

# On the Definition and Representation of the Quality of Service for Multimedia Systems

M. Diaz, K. Drira, A. Lozes, C. Chassot  
LAAS du CNRS

7, avenue du Colonel Roche - 31077 Toulouse - France

Phone number : 61 33 62 00 - Fax : 61 33 64 11 - e-mail : diaz, drira, lozes, chassot@laas.fr

## Abstract

This paper presents a new geometrical view of the concept of Quality of Service (QoS). QoS is of utmost importance and quite often referred to when handling information in distributed communications systems. Nevertheless, no formal definition exists and consequently no evaluation and no comparative analysis of QoS can be done. This paper is considered as a first step towards a well defined approach to access the QoS ; such approaches have to be developed to guarantee the quality of high performance multimedia data transfers.

Nowadays current protocols are either connectionless or connection-oriented protocols. A new concept, a Partial Order Connection (POC), has been proposed. More precisely, a POC is an end-to-end connection that provides a partial order service where the received SDU objects can be delivered to the users in a order that is different from the order they had when they were sent to the service. It is first shown how this new concept of order can be used to represent one aspect of the QoS. Furthermore it is shown how it can be extended to represent losses of messages and to handle time constraints in an integrated way. These two extensions lead to the definition of a space for representing general and specific services. Then the definition of the QoS is given as a volume inside this space and comparing QoSs means comparing the volumes they define within this space.

## Keywords

Partial order connection, quality of service, transport protocol, transport service, multimedia application

## 1 INTRODUCTION

This work presents a new geometrical view of the concept of Quality of Service (QoS). It has been conducted inside the CESAME (Conception formElle de Systèmes hAutS débits Multimédias coopEratifs) project [Diaz94a] [Annal94] when investigating methodologies and approaches able to define and represent the QoS of multimedia networks. QoS is of utmost

importance and quite often referred to when handling information in distributed communications systems. Nevertheless, no formal definition exists and consequently no evaluation and no comparative analysis of QoS can be done to guarantee the quality of high performance communications.

Present data transfer protocols use either connectionless (CL) or connection-oriented (CO) paradigms. A new concept, a Partial Order Connection (POC), has been proposed in CESAME [Amer93b] [Diaz94b] and is recalled in this paper. It will be shown in particular that connectionless and connection-oriented protocols are two specific cases of this new concept : connectionless and connection-oriented protocols appear to be two extreme cases of POC protocols, POC being a new and more general definition of a connection. An end-to-end partial order connection allows two users to define and use for transferring messages between their communicating entities any partial order services and protocols. In a POC, the objects can be delivered to a receiving user entity in an order that is different from the one that has been used by the sending entity. The acceptable difference between the serial emission of the sending user and the different possible receptions of the receiving user precisely results from the definition of the (selected) partial order. In particular, these different and acceptable possible receptions lead to transfer speed-up and resources savings at both sending and receiving sides.

[Pete89] defines a partial order on a set of ad-hoc defined messages communicated by a set of distributed processes. It implements a very expensive protocol Psync that encodes the partial ordering within each message.

Ahuja shows that some conclusions derived on the design of distributed algorithms need not have required FIFO ordering as a base assumption [Ahuj90]. He assumes that a sending process dynamically builds the partial order and that no objects are ever lost [Shaf92]. Also, Ahuja's four data types do not permit all possible partial orders as does our POC service and this solution does not conceptually extend the existing protocol approaches.

[Amer93a] presents a reliable version of POCs which requires all transmitted objects eventually to be delivered. Using Petri net based models to represent serial parallel partial orders, a metric capturing the complexity of a partial order,  $e(P)$ , is defined as the number of its linear extensions. [Amer93b] discusses an unreliable version of a POC which permits the service to lose a predefined subset of the objects. [Diaz93] extends the previous results and proposes new metrics that allow the designers to evaluate the consequences of the losses of messages during the transmissions, in terms of both entropy and uncertainty. These two metrics can be used to define the maximum number of losses that lead to the weakest (simplest) implementation of the underlying transmission network.

Reliability and timing constraints can be added to the POC to define a space of protocols. This space can then be used to establish some relationships between the set of the usual non timed protocols and the set of the timed protocols. In particular, it is proposed in this paper to use this space as a support for proposing a conceptual definition of the notion of quality of service. This definition allows the comparison of different services and provides the designer and the user with a method to define the relation "a service S1 conforms to a service S2".

Starting from connectionless (CL) and connection-oriented (CO) protocols, the concept of unreliable POCs will be recalled and discussed in Section 2. Then Section 3 shows how this concept can be extended to handle time constraints in an integrated way, leading to the definition of a complete (Order, Reliability, Time) space for designing general purpose and application dedicated services and communications. Then the definition of the Quality of Service is given in Section 4. It will be shown that QoSs can be defined as a volume inside this space. Consequently, it will follow that comparing different QoSs or checking the conformity of QoSs means comparing the volumes they define inside this space.

