

On the Efficiency of PGPS-based Packet and Cell Switching Technologies for Traffic with Guaranteed Delay

*Original*

On the Efficiency of PGPS-based Packet and Cell Switching Technologies for Traffic with Guaranteed Delay / Risso, FULVIO GIOVANNI OTTAVIO. - In: EUROPEAN TRANSACTIONS ON TELECOMMUNICATIONS. - ISSN 1124-318X. - 14:(2003), pp. 301-318.

*Availability:*

This version is available at: 11583/1405289 since:

*Publisher:*

Wiley

*Published*

DOI:

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

# On the Efficiency of PGPS-based Packet and Cell Switching Technologies for Traffic with Guaranteed Delay

FULVIO RISSO

Politecnico di Torino, C.so Duca degli Abruzzi, 24, I-10129 Torino  
*fulvio.risso@polito.it*

**Abstract.** Circuit switching, suited to providing real-time services due to the low and fixed switching delay, is not cost effective for building integrated services networks because is based on static allocation of resources which is not efficient with bursty data traffic. Moreover it cannot handle flow that are not integer multiple of 64 Kb/s, preventing the usage of low bit rate codecs.

This work explores the most suitable alternatives to the circuit switching technology (i.e. packet/cell switching) from the efficiency point of view, assuming that a PGPS scheduler is deployed in the network nodes. The paper defines an index to measure the efficiency of packet telephony, i.e. the volume of real-time traffic with *deterministically* guaranteed quality plus the amount of data carried related to the amount of network resources used. Furthermore it determines the maximum efficiency obtainable by packet networks, it compares different network technologies and it explores the problems of the deploying of low bit-rate codecs.

## 1 INTRODUCTION

Circuit switching is particularly suitable to provide real-time services, like video and telephony, because of its low and fixed switching delays. However, it is based on static allocation of resources which is not cost effective for bursty data traffic. Moreover, current circuit switching technologies handle flows at rates which are integer multiples of 64 Kb/s; this prevents from taking advantage of low bit rate voice encodings, unless multiple phone calls are aggregated in a single flow significantly increasing the complexity of the network and of call handling.

Packet/cell switching (simply referred as *packet switching* in the following) is appealing for carrying real-time traffic because it can benefit from aggressive and adaptive compression schemes and statistical multiplexing to more efficiently exploit network resources. Moreover, packet switching networks are able to carry voice and data at the same time, thus reducing costs and improving the utilization of the physical infrastructure.

Provision of quality of service (QoS) guarantees over packet switched networks requires deployment of advanced packet scheduling algorithms in the intermediate nodes, and a mechanism for call admission control (CAC). The former aims to guarantee the delay ensured to each flow in

a better way than simple first in first out (FIFO) queuing. The latter aims to control the amount of real-time traffic having access to the network and to reserve resources for real-time flows. These two components are strictly related since the amount of resources to be reserved for a real-time flow—and thus the amount of real-time traffic acceptable on the network—depends on the scheduling algorithm deployed.

The QoS provision framework must be completed with a signaling protocol (such as the Resource Reservation Protocol (RSVP) [1] on IP networks or through UNI signaling [2] on ATM) to carry users requests to the network, and policing functions to ensure that the actual traffic generated by users complies with their requests.

The work presented in this paper aims at a comparative analysis of packet, cell and circuit switching technologies for traffic with guaranteed delay under the assumption of the deployment of the Packet-by-Packet Generalized Processor Sharing (PGPS) [3, 4] scheduling algorithm into network nodes. This algorithm is able to guarantee a maximum delay bound to a given session provided that the input traffic is respectful of the service contract.

The described approach to QoS provision is conformant to the model for integrated services (IntServ) over the Internet [5], although this has been recognized having

scalability problems. A differentiated services (DiffServ) model [6] has been proposed as a more scalable solution because of its coarser granularity than the session level. Particularly, the Expedited Forwarding model [7] can be seen as a way to provide delay guarantees [8]. However, there are several problematic aspects in DiffServ. First, it does not assume the PGPS scheduler and the guaranteed delay bound is the sum of the maximum delay guaranteed by each network node, which is usually larger than the delay computed on the complete path (as PGPS does). Second, the DiffServ approach does not provide flow isolation and nobody can guarantee with absolute certainty that one session does not influence the other ones. Third, a signalling method for DiffServ (equivalent to RSVP in IntServ) does not exist. This means that DiffServ is usually deployed either on a pre-provisioned basis, or for some applications (like the ones that fits into the *Pseudo Wire Emulation Edge to Edge* [9] proposal) that span over long-term periods. It follows that the IntServ model is the most appropriate in case absolute guarantees are required and an explicit signalling for each call is required.

This work originates from [10], which compares circuit and packet telephony from a simulative point of view, and it extends [11, 12] that introduces some indexes to measure the ability of a packet network to transport real-time calls. However, the proposed index to measure the efficiency in transporting voice calls (*real-time efficiency*) supposed that the packet switched network was used to carry only voice traffic, which is not the case in modern networks. This paper defines a new quality index used to measure the efficiency of the network, called *generalized efficiency*. This parameter represents the bandwidth required to transport a certain amount of voice traffic with *deterministically* guaranteed quality (given a predetermined load due to data traffic) on packet/cell switching compared to the bandwidth required by circuit switching.

The maximum efficiency reached by each technology compared to circuit switching is only one of the goals of this paper. Next steps demonstrate the path to determine the best technology according to a given input (network topology, delay and bandwidth requirements of the guaranteed traffic, foreseen load due to data traffic) and to formalize the influence of the codec chosen by the packetization process. Although the formalization of the results in a mathematical form is one of the main goals of this paper, sometimes the mathematical elegance has to be abandoned in order to avoid convoluted forms; in some cases a simulating approach is used to show the results that is based on the ad-hoc call level simulator developed in [10].

IP (undoubtedly the most common packet switching technology) and ATM (the most important cell-based one) are taken into consideration as packet/cell switching technology for carrying compressed voice and they are compared to circuit switching carrying PCM (64 Kb/s) encoded voice; however the results are easily extensible to other technologies such as IPv6, Frame Relay, and oth-

ers. Two cases for ATM have been considered: when the voice payload fits into a single cell [13, 14], and when it requires more than one cell to be transported [15]. The former allows only PCM encoding with AAL1/5 data payload; the latter has been standardized recently and it applies to backbone traffic only (AAL2 payload, AAL2/5 control connections). Former technology is called *ATM single-cell* (in the following ATM\*), while the latter technology (*multiple-cells*) is called simply ATM throughout this paper. ATM multiple-cells looks promising to transport high bit-rate (and high quality) streams and to decrease the protocol overheads; however up to now there are only a few commercial devices that support this feature.

This paper does not investigate the service provided by some ATM service classes, notably the VBR-rt (Variable Bit Rate - Real Time) one. This is because this work assumed the existence of PGPS-aware nodes and there is no guarantee that the VBR-rt class makes use of such a scheduler.

While ATM single-cell can be seen as a particular case of ATM, they are analyzed separately in this paper. There are several reasons to do so. First of all, most of the ATM devices support only PCM encoding into AAL1/AAL5 frames that fits into a single cell. Then, ATM\* is simpler from the mathematical point of view. This work does not exclude that ATM\* devices can deploy codecs different from PCM; however it assumes that in any case the real-time payload (plus protocol headers) fits into a single cell. When more than one cell is required, the analysis refers to ATM. This difference makes the results easier to understand.

The paper is structured as follows. Section 2 presents the background of this work. It includes details about the Call Admission Control deployed, a brief overview of the PGPS scheduler, the codification and packetization processes and the network used for the simulations. Indexes used throughout the paper to evaluate the efficiency in utilization of network resources are introduced in section 3. Section 4 derives the maximum efficiency point for each technology; section 5 uses previous results to highlight the points in which a technology becomes convenient compared to the others. Since previous sections assumes that packets can be of any length, section 6 introduces a new degree of uncertainty due to the codec granularity and it gives some indications about the maximum obtainable efficiency when real codecs are deployed. Section 7 applies previous results to compare technologies one to the others and it derives some deployment guidelines. Finally, section 8 presents a brief discussion of the results and it gives some conclusive remarks.

## 2 BACKGROUND

This section summarizes the background needed in the following of the paper. Unless differently stated, following

variables are referred to a specified real-time session. The footer  $i$  (that means “session  $i$ ”) is indicated only when the notation could ingenerate some confusion.

## 2.1 THE CODIFICATION PROCESS

Codecs can be distinguished in *sample* and *frame* based. The first group includes the ones that are able to produce a distinct code (with length  $S$ ) at each step of the sampling process. The second group involves those that gather several samples together and return them into a single frame (with length  $F$ ).

According to [16], the length of the sample/frame needs a further adjustment to be compatible with the following packetization process. Therefore the length of the minimum sample (i.e. the minimum payload) is:

$$L_{PD}^{min} = S \cdot \min_{k \in \mathcal{N}, 1 \leq k \leq 8} \left\{ k : \left\lceil \frac{S \cdot k}{8} \right\rceil = \frac{S \cdot k}{8} \right\} \quad (1)$$

for sample-based codecs, and:

$$L_{PD}^{min} = \left\lceil \frac{F}{8} \right\rceil \cdot 8 \quad (2)$$

for frame-based codecs.

Since these parameters are codec properties and do not depend on the network technology,  $L_{PD}^{min}$  (instead of  $F$  or  $S$ ) will be considered the minimum sample permitted by the codec.

The *Packetization Delay* ( $D_{pack}$ ) is the amount of time chosen to gather samples (from the codec output) and put them into a single packet. This value must be an integer multiple of the minimum packetization delay ( $D_{pack}^{min}$ ) that is the one needed to gather the minimum sample  $L_{PD}^{min}$ .

Given a proper  $D_{pack}$ , the length of the data collected in this way is the *Data Payload* ( $L_{PD}$ ), i.e. the payload size for the real-time session, and it accounts also for the padding needed to align the frame to the byte length. The  $L_{PD}$  can be derived from the previous parameters since:

$$L_{PD} = \frac{D_{pack}}{D_{pack}^{min}} \cdot L_{PD}^{min} \quad (3)$$

The *Codec Granularity* is defined as the minimum packetization delay ( $D_{pack}^{min}$ ) permitted by the specified codec. For instance, PCM codec has the finest granularity because even a single sample (8 bits generated every 125  $\mu$ s) can be used to create a packet. GSM codec does not have a good granularity, because  $D_{pack}^{min} = 20$  ms (260 bits).

Finally, the *Effective Bandwidth* ( $B_{eff}$ ) of the real-time session is the bandwidth needed to transport the payload data, which is usually equal to the bit rate of the codec:

$$B_{eff} = \frac{L_{PD}}{D_{pack}} = \frac{L_{PD}^{min}}{D_{pack}^{min}} \quad (4)$$

## 2.2 THE PACKETIZATION PROCESS

The packetizer module gathers voice samples until the selected packetization delay has been reached. Appropriate headers ( $P_H$ ), depending on the selected network technology, are appended to the voice samples and eventually the packet is transmitted on the network.

