A TRANSPORT PROTOCOL FOR REAL TIME SERVICES ON A PACKET SWITCHING NETWORK

Internal Report C91-01

Gennaio/1991

Giuseppe Anastasi
Marco Conti
Enrico Gregori

A TRANSPORT PROTOCOL FOR REAL TIME SERVICES ON A PACKET SWITCHING NETWORK

G. Anastasi*, M. Conti**, E. Gregori**

* Graduated in Electronic Engineering
**CNR - Istituto CNUCE, Via S. Maria, 36
56100 Pisa, Italy
Tel: (50) 593111
E-Mail: Gregori @ ICNUCEVM.CNUCE.CNR.IT

Abstract

Originally network users required only a reliable transportation service but they have no other requirements. On the contrary, new applications may require some other performances such as an upper bound in delay and/or in delay variability, a packet loss rate not greater than a certain threshold, a minimum guarantied throughput and others. Therefore the need of a transportation system able to provide a service with a guarantied Quality of Service (QoS) is felt. In this paper a Transport Protocol for Real time services (TPR) is presented. It can be used to enhance the QoS of a packet switched network which provide some particular guaranties. The resultant transportation system is able to meet the needs of users with temporal constraints such as digitized video and voice, alarm messages and so on.

1. Introduction

Originally networks were designed to allow a user to send data or to access to a remote database. This users required only a reliable transportation service but they were not interested in other particular performances. With the development of new applications the situation has changed. Voice users requires an upper bound in delay with which information is tranferred from source to destination while they can tolerate a certain degree of delay variability and packet losses. Video users can tolerate a larger end to end delay but they are sensitive to delay variability and packet losses. The transmission of an alarm message needs both a delay constraint and an high reliability. Some file transfers applications could require a minimum guarantied throughput.

Therefore the need of a networks able to provide a service with a guarantied performance is felt. A communication system in which the user is allowed to express its requirements in terms of Quality of Service (QoS) is said carrying out a *real time service* ([FERR90c]). The required performance may not comprise delay constraints but, as example, only a minimum throughput.

Obviously even if a network is able to provide a real time service it may not satisfy completely user wishes. Therefore some software layers are placed on top of it in order to enhance the quality of service.

In this paper a Transport Protocol for Real time services (TPR) is presented. It can be used to enhance the Quality of Service (QoS) of a packet switched network which provide some particular guaranties. Such transport protocol is oriented especially towards users with temporal constraints and without stringent reliability requirements and particularly to video and/or voice applications

The resulting transportation system architecture is showed in figure 1.

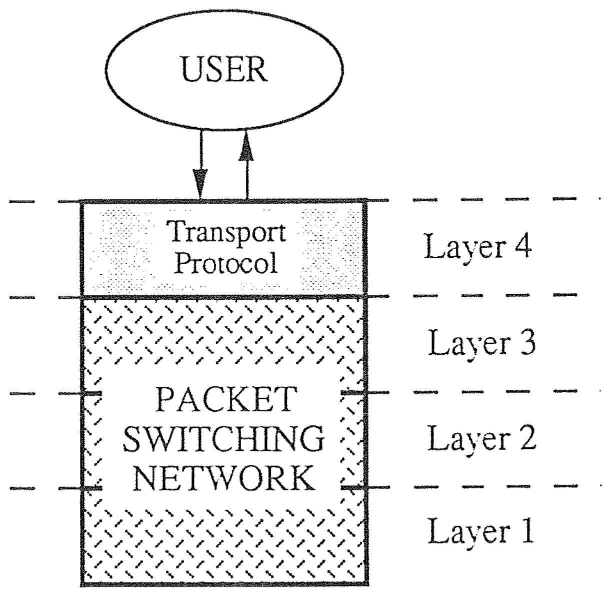

Fig. 1 - Architecture of the transportation system

In the following Section we will give a user classification based on quality of service needed for a correct information transfer. In Section 3 we will see the QoS a network must provide so that the TPR protocol can be used on top of it. The performances provided by TPR and the way such performances are obtained relying on underlying network services will be analyzed in Section 4. In Section 5 two cases of study will be presented: they will show the feasibility of using the TPR protocol with an FDDI (Fiber Distributed Data Interface) metropolitan network and with a geographical network. Our conclusion will be presented in section 6.