## 2 FROM CL AND CO PROTOCOLS TO PARTIAL ORDER CONNECTIONS

### 2.1 Order

Current data transfer protocols and services use either CL protocols, as UDP, or CO protocols, as TCP. In CL datagram services and protocols, data units are not related to each other and transmission errors are not recovered. On the other hand, CO protocols, using a sequential numbering, usually provide a reliable delivery of protocol data units. It follows that :

- a) CL services and protocols provide no guaranty on reliability and, as data units are independent, enforce no order.
- b) CO services and protocols guarantee a total reliability and a serial order due to the numbering, i.e. a total order.

Note that new high speed protocols such as XTP [XTP92] may be used in unreliable CO mode.

Consequently it has been proposed in [Amer94] [Diaz94b] to extend CL and CO services and protocols by introducing :

- one axis, R, dealing with reliability, where 0 means no reliability and 1 means full reliability,
- one axis, O, dealing with order, where 0 means no order and 1 means total order.

The resulting system is given in Figure 1(a), where CO services and protocols, having order=1 and reliability=1, define one point at the upper right corner, and CL protocols, having order=0 and reliability=0, define one point at the lower left corner.

It follows from Figure 1(a) that a large set of different services and protocols can be designed. These services and protocols are then characterized by their two parameters, reliability and order. As a consequence, reliability and order become two parameters, and the set of all partial order connections corresponds to the surface given in Figure 1(b).

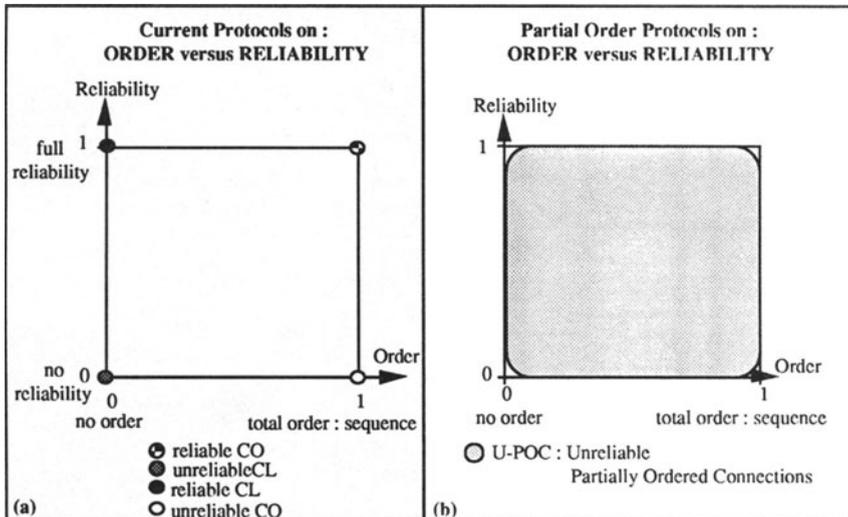


Figure 1 : Current protocols versus partial order protocols

Discussing order comes from the definition and handling of displayed objects. Let us assume that one piece of text must be received and that after receiving it, now, in parallel, one text and two sequences of audio and video data have to be presented to the user. Let us assume that the first PDU of the second text (the one to be presented in parallel with the audio and the video streams) is lost. Then, if a reliable CO delivery is implemented, retransmission is needed. It follows that the receiver may or may not store all PDUs received after the one that is lost. If it does not store them, then all of them need to be retransmitted again and the sending user has to keep the sent PDUs in its buffers. In the first case, where the receiver stores out-of-sequence PDUs, large memory is needed at the receiving side. It follows that storing PDUs requires extra memory while resending PDUs needs memory longer at the sending side, increases the needed bandwidth and delay. Of course, these constraints become quite important when required transfer speed increases, which is the case for high speed protocols.

Now, in the considered example, from the application point of view, it could be quite possible to start the display of the video before receiving the second text. If a CO protocol is used, after having noticed that the second text contains one lost PDU, it follows that the audio and the video PDUs cannot be displayed to the user, because of the total order. Waiting for receiving the text and delaying the audio and video information leads to unnecessary delays and/or use of buffers. Note that a delay of 0.5 seconds at 100 megabits/second means the receiver to receive  $50 \cdot 10^6$  bits (the numbers of bits of the PDUs) and to lose them or to store the resulting information in its buffer.

This is precisely what partial order avoids. In this simple case, the partial order between the texts, the audio and the video streams makes explicit that each of the second text, the audio and the video, are independent and so can be delivered in any order.

[Amer93a] proposes to characterize the partial orders from the point of view of the network, or more generally from the point of view of the service provider, in terms of sequences that can be accepted by the receiving user. The set of allowable sequences at the receiving side is defined as the set, denoted  $e(P)$ , of the linear extensions of the partial order  $P$ . It has been proposed to express the complexity of  $P$ , as dependent on the number,  $e(P)$ , of linear extensions associated to the partial order. It will be seen that the higher the number of possible linear extensions, the easier it is for the network to implement the requested connection. Let us comment on how the number of linear extensions of a partial order  $P$ ,  $e(P)$ , can be used to characterize a partial order service implementing partial order  $P$ .