In case of packet-network technology (e.g. IP or IPv6) the so created packet has a length  $L$  that is:

$$L = L_{PD} + P_H \quad (5)$$

In case of a cell-switched technology (e.g. ATM), the cell size  $C$  has a fixed value<sup>1</sup> and the data payload can span over multiple cells:

$$N = \left\lceil \frac{P_H + L_{PD}}{C_{PD}} \right\rceil, \quad C = C_{PD} + C_H \quad (6)$$

where  $N$  is the number of cells required to transport the payload,  $P_H$  are the headers appended to the real-time data (for example the AAL5 header),  $C_{PD}$  is the cell payload and  $C_H$  the cell header.

The raw capacity required for the transmission of the selected session into the network is defined *Real Bandwidth* ( $B_{real}$ ) and it depends on the overhead of the packetization process. This value is computed differently according to the underlying network technology:

$$B_{real,packet} = \frac{L}{D_{pack}}; \quad B_{real,cell} = \frac{N \cdot C}{D_{pack}} \quad (7)$$

where  $L$  is the packet size for that session,  $N$  is the number of cells (of size  $C$ ) required to carry the real-time data and  $D_{pack}$  is the selected packetization delay.

## 2.3 DELAY BOUND AND CALL ADMISSION CONTROL

PGPS is derived from the Generalised Processor Sharing (GPS) algorithm which assumes the *fluid flow* model of traffic: each active flow feeds a separate buffer and all the backlogged buffers are served concurrently. A GPS scheduler guarantees to each flow  $i$  a minimum service rate  $g_i$  which is a weighted share of the output link capacity. This rate is said to be *reserved* to flow  $i$ . Provided that a flow  $i$  is compliant with the traffic exiting a leaky bucket with an output rate  $\rho_i < g_i$  and depth  $\sigma_i$ , GPS guarantees an upper bound on the queuing delay of each flow  $Q_i^{GPS} = \sigma_i / g_i$ .

PGPS, also named *Weighted Fair Queuing* [17], extends GPS in order to handle packet-based flows. The idea behind PGPS is very simple: incoming packets are scheduled for transmission according to their equivalent GPS service time, i.e. the time in which the last bit of a packet would be sent by GPS.

<sup>1</sup> Also packet technologies have a maximum packet size. However this is quite a large value (maximum length of an IP packet is 64 KBytes) and it does not influence the results presented in this paper.

Assuming that a packet flow is compliant with the above leaky bucket (i.e. leak rate  $\rho_i$  and bucket depth  $\sigma_i$ ), the scheduling delay bound<sup>2</sup> is (equation 12.1 in [18]):

$$D_{i,sched} = \frac{\sigma_i}{g_i} + \frac{(H_i - 1) \cdot L_i}{g_i} + \sum_{m=1}^{H_i} \frac{L_M}{r_m} \quad (8)$$

where  $H_i$  is the number of hops on the path of flow  $i$ ,  $r_m$  is the service rate of the  $m^{th}$  node (usually the capacity of link  $m$ ),  $L_i$  is the maximum packet size for flow  $i$  and  $L_M$  is the maximum packet size allowed in the network. When a delay requirement is to be met by a flow  $i$ , the higher the burstiness of the source ( $\sigma_i$ ) and the number of intermediate nodes ( $H_i$ ), the larger the bandwidth  $g_i$  must be.

The scheduling delay (equation 8) is only a component of the overall end-to-end delay  $D_{e2e}$ . The other important terms of  $D_{e2e}$  are the packetization process ( $D_{pack}$ ) and the propagation delay  $D_m$  on the  $m^{th}$  link of the path; both the time needed for application level processing (e.g. audio compression) and protocol processing are considered negligible. The CAC is provided with a delay requirement  $D_{req}$  which is the network delay budget for the call, which is obtained by subtracting several terms from the delay acceptable by the user, and it uses the following inequality to decide whether to accept a new session:

$$D_{req} \geq D_{e2e} = D_{pack} + D_0 + \frac{\sigma_i + (H_i - 1) \cdot L_i}{g_i} + \sum_{m=1}^{H_i} \left( \frac{L_M}{r_m} + D_m \right) \quad (9)$$

The call admission deployed in the network accepts a new call whether each link on the call path has an appropriate amount of available (i.e. not yet reserved) bandwidth. This amount of bandwidth depends on  $B_{real}$ , which accounts for the real-time payload and the protocols overheads, and on  $PGPS$  Bandwidth ( $B_{PGPS}$ ), i.e. the bandwidth required to guarantee a specific end-to-end delay to the real-time session.  $B_{real}$  is given by equation 7 and  $B_{PGPS}$  is the minimum  $g_i$  value that satisfies inequality 9:

$$B_{PGPS} = \frac{\sigma_i + (H_i - 1) \cdot L_i}{D_{req} - D_{pack} - D_0 - \sum_{m=1}^{H_i} \left( \frac{L_M}{r_m} + D_m \right)} \quad (10)$$

Defining *Apparent Bandwidth* ( $B_{app}$ ) the bandwidth reserved by the CAC into the network, the CAC reserves to call  $i$  a bandwidth  $B_{app}$  that is:

$$B_{app} = \max(B_{real}, B_{PGPS}) \quad (11)$$

<sup>2</sup>Scheduling Delay is the sum of Queuing Delay and Transmission Delay.

When the amount of bandwidth  $B_{PGPS}$  needed to meet the QoS flow requirement is larger than the amount  $B_{real}$  required to transmit the flow  $i$  including protocol overheads, we say that *bandwidth over-allocation* is performed. When a call is torn down, the bandwidth previously reserved to it is released.

Bandwidth over-allocation, effective with the PGPS scheduler, can be less useful with other scheduler mechanisms. Particularly, there exist several schedulers in which effectiveness depends on other parameters (e.g. packet size is crucial with Weighted Round Robin and Class Based Queuing [19]), others that are specifically designed to provide guaranteed delay (e.g. Stop and Go queuing [20] or Jitter-EDD) and do not require over-allocation at all, and others that decouple bandwidth allocation and delay (Hierarchical Fair Service Curve [21] or Decoupled Class Based Queuing [22]).

## 2.4 NETWORK

Although this paper aims at providing a mathematical description of the packet switching efficiency, some simulation results are presented that originates from the network shown in Fig. 1; the topology has been designed after the one of a domestic telephone network. *Link C* has been split in several sublinks (connected by network nodes) in order to test paths using an arbitrary number of intermediate devices. This is used to simulate long-distance calls, which have several network devices along the path.

## 3 EFFICIENCY

The efficiency obtained by a session in a packet switched network can be split in several contributions. Since a voice channel has a bandwidth of  $B_{CS}$  (usually 64 Kbps) in a circuit switching network, the codification process has efficiency:

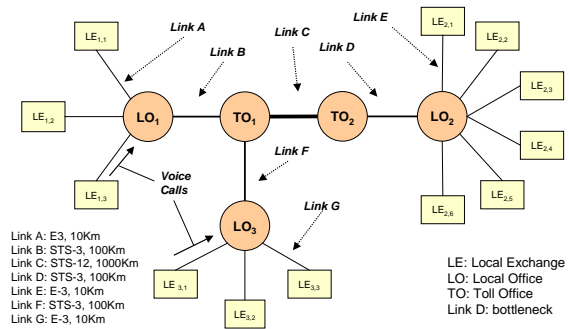


Figure 1: Network topology used in the simulations.

$$E_{coding} = \frac{B_{CS}}{B_{eff}} > 1 \text{ (usually)} \quad (12)$$

After the encoding process, the voice is to be packetized. The efficiency of this process is:

$$E_{pack} = \frac{B_{eff}}{B_{real}} < 1 \text{ (always)} \quad (13)$$

However the real-time session needs to be reserved a bandwidth equal to  $B_{app}$ ; therefore there is a further term that takes into account the allocation needed to transport the session. This term is the *Allocation Efficiency* ( $E_{alloc}$ ) and it can be defined as:

$$E_{alloc} = \frac{B_{real}}{B_{app}} \leq 1 \text{ (always)} \quad (14)$$

Since  $B_{real}$  is the “rough” bandwidth occupied by the call, we can define the *Transport Efficiency* ( $E_{TR}$ ):

$$E_{TR} = E_{coding} \cdot E_{pack} = \frac{B_{CS}}{B_{real}} \quad (15)$$

The  $E_{TR}$  gives an insight of the packet voice efficiency (in terms of bandwidth needed to transport the same call) compared to the circuit switching. The larger the  $E_{TR}$ , the higher the amount of traffic (data and real-time) that the network is able to transport. However this index does not take into account that a call can also need some *overallocated* bandwidth to meet its delay requirements.

The *Real-Time Efficiency* ( $E_{RT}$ ) can be defined as the overall efficiency in transporting real-time traffic:

$$E_{RT} = E_{TR} \cdot E_{alloc} = \frac{B_{CS}}{B_{app}} \quad (16)$$

The  $E_{RT}$  gives an indication of efficiency of the network when used to transport only real-time traffic. For instance, when the percentage of data traffic is significant the apparent bandwidth does not influence the overall efficiency because the bandwidth that is “reserved” (but not “used”) by real-time traffic can be exploited in carrying data traffic. Vice versa, the apparent bandwidth is important when part of that bandwidth is simply wasted because the amount of data is negligible. Unfortunately,  $E_{RT}$  is still limited since it takes into account the overallocation, but it does not consider that the overallocated bandwidth can be used to transport data traffic.

An overall index that takes into account both the Real-Time and Transport efficiency is the *Generalized Efficiency* ( $E_{gen}$ , shortened in *Efficiency* in the following of the paper) that can be defined as follows:

$$E_{gen} = \frac{\frac{B_{CS}}{1-d}}{\max\left(\frac{B_{real}}{1-d}, B_{app}\right)} = \quad (17)$$

$$\frac{\frac{B_{CS}}{1-d}}{\max\left(\frac{B_{real}}{1-d}, B_{PGPS}\right)} \quad (18)$$

where  $d$  is the percentage of the network load due to the data traffic (the last simplification is derived from equation 11). The numerator takes into account the bandwidth required to carry the imposed data/real-time traffic mix within a circuit switched network<sup>3</sup>. The denominator accounts the bandwidth required to carry the same traffic mix within a packet switched network<sup>4</sup>. In presence of a non-overallocating session, the network must leave “explicitly” free part of its bandwidth to transport also the imposed percentage of data. This is the case in which the  $\frac{B_{real}}{1-d}$  term is prevailing. Vice versa, the network can send data traffic for free using the large amount of bandwidth that has been overallocated (but not used) by a real-time session to meet its delay requirement. This is the case in which  $B_{app}$  is prevailing.

