i . So, if we consider n SAT rotations (from k to $n+k$), the maximum number of permission is given by: $n \cdot \left(\sum_{j=1}^N k_j \right)$.

Moreover, we know that packets, which have received the permission in a cycle, can be transmitted later. So, the packets which has received the permission in the $(k-1)$ -th cycle can be transmitted during the n successively rotation.

$$\text{Hence the number of transmissions in } n \text{ rotation can be } (n+1) \cdot \left(\sum_{j=1}^N k_j \right).$$

Furthermore, as showed in Theorem 3.2, station i and station $i-1$ can once hold the SAT (we're considering the worst case for station i).

In conclusion, if we consider the time for the n rotation of the SAT on the ring (nS), we have that time elapsed between n consecutive visits of the SAT at the same station i has an upper bound given by:

$$n \cdot S + l_i + l_{i-1} + (n+1) \cdot \sum_{j=1}^N k_j .$$

□

Proposition 4.1

If all the ring stations have the parameters l and k with the same value ($l_i=l_j$ and $k_i=k_j$ for each station j and each station i) then the maximum time elapsed between n consecutive visits of the SAT at the same station has an upper bound equal to:

$$n \cdot S + 2 \cdot l + (n+1) \cdot N \cdot k .$$

Proof

Trivial and therefore omitted.

□

Lemma 4.1

Let us consider the first synchronous packet in the output queue of station i and let $T_{wait_per}^i$ be the time that this packet has to wait before receiving the permission of transmission. The following holds:

$$T_{wait_per}^i = S + l_{i-1} + \sum_{\substack{j=1 \\ j \neq i-1}}^N 2 \cdot k_j - k_i$$

Proof

Let us consider a station i . Theorem 3.2 has shown that the upper bound to the SAT rotation is equal to:

$$SAT_MAX_i = S + l_i + l_{i-1} + \sum_{\substack{j=1 \\ j \neq i-1}}^N 2 \cdot k_j - k_i .$$

Station i has sent $l_i + k_i$ packets. These packets (in the worst case) are asynchronous packets. Let us suppose that at the end of these transmissions, station i generates its l synchronous packets.

When the SAT come back, these l packets have waited for a time equal to: $SAT_MAX_i - l_i - k_i$. □

Proposition 4.2

Let us suppose that each synchronous message of station i is divided into l_i packets. Let $T_{Wait_per}^i(n)$ be the time elapsed in the synchronous queue by the n -th message before receiving the permission of transmission. The following holds:

$$T_{Wait_per}^i(n) \leq SAT_TIME_i[n]$$

Proof

From Lemma 4.1 we know the maximum time elapsed before the first synchronous message receiving the permission. Then, the SAT signal, at each round, gives the permission to l_i packets. This mean that the first SAT round is used to give permission to the first group of l packets, and the k -th round is used to give the permission to the k group of l packets.

Analytically:

$$T_{Wait_per}^i(n) \leq n \cdot SAT_MAX_i \leq SAT_TIME_i[n]$$
□

Lemma 4.2

Let us consider station i and let $T_{Wait_tx}^i$ be the time elapsed between the permission and the complete transmission of the first group of l_i synchronous packets in the queue. The following holds:

$$T_{Wait_tx}^i \leq S + l_i + l_{i-1} + \sum_{\substack{j=1 \\ j \neq i-1 \\ j \neq i}}^N 2 \cdot k_j + k_{i-1}$$

Proof

Let us suppose that station i holds the SAT: it sends its l_i synchronous packets. The second group of l_i synchronous packets receives the permission when station i releases the SAT. Let us suppose that station i is not able to send any other packet. This mean that it will hold the SAT when it come back at station i . This mean that in this round station i doesn't transmit any packet and the l_i packets will be send in the next SAT round.

The first SAT round is equal to $SAT_MAX_i - l_i - k_i$ (station i doesn't transmit any packet after releasing the SAT).

Next round is used to send the group of l_i packets. So we have:

$$\begin{aligned}
T_{Wait_tx}^i &\leq SAT_MAX_i - l_i - k_i + SAT_MAX_i \leq \\
&\leq 2 \cdot SAT_MAX_i - l_i - k_i \leq \\
&\leq SAT_TIME_i[2] - l_i - k_i
\end{aligned}$$

However, if we consider that, when the SAT come back for the first time at station i , it is followed by only k_{i-1} packets, then station i , start transmitting after these packets, and it completes its transmission after l_i time slot. In conclusion:

$$\begin{aligned}
T_{Wait_tx}^i &\leq SAT_MAX_i - l_i - k_i + k_{i-1} + l_i \leq \\
&\leq S + l_i + l_{i-1} + \sum_{\substack{j=1 \\ j \neq i-1 \\ j \neq i}}^N 2 \cdot k_j + k_{i-1}
\end{aligned}$$

□

Proposition 4.3

If the first group of l_i synchronous packets in the queue of station i have waited for a time equal to $T_{wait_per}^i$, before having the permission of transmission, the maximum time elapsed between the permission and the complete transmissions is equal to $l_i + 2 \cdot k_{i-1}$.