First, for 4 unordered SDU service primitives, there are  $4! = 24$  delivery orderings ( $n!$  for  $n$  SDUs) that satisfy the service, as any order can be accepted by the receiving service user. On the other hand, a sequential (total) ordering means that only one delivery order - the same as the one that has been used by the sender - can be sent to the receiving user : for any other received ordering the sending and receiving entities will have to enforce the sequence, and so to take appropriate (resending) actions.

As the number of the acceptable delivery orderings is the number of linear extensions of the partial order, it is proposed to use  $e(P)$  as a characterization of the service, and more precisely of the facility to actually implement the connection : the larger the number of linear extensions is, the easier it is to implement the service.

Let us now consider the screen given in Figure 2(a). The four objects are displayed on this screen when they are received from a remote data base. Using TCP, the used service and protocol will be such that a sending numbering will be selected by the sending user, Figure 2(b), and enforced by the receiving entity. If one object is lost, then it will have to be transmitted again. If the lost object is the one whose number is 1, then a go back N protocol will have to retransmit all four objects. If the order is now the one of Figure 2(c), then the loss of object 1 will not lead to any constraint for the delivery of objects 2, 3 and 4, as it is made explicit that object 1 and objects 2, 3, 4 are independent. Of course, the example here will lead to consider the order given in Figure 2(d) as the "best" order for this remove presentation on a screen as 1, 2, the set {3, 4} are independently displayed and 3 is behind 4. The concept of

POC precisely encapsulates this notion of relative ordering between displayed or presented objects

When considering the network, it is easier to implement order 2(d), as it is less constrained than order 2(b), which is more constrained. This intuition is captured by the number  $e(P)$ , the number of linear extensions of a partial order.  $e(P)$  for Figure 2(b) is 1 (order 1;2;3;4), while it is 4 for Figure 2(c) (1;2;3;4 2;1;3;4 2;3;1;4 and 2;3;4;1) and 6 for Figure 2(d). Considering an imperfect network as a network that can modify the sending order, it follows that the more orders can be considered as correct at the receiving side, the easier it is for the network to ensure the transfer, as less wrong orders have to be corrected.

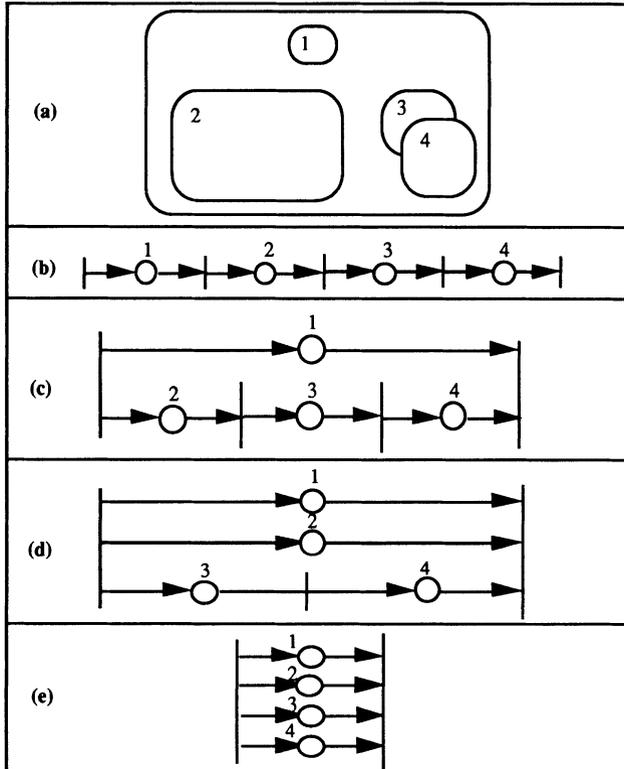


Figure 2

$e(P)$  also allows one to quantify and compare two or more POCs. Unfortunately, there is no formula for calculating  $e(P)$  for an arbitrary partial order. A formula to compute  $e(P)$  exists in the case where the partial orders under consideration are limited only to those that can be composed by serial-parallel partial orders. Such a way of handling computations and leading to a serial-parallel composition appears to be coherent for instance with a serial-parallel Petri net based modelling. Of course other representations can be selected but timed serial-parallel Petri nets have been used for modelling the synchronization and ordering of multimedia complex objects in [Litt90a] [Litt90b] (Figure 3). This model has been extended to include the representation and specification of imperfect synchronisation and imperfect timing, including delays and jitters, in [Diaz93] and to express hypermedia objects in [Sena95].

