SICS Technical Report
T92/02

ISRN : SICS-T—92/02-SE
ISSN : 1100-3154

Multimedia Transport Service and Protocol Issues

by

Bengt Ahlgren and Mats Björkman

December 1991

Swedish Institute of Computer Science
Box 1263, S-164 29 KISTA, SWEDEN

Multimedia Transport Service and Protocol Issues*

Bengt Ahlgren[†] Mats Björkman[‡]

(bengta@sics.se) (Mats.Bjorkman@DoCS.UU.SE)

SICS technical report T92:02

[†]Swedish Institute of Computer Science,
Box 1263, S-164 28 Kista, Sweden

[‡]Dept. of Computer Systems, Uppsala University,
Box 520, S-751 20 Uppsala, Sweden

December 5, 1991

Abstract

This paper considers the issues of a real time transport service needed by multimedia applications for transferring digital video and audio. Three classes of transport service are defined with different levels of real time constraints. Methods for error control are considered for the classes, and the classes are discussed with respect to the application requirements.

*Also appearing in the Proceedings of the 3rd MultiG Workshop, KTH, Stockholm, December 17, 1991.

1 Introduction

Traditional transport service, such as the service from TCP, was defined to support applications like file transfer and electronic mail. These applications do not depend on real time guarantees to work properly. The growing field of multimedia applications imposes new requirements on the transport service. The new applications require that a certain amount of bandwidth is available and that the time to deliver a packet from source to destination is bounded by some value. These are real time requirements that the traditional transport service was not designed to support.

The purpose of this paper is to discuss and suggest real time properties that the transport service could provide, as well as discuss what service a real time application needs from the transport provider.

This work is a part of the MultiG research program [3] in Sweden. MultiG is a collaborative program for research in the area of multimedia applications and high speed networks. The program contains many projects ranging from optical fiber interfaces to distributed virtual world applications and applications using digital audio and video.

This paper is based on a series of discussions held at SICS mainly in the period from September 1990 to February 1991. The following persons have participated one or more times: Bengt Ahlgren, Mats Björkman, Stephen Pink, Peter Sjödin and James Kistler (guest from CMU).

2 New application requirements

Table 1 summarizes the real time properties that multimedia applications require. The table is taken from an article by Hehmann, Salmony and Stüttgen [2].

QoS	Maximum delay (s)	Maximum delay jitter (ms)	Average throughput ($Mbit/s$)	Acceptable bit error rate	Acceptable packet error rate
Voice	0.25	10	0.064	$< 10^{-1}$	$< 10^{-1}$
Video (TV quality)	0.25	10	100	10^{-2}	10^{-3}
Compressed video	0.25	1	2–10	10^{-6}	10^{-9}
Data (file transfer)	1	—	2–100	0	0
Real time data	0.001–1	—	< 10	0	0
Image	1	—	2–10	10^{-4}	10^{-9}

Table 1: New application requirements.

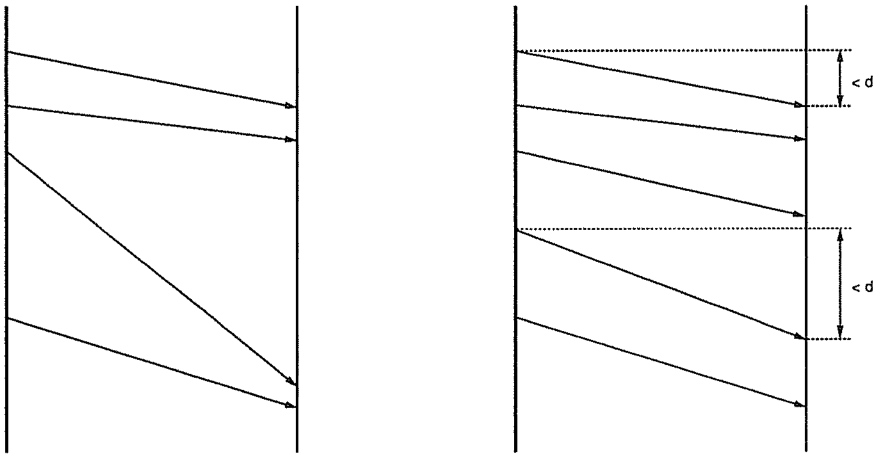

Figure 1: Asynchronous and synchronous service.