Proof

We've supposed that the first group of l_i synchronous packet in the queue of station i have waited for a time equal to $T_{wait_per}^i$, before having the permission of transmission. This means that the SAT rotation time (from station i to itself) is equal to its own maximum, and so, when the SAT come back it gives the permission at the synchronous packets, but it is not held by station i , because the station is satisfied. The k_{i-1} packets of station $i-1$ follow SAT. In the next round the SAT allow all the stations to send other k packets ($\sum_{j=1}^N k_j$). However, only station $i-1$ packets have to be considered. So, when the SAT come back at station i , it can be followed by $2 \cdot k_{i-1}$ packets.

At the end of this packets, station i can transmit its l_i synchronous packets.

□

Lemma 4.3

Let us consider the first synchronous message in the output queue of station i (divided into l_i packets) and let T_{wait}^i be the time elapsed between its arrival in the first queue position and its complete transmission. The following holds:

$$T_{wait}^i \leq S + l_i + l_{i-1} - k_i + 2 \cdot \sum_{j=1}^N k_j$$

Proof

Form Lemma 4.1 we know that $T_{wait_per}^i$ uses only one SAT round, and from Lemma 4.2 we know that $T_{wait_tx}^i$ uses two SAT round. So, we have:

$$T_{Wait}^i \leq 3 \cdot SAT_MAX_i \leq SAT_TIME_i [3].$$

However, from Proposition 4.3 we know that if the first synchronous packets in queue has wait the permission for a time equal to $T_{Wait_per}^i$ it is transmitted after a time equal to $l_i + 2 \cdot k_{i-1}$.

Hence we have:

$$\begin{aligned} T_{Wait}^i &\leq T_{Wait_per}^i + l_i + k_{i-1} = \\ &= S + l_i + l_{i-1} - k_i + 2 \cdot \sum_{j=1}^N k_j \end{aligned}$$

□

Theorem 4.2

Let us suppose that each synchronous message of station i is divided into l_i packets. Let $T_{Wait}^i(n)$ be the time elapsed between its arrival in the queue and its complete transmission by the n -th message. The following holds:

$$T_{Wait}^i(n) \leq (n+2) \cdot S + l_i + l_{i-1} + (n+3) \cdot \sum_{j=1}^N k_j$$

Proof

From Lemma 4.1 we know that $T_{Wait_per}^i$ uses one SAT round and from Lemma 4.2 we know that $T_{Wait_ex}^i$ uses two SAT round. Then, at each SAT round, station i sends l_i packets. So, the transmission of the n -th synchronous message is bounded by:

$$\begin{aligned} T_{Wait}^i(n) &\leq (n+2) \cdot SAT_MAX_i \leq \\ &\leq SAT_TIME_i [n+2] \leq \\ &\leq (n+2) \cdot S + l_i + l_{i-1} + (n+3) \cdot \sum_{j=1}^N k_j \end{aligned}$$

□

References

- [AGRA94] G. Agrawal, B. Chen, W. Zhao, S. Davari, "Guaranteeing Synchronous Message Deadlines with the Timed Token Medium Access Control Protocol", IEEE Transactions on Computers, vol. 43, no. 3, pp. 327-339, March 1994.
- [ANSI87] American National Standards Institute, "FDDI Token Ring Media Access Control", draft proposed, ASC X3T9.5, 1987.