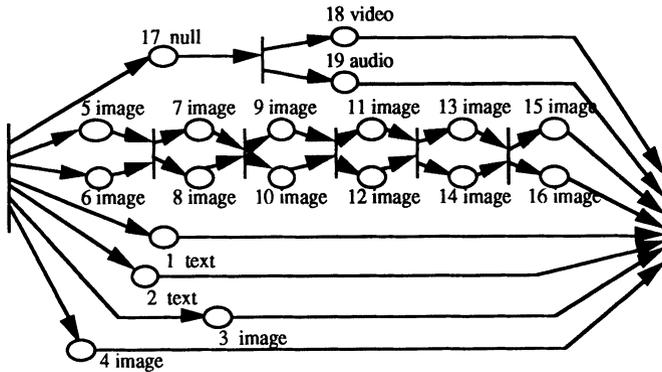


Figure 3 : 5 to 16 : two rotatives view of heart - 18, 19 : sound & video of beating heart

## 2.1 Reliability

Reliability is understood as the possibility of the network to lose messages while always providing the requested service. It can for instance be proposed to define the reliability as a number of losses of messages, of PDUs or of SDUs. Then, if 0 loss is allowed, no PDU or SDUs can be lost, and if  $k$  losses are allowed, then  $k$  messages can be lost while still providing the service.

Note that if  $n < k$  messages are lost, the quality of service related to reliability is enforced. Of course, the interest of such a weak definition is of interest as new applications, such as the transfer of voice and video, precisely do not require full reliability.

[Amer93b] [Diaz94] give a way to compute  $e(P)$  for serial-parallel partial orders extended with losses, noted as  $e_k(P)$ , the number of linear extensions permitted by a partial order  $P$  that tolerates the loss of exactly  $k$  objects, where  $e_0(P)$  represents what was previously denoted  $e(P)$ .

For example, the partial orders  $a;(b//c);d$  and  $(a;b)//(c;d)$  (where  $a;b$  means "a must be delivered before b" and  $a//b$  means "a and b are independant") permit two and six linear extensions. In the first case, the linear extensions with no loss are :  $(a;b;c;d)$  and  $(a;c;b;d)$ . If now one of the objects can be lost, then the number of linear extensions the receiver can accept becomes :  $(a;b;c;d)$ ,  $(a;c;b;d)$  without loss,  $(a;b;c)$ ,  $(a;b;d)$ ,  $(a;c;b)$ ,  $(a;c;d)$ ,  $(b;c;d)$ , and  $(c;b;d)$  with one error (Figure 4(c)). It follows, as expected, that delivering objects to the users becomes simpler when missing objects can be tolerated. Intuitively, when composing  $r$  Petri-nets, either sequentially or in parallel, with  $k$  losses, it is possible to have 1 loss in each of  $k$  nets and none in the others ; or 2 losses (if possible) in a single net, 1 loss in each of  $k-2$  nets and none in the others ; or 2 losses (if possible) in each of the first 2 nets, 1 loss in each of  $k-4$  nets and none in the others ; ...; or  $k$  losses (if possible) in 1 net and none in the others.

[Amer93] and [Diaz94] define four metrics and give a way to compute the number of linear extensions and evaluate these metrics. [Diaz94] jointly uses two complementary metrics, uncertainty and entropy.

### Establishing a POC

Note that implementing a POC first means defining the way the user will make known the POC to the communicating entities. In particular, the receiving entity must be aware of the partial order to enforce it.

It has been seen in Figure 3 that Petri nets can be used by the multimedia application. This means that this model is known at the application layer. In order to implement the most adequate partial order connection, it is proposed to transmit this user defined partial order, i.e., in this case an adequate coding of the Petri net, to the transport entity. This partial order will be, when establishing the PO connection, sent to the remote receiving entity, as one of the connection parameters (coding it in the best way, for instance as a graph). Upon receiving the graph, the receiving entity will become able to build a local representation of the Petri net, and this representation will be used to accept and deliver, to the user, in the adequate order, the received objects. Of course, the objects are delivered to the user not in the order they have been sent but in accordance with the partial order defined by the Petri net.

### 3 TIME AND THE SPACE OF PROTOCOLS

Up to now, only logical ordering has been addressed, and no explicit time constraints have been taken into account. From an application point of view, high speed protocols and services will be used for multimedia communications. Of course, music and video sequences need time constrained transmissions. As a consequence, very general synchronous distributed multimedia applications imply the resulting communication services have to enforce timing constraints, for instance for ensuring isosynchrony. Multimedia presentations of multimedia objects in particular may be defined to contain application related information defining precise temporal interval values [Diaz93].

As a consequence, the following question emerges : how can services be characterised in a general way in terms of time? Starting from the previous basis, it is proposed here to extend the couple (order, reliability) by a third axis related to time.

Now, let us start by considering one of the most well known examples of time constrained service, the PCM (Pulse Coded Modulation) used in Telecommunication systems to transfer voice. As this modulation provides 8 bits of information every 125 microseconds, locating it on the time axis suggests to scale this time axis by a specific value, the time that lasts between the sending of the two consecutively sent bytes (messages). If a PCM service is to be directly implemented, it has to provide the transfer of one message of 8 bits every 125 microseconds in and out of the underlying PCM connections. For instance, a point located at  $t=125$  on the T axis will mean that consecutive messages are delivered to the user of the service every 125 units of time (microseconds for instance).