2. User requirements

Users of a generic transportation system have been divided in four classes as shown in table 1. Each class can be characterized by the quality of service required as it appear from table 2 [VERB90].

- Class 1	Continuous Bit stream Oriented (CBO)
- Class 2	Variable Bit Rate video and audio (VBR)
- Class 3	Connection oriented data transfer
- Class 4	Connectionless data transfer

Table 1: User classes

	Class1	Class 2	Class 3	Class 4
Timing between source and destination	Related		Not related	
Bit rate	Constant	Variable		
Connection mode	Connection Oriented			Connection-less

Table 2: Characteristic of the user classes.

Information is passed to the transportation system in blocks called *Service Data Units* (SDUs). The time between two consecutive SDUs is the *interarrival time*.

Both class 1 and class 2 users need a synchronization between source and destination. In other words they require the bit patterns at the sending and the receiving sides must be the same with only a temporal shift. Thus every SDUs needs to experience a constant delay from the source to the destination (end to end delay). Besides the frequency with which the SDUs are delivered to the destination by the transportation system must be the same with which they had been passed by the source.

As it appears from table 2 synchronization is not required by data transfers. These have generally a variable bit rate and can require either a connection oriented service (class 3) or a datagram one (class 4).

In addition to those outlined above, which are common to all users of the same class, users can have other particular needs .

In the case of a telephone application which generally belongs to class 1 not only the end to end delay must be constant for each SDU (i.e. for each sample) but it must be less than a certain threshold in order to avoid a degradation of the communication quality.

For a file transfers (class 3) a high reliability of the transport service is required while delay usually is not important.

In a distributed real time system in which sensors send alarm messages to a control device (class 3 or class 4 application) both a reliable transport service and a bounded end to end delay are required.

3. Network description

Generally speaking the quality of service provided by a packet switched network is very far away from meeting user requirements. Therefore it is necessary to place one or more software layers in order to enhance the quality of service offered by the network. But not every user needs can be satisfied relying on a generic network. As an example with a CSMA/CD LAN it never will be possible to guarantee an upper bound for delay and delay jitter even adding upper layers.

In general we can say that there is no possibility to provide a real time service if the underlying layer doesn't offer some guaranties. In the following of this section we will analyze the QoS of a network required by TPR protocol:

Maximum packet loss rate: it is necessary to know with the best possible precision the maximum packet loss rate of the network.

Upper/lower bound on packet transfer delay: the network must provide an upper and lower bounds on packet transfer delay. This allow us to handle users that want to transfer messages with stringent temporal constraints.

Guarantied bandwidth: only if the network guaranties the bandwidth on an interval I the TPR can avoid congestion on its users.

In conclusion our network must be able to guarantee un upper bound on packet transfer delay and the absence of congestions at least for high priority packets. It is also important to estimate the minimum packet transfer delay with the best possible approximation.

4. Transport protocol

The TPR protocol provide a connection oriented service. In other words before proceeding with information transfer a user must request to the transport protocol to set up a connection. This will be closed at the end of the communication.
During the set up phase the user notifies its requirement to the transport protocol by means of a set of QoS parameters. These specify [1]:

- maximum end to end delay D_{umax}
- maximum delay jitter J_{umax}
- maximum packet loss rate L_{umax}

Two additional services can be required by the user:

---

[1] The subscript u in the follwing QoS parameters indicates that the related quantity is related to the user.

- Reconstruction of the sender frequency at the destination side in order to eliminate the *drift* between the sender's and receiver's frequencies.