3 Synchronicity

In a real time system it is necessary to know the time it takes to perform a certain task. For a communication service that is part of the real time system, it is the time to deliver a packet of data from the source to the destination—the *delay*—that is the most important parameter. The delay variation, or *jitter*, is also important. The following terms are used to describe communication services with different delay characteristics (illustrated in figures 1 and 2):

- **Asynchronous**: has unspecified delay, which implies unspecified delay variation.
- **Synchronous**: has an upper bound on the delay, which implies a coarse bound on the jitter.
- **Isochronous**: has specified delay and delay variation.

There is a problem with defining what is meant by "delay" for some kinds of transport service. If a data unit delivered to the receiver corresponds to exactly one data unit provided by the sender, there is no problem. However, if the data unit delivered to the receiver can originate from more than one data unit provided by the sender (as in, e.g., TCP), then the delay becomes different for different parts of the data unit at the receiver. Protocols with this property can therefore not provide an isochronous service.

Another characteristic that may be important to real time applications is **fixed rate**. With fixed rate we mean that the flow of data is constant over time on a stream (see figure 2). The sender must always provide data to the stream at this rate, maybe on request from the transport provider. If the sender cannot provide data, "random" data will be delivered. The receiver

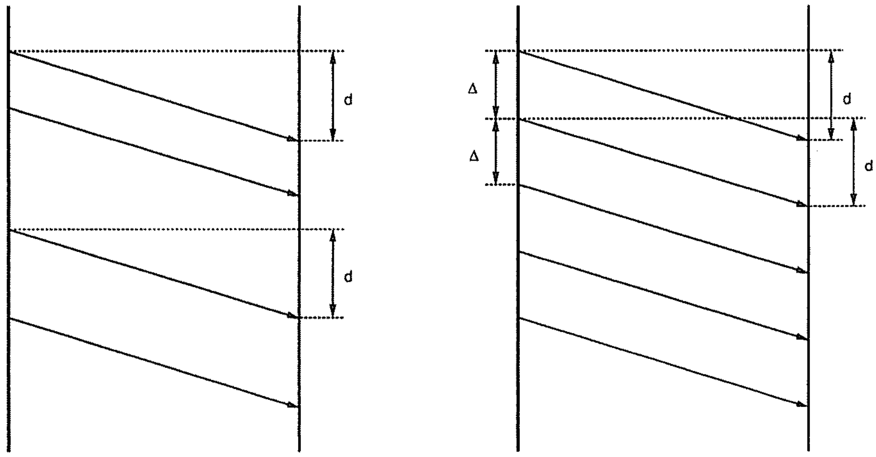

Figure 2: Isochronous service and isochronous service with fixed rate.

of the stream must receive at the rate of the stream, and cannot throttle the stream without data being lost.

How is the notion of "synchronicity" related to fixed rate? Fixed rate is a very strict form of service that needs at least bounded delay to work properly. It is therefore incompatible with asynchronous service.

4 Transport service

4.1 Classes of transport service

With the previous notions as a basis we define three classes of transport service.

4.1.1 Asynchronous

This service is the traditional non-realtime service. Since the service does not guarantee any upper bound on the delay, bandwidth guarantees cannot be made either. The sender may try to send at any rate. The receiver may do flow control, that is, it may throttle the stream.

4.1.2 Synchronous/isochronous

This service class includes both synchronous and isochronous, because we believe that there is no sharp division between them. The difference is just how much the delay is allowed to vary. By introducing transport layer buffering, a synchronous service with a maximum delay of d can be converted to an isochronous service with delay d and some specified variance.

With this service, bandwidth and maximum delay can be guaranteed. If bounds on the delay variance are specified, the service is isochronous.