Of course, such a choice only models perfect systems, as the real time that will exist between two messages cannot be enforced to be exactly 125 microseconds in actual implementations.

It is then proposed to scale the T axis by new couples ( $t_{min}, t_{max}$ ) ; these couples will mean that the sendings of two consecutive PDUs can be separated by an interval of time comprised between  $t_{min}$  and  $t_{max}$ . Then, (125, 125) means that messages are handled exactly every 125 units of time and  $(0, \infty)$  means that no constraint exists on the messages as a message can be sent right after the previous one (i.e. for  $t_{min}=0$ ) or never be sent (i. e. the previous one was the last of the transfer when  $t_{max}=\infty$ ), or will be received right after the previous one, or never be received.

A couple as (120,130) means that the transfer of messages is almost isosynchronous, i.e. occurs every 125 units of time, but with a jitter of 5 :  $(125-5, 125+5)$ .

Of course, usual non timed services are defined by the couple  $(0, \infty)$ . Also,  $(\infty, \infty)$  is the trivially satisfied timed service as messages are never sent, while  $(0, 0)$  is the impossible totally perfect timed service as it implies that the transmission delay should be 0.

It follows that a space can be defined to design first services, then protocols and connections, according to the system of three coordinates (Reliability, Order, Time). Classical non timed protocols are located on the  $(0, \infty)$  interval, timed protocols are located on the other intervals, and  $(t, t)$  timed protocols define one plan. For instance, a perfectly defined and implemented

Pulse Coded Modulation service or protocol will be located at the intersection of the timed plan (125,125), with the total order where reliability equals 1 (Figure 4). An imperfect service will allow the protocol to deliver using a couple (120,130) for instance, using an unreliable total order, where its unreliability can then be defined along the R axis.

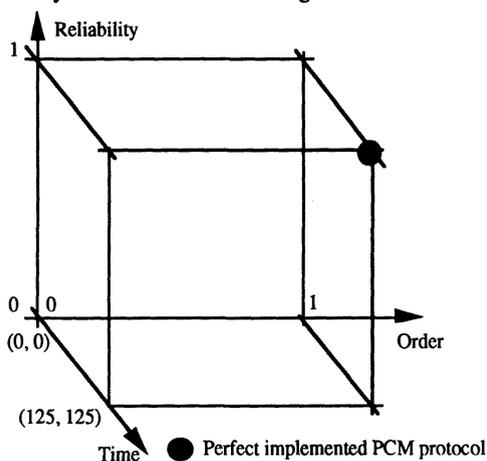


Figure 4 : The space of protocols (Reliability, Order, Time)

General monomedia or multimedia services, protocols and connections should be characterised as being able to be located inside a volume anywhere on the (R,O,T) space.

## 4 DEFINITION OF THE QUALITY OF SERVICE

From the previous sections, it follows that three main basic parameters have been introduced in order to conceptually define services, and so by implication protocols and connections : order, reliability and time.

As a consequence, this new order is no more a functional aspect of services but also becomes a qualitative aspect of services that has to be provided by protocols implementing a new general concept of connection. Handling together these three parameters and comparing them should lead to compare services, protocols and ultimately could provide a strong basis to define the notion of conformity with respect to a given QoS. A first attempt will be illustrated in what follows.

### 4.1 Motivations

QoS [Blai92] [Dant92] [Leop92] is of utmost importance in future communication systems. QoS is traditionally defined by means of several parameters such as the "transit delay", "transfer failure probability", etc. (for an extensive list the reader can refer to [ISO92]). These parameters lack formality and consequently, defining and then comparing services w.r.t. a given QoS is not possible. This section proposes the definition of the QoS using the previous three parameters. As a consequence, these three parameters are considered to be basic for multimedia networks.

It is then given a general framework to define and compare the QoS of multimedia systems. Three main notions will be used for this purpose and will be defined using the three previously

defined concepts : **ordering**, **reliability** and **timing**. These are suggested here to be the three most basic parameters of the concept of QoS. These parameters allow the particular cases corresponding to classical transport services to be represented.

## 4.2 QoS Definitions

We propose here to use services and protocols with various degrees of order, time and reliability. Also, these definitions should cover the existing well-known classes, namely : Connection-Oriented and ConnectionLess transport protocols. It is shown in [Amer94] that these two orthogonal families of protocols are not the most appropriate for multimedia applications. This explains why it is proposed to define a space of QoS. Such a space should make the meaning of QoS more understandable for designing future transport services and protocols.

### *Conformance to ordering*

We will consider that the QoS orderings provided by the service are described as binary relations over  $E \times E$  where  $E$  denotes the set of transmitted objects. This means that orderings are described by a set of couples. For example, at the receiving site, a couple  $(a,b) \in S$  means that *object 'a' must be delivered before object 'b'*. A set of couples will define a partial order. When comparing them, it will be assumed that the representation of partial orders as a set of couples is closed under transitivity. This expresses the intuition that if object  $a$  must be delivered before object  $b$ , and if object  $b$  must be delivered before object  $c$  then object  $a$  must also be delivered before object  $c$ . Formally this means that if  $(a,b) \in S$  and  $(b,c) \in S$  then  $(a,c) \in S$ .