- Lost packet compensation

A user can indicate only the QoS parameters in which it is interested. So a user that needs only a bounded end to end delay has to specify only a value for D_{umax} but can not to specify the other parameters.
During the set up phase the user notifies also its characteristic in terms of: bit rate, packet length and packet interarrival time. On the basis of this information the TPR protocol decides if it is able to satisfy user requirements. If so the connection is set up and the information transfer can take place. Otherwise user is informed about failure of its connection request.
The TPR protocol has a *bandwidth allocation* mechanism to assign a user with the necessary bandwidth for satisfying its requirements without deteriorate the quality of service that must be provided to the other users. If every "time constrained" user is assigned with a bandwidth greater than or equal to its maximum bit rate and if allocated bandwidth is, in all, not greater than network capacity then it is possible to satisfy temporal constraints while avoiding congestions
Users without temporal requirements are not assigned with a guarantied bandwidth. So they can transmit their traffic only when network capacity is not completely used by "guarantied bandwidth" users.
For variable bit rate user (class 2 and class 3) bandwidth allocation can be made following a probabilistic policy based on *statistical multiplexing* . The transport protocol has a *flow control* mechanism whose function is to block some users when all would wants to transmit simultaneously.
In order to allow the transport protocol to properly operate every user must operate in accordance with its declared characteristic. In any case the TPR protocol has a *rate control* mechanism to avoid that a user can send information with a bit rate greater than declared.
In the following we will try to see how it is possible to guarantee a given quality of service using services of the underlying network. In order to avoid confusion we will indicates quantities with the same name but related to different layers of the transportation system architecture with different subscripts: so "u" is for user, "t" indicates the transport layer and "n" is related to the underlying network.

4.1 Delay bound

We know the underlying network guarantee an upper bound in transfer delay:

$$D_n \leq D_{nmax} \tag{4.1}$$

The total delay at the interface between the user and the transport protocol is given by:

$$D_t = D_{tp} + D_n \tag{4.2}$$

where D_{tp} is the *packetization delay* that is the time necessary to form a TPR packet. Introducing the (4.1) in the (4.2) we obtain the following relation:

$$D_t \leq D_{tp} + D_{nmax} \tag{4.3}$$

that expresses the delay bound guarantied by TPR.
Obviously necessary condition to accept a connection request forwarded by an user is that:

$$D_{tmax} = D_{tp} + D_{nmax} \leq D_{umax}$$

We can observe that in the preceding relation D_{nmax} is related to the underlying layers but the TPR protocol has the possibility to choose a smaller packet size in order to lower the packetization delay D_{tp}. But doing do the network overhead increases. So a compromise between TPR packet length and D_{tp} value has to be reached.

4.2 Delay jitter Bound

In a packet switched network delays vary from packet to packet. In addition to a delay bound in packet delivery a user can require that delay variability is not greater than a certain threshold. Therefore it must be:

$$E\{D_t\} - J_{umax} \leq D_t \leq E\{D_t\} + J_{umax} \tag{4.4a}$$

with the condition:

$$E\{D_t\} + J_{umax} \leq D_{umax} \quad (4.4b)$$

where J_{umax} indicates the maximum jitter the user can tolerate while $E\{D_t\}$ is the average end to end delay introduced by the transportation system.
Every packet is transferred by the underlying packet switched network with a delay:

$$D_{nmin} \leq D_n \leq D_{nmax} \quad (4.5)$$

Calling $E\{D_n\}$ the average network delay if it is:

$$J_{umax} \geq \max\ [(D_{nmax} - E\{D_n\})\ ;\ (E\{D_n\} - D_{nmin})] \quad (4.6a)$$

and simultaneously:

$$E\{D_n\} + J_{umax} \leq D_{umax} \quad (4.6b)$$

then the user requirement on delay jitter is directly satisfied by the underlying network (see figure 2).

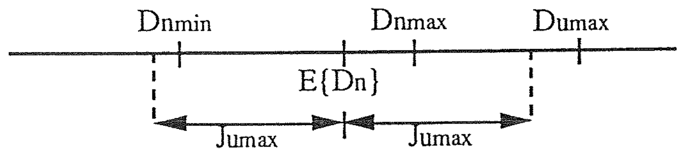

Fig. 2 - User requirement on delay jitter is directly satisfied by the underlying network.