The receiver's ability to throttle the stream is limited because of the delay and delay variance guarantees. The sender may try to send at any rate. If the sending rate exceeds the allocated bandwidth, delay guarantees might not be met.

4.1.3 Isochronous with fixed rate

This service class is a combination of isochronous service and the notion of a fixed rate stream.

The rate of a stream is defined by the requested bandwidth. The delay and delay variance are appropriate parameters to specify for the transport service user. Note that with this service class, the delay is related to the rate in the sense that an increase (decrease) in delay will result in a temporary rate decrease (increase).

The receiver may not do flow control. The sender must send at the requested rate.

4.2 Quality of Service interaction

The Quality of Service (QoS) user requirements supplied in a connection establishment request can either be simple values for upper or/and lower bounds on one or more parameters, or more elaborate ones, like probability bounds on the parameters [1].

The actions taken by the transport provider if it cannot provide the requested Quality of Service can be either to just refuse to set up the connection, possibly indicating which requirements could not be met, or to set up a connection with a degraded QoS, indicating the actual QoS.

If, during the lifetime of a connection, QoS changes, the transport provider could either notify the user about the change, possibly closing the connection if the degradation in QoS crosses some bound(s), or give the user the possibility to query the transport provider about the current QoS at any time. In the latter case, the user must find out about QoS degradations for himself.

Which of the models of QoS interaction to use may itself be a part of the QoS.

4.3 Discussion

In most non-realtime environments (such as UNIX), isochronous transport service is a wasted effort, since the operating system introduces so much jitter that the application must smooth it out by buffering anyhow. Thus, the transport service might as well be synchronous, with all buffering taking place in the application.

Fixed rate as a transport service is in principle not needed, because the combination of isochronous service and guaranteed bandwidth gives the sender the possibility to send at a fixed rate, which will be received by the receiver at a fixed rate because of the constant delay.

The reason for introducing the notion of fixed rate is that it could be easier for the transport provider to guarantee low jitter. The transport provider could also compensate for jitter introduced by the sender, because the receiving transport entity knows when a packet should be delivered to the receiver *regardless* of when the packet was sent by the sender.

5 Error control

Depending on user application requirements and the transport layer's knowledge about the error characteristics of the underlying network, different error control schemes may be applied at the transport layer. If the application can accept the error rate of the network, transport error control does not need to reduce the error rate. Often, however, control schemes are used to reduce the network error rate to a rate acceptable to the transport user.

5.1 Error control scheme types

We divide the error control schemes into Automatic Repeat Request and Forward Error Correction type schemes. Depending on the characteristics of the application and its requirements, one or both of these schemes may be applied.

5.1.1 Automatic Repeat reQuest (ARQ)

ARQ is the type of error control where the receiving transport entity detects lost or damaged packets and requests retransmission.

Retransmission requests are generated when losses/damages are detected, so the performance cost of ARQ cannot be calculated in advance, but only statistically predicted. The retransmission must be able to take place within the limit of the maximum allowed delay. In order for ARQ to work efficiently, the receiver must have enough buffer space to buffer all packets from the lost/damaged one until the retransmitted packet has reached the receiver. Otherwise, each retransmission will cause one or more of the packets following the lost/damaged one to be dropped and each in turn cause a retransmission.

ARQ requires a two-way (duplex) connection between transport entities.

5.1.2 Forward Error Correction (FEC)

FEC is the type of error control where enough redundant information is transmitted for errors to be corrected at the receiving side.