A Transport service is said to be conforming to the ordering constraints of a given QoS definition (shortly, the service is said to be ordering-conforming) when all possible orderings (provided by this service) are valid with respect to the  $LE(P = S)$ , the set of linear extensions of the QoS. That is all linear extensions,  $LE(P = S)$ , of the service must be also linear extensions of the partial order describing the requested QoS.

#### **Definition 1 (ordering-conformance)**

A service  $S$  is ordering-conforming to a QoS  $Q$ , denoted  $S$  **o-conf**  $Q$ , iff it verifies  $LE(P=Q) \supseteq LE(P=S)$ . This intuitively means that all constraints of  $S$  are coherent with the ones of  $Q$ .

Let us consider a connectionless (unordered) protocol used to send 4 messages. The set  $LE(P=CL)$  of its 4! linear extensions is the set of all sequences : { 1234, 1243, 1324, 1342, ..., 4321 }. If a connection oriented service CO is used, then  $LE(P=CO) = \{1234\}$ . It follows that  $LE(P=CL) \supseteq LE(P=CO)$  and CO **o-conf** CL.

Considering now Figure 2, it can also be deduced that Figure 2(b) **o-conf** Figure 2-c, Figure 2(c) **o-conf** Figure 2(d) and Figure 2(d) **o-conf** Figure 2(e).

### *Conformance to reliability*

Several definitions can be used to describe the loss of services and QoSs. A few consider the maximal number of objects that can be lost, others consider sets of objects that the service is

allowed to loss. We will consider here a more refined definition which allows the definition of tolerated losses to be described with accuracy. This will be achieved by considering that, for a set of messages, the loss is not specified by a unique subset of messages but by several subsets, each of which declares a set of **objects that can be lost together**.

Formally, we consider that the accepted loss is specified by a *loss set* defined as a "regular" set of sets of messages,  $Q \subseteq P(E)$ , where  $P(E)$  denotes the powerset of  $E$ , and  $E$  denotes a set of transmitted objects. The regularity of loss sets is introduced to discard senseless specifications : if  $\{a,b,c\}$  may be lost together then so do  $\{a\}$ ,  $\{a,b\}$ ,  $\{a,c\}$ ,  $\{b\}$ ,  $\{b,c\}$  and  $\{c\}$ , that is the subsets of  $\{a,b,c\}$ . The regularity property expresses then that the loss set must be subset-closed. Formally, a loss set  $S$  verifies : for all  $A \in S, A \supseteq B$  implies  $B \in S$ . Elements of this set are considered as a set of objects which can be lost together.

A not formal definition of the reliability conformance can be stated as follows :

A service is said to be conforming to the reliability constraints of the QoS definition (shortly, the service is said to be reliably-conforming) when losses of objects never exceed the limits fixed by the QoS.

Under the representation we chose here, this definition can be formally instantiated by the following equivalent definition.

**Definition 2 (reliability-conformance)**

A service  $S$  is reliably-conforming to a QoS  $Q$ , denoted  $S$  **r-conf**  $Q$ , iff it verifies  $\forall A \in S, \exists B \in Q : A \subseteq B$ . This means that whenever  $S$  may lose all objects represented by set  $A$ , then  $Q$  contains a set,  $B$ , which allows these objects to be lost together.

The above relation was introduced in [Drira 92] and denoted  $S \ll Q$ .

As an example, let us consider a service which never loses simultaneously more than one object. Of course, it conforms to a QoS accepting simultaneous losses of any couple of consecutive objects, since :  $\{\{1\}, \dots, \{n\}\} \subset \{\{1\}, \dots, \{n\}, \{1,2\}, \{2,3\}, \dots, \{n-1,n\}\}$ .

**Conformance to timing**

A service is said to be conforming to the timing constraints of the QoS definition (shortly, the service is said to be timing-conforming) when the possible interval between the sending of two SDUs of the service is included in the interval defined by the QoS, i.e. equals at least the minimal delay and never exceed the maximal delay fixed by the QoS. This can be formally expressed using the following partial order between couples of real values (or integer values) :  $(a,b) \leq (a',b')$  when :  $a \geq a'$  and  $b \leq b'$ .

**Definition 3 (timing-conformance)**

A service  $S$  is timing-conforming to a QoS,  $Q$ , denoted  $S$  **t-conf**  $Q$ , when  $(a_s, b_s) \leq (a_Q, b_Q)$  according to the above interval inclusion, that is when  $a_s \geq a_Q$  and  $b_s \leq b_Q$ .

For example  $(t_{\min}, t_{\max})$  **t-conf**  $(0, \infty)$  for all positive real values  $t_{\min}, t_{\max}$  such that  $t_{\min} \leq t_{\max}$ .