#### 4 MAXIMIZING THE EFFICIENCY

The maximization of the Generalized Efficiency can be obtained when the denominator of equation 18 is minimized, that consists in the optimization of the only parameter that can be tuned,  $D_{pack}$ . For instance, the call properties (codec type, end-to-end path characteristics in term of nodes and links) and the amount of network resources  $d$  that have to be dedicated to the data traffic are inputs of the problem and cannot be changed. At this point the maximum efficiency is obtained when:

$$\min_{0 < D_{pack} \leq D_{pack}^{max}} \left[ \max\left(\frac{B_{real}}{1-d}, B_{PGPS}\right) \right] \quad (19)$$

where  $D_{pack}^{max}$  is the maximum packetization delay admissible for that call, i.e. supposing that  $g_i = r_m = \infty$  (derived from equation 9):

$$D_{pack}^{max} = D_{req} - D_0 - \sum_{m=1}^H \left( \frac{L_M}{r_m} - D_m \right) \quad (20)$$

Supposing  $D_{pack}$  a continuous function, a closer look at equation 19 permit to enunciate the following properties.

**Property 1** When both  $B_{PGPS}$  and  $B_{real}$  are continuous functions,  $D_{pack}$  that minimizes the efficiency is the value derived when the following holds:

$$\min_{0 < D_{pack} \leq D_{pack}^{max}} \left( \frac{B_{real}}{1-d} = B_{PGPS} \right) \quad (21)$$

<sup>3</sup>Supposing that  $B_{CS} = 64$  Kbps and  $d = 50\%$ , the network needs 128 Kbps for each accepted session: 64 Kbps are used to transport the session while the others are devoted to data traffic.

<sup>4</sup>Both terms are normalized to the bandwidth required by a single call but it can be proved that, in presence of calls with the same characteristics, the same result can be obtained considering the overall network bandwidth instead of a single call.

*Proof is trivial since both functions are continuous: equation 21 is minimized when neither the first term nor the second one are prevailing, i.e. when they have the same value. It is worthy noticing that this property does not exclude the existence of several points in which  $\frac{B_{real}}{1-d} = B_{PGPS}$ .*

△

**Property 2** When  $B_{real}$  is a decreasing monotonic function,  $B_{PGPS}$  is an increasing monotonic function and both are continuous, the optimal packetization delay is obtained when:

$$\frac{B_{real}}{1-d} = B_{PGPS} \quad (22)$$

*Proof can easily derived keeping in mind property 1, that  $B_{real} > 0$  always and that  $0 \leq B_{PGPS} < \infty$ . Since both functions are monotonic and continuous, there will be a unique point  $D_{pack}^{opt}$  in which  $\frac{B_{real}}{1-d} = B_{PGPS}$ , and this point will be certainly  $0 < D_{pack}^{opt} \leq D_{pack}^{max}$ .*

*This gives a simple method to derive the highest efficiency of the network.*

△

**Property 3** When  $D_{pack} \leq D_{pack}^{opt}$  and the previous hypotheses still hold, following equation is valid:

$$\left[ \max \left( \frac{B_{real}}{1-d}, B_{PGPS} \right) \right]_{D_{pack}} = \left[ \frac{B_{real}}{1-d} \right]_{D_{pack}}$$

*In fact, when  $D_{pack} < D_{pack}^{opt}$  the real bandwidth is prevailing because of the same hypotheses of property 2. When  $D_{pack} = D_{pack}^{opt}$  the proof is trivial thanks to property 2:  $\frac{B_{real}}{1-d}$  and  $B_{PGPS}$  have the same value; therefore one of them is enough. In other words, as long as  $D_{pack} \leq D_{pack}^{opt}$ , the term containing  $B_{real}$  is the most important one.*

*For instance the computation of the  $B_{real}$  is also simpler than the computation of  $B_{PGPS}$ .*

△

Following sections will derive the Maximum Efficiency for ATM and IP technologies under the assumption that the packetization delay is a continuous function. At this stage, these technologies are supposed to be able to create packets that might not have an integer multiple of byte.

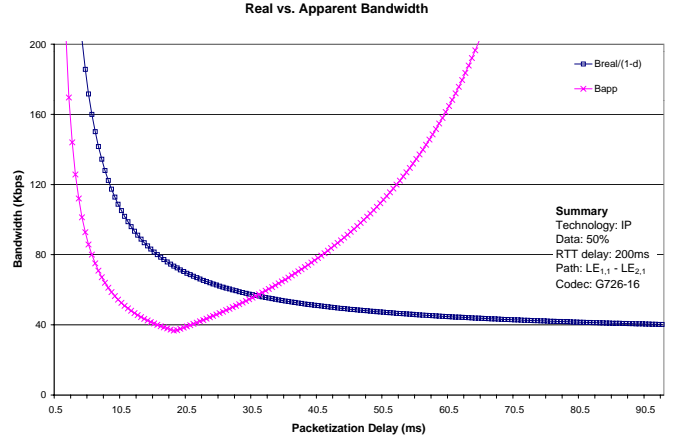


Figure 2: Maximizing the  $E_{gen}$ : Real vs Apparent Bandwidth with IP technology.

#### 4.1 IP

$B_{real,IP}$  is a continuous and monotonic decreasing function (figure 2). Deriving the maximum efficiency is straightforward and it involves solving equation 22 in order to derive  $D_{pack}^{opt}$ . This is obtained by merging equations 7 and 10 into 22 and taking into account that, for IP,  $\sigma_i = L_i = L$ .

$$D_{pack,IP}^{opt} = \frac{D_{req} - D_0 - \sum_{m=1}^H \left( \frac{L_m}{r_m} + D_m \right)}{H \cdot (1-d) + 1} \quad (23)$$

In presence of paths with limited number of nodes and large links, previous equation can be approximated as follows:

$$D_{pack,IP}^{opt} \simeq \frac{D_{req}}{H \cdot (1-d) + 1} \quad (24)$$

Previous equations are congruent with the result obtained in [10]. Figure 3 shows the optimal packetization delay  $D_{pack,IP}^{opt}$  for IP technology, with different percentage of data traffic and with different end-to-end requirements.

Maximum efficiency of IP, computed according to equation 18, requires  $B_{real,IP}$  that can be easily derived taking into account equations 7,5,4:

$$B_{real,IP} = B_{eff} + \frac{P_{H,IP}}{D_{pack,IP}^{opt}} \quad (25)$$

Efficiency can be derived by substituting equation 25 into 18 and taking into account both equation 23 and property 3:

$$E_{gen,IP}^{max} = \frac{B_{CS}}{B_{eff} + \frac{P_{H,IP}}{D_{pack,IP}^{opt}}} \quad (26)$$

The efficiency is proportional to  $D_{pack,IP}^{opt}$ , so it is higher when the number of nodes on the path is limited,

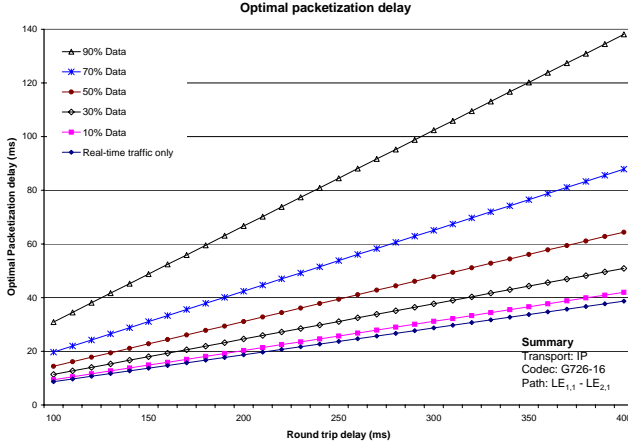


Figure 3: Optimal delay varying the percentage of data traffic.

the data percentage is not negligible and the end-to-end requested delay is not too strict.

It is worth noticing that the point  $D_{pack}^{opt}$  that maximizes the efficiency is independent from the utilized codec (i.e.  $B_{eff}$ ) while the value of the Generalized Efficiency depends on it: the more effective the compression, the higher the efficiency. In other words, even if the relative weight of the protocol headers against the real-time data payload increases, this is not a problem from the efficiency point of view.

#### 4.2 ATM SINGLE-CELL (ATM\*)

Due to the excellent delay properties of ATM\* (small cell length, small packetization delays required to fill a cell), we can assume that ATM\* does not have overallocation<sup>5</sup>. According to this assumption, maximum efficiency  $E_{gen,ATM*}^{max}$  is reached when the real bandwidth is minimized, i.e. when the cell is completely full ( $C = C_{PD} + C_H = L_{PD} + P_{H,ATM*} + C_H$ ).

Optimal packetization delay  $D_{pack,ATM*}^{opt}$  can be derived from equations 4,6 taking into account that  $N = 1$ :

$$D_{pack,ATM*}^{opt} = \frac{C_{PD} - P_{H,ATM*}}{B_{eff}} \quad (27)$$

Efficiency can be found in a similar way of the IP case, substituting equation 27 into 18 and taking into account property 3<sup>6</sup>.  $B_{real,ATM*}$  can easily be computed by means of equations 7,4 and keeping in mind that the payload fits into a single cell that is completely full:

$$B_{real,ATM*} = \frac{C \cdot B_{eff}}{C_{PD} - P_{H,ATM*}} \quad (28)$$

<sup>5</sup>Simulations showed that in a typical network topology (like the one in figure 1) ATM single-cell might require more than 100 network nodes before needing overallocation, that is unlikely in normal operating conditions.

<sup>6</sup>In ATM\* both  $B_{real}$  and  $B_{PGPS}$  are continuous functions (in the range of interest  $0 < D_{pack,ATM*} \leq D_{pack,ATM*}^{opt}$ ).

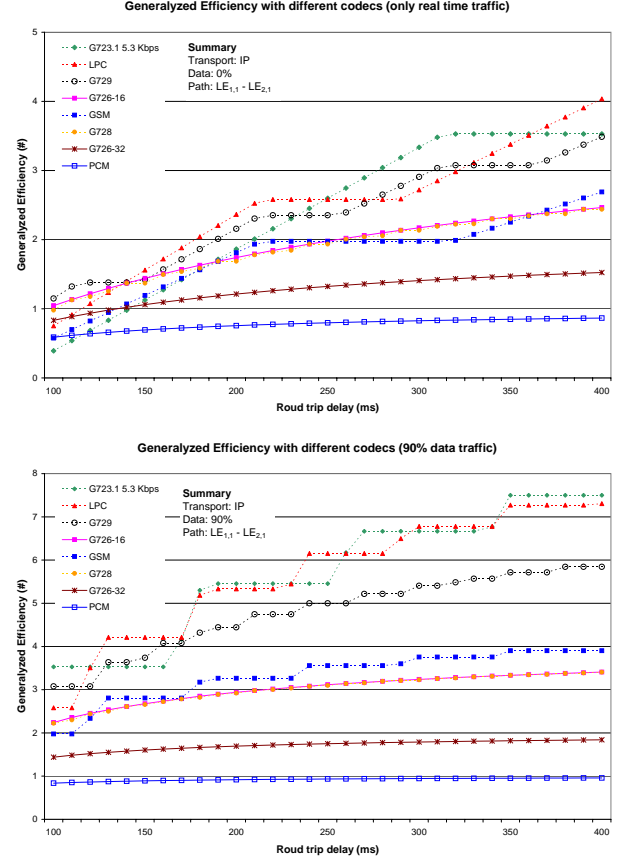


Figure 4: Generalized Efficiency varying the percentage of data traffic, with IP technology.