- [ARVI91] K. Arvind, K. Ramamritham, J. A. Stankovic, "A Local Area Network Architecture for Communication in Distributed Real-Time Systems", *J. Real-Time Systems*, vol.3, no. 2, pp.115-147, May 1991.
- [BEEL90] R. Beeler, M. Potts and S. Rao, "D3Q-the dynamic distributed dual queue", *Proc. ISS'90*, 13th International Switching Symposium, Stockholm, Sweden 1990.
- [CHEN92] B. Chen, G. Agrawal, W. Zhao, "Optimal Synchronous Capacity Allocation for Hard real-time Communications with the Timed Token Protocol", *Proceedings of the 13th IEEE real-time Systems Symposium*, pp. 198-207, December 1992.
- [CHEN92b] B. Chen, G. Agrawal, W. Zhao, "Optimal Synchronous Capacity Allocation for hard real-time communications with the timed token Media Access Control Protocol", *Technical Report*, Computer Science Department, Texas A&M University, May 1992.
- [CIDO93] I. Cidon, Y. Ofek, "MetaRing - A Full-Duplex Ring with Fairness and Spatial Reuse", *IEEE Transactions on communications*, vol. 41, no. 1, pp. 110-120, January 1993.
- [CONT93] M. Conti, E. Gregori, L. Lenzini, "Metropolitan area networks (MANs): protocols, Modeling and Performance Evaluation", *Lecture Notes in Computer Science 729*, L. Donatiello and R. Nelson (Editors), 1993, pp. 81-120.
- [CONT95] M. Conti, L. Donatiello and M. Furini, "MetaRing+: an enhancement of MetaRing Access Control Protocol for supporting Real-Time applications", *Proceedings of the eighth Euromicro Workshop on Real-Time Systems*, L'Aquila, Italy, June 12-14, 1996, IEEE Computer Society 1996.
- [CONT96] M. Conti, S. Ghezzi, E. Marcacci, "Design and analysis of MetaRing++: an efficient protocol for supporting real-time distributed systems", *Proceedings of the ISCA international conference - parallel and distributed computing systems*. Dijon France, September 1996.
- [DIKE88] D. Dikeman, W. Bux, "Analysis and Tuning of the FDDI Media Access Control Protocol", *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 6, pp. 997-1010, July 1988.
- [FERR90] D. Ferrari, "Client Requirements for Real-Time Communication Services", *IEEE Communications Magazine*, November 1990, pp. 65-72.
- [FURI95] M. Furini, "Sistemi Distribuiti Real-Time: analisi del protocollo di accesso MetaRing", *Tesi di laurea*, Facoltà di Scienze MM.FF.NN., corso di laurea di Scienze dell'Informazione, Università di Bologna, Marzo 1995.
- [IEEE82] IEEE Standards for Local Area Network, "Token-Passing Bus Access Method", IEEE, New York, 1984.
- [LANN87] G. Le Lann, "The 802.3D Protocol: A Variation on the IEEE 802.3 Standard for Real-Time LANs", *INRIABP 105*, Le Chesnay Cedex, France 1987.
- [LAZA90] A. A. Lazar, A. T. Temple and R. Gidron, "MAGNET II: A metropolitan area network based on asynchronous time sharing", *IEEE J. Selected Areas Comm.* 8(8) 1990, pp. 1582-1594.
- [OHNI89] H. Ohnishi, N. Morita and S. Suzuki, "ATM ring protocol and performance", *Proc. ICC'89*, 1989, pp. 394-398.
- [MALC94] N. Malcolm, W. Zhao, "The Timed-Token Protocol for Real-Time Communications", *IEEE Computer*, January 1994, pp.35-41.

REFERENCES

- [MART67] J. Martin, "Design of real-time computer systems", Englewood Cliffs, NJ: Prentice Hall, 1967.
- [OFEK91] Y. Ofek, K. Sohraby, Ho-Ting Wu, "Integration of Synchronous and Asynchronous Traffic on the MetaRing Architecture and its Analysis", IBM Research Report, RC 17718, December 1991.
- [OFEK92] Y. Ofek, K. Sohraby, Ho-Ting Wu, "Integration of Synchronous and Asynchronous Traffic on the MetaRing Architecture and its Analysis", International Conference on Communications, 1992.
- [OFEK94] Y. Ofek, "Overview of the MetaRing architecture", Computer Networks and ISDN Systems 26, pp. 817-819, 1994.
- [PRYC91] M. De Prycker, "Asynchronous Transfer Mode", Ellis, Horwood Inc., 1991
- [STAN88] J. A. Stankovic, "Misconceptions about Real-Time Computing", IEEE Computer, October 1988, pp. 10-19.
- [STAN93] J. A. Stankovic, K. Ramamritham, "Advances in Real-Time Systems", IEEE Computer Society Press, 1993.
- [STRO88] J. K. Strosnider, "Highly Responsive real-time token rings", Ph. D. Thesis, Carnegie-Mellon University, Pittsburgh, PA, 1988.
- [ZHAN94] S. Zhang, A. Burns, "EMCA: an optimal Synchronous Bandwidth Allocation scheme for guaranteeing Synchronous Message Deadlines with the timed token protocol in a FDDI Network", Department of Computer Science, University of York, Technical Report YCS244, 1994.