Now the global conformance can be defined. A service is said to be conforming to the QoS it implements when this service is ordering-conforming, reliably-conforming and timing-conforming.

### Global conformance

Then a service  $S$  conforms to a QoS  $Q$  if it conforms to its order, reliability and time constraints.

#### Definition 4 (conformance)

$S$  **conf**  $Q$  iff ( $O_S$  **o-conf**  $O_Q$ ) and ( $R_S$  **r-conf**  $R_Q$ ) and ( $T_S$  **t-conf**  $T_Q$ ).

- $1 \leq o_S \leq o_Q$  : means that (as a consequence of the ordering-conformance) the (number of possible) orderings of the received objects are limited by the ones fixed by the QoS. That is transmission ordering may change within a partial-order-connection, but must never go outside the ordering possibilities fixed by the QoS. The common lower bound of the QoS ordering, 1, means that the QoS requires a total ordering which corresponds to  $o_Q=1$ .  $o_S=1$  would mean that ordering remains unchanged within the whole connection.

- $\emptyset \leq r_S \leq r_Q$  : means that (as a consequence of the reliability-conformance) the (number of possible) losses of objects are limited by the ones fixed by the QoS.

- $0 \leq a_S \leq a_Q \leq b_S \leq b_Q$  : means that the timing conformance of the service fulfils the interval constraint of the quality of service.

#### Discussion

From the previous concepts, the unordered, unreliable and untimed QoS (shortly uuu-QoS) is defined by :

- its set of linear extensions is the  $n!$  total orders over  $E$ . Using simple notations, this set is represented by :  $\{1;2\dots;n, \dots, n;\dots;1\}$ .

If  $n = 3$ , then  $\{1;2;3, 1;3;2, 2;1;3, 2;3;1, 3;1;2, 3;2;1\}$  is the set of accepted total orders.

- its set of possible losses, i.e. all subsets of the set of the sent objects is :  $\{P(E)\} = \{\emptyset, \{1\}, \dots, \{n\}, \dots, \{1, \dots, n\}\}$ . The whole set of objects may be lost.

If  $n = 3$ , then  $\{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$  is the loss set.

- its time interval which is  $(0, \infty)$ .

It follows that any implemented service conforms to the uuu-QoS.

For instance it becomes obvious that a totally ordered, reliable and timed service conforms to the uuu-QoS as it is defined, e.g., by :  $(\{1;2;\dots;n\}, \{\emptyset\}, (t_{min}, t_{max}))$  and as :

$\{1;2;\dots;n\} \subseteq \{1;2;\dots;n, \dots, n;\dots;2;1\}$  and  $\{\emptyset\} \subseteq \{P(\{1,2,\dots,n\})\}$  and  $(t_{min}, t_{max}) \leq (0, \infty)$ .

Also any reliable PCM service conforms to the QoS of any untimed TCP service as if the PCM service is perfect  $(125, 125) \leq (0, \infty)$  and as even if its timing has some jitter, then  $(125-j, 125+j) \leq (0, \infty)$ .

The previous comparisons become possible as a formal basis has been given for defining an overall framework allowing the description and the comparison of two definitions of services. This approach seems able to consider at least three important parts of any particular service, and thus have been avoided :

- lack of completeness : all ordering, reliability, timing cases are covered,
- lack of consistency : the descriptions are dependent of the services and are meaningful whatever the service is. The proposed parameters are not imaginary, but, on the contrary, are suitable for high speed network based systems where latency is significant with respect to transmission time.

## 5 CONCLUSION

The ideas underlying partial order connections have been presented and it has been shown how these partial order connections are much more general than the usual CL or CO connections.

This work has investigated the general notion of quality of service. Starting from a new concept, the concept of partial order in partial order connections, it has given a three dimensional representation of the QoS by introducing a reliability axis and a time axis. This representation allows a formal definition of the QoS to be given.

From a (Reliability, Order, Time) space of protocols, a formal framework for defining and handling the quality of service of protocols has been provided, where quality of service and conformance to a given quality of service can be defined.

A lot of work is still to be done for expressing and proving the conformity of protocols with respect to a given quality of service by using in a first phase the proposed set of three parameters.

### Acknowledgements

This work has been supported by CNET-FRANCE TELECOM under Grant 92 1B 178 as part of the CNET-CNRS CESAME project on the design of High Speed Multimedia Cooperative Systems.