Maximum efficiency for ATM\* is:

$$E_{gen,ATM*}^{max} = \frac{B_{CS} \cdot (C_{PD} - P_{H,ATM*})}{B_{eff} \cdot C} \quad (29)$$

Both values  $D_{pack,ATM*}^{opt}$  (i.e. where the efficiency is maximized) and  $E_{gen}$  are dependent on the codec. For instance, halving the  $B_{eff}$  of the codec corresponds to an equal cut of the packetization delay and a two-times increasing of efficiency<sup>7</sup>.

Under the assumption that ATM does not have overallocation, ATM\* has several differences from IP: the efficiency does not depend on the end-to-end required delay, on the length of the path and on the percentage of data traffic. This makes ATM\* suitable for networks with mainly real-time traffic, long paths and strict end-to-end delays.



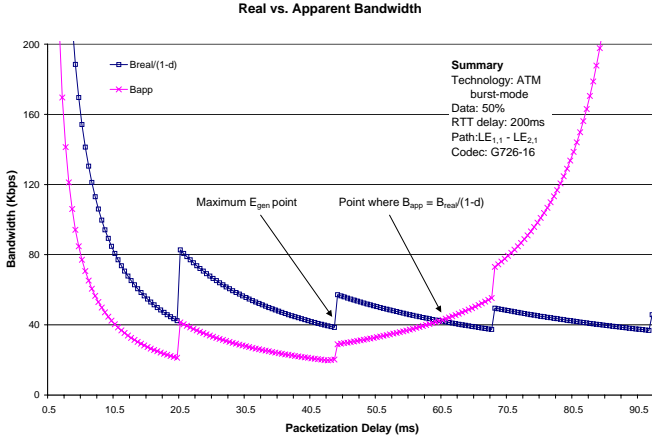


Figure 5: Maximizing the  $E_{gen}$ : Real vs. Apparent Bandwidth with ATM technology.

### 4.3 ATM

This case is different from the previous ones because  $B_{real,ATM}$  is no longer a continuous nor decreasing monotone function (figure 5) due to the small and fixed-length cells. For that reason equation 22 cannot be used alone to derive the maximum efficiency.

However it can be demonstrated the existence of a new set of properties that hold in the ATM case.

**Property 4** the maximum efficiency in ATM can be reached either (1) when equation 22 is satisfied (but there is no guarantee of an unique solution), or (2) when a cell is completely full.

A closer look at  $B_{real}, B_{PGPS}$  functions shows that these are monotonic and continuous except for the points in which the increasing packetization delay brings about the need of a new cell. If these intervals are considered separately, properties 1,2 are still valid, therefore demonstrating first case of this property. However this does not exclude equation 22 from being verified in more than one interval, that is an event that can happen indeed.

An intuitive proof of the second point can be easily seen in figure 5. Since both  $\frac{B_{real}}{1-d}$  and  $B_{PGPS}$  jump up as soon as a new cell is required (then  $B_{real}$  decreases again while  $B_{PGPS}$  continues increasing), the point in which a cell is full can be the optimal point because:

$$\left[ \max \left( \frac{B_{real}}{1-d}, B_{PGPS} \right) \right]_{N \text{ cells}} < \left[ \max \left( \frac{B_{real}}{1-d}, B_{PGPS} \right) \right]_{N \text{ cells} + 1 \text{ byte}}$$

<sup>7</sup>Taking into account the definition of Coding Efficiency ( $E_{coding}$ ) in equation 12, equation 29 can be rewritten as:

$$E_{gen,ATM}^{max} = E_{coding} \cdot \frac{C_{PD} - P_{H,ATM}}{C}$$

i.e. the maximum efficiency point calculated when  $N$  cells are completely full is larger than the value computed when the payload is one byte larger than before (therefore it spans over  $(N + 1)$  cells). It follows that the maximum efficiency point can be reached, aside from case (1), also when a cell is completely full.

△

**Property 5** when maximum efficiency is due to case (2) of property 4, the number of cells that maximize the efficiency is one less compared to the number of cells needed that maximize the efficiency according to case (1) of property 4.

Proof follows directly from the proof of case (2) of property 4. If the best result according to case (1) is reached with  $(N^{opt} + 1)$  cells, the best efficiency point according to case (2) computed when  $N \leq N^{opt}$  is the one that refers to  $N^{opt}$  cells. In fact:

$$\left[ \max \left( \frac{B_{real}}{1-d}, B_{PGPS} \right) \right]_{N^{opt} \text{ cells}} < \left[ \max \left( \frac{B_{real}}{1-d}, B_{PGPS} \right) \right]_{N < N^{opt} \text{ cells}}$$

because  $B_{real}$ , while computed only in correspondence of a full cell, is a decreasing function (therefore  $B_{real,N^{opt}} < B_{real,N < N^{opt}}$ ); therefore previous inequality is always verified for  $N \leq N^{opt}$  no matter of  $B_{PGPS}$ .

Using the same method it is possible to demonstrate that there are no  $N \geq (N^{opt} + 1)$  in which the efficiency is better than the value obtained using  $N^{opt}$ . The most important term becomes  $B_{PGPS}$  which is an increasing function; therefore there cannot exist any value  $N \geq (N^{opt} + 1)$  in which  $B_{PGPS}$  is smaller than the value computed according to case (1) of property 4. It follows that the optimal efficiency point due to case (2) has a payload that spans a number of cells that is one less compared to the number of cells needed that maximize the efficiency according to case (1) of property 4.

△

**Property 6** when  $D_{pack} \leq D_{pack,ATM}^{opt}$  the following equation is valid for ATM technology:

$$\left[ \max \left( \frac{B_{real,ATM}}{1-d}, B_{PGPS} \right) \right]_{D_{pack}} = \left[ \frac{B_{real,ATM}}{1-d} \right]_{D_{pack}}$$

The first part of the proof demonstrates that this property is valid when  $D_{pack} = D_{pack,ATM}^{opt}$ . If the optimum point is reached when  $\frac{B_{real}}{1-d} = B_{PGPS}$  holds (case (1) of property 4), both the first and the second term have the same value and this property is verified. Vice versa, if the optimum point is reached when a cell is completely full (case (2)) the term  $\frac{B_{real}}{1-d}$  will prevail over  $B_{PGPS}$  (see property 5) and this property is verified again.

Second part (when  $D_{pack} < D_{pack,ATM}^{opt}$ ) can be demonstrated taking into account that  $B_{PGPS}$  is a monotone increasing function. For any packetization delay smaller than the optimal one,  $B_{PGPS}$  computed in that point is smaller than the value of  $B_{PGPS}$  in the optimal point. If would exist a point  $D_{pack}^{absurd} < D_{pack,ATM}^{opt}$  in which  $\frac{B_{real}}{1-d} < B_{PGPS}$  then this will be the optimal point, which makes no sense.

This property extends property 3 and it is of great advantage to simplify the computation of the  $E_{gen,ATM}^{max}$ .

△

The derivation of  $E_{gen,ATM}^{max}$  needs to compute the efficiency into both points expressed by property 4 and to take the best result among them.

Obtaining the packetization delay that satisfies equation 22 is harder than in previous sections. While  $E_{gen}$  might be derived merging equations 7 and 10 into 22 and taking into account that (for ATM)  $\sigma_i = N \cdot C$  and  $L_i = L_M = C$ , the solution is not in a close-form because the number of cells  $N$  depends on  $D_{pack}$  and there are not any feasible simplifications; the best result will be:

$$D_{pack,ATM}^{opt} = \frac{N}{(1-d) \cdot (N+H-1) + N} \cdot \left[ D_{req} - D_0 - \sum_{m=1}^H \left( \frac{C}{r_m} + D_m \right) \right] \quad (30)$$

The optimal packetization delay can instead be obtained with a simple iterative method based on equation 19:

$$D_{pack,ATM}^{opt} = D_{pack} : \min_{0 < D_{pack} \leq D_{pack}^{max}} \left\{ \max \left( \frac{B_{real}}{1-d}, B_{PGPS} \right) \right\} \quad (31)$$

where  $D_{pack}^{max}$  is the value derived from equation 20.

The value of maximum  $E_{gen}$  reached by ATM can be obtained with the usual method, computing its  $B_{real,ATM}$  by means of equations 7,6,4:

$$B_{real,ATM} = \frac{C}{D_{pack,ATM}^{opt}} \cdot \left[ \frac{P_{H,ATM} + B_{eff} \cdot D_{pack,ATM}^{opt}}{C_{PD}} \right] \quad (32)$$

Taking into account property 6 and equation 18, ATM efficiency is:

$$E_{gen,ATM}^{max} = \frac{B_{CS} \cdot D_{pack,ATM}^{opt}}{C \cdot \left[ \frac{P_{H,ATM} + B_{eff} \cdot D_{pack,ATM}^{opt}}{C_{PD}} \right]} \quad (33)$$

As expected,  $E_{gen,ATM}^{max}$  is inversely proportional to the efficient bandwidth of the codec and proportional to the value of the optimal packetization delay. Figure 12 shows that  $E_{gen,ATM}^{max}$  increases at the increase of the end-to-end requested delay: this suggests that the optimal packetization delay is somehow proportional to the end-to-end requested delay.

## 5 IDENTIFYING THE BEST TECHNOLOGY

This section aims at the determination of the best technology for a given usage environment (call properties, network characteristics, end-to-end requested delay). Obviously there is not a clear answer to this question since each technology performs the best under some circumstances. However it is possible to determine the best operating range for each technology, i.e. when:

$$E_{gen,tech1} > E_{gen,tech2} \quad (34)$$

Following sections will answer to this question.

### 5.1 IP vs. ATM\*

Before comparing IP and ATM\*, we need to demonstrate the following property.