Otherwise TPR has a *dejittering* mechanism to completely eliminate delay variability. But, as we will see, this is possible only at the cost of a greater delay bound.
When it reaches the destination side the first packet of a communication is bufferized and delayed by a quantity D_{tj} equal to:

$$D_{nmax} - D_n$$

if D_n is known or equal to:

$$D_{nmax} - D_{nmin}$$

when we don't know the actual delay. For successive packets if a packet has been sent x seconds after the preceding the same packet will be delivered to the destination exactly x seconds after the preceding. The destination TPR needs to know the temporal distance of a packet from its preceding: if this is fixed it can be communicate at the connection set up phase otherwise it must be communicate with the same packet. Using the technique described above every packet experiences a constant delay whose value is equal to:

$$D_{tp} + D_{nmax} + D_{tj}$$

So jitter absorption process increases the delay bound. There is no need to say that, in this case, condition to be verified in order to accept or reject a connection request is:

$$D_{tp} + D_{nmax} + D_{tj} \leq D_{umax} \quad (4.7)$$

It is also obvious that the dejittering buffer size must be computed in order to avoid buffer underflow or overflow.

4.3 Reliability guaranties

As regards the reliability problem the TPR can rely only upon underlying network service due to the fact that it has no mechanism to recover lost packets. The technique of retransmission usually used for recovering missing packets is unuseful for a protocol like TPR whose users are generally time constrained.

Only in the case of users which have to sent short messages with stringent delay constraints (e. g. alarm messages) the TPR protocol can improve the underlying network reliability by sending more copies of the message. At the destination side the first received copy will be delivered while the others discarded.

In all other cases a connection request specifying a threshold L_{umax} for packet loss rate can be satisfied if and only if it is:

$$L_{nmax} \leq L_{umax}$$

where L_{nmax} is the maximum packet loss rate the underlying network may introduce.
We have to point out that the packet loss rate provided by the TPR protocol to the user can be higher than that at the interface between TPR and underlying network if some packet can be lost at the transport level. As example this is the case when statistical multiplexing is adopted.

4.4 Lost packet compensation

This mechanism is needed exclusively by class 1 and class 2 users. For such an user a lost packet can lead to the loss of synchronization between sender and receiver. Besides in the cases in which compression techniques in signal coding are adopted a missing sample doesn't allow a correct decoding of the next ones. In addition we have to consider that a lost packet means that a piece of information is not received by destination user.
Since we cannot recover a missing packet by retransmission we must try to reduce the negative effects of a lost packet as much as possible. Fortunately both video and voice applications can tolerate the alteration of some information without appreciable damage for quality communication. Therefore concealment techniques are used. That is a lost packet is replaced with a silence packet in voice applications and by a preceding packet in video applications. In the second case the replacing packet can be the corresponding packet in the previous line of the same frame (*time masking*) or the corresponding packet in the same line of the previous frame or field (*time masking*) or a more complex combination of preceding packets. A user can choose among the concealment techniques provided by the TPR protocol the most adapt with its coding.

4.5 Drift correction mechanism

This mechanism is oriented only to class 1 and class 2 users and, among these, in particular to those with a larger bit rate. It may be useful to remember such users pass their own information periodically in SDUs which have variable length in the case of VBR applications.
The SDUs are sent by the transmitting user to the TPR with an f_t frequency nominally equal to the inverse of the interarrival time, T_{SDU}. They are delivered to the receiving user with an f_r frequency of equal nominal value. But since the two frequencies are obtained, in general, from different