FEC control is applied to all packets, so the cost of using FEC is known in advance. FEC calculations are more complex than ARQ calculations. For an

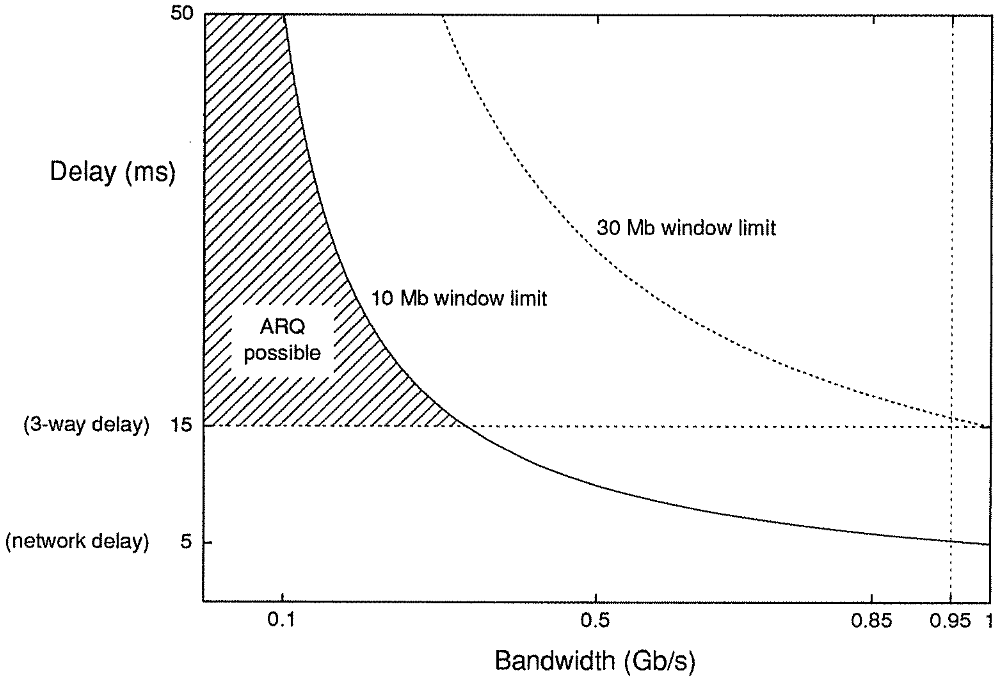

Figure 3: Applicability of ARQ.

implementation to be efficient (and to meet delay limits), hardware support may be needed.

FEC can be used on one-way (simplex) connections.

5.2 Applicability of ARQ and FEC

We have tried to understand when the different types of error control, like ARQ and FEC are applicable.

Figure 3 shows when ARQ is possible for a network with a bandwidth of 1 Gbit/s and 5 ms delay in both directions. It is assumed that one retransmission is sufficient to keep the requested residual error rate. This gives 15 ms as a theoretical lower limit on the delay guarantee. ARQ protocols have some overhead that is assumed to be 5%, hence 0.95 Gb/s for data. Another parameter is the size of the retransmission window. If the retransmission window is too small, some of the packets following a lost/damaged packet has to be thrown away when the receiving transport entity is waiting for the lost/damaged packet to be retransmitted. In the figure, both 10 Mbit and 30 Mbit window limits have been drawn. The shadowed area shows when ARQ is applicable if the window limit is 10 Mbit.

Figure 4 shows when it is possible to use FEC for error correction for a network with the same bandwidth and delay as for ARQ. Here we assume that the FEC uses 15% of the bandwidth for error correction.