**Property 7** *IP outperforms ATM\* when the real bandwidth of IP, computed in the IP optimal point, is smaller than the real bandwidth of ATM\*, computed in the ATM\* optimal point:*

$$[B_{real,IP}]_{D_{pack,IP}^{opt}} < [B_{real,ATM*}]_{D_{pack,ATM*}^{opt}} \quad (35)$$

*Proof is trivial and it follows directly from inequality 34 (computed in the maximum efficiency point of IP and ATM\*), equation 18 and property 3 (which holds for both IP and ATM\*).*

*In order to simplify the notation, the real bandwidth computed in the optimal IP point will be shortened as  $B_{real,IP}^{opt}$ ; the same notation will be used for ATM\* and ATM.*

△

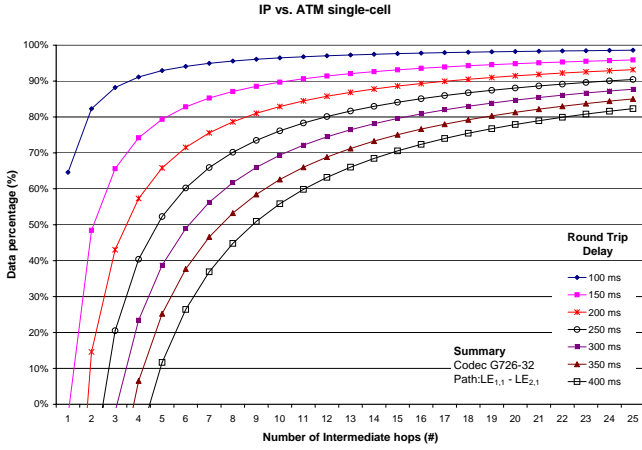


Figure 6: When IP is convenient over ATM\*: the needed percentage of data traffic.

Intuitively, IP is convenient over ATM\* when the overhead (percentage) due to the IP headers becomes smaller than the overhead due to the ATM\* headers. The precise comparison between IP and ATM\* can be done by means of property 7 and by taking into account that  $B_{real,IP}$  and  $B_{real,ATM*}$  are known from equations 25,28. It is easy to derive when IP outperforms ATM\*:

$$D_{pack,IP}^{opt} > \frac{P_{H,IP}}{\frac{C \cdot B_{eff,ATM*}}{C_{PD} - P_{H,ATM*}} - B_{eff,IP}} \quad (36)$$

However the  $D_{pack,IP}^{opt}$  is known from equation 23:

$$\begin{aligned} \frac{D_{req} - D_0 - \sum_{m=1}^H \left( \frac{L_M}{r_m} + D_m \right)}{H \cdot (1 - d) + 1} &> \\ &> \frac{P_{H,IP}}{\frac{C \cdot B_{eff,ATM*}}{C_{PD} - P_{H,ATM*}} - B_{eff,IP}} \end{aligned} \quad (37)$$

and, if the same approximations of equation 24 hold<sup>8</sup>:

$$\frac{D_{req}}{H \cdot (1 - d) + 1} > \frac{P_{H,IP}}{\frac{C \cdot B_{eff,ATM*}}{C_{PD} - P_{H,ATM*}} - B_{eff,IP}} \quad (38)$$

Inequalities 37,38 can be simplified if the codec deployed by IP and ATM\* is the same. For instance, inequality 38 becomes:

$$\frac{D_{req}}{H \cdot (1 - d) + 1} > \frac{P_{H,IP}}{B_{eff} \cdot \left( \frac{C}{C_{PD} - P_{H,ATM*}} - 1 \right)} \quad (39)$$

Figure 6 shows when IP has a better efficiency than ATM\*. It can be noted, for example, that IP is still the

<sup>8</sup>Inequality 38 is a superior approximation of 37; while IP is certainly better than ATM\* when inequality 38 is verified, there might exist some points in which IP is better and that inequality does not hold.

best choice in a path with 20 hops (G.726-32 codec, round trip delay of 200 ms) when the percentage of data is more than 90%.

Deriving  $D_{pack}$  from equation 4, applying the result to equation 36 and substituting the proper value to the variables ( $P_{H,IP} = 48$  bytes due to RTP, UDP, IP, PPP headers;  $P_{H,AAL5} = 8$  bytes,  $C = 53$  bytes,  $C_{PD} = 48$  bytes), we can obtain the minimum Data Payload ( $L_{PD}$ ) in which  $E_{gen,IP} < B_{gen,ATM*}$ :

$$L_{PD} > \frac{48 \cdot B_{eff,IP}}{\frac{53}{40} \cdot B_{eff,ATM*} - B_{eff,IP}}$$

If  $B_{eff,IP} = B_{eff,ATM*}$ , it is trivial to derive that IP outperforms ATM\* when  $L_{PD} > 147$  bytes, that is equivalent to a packetization delay of 18.4 ms with a PCM codec (36.8 ms with an G.726-32 and 73.5 ms with an G.726-16). High bit rate codecs make this quite a small value, confirming that in this case IP can be a better choice than ATM\*.

Summarizing, ATM\* is the best technology when the end-to-end delay requirement is strict, the end-to-end path has a high number of intermediate nodes and with high bandwidth codecs. Low bit rate codecs means that the overhead of the packet headers is quite high compared to the cell overhead, so that ATM\* is the best choice for them. IP can be the best technology when the network is intended to carry mainly data traffic, when the number of nodes is limited and in presence of high bandwidth real-time sessions (large  $B_{eff}$ ).

## 5.2 IP vs. ATM

The comparison between ATM and IP is not easy due to the problems related to the computation of the optimal point for an ATM network.

An indication can however be obtained exploiting the following set of properties.

**Property 8** When the optimal packetization delay for ATM is due to case (1) of property 4, optimal packetization delay for IP is always less than the ATM one:

$$D_{pack,IP}^{opt} < D_{pack,ATM}^{opt}$$

*Proof can be obtained by comparing equations 23, 30, i.e. the points in which equation 22 holds. For instance the first term is always larger in ATM because of the smallest cell size ( $C < L_M$ ):*

$$\begin{aligned} &\left[ D_{req} - D_0 - \sum_{m=1}^H \left( \frac{C}{r_m} + D_m \right) \right]_{ATM} > \\ &> \left[ D_{req} - D_0 - \sum_{m=1}^H \left( \frac{L_M}{r_m} + D_m \right) \right]_{IP} \text{ always} \end{aligned}$$

Moreover, also the second term is always larger in ATM:

$$\frac{N}{(1-d) \cdot (N+H-1) + N} \geq \frac{1}{H \cdot (1-d) + 1} \text{ always}$$

Since both factors that appear in the  $D_{pack,ATM}^{opt}$  computation are bigger or equal to the equivalent terms in the  $D_{pack,IP}^{opt}$ , it means that  $D_{pack,ATM}^{opt} > D_{pack,IP}^{opt}$  (when it is derived within the case (1) of property 4).

This result shows (surprisingly) that the optimal packetization delay in ATM is always larger than the IP one, even when ATM has larger protocol overheads compared to IP (in ATM you have a fixed 5 bytes fee for each cell).

**Property 9** When the optimal packetization delay for ATM is due to case (2) of property 4, there is no guarantees that optimal packetization delay for IP is always less than the ATM one.

Proof follows from property 8 which does not specify how big is the difference  $D_{pack,ATM,case(1)}^{opt} - D_{pack,IP}^{opt}$  that can be a very small value  $\epsilon$ . Therefore, if  $D_{pack,ATM,case(1)}^{opt}$  corresponds to a payload spanning over  $M$  cells,  $D_{pack,ATM,case(2)}^{opt}$  might correspond to a payload filling completely  $(M-1)$  cells, giving no assurances that still  $D_{pack,IP}^{opt} < D_{pack,ATM,case(2)}^{opt}$ .

**Property 10** ATM outperforms IP when  $B_{real,ATM}^{opt}$  is smaller than the IP one ( $B_{real,ATM}^{opt} < B_{real,IP}^{opt}$ ).

This can be easily demonstrated in a similar way of the one of property 7, applying property 6 to inequality 34.

Unfortunately the exact points in which ATM outperforms IP are not easily derivable; therefore there will be derived three ranges, the first one where ATM outperforms IP, the second one where IP outperforms ATM, and the third one that is an uncertainty range (*limbo*) in which we are not able to derive analytically which technology is the best.

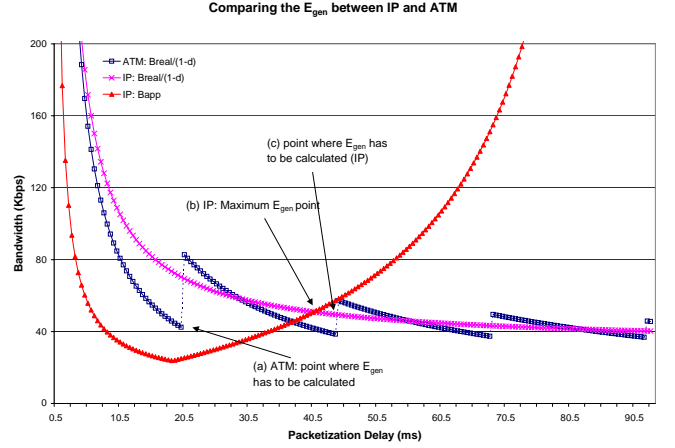


Figure 7: When ATM outperforms IP: how approximations have been made.

### 5.2.1 When ATM is better

The comparison between ATM and IP requires the use of a series of approximations that can be seen in figure 7 unless using an iterative method that has to be invoked each time.

According to property 4, the maximum efficiency point can be derived keeping the best value among case (1) and case (2). Calling  $E_{gen,ATM,case(2)}^{opt}$  the maximum efficiency due to case (2),  $E_{gen,ATM,case(2)}^{opt} \leq E_{gen,ATM}^{opt}$  always. Since, due to property 6,  $B_{real}$  is the most important term<sup>9</sup>,

$$B_{real,ATM,case(2)}^{opt} \geq B_{real,ATM}^{opt} \text{ always} \quad (40)$$

According to property 10, ATM outperforms IP when  $B_{real,ATM}^{opt} < B_{real,IP}^{opt}$  for a given range of  $D_{pack}$ . Due to inequality 40, IP will certainly outperform ATM when  $B_{real,ATM,case(2)}^{opt} < B_{real,IP}^{opt}$ .

If  $N^{opt}$  is the number of cells required to derive  $B_{real,ATM,case(2)}^{opt}$ , the real-time payload correspondent to  $D_{pack} = D_{pack,ATM,case(1)}^{opt}$  will require  $(N^{opt} + 1)$  cells (Property 5).

Since property 8 assures that  $D_{pack,IP}^{opt} < D_{pack,ATM}^{opt}$  and that  $B_{real,IP}$  gets even smaller when it is computed using  $D_{pack*} \geq D_{pack,ATM}^{opt}$ , previous inequality can be approximated:

$$ATM \text{ better} \iff B_{real,ATM,case(2)}^{opt} \leq B_{real,IP}^{D_{pack*}} \quad (41)$$

Particularly,  $D_{pack*}$  might be the value that corresponds to a data payload that will fill completely  $(N^{opt} + 1)$  cells in ATM technology. It follows that as long as:

$$[B_{real,ATM*}]_{D_{pack}^{N^{opt}}} < [B_{real,IP}]_{D_{pack}^{N^{opt}+1}} \quad (42)$$

<sup>9</sup>As long as  $D_{pack} \leq D_{pack,ATM}^{opt}$ .

ATM will be certainly a better choice than IP.

It can be proved that this is always verified if the IP optimal point corresponds to a data payload  $L_{PD}$  that is contained into max 7 cells, i.e.  $L_{PD,IP} = 308$  bytes. This corresponds to a packetization delay of 38.5 ms for a PCM codec, 77 ms for an G.726-32 one and 154 ms for an G.726-16 one.