clocks (each of which has a drift from the nominal value $f_n = \frac{1}{T_{SDU}}$) the two frequencies are not equal.
Now even a slight difference between f_t and f_r might create relevant problems especially if the bit rate of the applications is high. In fact, if f_t is greater than f_r, the information tends to accumulate in the TPR outgoing (dejittering) buffer which might overflow causing information loss. On the contrary, if f_t is lower than f_r, the buffer tends to empty itself. The continuity of the information flow towards the receiving user may then no longer be guarantied. One of the solution might be to extend the waiting time in the buffer in order to avoid it from emptying and to lengthen the buffer in order to avoid overflow. This solution is feasible enough for applications with a very low bit rate, as for example, the telephone but, in the case of video, the buffer, which is already long, would reach very large dimensions. For these applications we could absorb the difference between the two frequencies by repeating or skipping a whole frame occasionally. This technique is used (together with some other techniques) in the Prelude network ([COCH85], [DEVA88]). The main disadvantage of this method is that the buffer must be lengthened to hold an extra frame. Moreover, skipping or repeating a frame causes vision problems. Hence it must be rarely done, let say, once every transmission hour. The problem is that we might not have a clock able to guarantee such a performance.
These considerations led us to define a drift compensation mechanism to render the two frequencies as equal as possible.
There are two ways to obtain this result. The first is based on the observation of packets arrival sequence and the second is based on the observation of the buffer filling level. A first type method was experienced in the Prelude network [COCH85] but it cannot be used in cases of very wide jitter. Another proposal of the same kind is presented in [VAKI89]. It is valid for any amplitude of jitter but it is intended for a configuration in which the passage of information from and to the user comes about bit by bit. Moreover, its applications scheme is not very simple.
The methods based on the observations of the buffer filling level do not reconstruct the f_t frequency but the difference between the transmitter's frequency and the receiver's frequency. In these methods the problem is that the buffer level varies not only due to frequency drift but also because of the jitter effect. The various proposals ([COCH85], [DEPR87], [KIM88]) are different in the way they face this aspect but they all suppose that the

jitter has a small value with respect to the total delay and to the packet generation period. Obviously this condition is not a general one. As example in an FDDI metropolitan area network with a 20 Km ring delay jitter must be in the order of tens of milliseconds if a good network utilization is required.

We have developed a different method which is not only applicable in very wide jitter situations but it is simple and robust as well. This method is based on the observation of the buffer filling level expressed as the number of packets present. This level is recorded each time a packet has been delivered. We indicate with Xi the buffer filling level at the time the receiver has received i-th packet. If we consider the Xi values inside a certain time interval we can see that they may greatly vary due to the dispersal of packet arrivals that might be very great. We can consider the mean level of the buffer, X^*, during this interval given by the following expressions:

$$X^* = \sum_{i=n+1}^{n+F} \frac{X_i}{F} \tag{4.8}$$

In the preceding relation the extremes of the sum denote the sequence number of the packets absorbed by the receiver, respectively first and last, during the measurement interval. The length of this interval is given by:

$$T_m = F \cdot T_P \tag{4.9}$$

where T_P is the TPR packet period. In this way, an average level of buffer X^* is associated to each temporal window of T_m duration. If ΔX^* is the variation of the mean level with respect to the preceding interval the (relative) drift between the frequencies is:

$$\Delta = \frac{f_r - f_t}{f_n} = -\Delta X^* \tag{4.10}$$

In this way it is possible to reconstruct the transmitter's frequency with the following relation:

$$f_t' = f_r + \Delta X^* \bullet f_n \tag{4.11}$$

In the preceding relationship we put f_t' and not f_t because, due to the inevitable errors, the reconstructed frequency does not exactly correspond to the transmitter's frequency. Nevertheless, since the mechanism is constantly active, eventual errors can be determined in the next step. The accuracy of the method depends on how long one choses the interval duration of the T_m observation window. For certain applications the instantaneous variation of the receiver's frequency could cause very evident problems.[2] In these cases it is better to carry out a single accurate correction for which the T_m value must be sufficiently great. In the cases in which the correction of the f_r does not cause significant problems we can choose a smaller value for T_m and carry out more corrections.
Since the first correction is performed after a time equal to $2 \bullet T_m$, in order to avoid a buffer underflow, we need to delay the first packet exit at the beginning of the transmission by an adequate interval D_{td}.
Therefore the new delay bound in presence of drift and when jitter mechanism is activated is:

$$D'_{tmax} = D_{tp} + D_{nmax} + D_{tj} + D_{td} \tag{4.12}$$

Obviously also the outgoing buffer must be opportunely lengthened to avoid buffer overflow in the initial period (see [ANAS90a] for details).
The method herein described is extremely simple and can be rendered robust as much as we want by increasing the observation interval. It is generally applicable in so much as it does not place limits on the jitter amplitude.
The effectiveness of the method has been proved via a simulation test ([ANAS90a]).

5. Cases of study

In this section we will analyze two networks with which it is possible to use the TPR protocol. The former is a metropolitan FDDI network while the latter is a wide area network with a general enough architecture.

---

[2] In the case of video it is better to postpone the correction of the receiver's frequency at the begining of the new frame.

5.1 FDDI network

FDDI is a fiber optic large band (100 Mbit/sec) network with a *dual ring* configuration and a *timed token* Medium Access Control (MAC) protocol ([ANSI87]). This protocol foresee two classes of traffic:

- synchronous traffic
- asynchronous traffic

The former can be sent every time a station has received a token and is transmitted with guarantied bandwidth while asynchronous traffic can be transmitted only if the token rotation time[3] has been less than a the value of a parameter, *T_Opr*, that can be negotiated by the stations.
The MAC protocol is deviced so that the average token rotation time is less than or equal to T_Opr and the time between two successive token arrivals to a station is always less than or equal to 2•T_Opr [SEVC87].
From these properties it is possible demonstrate that:

If information is passed to FDDI in packets with fixed interarrival time, T, and if such packets are sent as synchronous traffic and the bandwidth allocated to the user is greater than or equal to the maximum bit rate of the user then the maximum queuing delay for a packet waiting transmission results:

$$0 \leq D_{nq} \leq 2 \cdot T\_Opr + T \qquad (5.1)$$

(See [ANAS90b] for demonstration).
Other delays experienced by a packet in the FDDI network are related to transmission (D_{ntr}) and propagation across the fiber (D_{npr}). Both of them can be bounded and considered constant so that the total delay results:

$$D_{ntr} + D_{npr} \leq D_n \leq 2 \cdot T\_Opr + T + D_{ntr} + D_{npr} \qquad (5.2)$$

From (5.2) we can see that the difference the delay variability

[3]This is the time between the arrival of the last token and that of the present token to the station.

$$D_{nmax} - D_{nmin} = 2 \cdot T\_Opr + T \qquad (5.3)$$

is influenced by T_Opr. We could choose a small value for it but in this way also network utilization decreases. Therefore a compromise need to be reached.
As regards reliability an FDDI network has a Bit Error Rate (BER) within the range $(2.5 \cdot 10^{-10}; 10^{-12})$.
The feasibility of using the TPR protocol with an FDDI network has been confirmed by a set of simulation tests ([ANAS90a]).

5.1 Wide Area Network

A scheme for obtaining real time services on top of a wide area network is presented in [FERR90a] and [FERR90b]. The network considered there may have an arbitrary topology and may consist of several networks of various type (LANs and WANs). In it a message can go from a source host to a destination host passing through a number of intermediate nodes some of which may be gateways connecting different nodes. Some of the links between store-and-forward networks may not be store-and-forward networks, provided that their total delay can be bounded. Therefore they can be very high speed circuit switched trunks or FDDI networks while it is not possible to have contention based networks such as Ethernet.
Using this scheme a real time channel can be established between two users. During the channel set up phase the requesting user must declare its traffic characteristics and performance requirements. These are given by:

- the source to destination delay bound

- the maximum loss probability

Two types of real time channels are possible:

- deterministic channels, in which the packet delay D_n is absolutely bounded ($D_n \leq D_{nmax}$).