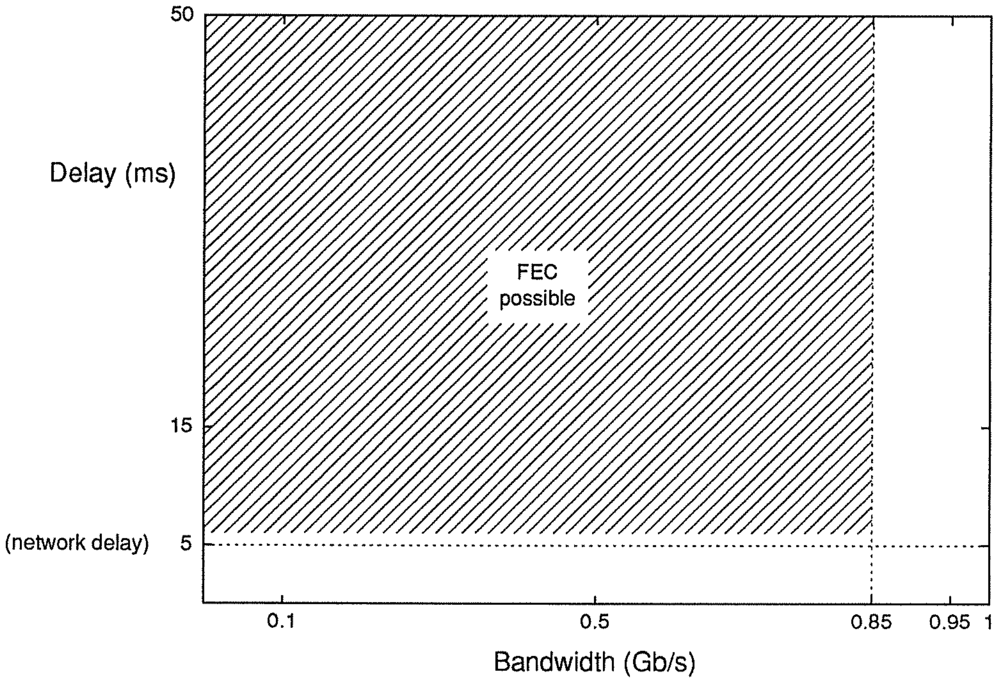

Figure 4: Applicability of FEC.

5.3 Discussion

There is a delay/bandwidth tradeoff between ARQ and FEC. In ARQ, retransmission delays the delivery of the lost/damaged packet with at least one round trip. A FEC packet, on the other hand, requires extra transmission bandwidth even in the absence of errors.

FEC costs can be calculated in advance, ARQ costs only estimated.

FEC can be applied to simplex connections, ARQ only to duplex connections.

FEC algorithms are often more complex than ARQ algorithms.

5.4 Error notification

When an error cannot be masked by the transport layer, two choices have to be made:

- Whether or not to deliver the bad packet.
- Whether or not to notify the user about the bad packet.

Both choices depend on application requirements. For some applications, a bad packet is better than no packet, but for other applications a bad packet may be disastrous. In the case of notification, some applications may want to be notified in order to apply their own error compensation or recovery, while other applications do not care (or do not have the time!) to compensate for residual errors.

Note: Failure to meet delay limits is an error, and late packets may be treated as bad packets above.

6 Conclusions

The paper discusses three classes of transport service. Although the properties of these classes are appropriate for real time applications, it is unclear if it is appropriate to place them in the transport layer.

The application needs at least a synchronous transport service, but the application may not have any use for an isochronous and/or fixed rate transport service, because jitter in the application may force the application to redo the synchronization in the end.

From the previous sections we conclude that the following quality of service parameters are needed to control the transport service:

- Bandwidth
- Error rate and recovery model (error recovery, error notification or none).
- Maximum delay and delay variation (jitter).
- Tradeoffs between bandwidth, error rate, delay and jitter.

We see two models of quality of service interactions between the transport user and provider at connection setup time:

1. The application requests a certain QoS. The transport provider sets up the connection only if the requested QoS can be met.
2. The application requests a certain QoS. The transport provider sets up the connection and tells the application what QoS it got.

During the lifetime of a connection, QoS may change due to network or host dynamics. Here we also see two models of interaction:

1. If (and when) the transport provider no longer can provide the requested QoS, the application is notified and/or the connection is closed.
2. The application can at any time query the transport provider as to what the QoS currently is.

7 Future work

Future work includes:

- Relative synchronization of multiple streams
- Multicast

References

[1] Domenico Ferrari. Client requirements for real-time communication services. *IEEE Communications Magazine*, 28(11):65–72, November 1990.

[2] Dietmar B. Hehmann, Michael G. Salmony, and Heinrich J. Stüttgen. Transport services for multimedia applications on broadband networks. *computer communications*, 13(4):197–203, May 1990.

[3] Björn Pehrson and Stephen Pink. Multimedia and high speed networking in MultiG. *Computer Networks and ISDN Systems*, 21(4):315–319, June 1991.