### 5.2.2 When IP is better

According to property 10, IP is better than ATM when  $B_{real,IP}$  becomes smaller than any possible value of  $B_{real,ATM}$ . This is possible because ATM has an extra-overhead of  $C_H$  bytes every  $C_{PD}$  bytes of the data payload, while IP does not. Therefore the real bandwidth of ATM cannot decrease under a certain value that is bigger than the corresponding limit for IP. Therefore IP outperforms ATM when, whatever  $B_{real,IP}$  is, the following holds:

$$B_{real,IP} < \lim_{D_{pack,ATM} \rightarrow \infty} B_{real,ATM}$$

IP becomes certainly convenient over ATM when the  $L_{PD,IP} > 460$  bytes. This is a quite high packetization delay for codecs usually deployed in voice compression. This value corresponds to 57.5 ms for PCM, 115 ms for G.726-32 and 230 ms for G.726-16.

### 5.2.3 The Limbo

There is a region where is not possible to determine analytically whether IP is convenient over ATM. It has been determined the convenience of ATM when the IP optimal point is less than 308 bytes and the convenience of IP when IP optimal point is more than 460 bytes. The region in the middle can be considered a sort of “limbo” because it is not possible to determine analytically, a priori, which technology is the best; therefore the iterative method is required.

For instance, simulations confirm that IP becomes convenient over ATM in presence of a round trip delay of 290 ms and a G726-32 codec ( $E_{gen,IP} = 1.781$ ,  $E_{gen,ATM} = 1.776$  with  $L_{PD,IP} = 395$  bytes and packetization delay of 98.75 ms), that becomes 160 ms with a PCM codec ( $E_{gen,IP} = 0.897$ ,  $E_{gen,ATM} = 0.892$  with  $L_{PD,IP} = 419$  bytes and packetization delay of 52.4 ms).

## 6 CODEC GRANULARITY

Packetization delay has always been considered being a continuous function; therefore previous equations define the optimal packetization delay without caring whether this leads to an acceptable packet size nor the selected codec permits that value. High granularity codecs (usually the sample-based ones) are able to approximate the optimal packetization delay in a reasonable way. Vice versa, results

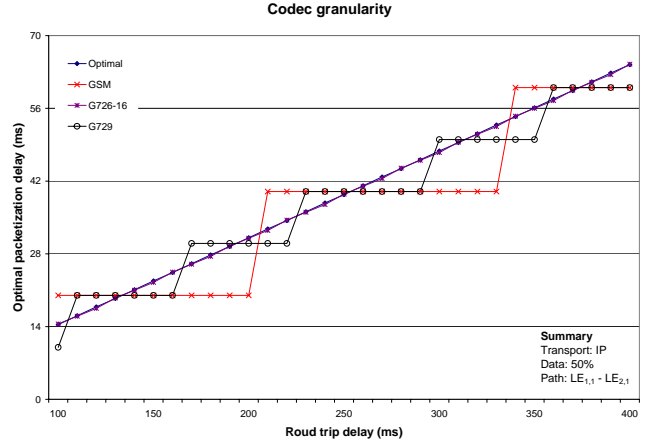


Figure 8: Codec Granularity: the approximation of the optimal packetization delay.

obtained with low granularity codecs can be substantially different from the optimal values. Figure 8 shows the optimal packetization delay for several codecs pointing out that  $D_{pack}^{opt}$  for G726-16 is almost indistinguishable from the ideal one, while GSM and G729 (with  $D_{pack}^{min}$  of 20 and 10 ms respectively) differ significantly from the ideal trace.

The optimal packetization delay computed in previous sections can be seen an upper bound of the maximum efficiency obtainable by that network configuration and it can be used to a preliminary comparison among the various technologies. The computation of the real optimal packetization delay can be done by defining  $D_{pack}^{opt-}$  and  $D_{pack}^{opt+}$  as the first admissible packetization delay below and above the theoretical one:

$$D_{pack}^{opt-} = \left\lfloor \frac{D_{pack}^{opt}}{D_{pack}^{min}} \right\rfloor \cdot D_{pack}^{min} \quad (43)$$

$$D_{pack}^{opt+} = \left\lceil \frac{D_{pack}^{opt}}{D_{pack}^{min}} \right\rceil \cdot D_{pack}^{min} \quad (44)$$

Some technologies (notably ATM\* and IP) are able to exploit these values in a very simple way; next sections will show the impact of the *real* optimal packetization delay on various technologies.

### 6.1 IP

IP makes the adaptation between the ideal and real  $D_{pack}^{opt}$  simpler. With the previous definition, the maximum efficiency is determined by:

$$E_{gen,IP} = \frac{\frac{B_{CS}}{1-d}}{\max \left( \left\lceil \frac{B_{real}}{1-d} \right\rceil D_{pack}^{opt-}, [B_{PGPS}] D_{pack}^{opt+} \right)} \quad (45)$$

A packetization delay that is smaller than the ideal one gives a predominance of the  $B_{real}$  term because of the



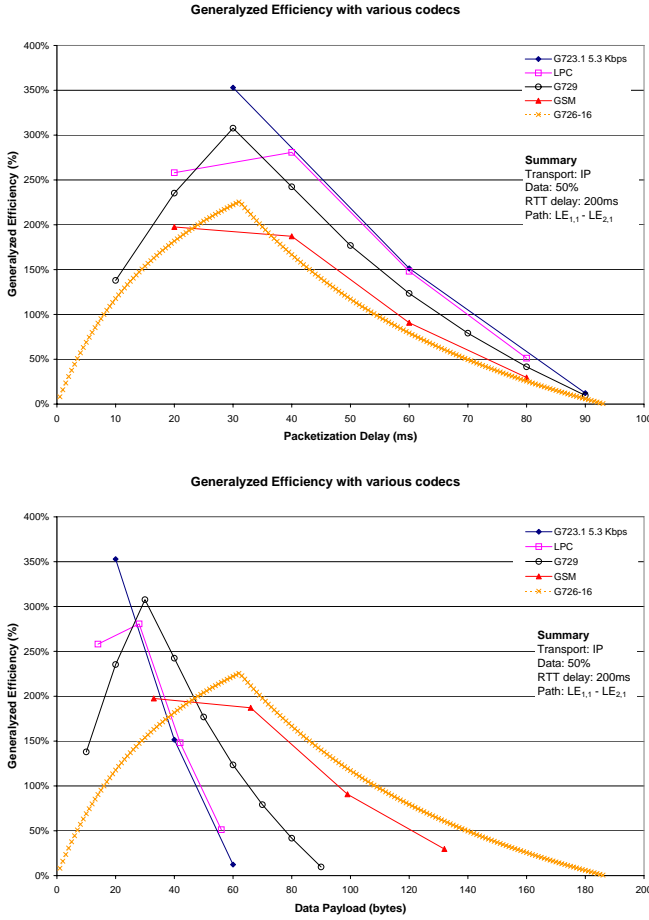


Figure 9: Variation of the Generalized Efficiency with different codecs, with IP technology. The  $E_{gen}$  is plotted in function of the packetization delay and the data payload.

larger header overheads. Vice versa, a packetization delay bigger than the optimal value leads to a predominance of the  $B_{PGPS}$  term.

Equation 45 gives also a method to determine the best codec. Since  $D_{pack}^{opt}$  is known by means of equation 23, the comparison of the results of equation 45 highlights the codec that has the best efficiency in the given conditions (network characteristics, real-time session properties).

Figure 9 shows the  $E_{gen}$  obtained with different codecs; traces are meaningful only in presence of a mark that indicates the valid points according to the codec granularity. This figure shows that the codec characteristics are of primary importance in getting the highest efficiency. For instance, G.729 has the highest efficiency even if LPC codec has the best efficient bandwidth ( $B_{eff,LPC} = 5.6Kbps$  and  $B_{eff,G.729} = 8Kbps$ ). LPC granularity is quite low, thus this codec is not able to approximate the optimal packetization delay with a sufficient precision. G.729 has smaller  $E_{gen}$ , but its better granularity permits the selection of a packetization delay closer to the optimal value.

## 6.2 ATM\*

In case of codecs with small granularity, the  $B_{real,ATM^*}$  is computed taking into account equations 7,3 and 43, since the requirement is to use no more than one cell. In this case:

$$B_{real,ATM^*} = \frac{C}{\left\lceil \frac{C_{PD}-P_H}{L_{PD}^{min}} \right\rceil \cdot D_{pack}^{min}} \quad (46)$$

This value is used (instead of the one presented in equation 28) to obtain the  $E_{gen,ATM^*}^{opt}$ , showed in figures 10,11. Although these figures refer to ATM, maximum ATM\* efficiency can be seen when the data payload is  $L_{PD} \leq 40$  bytes, i.e. the payload is contained in a single cell.

## 6.3 ATM

As it was already shown, it is not easy to derive the  $D_{pack,ATM}^{opt}$  in the ATM case. The codec granularity inserts another degree of freedom so that the optimal packetization delay cannot be derived analytically. Moreover equations 43,44 cannot be used for ATM because they do not guarantee that neither  $D_{pack}^{opt-}$  nor  $D_{pack}^{opt+}$  are optimal efficiency points. In fact these points might lead to a non-optimal  $E_{gen}$  because of the different number of cells (possibly) required compared to the  $D_{pack}^{opt}$ .

The best way to cope with the problem is to re-apply the iterative method already shown in section 4.3 (equation 31), although in this case the packetization delay varies according to the codec granularity:

$$D_{pack,ATM}^{opt} = D_{pack} : \min_{0 < D_{pack} \leq D_{pack}^{max}, D_{pack} = K \cdot D_{pack}^{min}} \left\{ \max \left( \frac{B_{real}}{1-d}, B_{PGPS} \right) \right\} \quad (47)$$

Once derived the  $D_{pack,ATM}^{opt}$ , the maximum efficiency guaranteed by the network can be computed applying its definition (equation 18). No simplifications are available in this case.

Figures 10,11 show the generalized efficiency with various codecs. It can be noted that almost all the codecs reach the maximum efficiency when the payload is 40 bytes, that is the maximum allowed into a single cell. This means that in these conditions ATM\* and ATM perform the same.