- statistical channels, in which it is guarantied that the probability that the delay D_n exceed the delay bound D_{nmax} is greater than a certain value P_{min} (Prob $[D_n < D_{nmax}] \geq P_{min}$).

Obviously in this paper we will not analyze the scheme introduced above for which all the details can be found in the bibliography.
The possibility of using the TPR protocol with such a network is quite evident. The only condition is that the network must also be able to provide a minimum value for packet delay or an estimation as good as possible of it. This estimate can be computed considering in each node a packet processing time equal to zero.

6. Conclusions

In this paper we have presented a Transport Protocol for Real time services (TPR). Such a protocol can be used with any network that guaranties: a packet loss rate not greater than a certain value and an upper and a lower bound on packet transfer delay. In addition the network must guarantee the bandwidth on an interval I at least for a class of users.
Relying on services provided by such a network the TPR protocol is able to guarantee, in turn:

- a bounded end to end delay
- a null or bounded delay jitter
- a packet loss rate not greater than a certain threshold

In addition the user can also require:

- the reconstruction of the sender frequency at the destination side.
- a lost packet compensation

From the QoS provided we can see that this transport protocol is especially oriented towards video and/or voice users. However it can be used in general with users with some temporal constraints and no stringent requirements as regard reliability. In the cases of short messages it is possible obtain a high reliability too by sending more copies of the message.

References

[ANAS90a] Anastasi G., Conti M., Gregori E. - "TPR: a Transport Protocol for Real Time Services in an FDDI Environment" - Proceeding of the Second IFIP WG6.1/WG6.4 International Workshop on Protocols for High Speed Networks, November 1990.

[ANAS90b] Anastasi G. - "La Trasmissione del Video e della Voce su una Rete FDDI: Definizione e Valutazione del Protocollo di Trasporto" - Tesi di laurea in Ingegneria Elettronica, Universita` di Pisa, 1990, in Italian.

[ANSI87] American National Standards Institute - "FDDI Token Ring Media Access Control", draft proposed - ASC X3T9.5 1987.

[COCH85] Cochennec J. Y., Adam P. and Houdoin T. - "Asinchronous Time Division Networks: Terminal Synchronization for Video and Sound Signals" - Proc. IEEE GLOBECOM 1985, pp. 25.6.1-25.6.4.

[DEPR87] De Prycker M., Ryckebush M., Barri P. - "Terminal Synchronization in Asynchronous Network" - Proc. ICC '87, Seattle, June 1987, pp. 22.7.1-22.7.8.

[DEVA88] Devault M., Cochennec J. Y. and Servel M.- "The "Prelude" Experiment: Assessments and Future Prospects" - IEEE JSAC vol 6, n. 9, December1988, pp. 1528-1536.

[FERR90a] Ferrari D., Verma D. C. - "A Scheme for Real Time Channel Estabilishment in Wide Area Networks" - IEEE JSAC vol 8, n. 3, April 1990, pp. 368-379.

[FERR90b] Ferrari D., Verma D. C. - "Real Time Communication in a Packet Switching Network" - Proc. Second IFIP WG6.1/WG6.4 International Workshop on Protocols for High Speed Networks, November 1990.

[FERR90c] Ferrari D. - "Client Requirements for Real Time Communication Services" - IEEE Communication Magazine, November 1990, pp. 65-72.

[KIM88] Kim C. K., Lee S.H., Wu L. T. - "Circuit Emulation" - International Journal of Digital and Analog Cabled Systems, vol. 1, 1988, pp. 245-256.

[SEVC87] K. C. Sevcik, M. J. Johnson, "Cycle Properties of the FDDI Token Ring Protocol" - IEEE Transaction on Software Engineering, vol. SE-13, n.3, March 1987.

[VAKI89] Vakil F. - "On Source Timing Recovery for Circuit Emulation in ATM Networks" - Proc. IEEE GLOBECOM 1989, pp. 50.4.1-50.4.8.

[VERB90] Verbiest W. Elewaut L. - "Packet video" - IEEE INFOCOM'90, Tutorial #2, June 1990.