## 6 REFERENCES

- [Ahu90] Ajuha (M.). FLUSH Primitives for Asynchronous Dist'd Systems. *Info Processing Letters*. (February 1990), 34(1), 5-12.
- [Amer93a] Amer (P.D.), Chassot (C.), Connolly (T.), Diaz (M.). Partial Order Transport Service for Multimedia Applications : Reliable Service - *2nd High Performance Distributed Computing Conf.* (July 1993) - Spokane, Wash.
- [Amer93b] Amer (P.D.), Chassot (C.), Connolly (T.), Diaz (M.). Partial Order Transport Service for Multimedia Applications : Unreliable Service - *INET '93 3rd International Conference, San Francisco, CA*, (August 17-20 1993).
- [Amer94] Amer (P. D.), Chassot (C.), Connolly (C.), Conrad (P.), Diaz (M.). Partial Order Transport Service for Multimedia and other Applications. *IEEE/ACM Transactions on Networking*, Oct. 1994, vol.2, n° 5.
- [Ande91] Anderson (D.), Homsy (G.). A Continuous Media I/O Server and its Synchronization Mechanism. *IEEE Computer* (October 1991), 24(10), 51-57.
- [Annal94] Special Issue on CESAME, *Annals of Telecommunications*, May-June 1994.
- [Berk67] Coll. Statistical Physics. Berkeley Physics Course, Vol. 5, McGraw-Hill (1967)
- [Blai92] Blair (G.S.), Garcia (F.), Hutchison (D.), Shepherd (W.D.). Towards New Transport Services to Support Distributed Multimedia Applications. *4th IEEE COMSOC International workshop*, Monterey, USA, April 1-4, 1992.
- [Camp93] Campbell (A.), Coulson (G.), Hutchison (D.). A Multimedia Enhanced Transport Service in a Quality of Service Architecture. *4th International workshop on Network and Operating Systems Support for Digital Audio and Video*, Nov. (3-5) 1993, Lancaster, UK.
- [Dant92] Danthine (A.), Baguette (Y.), Leduc (G.), Leonard (L.). The OSI95 connection-mode Transport service - the enhanced QoS. *4th IFIP Conference on High Performance Networking*, Liège, Belgium, Dec. (14-18) 1992.

- [Diaz93] Diaz (M.), Sénac (P.). Time Stream Petri Nets, A Model for Multimedia Streams Synchronization. *Proc of the 1rst International Conf. on MultiMedia Modelling*. World Scientific. (Nov. 1993 ), 257-273.
- [Diaz94a] Diaz (M.), Pays (G.), The CESAME Project, *Annals of Telecommunications*, May-June 1994
- [Diaz94b] Diaz (M.), Lozes (A.), Chassot (C.), Amer (P.), Partial Order Connections, A new concept for High Speed and Multimedia Services and Protocols, *Annals of Telecommunications*, May-June 1994
- [Drira92] Drira (K.). The refusal graph a Tradeoff between Verification and Test. *Proc. of the 6th International IFIP Workshop on Protocol Test Systems. IWPTS VI*. Pau - Sept. 1993.
- [Gall68] Gallager (R.G.). Information Theory and Reliable Communication. Jon Wiley and Sons inc, (1968).
- [Hard91] Hardt-Kornaci (S.L.), Ness (L.A.). Optimization Model for the Delivery of Interactive Multimedia Documents. In *Proc Globecom 91*, Phoenix (Dec. 1991), 669-674.
- [ISO92] A suggested QoS architecture for multimedia communications. *ISO/IEC JTC1/SC 21/WG 1 N - OSI Architecture*. September 92.
- [Lamp78] Lamport (L.). Time, Clocks and the Ordering of Events in a dist'd System. *CACM*, (July 1978), 21(7),558-565.
- [Leop92] Leopold (H.), Campbell (A.), Hutchinson (D.), Singer (N.). Towards an integrated quality of service architecture (QoS\_A) for distributed multimedia communications. *4th IFIP Conference on High Performance Networking*, Liège, Belgium, Dec. (14-18) 1992.
- [Litt90a] Little (T.), Ghafoor (A.). Network Considerations for Distributed Multimedia Objects Composition and Communication. *IEEE Network Magazine*, (November 1990), 32-49.
- [Litt90b] Little (T.), Ghafoor (A.). Synchronization and Storage Models for Multimedia Objects. *IEEE J on Selected Areas in Comm.*, (April 1990), 8(3), 413-427.
- [Neig87] Neiger (G.), Toueg (S.). Substitution of Real Time and Common Knowledge in Asynchronous dist'd Systems. In *Proc 4th Symp. on Principles of Dist'd Computing*, (1987), 281-293.
- [Pete89] Peterson (L.), Buchholz (N.), Schlichting (R.). Preserving and Using Context Information in Interprocess Communication. *ACM Trans. on Computer System*, 7(3), (August 1989), 217-246.
- [Sena95] Sénac (P.), Diaz (M.), Léger (A.), de Saqui-Sannes (P.), Modelling Logical and Temporal Synchronisation in Hypermedia Systems, to be published, *IEEE Trans on Selected Areas in Communications*, 1995
- [Shaf92] Shafer (K.), Ahuja (M.). Process-channel(agent)-Process Model of Asynchronous dist'd Communication. In *Proc. ICDCS 12*, Yokohama, Japan, (June 1992), 4-11.
- [XTP92] protocol definition, Revision 3.6. *Protocol Engines Incorporated* (Jan. 1992).