## 7 COMPARING DIFFERENT TECHNOLOGIES

Figures 12,13 show the comparison among IP, ATM\* and ATM when the data percentage is 50% and 90%. The first result is that ATM seems not to have substantial advantages over ATM\* (single cell). ATM gets slightly better when the payload does not fit into a single cell, for example when high bit rate codecs are deployed: ATM outperforms ATM\* when a G726-32 (32 Kbps) is deployed, even

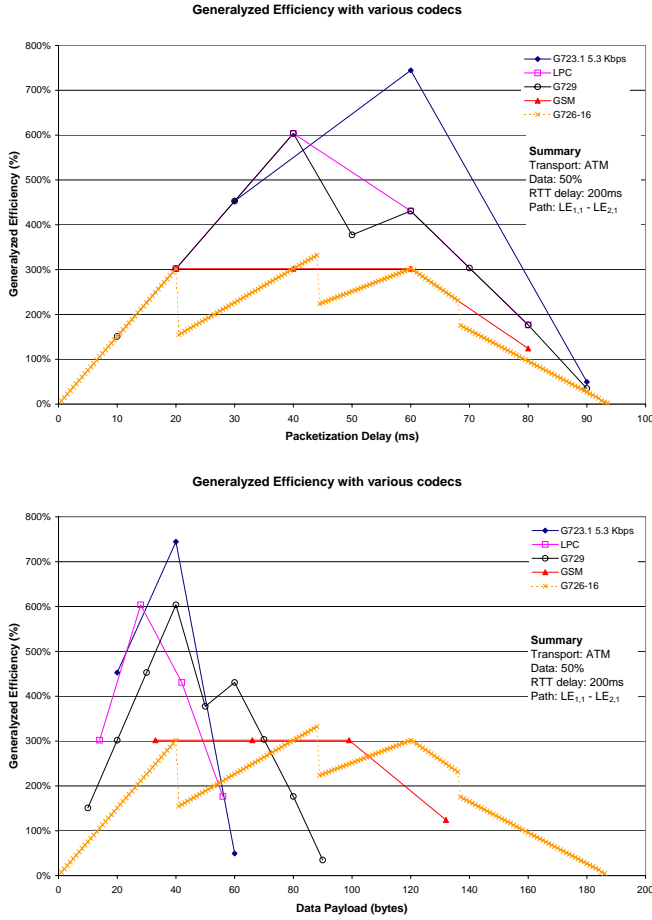


Figure 10: Variation of the Generalized Efficiency with different packetization delays, with ATM technology. The  $E_{gen}$  is plotted in function of the packetization delay and the data payload.

with strict delay requirements (G726-32 codec, 100 ms round trip delay, 50% best effort:  $E_{gen,ATM^*} = 1.51$ ,  $E_{gen,ATM} = 1.66$ ). Low bit rate codecs usually reach the maximum efficiency using no more than a cell unless the delay requirement is really large; for instance, a G723.1 - 5.3Kbps codec needs a packetization delay of more than 60 ms in order to span among multiple cells. Low bit rate codecs clear any differences between ATM\* and ATM.

Increasing the percentage of data traffic (figure 13) does not bring to any substantial improvement in ATM. Low bit rate codecs still prevent this technology from using multiple cells unless the delay requirement is fairly large; high bit rate codecs performs almost the same as before (for example a G.726-32 codec, 100 ms round trip delay and 90% data has  $E_{gen,ATM} = 1.66$  compared to 1.71 when data was 50%). Comparing figures 10,11 it becomes clear that the maximum efficiency is reached in correspondence of a single cell (Data Payload equal to 40 bytes) for most of low bit rate codecs, because the main component of the end-to-end delay is the packetization process. This often prevents low bit rate codecs from spanning over multiple cells.

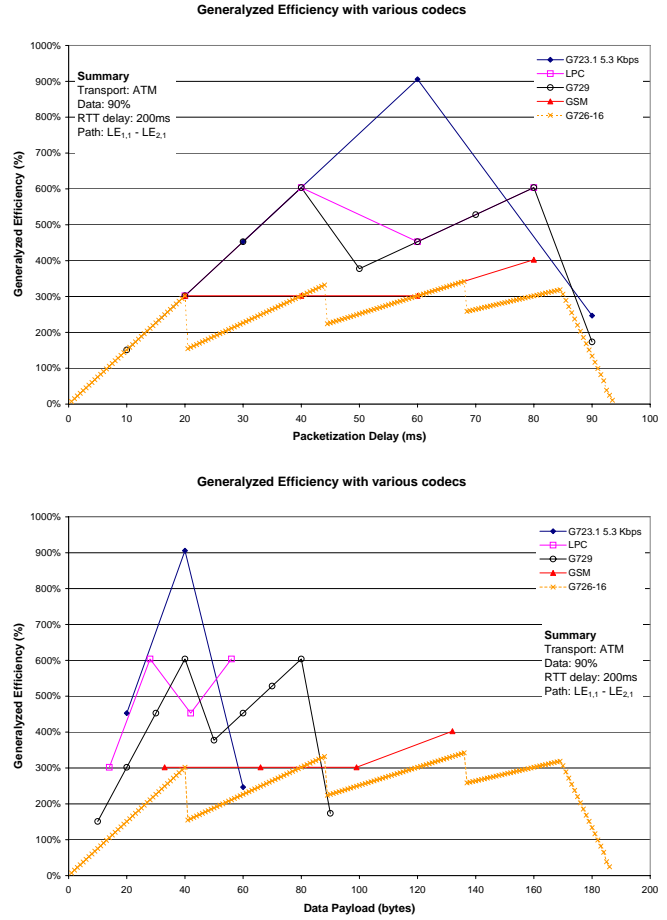


Figure 11: Variation of the Generalized Efficiency with different packetization delays, with ATM technology. The  $E_{gen}$  is plotted in function of the packetization delay and the data payload.

Deploying ATM (when the payload can spans over multiple cells) can be a risky choice. Low bit rate codecs make ATM\* and ATM almost the same. However when high bit rate codecs are deployed (maybe together with loose end-to-end delays and high volumes of best effort traffic) the advantage of ATM cannot be worthwhile because IP performs quite well too (for instance a G.726-32 codec, 300 ms round trip delay and 90% best effort shows that  $E_{gen,IP} = 1.79$ ,  $E_{gen,ATM} = 1.78$ ). In fact, as shown in section 5.2.2, IP certainly outperforms ATM when the data payload exceeds 460 bytes.

Developing new equipments for ATM (multiple cell) can be costly (there are no devices that use more than one cell to carry voice traffic), while both ATM\* and IP can exploit standard (and existing) devices. Moreover ATM is hard to tune because of several factors. Computation of the ATM optimal efficiency point is not trivial, codecs insert another degree of uncertainty, the behavior of the  $E_{gen}$  is less predictable due to the several interactions among the small cell size, the codec granularity and the non-fixed overhead due to the use of more than one cell. For these

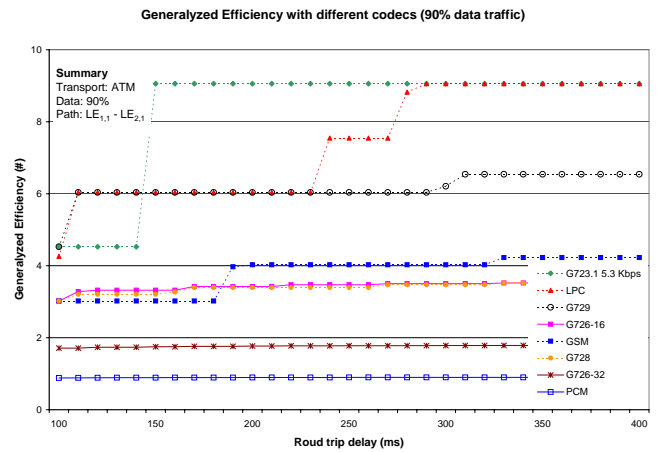
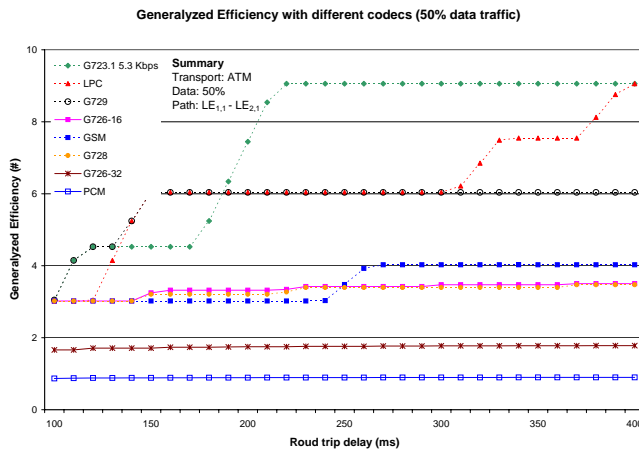
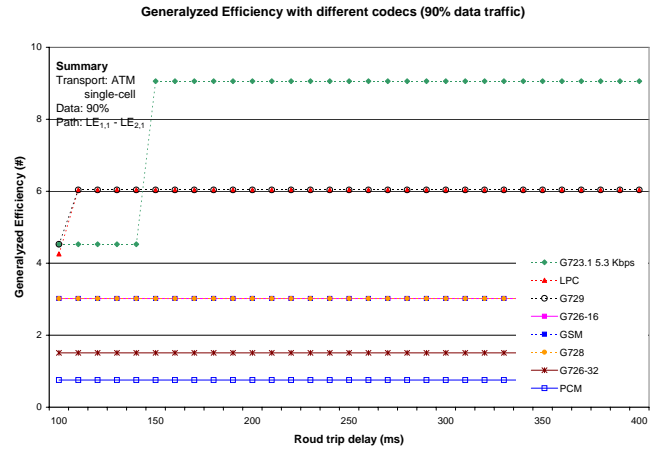
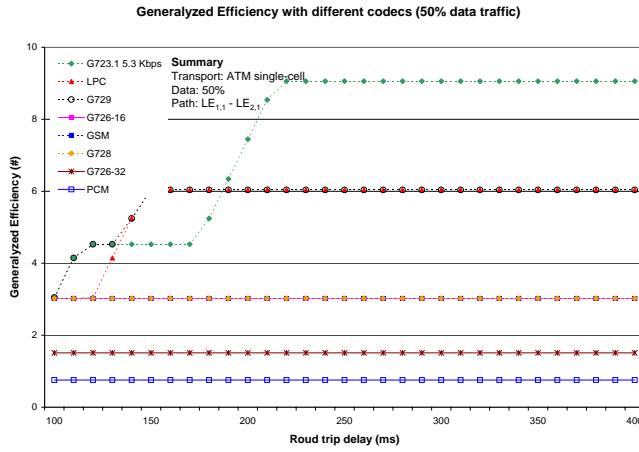
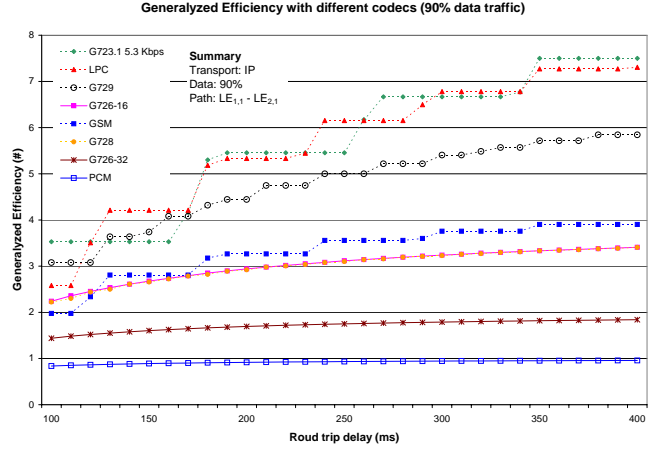
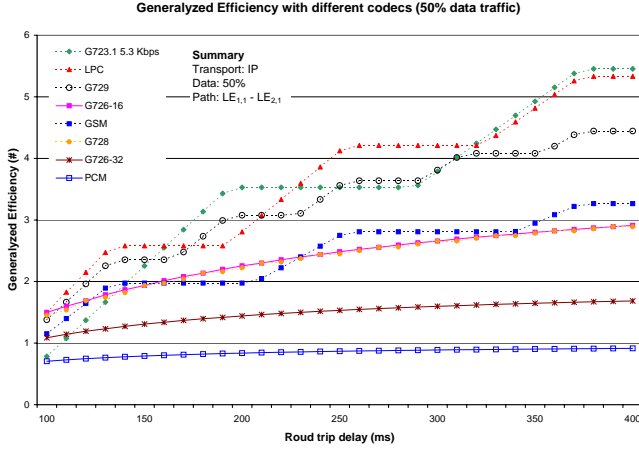


Figure 12: Comparing the Generalized Efficiency using different end-to-end requirement, according to different network technology.

Figure 13: Comparing the Generalized Efficiency using different end-to-end requirement, according to different network technology.



reasons the risk of using ATM in a sub-optimal point because of the tuning difficulties is real, showing once more the lack of convenience of ATM compared to the competing technologies.

Figure 12b shows also that a cell might not be filled completely especially in case of strict end-to-end delay requirements and low bit rate codecs (G.729, LPC, G.723.1-5.3). This invalidates one of assumptions of the ATM\* case, i.e. that this technology operates always when the cell is completely full. Although from a pure mathematical point of view this could affect some results, it does not influence the findings from a quantitative point of view; ATM\* is still the best technology when low-bit rate codecs are deployed, even if the maximum efficiency could be slightly less than the value derived from the equations presented in this paper<sup>10</sup>.

Supposing that all the considered codecs are able to provide an equivalent degree of quality for voice calls, an overall look at the graphs shows that G.729 is the best codec because of its compromise between low bit rate (8 Kbps) and granularity (10 ms). Obviously, high quality stereo multimedia will require other codecs, although the general ideas presented in this paper are still valid.

## 8 CONCLUSIONS

This paper focuses on the comparison of packet and cell switching technologies to the “legacy” circuit switching in carrying real-time traffic with guaranteed delay. The network is engineered with PGPS-aware nodes and per-flow guarantees, according to the Integrated Services model.

For the packet/cell switching arena, IP and ATM have been selected as the most important example of packet and cell switching technologies respectively. These technologies have been examined in depth, although the analysis can be easily extended to other technologies like IPv6 and Frame Relay. The maximum efficiency point, i.e. the point in which the utility of the network is to be maximized, has been derived for each technology. This can be used by a network provider to select the appropriate setup for each incoming call in order to maximize the overall amount of transported traffic.

The second step has been the comparison among the selected technologies in order to determine which one will be the best successor to circuit switching from a pure efficiency point of view. The comparison shows that ATM\* is the best technology for a network provider (i.e. telecom provider) intended to offer a low-delay voice service only, where data is a negligible part of the overall traffic and real-time flows can be compressed by using low bit rate codecs. IP can be the best technology for a network provider (i.e.

internet provider) intended to add the capability to transport high quality real-time traffic (i.e. high quality multimedia) to its best-effort service. Summarizing, IP will be the best choice when the provision of data traffic is an important part of the overall traffic (figure 14), when high bit rate codecs (for example video codecs or the ones providing CD-quality audio) are selected and when the end-to-end requirement is large. IP can have a better packetization efficiency ( $E_{pack}$ ) because of the ability to create bigger packets; this capability is useful particularly in case of non-interactive multimedia streaming which is expected to play a key role in the future digital networks. ATM\* (single-cell) can be the better choice at present, when the network is used to carry data and phone calls with high compression codecs. IP can be the best choice in the future when high quality multimedia and the prevailing of data traffic will become a reality. Surprisingly, ATM results the clear loser among all the considered technologies: expensive, similar to ATM\* when real-time requirements are strict, too similar to IP in the other cases. By the way, ATM (multiple cell) might be seen as a nonsense because one of the most important points of the ATM technology was the small cell size (compared to the larger frame size in Frame Relay) in order to decrease the end-to-end delay. For instance, ATM (multiple cell) tends to invert this process by aggregating several cells (i.e. large “virtual cells”) to decrease the protocol overheads.

The last step examined the importance of the codec used to encode real-time flows. The codec granularity can be of high importance in deriving the maximum efficiency point for each technology, particularly in case of low granularity codecs. Vice versa, the impact of the high granularity codecs on the maximum efficiency can be negligible. Nevertheless the correct optimal point, if needed, can be easily derived for IP and ATM\* technologies; once more ATM has been proved being hard to tune.

This paper does not consider the possibility to allocate different amount of bandwidth on different links (proposed in [12]). In other words, an overallocating call might use more bandwidth on underloaded links and it might allocate the minimum bandwidth ( $B_{real}$ ) on the congested ones. This process increases the efficiency; however the Author’s belief is that this solution increases the complexity of the reservation mechanism (a session does not know, “a priori”, the amount of bandwidth that has to be reserved on each link) without offering a large gain. Studies in [12] report some 20% efficiency improvement, therefore this solution has considered not being worthwhile.

Present trends seem to go beyond the results presented in this paper. Particularly the DiffServ Quality of Service model is getting stronger. Author’s belief is that even if the market is moving away from the model presented in this paper, there are no alternatives to the IntServ when per-flow absolute guarantees are required. Investigations about guarantees with flow aggregation [23] are still in an early stage. While also the choice of the PGPS scheduler is ques-

<sup>10</sup>The Author believe that the increased mathematical complexity needed to derive the correct results is not worthwhile according to the purposes of this paper.

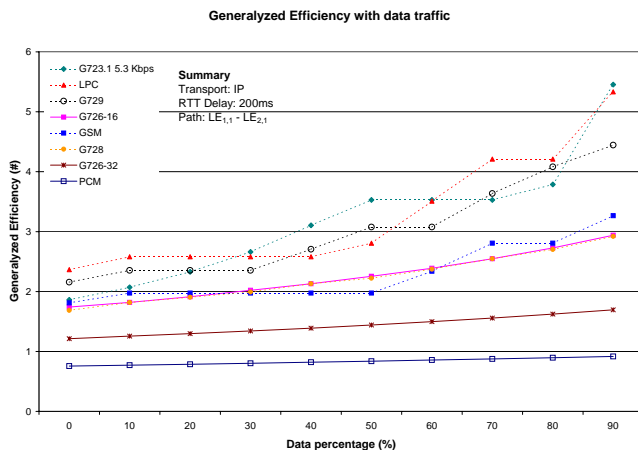


Figure 14: Maximum Generalized Efficiency of IP varying the percentage of data traffic.

tionable, there are no valid alternatives. For instance, while HFSC [21] looks like an excellent scheduling algorithm for data and real-time traffic integration, it is not available on any commercial product; moreover it has a complexity that is far beyond the PGPS one.

From the perspective of this paper, the circuit switching is the clear loser among the competing technologies. Among the winners, the packet switching seems to be the best choice for the next generation integrated networks over the long term period.

## ACKNOWLEDGMENTS

This work has been partially supported by Telecom Italia Lab, Torino, Italy. The Author thanks all the friends who made this work possible, particularly Mario Baldi for his several suggestions.

Manuscript received on October 11, 2001.

## REFERENCES

- [1] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation protocol (RSVP) - version 1 functional specification. Standard Track RFC 2205, Internet Engineering Task Force, September 1997.
- [2] The ATM Forum. *ATM User-Network Interface Specification - Version 3.1.1*. The ATM Forum, September 1994.
- [3] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single-node case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, June 1993.
- [4] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The multiple node case. *IEEE/ACM Transactions on Networking*, 2(2):137–150, April 1994.
- [5] S. Shenker R. Braden, D. Clark. *Integrated Service in the Internet Architecture: an Overview*. Internet Engineering Task Force, July 1994.
- [6] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. *RFC 2475: An Architecture for Differentiated Services*. Internet Engineering Task Force, December 1998.
- [7] B. Davie, A. Charny, J.C.R. Bennett, K. Benson, J.Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis. *RFC 3246: An Expedited Forwarding PHB (Per-Hop Behavior)*. Internet Engineering Task Force, March 2002.
- [8] G. Armitage, B. Carpenter, A. Casati, J. Crowcroft, J. Halpern, B. Kumar, and J. Schnizlein. *RFC 3248: A Delay Bound alternative revision of RFC 2598*. Internet Engineering Task Force, March 2002.
- [9] IETF. Pseudo Wire Emulation Edge to Edge (pwe3). URL=<http://www.ietf.org/>.
- [10] M. Baldi, D. Bergamasco, and F. Risso. On the efficiency of packet telephony. In *7<sup>th</sup> IFIP International Conference on Telecommunication Systems*, March 1999.
- [11] M. Baldi and F. Risso. Comparing the efficiency of ip and atm telephony. In *2<sup>nd</sup> International Conference on ATM*, June 1999.
- [12] M. Baldi and F. Risso. Efficiency of packet voice with deterministic delay. In *IEEE Communications Magazine*, pages 170–177, May 2000.
- [13] The ATM Forum. Voice and telephony over atm to the desktop. Approved Specification af-vtoa-0083.001, The ATM Forum, February 1999.
- [14] The ATM Forum. Atm trunking using aal1 for narrow band services v1.0. Approved Specification af-vtoa-0089.000, The ATM Forum, July 1997.
- [15] The ATM Forum. Atm trunking using aal2 for narrow-band services. Approved Specification af-vtoa-0113.000, The ATM Forum, February 1999.
- [16] H. Schulzrinne. Rtp profile for audio and video conferences with minimal control. Standard Track RFC 1890, Internet Engineering Task Force, January 1996.
- [17] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queuing algorithm. *ACM Computer Communication Review (SIGCOMM'89)*, pages 3–12, 1989.
- [18] C. Partridge. *Gigabit Networking*. Addison Wesley, October 1993.
- [19] S. Floyd and V. Jacobson. Link sharing and resource management models for packet networks. *IEEE/ACM Transaction on Networking*, 3(4), August 1995.
- [20] S. Golestani. A stop-and-go queuing framework for congestion management. In *ACM SIGCOMM '90*, pages 8–18, September 1990.
- [21] I. Stoica, H. Zhang, and T. S. Eugene Ng. A hierarchical fair service curve algorithm for link-sharing, real-time and priority service. In *ACM SIGCOMM'97*, 1997.
- [22] F. Risso. Decoupling bandwidth and delay properties in class based queuing. In *6th IEEE Symposium on Computers and Communications (ISCC 2001)*, pages 524–531, July 2001.
- [23] A. Blanc and R. Cruz. Private communication. Technical report, University of California at San Diego, December 2000.

